The

Memoir

Class

# The Memoir Class

for

# Configurable Typesetting User Guide

Peter Wilson

 $\label{eq:maintained} \text{Maintained by Lars Madsen}$  Corresponding to memoir version v3.7h, 2018/12/12



© 2001-2010 Peter R. Wilson © 2011- Lars Madsen

All rights reserved

The Herries Press, Normandy Park, WA.

Printed in the World

The paper used in this publication may meet the minimum requirements of the American National Standard for Information Sciences — Permanence of Paper for Printed Library Materials, ANSI Z39.48--1984.

#### 

First edition: 3 June 2001 2 July 2001 Second impression, with corrections: Second edition: 14 July 2001 3 August 2001Second impression, with corrections: Third impression, with minor additions: 31 August 2001 Third edition: 17 November 2001 Fourth edition: 16 March 2002 Fifth edition: 10 August 2002 Sixth edition: 31 January 2004 Seventh edition: 10 May 2008 12 July 2008 Eighth impression, with very minor corrections: 8 July 2009 Ninth impression, with additions and corrections: August 2009 Eighth edition: Tenth impression, with additions and corrections: 11 November 2015 memoir, n. a written record set down as material for a history or biography: a biographical sketch: a record of some study investigated by the writer: (in pl.) the transactions of a society. [Fr. mémoire — L. memoria, memory — memor, mindful.]

Chambers Twentieth Century Dictionary, New Edition, 1972.

memoir, n. [Fr. mémoire, masc., a memorandum, memoir, fem., memory < L. memoria, memory] 1. a biography or biographical notice, usually written by a relative or personal friend of the subject 2. [pl.] an autobiography, usually a full or highly personal account 3. [pl.] a report or record of important events based on the writer's personal observation, special knowledge, etc. 4. a report or record of a scholarly investigation, scientific study, etc. 5. [pl.] the record of the proceedings of a learned society

Webster's New World Dictionary, Second College Edition.

memoir, n. a fiction designed to flatter the subject and to impress the reader.

With apologies to Ambrose Bierce

# 목차 (略)

```
목차 (略) · vii
                  목차 · ix
               圖 목차 · xvi
               表 목차 · xix
           조판 例 목록·xxi
              프레훠스 · xxiii
           1 시작하기·1
2 \; Laying out the page \cdot 7
      3 텍스트와 글꼴 · 35
               4 표제 · 57
          5 Abstracts \cdot 71
 6 Document divisions \cdot 75
7 페이지 매김과 머릿글 · 105
      8 문단과 리스트 · 121
      9 Contents lists \cdot 139
 10 Floats and captions \cdot 167
  11 Rows and columns \cdot 205
         12 Page notes \cdot 229
     13 Decorative text \cdot 249
```

viii 목차 (略)

- 14 Poetry  $\cdot$  255
- 15 Boxes, verbatims and files  $\cdot$  271
  - 16 Cross referencing  $\cdot$  293
    - 17 Back matter  $\cdot$  297
    - 18 Miscellaneous  $\cdot$  317
  - 19 For package users  $\cdot$  341
  - 20 An example book design  $\cdot$  345
- 21 An example thesis design  $\cdot$  353
  - A Packages and macros  $\cdot$  375
    - B Showcases  $\cdot$  379
      - C Sniplets  $\cdot 397$
      - D Pictures  $\cdot 405$
    - E LaTeX and TeX  $\cdot$  425
  - F The terrors of errors  $\cdot$  443
    - G Comments  $\cdot$  467
      - $Notes \cdot 469$
      - 문헌목록 · 471
        - **색인** · 479

# 목차

목	차 (略)		vii
목	차		ix
몹	목차		xvi
表	목차		xix
조	판 例 목	록	xxi
<u>ш</u> ;	레훠스		xxiii
1	시작하 1.1 1.2 1.3 1.4 1.5	기 용지 크기 옵션	1 1 2 4 5 5
2	Laying 2.1 2.2 2.3 2.4	out the page Introduction	7 7 7 8 13
	2.5 2.6 2.7 2.8 2.9 2.10	block size 16 Headers, footers and marginal notes Other Putting it together Side margins Emitting the page size Example 2.10.1 The page layout of this manual 27	21 22 22 26 26 26
	2.11	Predefined layouts	29

x 목차

3.1 글꼴 3.2 Font sizes 3.3 Spaces 3.3.1 Paragraphs 49, 3.3.2 Double spacing 50 3.4 Overfull lines 3.5 Sloppybottom 3.6 Text case 3.6.1 Nested text 55  4 표제 4.1 표지 양식화 4.2 감사의 말 양식화하기  5 Abstracts 5.1 Styling 5.2 One column abstracts  6 Document divisions 6.1 Logical divisions 6.2 Sectional divisions 6.2 Sectional divisions 6.2.1 Appendices 78 6.3 Numbering 6.4 Book and part headings 6.4.1 Leadpage 83 6.5 Chapter headings 6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles  7 페이지 매김과 머릿글 7.1 페이지 매김과 폴리오 7.2 페이지 매김과 폴리오 7.3 머릿글과 바닥글 만들기 7.3.1 에서 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showloes pagestyle 7.5 Other things to do with page styles			
3.2 Font sizes 3.3 Spaces 3.3.1 Paragraphs 49, 3.3.2 Double spacing 50 3.4 Overfull lines 3.5 Sloppybottom 3.6 Text case 3.6.1 Nested text 55  4 표제 4.1 표지 양식화 4.2 감사의 말 양식화하기  5 Abstracts 5.1 Styling 5.2 One column abstracts  6 Document divisions 6.1 Logical divisions 6.2 Sectional divisions 6.2 Sectional divisions 6.3 Numbering 6.4 Book and part headings 6.4.1 Leadpage 83 6.5 Chapter headings 6.4.1 Leadpage 83 6.5 Chapter headings 6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles  7 페이지 매김과 머릿글 7.1 페이지 매김과 플리오 7.2 페이지 매김과 플리오 7.3 머릿글과 바닥글 만들기 7.3.1 예시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles  8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 Flush and ragged 8.3 Quotations	3	텍스트	와 글꼴
3.3 Spaces 3.3.1 Paragraphs 49, 3.3.2 Double spacing 50 3.4 Overfull lines 3.5 Sloppybottom 3.6 Text case 3.6.1 Nested text 55 4 표제 4.1 표지 양식화 4.2 감사의 말 양식화하기 5 Abstracts 5.1 Styling 5.2 One column abstracts 6 Document divisions 6.1 Logical divisions 6.2.1 Appendices 78 6.3 Numbering 6.4 Book and part headings 6.4.1 Leadpage 83 6.5 Chapter headings 6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles 7 페이지 매김과 머릿글 7.1 페이지 매김과 플리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 예시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles 8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 Flush and ragged 8.3 Quotations		3.1	글꼴
3.3.1 Paragraphs 49, 3.3.2 Double spacing 50 3.4 Overfull lines 3.5 Sloppybottom 3.6 Text case 3.6.1 Nested text 55  4 표제 4.1 표지 양식화 4.2 감사의 말 양식화하기  5 Abstracts 5.1 Styling 5.2 One column abstracts  6 Document divisions 6.1 Logical divisions 6.2 Sectional divisions 6.2 Sectional divisions 6.2 Numbering 6.4 Book and part headings 6.4.1 Leadpage 83 6.5 Chapter headings 6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles  7 페이지 매김과 머릿글 7.1 페이지 매김과 플리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 에시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles  8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations		3.2	Font sizes
3.4 Overfull lines 3.5 Sloppybottom 3.6 Text case 3.6.1 Nested text 55  4 표제 4.1 표지 양식화 4.2 감사의 말 양식화하기  5 Abstracts 5.1 Styling 5.2 One column abstracts  6 Document divisions 6.1 Logical divisions 6.2 Sectional divisions 6.2 Sectional divisions 6.3 Numbering 6.4 Book and part headings 6.4.1 Leadpage 83 6.5 Chapter headings 6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles  7 페이지 매김과 머릿글 7.1 페이지 매김과 플리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 에서 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles  8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 Flush and ragged 8.3 Quotations		3.3	•
3.5 Sloppybottom 3.6 Text case 3.6.1 Nested text 55  4 표제 4.1 표지 양식화 4.2 감사의 말 양식화하기  5 Abstracts 5.1 Styling 5.2 One column abstracts  6 Document divisions 6.1 Logical divisions 6.2 Sectional divisions 6.2 Sectional divisions 6.3 Numbering 6.4 Book and part headings 6.4.1 Leadpage 83 6.5 Chapter headings 6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles  7 페이지 매감과 머릿글 7.1 페이지 매감과 田릿글 7.1 페이지 매감과 플리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 에시 페이지 양식 1.1, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles  8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations			3.3.1 Paragraphs 49, 3.3.2 Double spacing 50
3.6 Text case 3.6.1 Nested text 55 4 표제 4.1 표지 양식화 4.2 감사의 말 양식화하기 5 Abstracts 5.1 Styling 5.2 One column abstracts 6 Document divisions 6.1 Logical divisions 6.2 Sectional divisions 6.2.1 Appendices 78 6.3 Numbering 6.4 Book and part headings 6.4.1 Leadpage 83 6.5 Chapter headings 6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles 7 페이지 매김과 머릿글 7.1 페이지 매김과 플리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 에서 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles 8 문단과 리스트 8.1 문단 8.1. 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations		3.4	
3.6.1 Nested text 55 4 표제 4.1 표지 양식화 4.2 감사의 말 양식화하기 5 Abstracts 5.1 Styling 5.2 One column abstracts 6 Document divisions 6.1 Logical divisions 6.2 Sectional divisions 6.2 Sectional divisions 6.2 Abstracts 6.3 Numbering 6.4 Book and part headings 6.4.1 Leadpage 83 6.5 Chapter headings 6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles 7 페이지 매김과 머릿글 7.1 페이지 매김과 田릿글 7.1 페이지 매김과 바닥글 만들기 7.3.1 에서 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles 8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations			Sloppybottom
4.1 표지 양식화 4.2 감사의 말 양식화하기  5 Abstracts 5.1 Styling 5.2 One column abstracts  6 Document divisions 6.1 Logical divisions 6.2 Sectional divisions 6.2.1 Appendices 78 6.3 Numbering 6.4 Book and part headings 6.4.1 Leadpage 83 6.5 Chapter headings 6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles  7 페이지 매김과 머릿글 7.1 페이지 매김과 플리오 7.3 머릿글과 바닥글 만들기 7.3.1 예시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles  8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations		3.6	
4.1 표지 양식화 4.2 감사의 말 양식화하기  5 Abstracts 5.1 Styling 5.2 One column abstracts  6 Document divisions 6.1 Logical divisions 6.2 Sectional divisions 6.2.1 Appendices 78 6.3 Numbering 6.4 Book and part headings 6.4.1 Leadpage 83 6.5 Chapter headings 6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles  7 페이지 매김과 머릿글 7.1 페이지 매김과 플리오 7.3 머릿글과 바닥글 만들기 7.3.1 예시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles  8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations	1	표제	
4.2 감사의 말 양식화하기 5. Abstracts 5.1 Styling	4		표기 야시하
5.1 Styling 5.2 One column abstracts  6.1 Logical divisions 6.1 Logical divisions 6.2 Sectional divisions 6.2.1 Appendices 78 6.3 Numbering 6.4 Book and part headings 6.4.1 Leadpage 83 6.5 Chapter headings 6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles  7 페이지 매김과 머릿글 7.1 페이지 매김과 플리오 7.2 페이지 애김과 플리오 7.3 머릿글과 바닥글 만들기 7.3.1 예시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles  8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations			
5.1 Styling 5.2 One column abstracts  6 Document divisions 6.1 Logical divisions 6.2 Sectional divisions 6.2.1 Appendices 78 6.3 Numbering 6.4 Book and part headings 6.4.1 Leadpage 83 6.5 Chapter headings 6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles  7 페이지 매김과 프리오 7.1 페이지 매김과 폴리오 7.3 머릿글과 바닥글 만들기 7.3.1 예시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showloos pagestyle 7.5 Other things to do with page styles  8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations		4.2	14시의 글 6의의에서 · · · · · · · · · · · · · · · · · · ·
5.2 One column abstracts 6 Document divisions 6.1 Logical divisions 6.2 Sectional divisions 6.2.1 Appendices 78 6.3 Numbering 6.4 Book and part headings 6.4.1 Leadpage 83 6.5 Chapter headings 6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles 7 페이지 매김과 머릿글 7.1 페이지 매김과 플리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 메시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles 8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations	5	Abstra	
5.2 One column abstracts 6 Document divisions 6.1 Logical divisions 6.2 Sectional divisions 6.2.1 Appendices 78 6.3 Numbering 6.4 Book and part headings 6.4.1 Leadpage 83 6.5 Chapter headings 6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles 7 페이지 매김과 머릿글 7.1 페이지 매김과 플리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 메시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles 8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations		5.1	Styling
6.1 Logical divisions 6.2 Sectional divisions 6.2.1 Appendices 78 6.3 Numbering 6.4 Book and part headings 6.4.1 Leadpage 83 6.5 Chapter headings 6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles 7 페이지 매김과 머릿글 7.1 페이지 매김과 플리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 에서 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles 8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations		5.2	One column abstracts
6.1 Logical divisions 6.2 Sectional divisions 6.2.1 Appendices 78 6.3 Numbering 6.4 Book and part headings 6.4.1 Leadpage 83 6.5 Chapter headings 6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles 7 페이지 매김과 머릿글 7.1 페이지 매김과 플리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 에서 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles 8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations	6	D	and Profession
6.2 Sectional divisions 6.2.1 Appendices 78 6.3 Numbering 6.4 Book and part headings 6.4.1 Leadpage 83 6.5 Chapter headings 6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles 7 페이지 매김과 머릿글 7.1 페이지 매김과 폴리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 예시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles 8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations	0		
6.2.1 Appendices 78 6.3 Numbering 6.4 Book and part headings 6.4.1 Leadpage 83 6.5 Chapter headings 6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles 7 페이지 매감과 머릿글 7.1 페이지 매감과 플리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 예시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles 8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations			
6.3 Numbering 6.4 Book and part headings 6.4.1 Leadpage 83 6.5 Chapter headings 6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles 7 페이지 매김과 머릿글 7.1 페이지 매김과 플리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 예시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles 8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations		6.2	
6.4 Book and part headings 6.4.1 Leadpage 83 6.5 Chapter headings 6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles 7 페이지 매김과 머릿글 7.1 페이지 매김과 폴리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 예시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles 8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations		0.0	
6.4.1 Leadpage 83 6.5 Chapter headings			
6.5 Chapter headings 6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles 7.1 페이지 매김과 머릿글 7.1 페이지 매김과 폴리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 예시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations		6.4	· · · · · · · · · · · · · · · · · · ·
6.5.1 Defining a chapter style 87, 6.5.2 Further chapterstyles 90, 6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles 7 페이지 매김과 머릿글 7.1 페이지 매김과 폴리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 예시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles 8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations		0.5	· •
6.5.3 Chapter precis 92 6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles 7 페이지 매김과 머릿글 7.1 페이지 매김과 폴리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 예시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles 8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations		6.0	-
6.6 Lower level headings 6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles  7 페이지 매김과 머릿글 7.1 페이지 매김과 폴리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 예시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles  8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations			
6.7 Fancy anonymous breaks 6.8 Footnotes in division headings 6.9 Predefined heading styles  7 페이지 매김과 머릿글 7.1 페이지 매김과 폴리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 예시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles  8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations		0.0	
6.8 Footnotes in division headings			
6.9 Predefined heading styles 7 페이지 매김과 머릿글 7.1 페이지 매김과 폴리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 예시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles 8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations			
7.1 페이지 매김과 머릿글 7.1 페이지 매김과 폴리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 에시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles  8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations			<u> </u>
7.1 페이지 매김과 폴리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 예시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles 8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations		6.9	Predefined heading styles
7.1 페이지 매김과 폴리오 7.2 페이지 양식 7.3 머릿글과 바닥글 만들기 7.3.1 예시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle 7.5 Other things to do with page styles 8 문단과 리스트 8.1 문단 8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged 8.3 Quotations	7	페이지	매김과 머릿글
7.2 페이지 양식			
7.3 머릿글과 바닥글 만들기			
7.3.1 예시 페이지 양식 111, 7.3.2 Index headers 116, 7.3.3 Float pages 117 7.4 The showlocs pagestyle			머릿글과 바닥글 만득기
7.4 The showlocs pagestyle		•••	7 3 1 예시 페이지 양식 111 7 3 2 Index headers 116 7 3 3 Float pages 117
7.5 Other things to do with page styles		7 4	
8 문단과 리스트 8.1 문단			
8.1 문단		1.0	concretings to do with page styles
8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged	8	문단과	리스트
8.1.1 블록 문단 121, 8.1.2 Hanging paragraphs 122 8.2 Flush and ragged		8.1	문단
8.2 Flush and ragged			
8.3 Quotations		8.2	, , , , , , , , , , , , , , , , , , , ,

*목차* xi

	8.5 8.6	8.4.1 Last line not short 125, 8.4.2 Russian typography 126, 8.4.3 Fill with rules 126, 8.4.4 Some ragged paragraphs 127, 8.4.5 Left spring right 128 Changing the textwidth
9	Conte	nts lists
	9.1	General ToC methods
	9.2	The class ToC methods
	9.3	New 'List of' and entries
	9.4	Chapter precis
	9.5	Contents lists and bookmarks
10	Floats	and captions
	10.1	New float environments
	10.2	Setting off a float
	10.3	Multiple floats
	10.4	Where LaTeX puts floats
	10.5	Captions
	10.6	Caption styling
	10.7	Continuation captions and legends
	10.8	Bilingual captions
	10.9 10.10	Subcaptions
	10.11	How LaTeX makes captions
	10.12	Footnotes in captions
	10.13	The class versus the caption package (and its friends)
11	Rows	and columns
	11.1	General
	11.2	The preamble
		11.2.3 Surprises 211
	11.3	The array environment
	11.4	Tables
	11.5	Fear's rules
	11.6	Tabular environments
	11.7	Spaces and rules

xii 목차

	11.8	Free tabulars	22
12	Page r	notes	22
	12.1	Footnotes	22
		12.1.1 A variety of footnotes 230, 12.1.2 Styling 232	
	12.2	Marginal notes	23
	12.3	Side notes	23
	12.4 12.5	Sidebars	23 24
	12.6	Endnotes	24
13	Decora	ative text	24
	13.1	Epigraphs	24
	13.2	General	25
	13.3	Epigraphs before chapter headings	25
14	Poetry		25
	14.1	Classy verse	25
	14.2	Titles	26
	14.3	Examples	26
15	Boxes,	verbatims and files	27
	15.1	Boxes	27
	15.2	Long comments	27
	15.3	Verbatims	27
	15.4	Files	28
	15.5	Answers	29
16	Cross	referencing	29
	16.1	Labels and references	29

*목*차 xiii

	16.2	Reference by name
17	Back	matter
11	17.1	Bibliography
	11.1	17.1.1 BibTex 299
	17.2	Index
	11.2	17.2.1 Printing an index 300, 17.2.2 Preparing an index 301,
		17.2.3 MakeIndex 304, 17.2.4 Controlling MakeIndex output 307,
		17.2.5 Indexing and the <i>natbib</i> package 310
	17.3	Glossaries
		17.3.1 Controlling the glossary 311
18	Miscel	laneous
	In wh	ich we talk of many things, but not shoes or ships or sealing wax, nor
		ges and kings.
	18.1	Draft documents
	18.2	Change marks
	18.3	Trim marks
	18.4	Sheet numbering
	18.5	Gatherings or signatures
	18.6	Time
	18.7	Page breaks before lists
	18.8	Changing counters
	18.9	New new and provide commands
	18.10	Changing macros
	18.11	String arguments
	18.12	Odd/even page checking
	18.13	Moving to another page
	18.14	9
		18.14.1 Numeric numbers 327, 18.14.2 Named numbers 328, 18.14.3 Frac-
		tions 330
		An array data structure
	18.16	Checking the processor
		18.16.1 Checking for pdfLaTeX 332, 18.16.2 Checking for etex 332,
	10.15	18.16.3 Checking for XeTeX 333, 18.16.4 Checking for LuaTeX 333
		Leading
		Minor space adjustment
	18.19	Adding a period
	18.20	Words and phrases
	18.21	Symbols
	18.22	Two simple macros
	18.23	Vertical centering
	18.24	For package writers
		18.24.1 Emulating packages 336, 18.24.2 Inserting code before and after
	10 05	a file, package or class 337
	18.25	Heading hooks
	10.20	Documenting LateA commands

19	For package users	341	
	19.1 Class/package name clash	. 341	
	19.2 Support for bididirectional typesetting		
20	An example book design	345	
	20.1 Introduction	. 345	
	20.2 Design requirements	. 345	
	20.3 Specifying the page and typeblock		
	20.4 Specifying the sectional titling styles		
	20.4.1 The chapter style 348, 20.4.2 Lower level divisions 348		
	20.5 Specifying the pagestyle		
	20.6 Captions and the ToC	. 351	
	20.7 Preamble or package?	. 351	
21	An example thesis design	353	
	21.1 Example US thesis typographic requirements	. 353	
	21.1.1 General 353, 21.1.2 Preliminary matter 354, 21.1.3 Table of con-		
	tents 355, 21.1.4 Lists 356, 21.1.5 Main text 356, 21.1.6 Backmatter 357		
	21.2 Code		
	21.2.4 The ToC and friends 360, 21.2.5 Chapter styling 361, 21.2.6 Section at a styling 362, 21.2.7 Continue 362, 21.2.8 The hilliam when 362		
	tion, etc., styling 362, 21.2.7 Captions 363, 21.2.8 The bibliography 363, 21.2.9 End notes 363, 21.2.10 Preliminary headings 364, 21.2.11 Compo-		
	, , , , , , , , , , , , , , , , , , , ,		
	nents of the title and approval pages 364, 21.2.12 The title and approval pages 365, 21.2.13 The last bits 369		
	21.3 Usage	. 369	
	21.4 Comments		
		. 011	
A	Packages and macros	375	
	A.1 Packages	. 375	
	A.2 Macros	. 376	
В	Showcases	379	
	B.1 Chapter styles		
	B.1.1 Chappell 392, B.1.2 Demo, Demo2 and demo3 393, B.1.3 Pedersen 393, B.1.4 Southall 394, B.1.5 Veelo 395		
C	Sniplets	397	
	Sniplet C.1 (Mirroring the output)		
	Sniplet C.2 (Remove pagenumber if only one page)		
	Sniplet C.3 (A kind of draft note)	. 398	
	Sniplet C.4 (Adding indentation to footnotes)	. 399	
	Sniplet C.5 (Background image and trimmarks)		
	Sniplet C.6 (Autoadjusted number widths in the ToC)		
	Sniplet C.7 (Using class tools to make a chapter ToC)		
	Sniplet C.8 (An appendix ToC)		
D	Pictures	405	

*목차* xv

	D.1	Basic principles
	D.2	Picture objects
		D.2.1 Text 407, D.2.2 Boxes 408, D.2.3 Lines 413, D.2.4 Arrows 415,
	D.3	D.2.5 Circles 415 Repetitions
	D.3 D.4	Bezier curves
	D.4	Deziel curves
Ξ	LaTeX	Cand TeX
	E.1	The TeX process
	E.2	LaTeX files
	E.3	Syntax
	E.4	(La)TeX commands
	E.5	Calculation
		E.5.1 Numbers 432, E.5.2 Lengths 433
	E.6	Programming
	The te	errors of errors
	F.1	TeX messages
		F.1.1 TeX capacity exceeded 452
	F.2	LaTeX errors
	F.3	LaTeX warnings
	F.4	Class errors
	F.5	Class warnings
3	Comm	
	G.1	Algorithms
		G.1.1 Autoadjusting \marginparwidth 467
Vo.	tes	
		er 3 텍스트와 글꼴
	-	er 12 Page notes
<b>_</b>	허목록	
Ξì	275	
백인	<u> </u>	

## 圖 목차

2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 2.10 2.11 2.12 2.13	LaTeX page layout parameters for a recto page The memoir class page layout parameters for a verso page The memoir class page layout parameters for a recto page The recto page layout for this manual Default layout for letterpaper Letterpaper layout: Left \medievalpage, Right \medievalpage[12] Letterpaper layout: Left \isopage, Right \isopage[12] Letterpaper layout: Left \semiisopage, Right \semiisopage[12] Default layout for a4paper A4paper layout: Left \medievalpage, Right \medievalpage[12] A4paper layout: Left \isopage, Right \isopage[12] A4paper layout: Left \isopage, Right \isopage[12] Example of the nine \setpageXX macros for placing a trimmed page on the stock. These are all odd pages, under twoside, even pages will have the left/right trims reversed.	9 10 11 28 30 30 30 31 31 31 31
4.1 4.2 4.3 4.4 4.5	타이포그래피에 관한 책의 표지 레이아웃 박사 학위 논문의 지정 표지 양식의 예시	58 59 60 61 62
6.1 6.2 6.3	Class layout parameters for chapter titles. Working with \beforechapskip need a little thought, see the text	85 94 95
7.1	머릿글과 바닥글 영역	109
8.1 8.2	문단의 매개변수	121 135
9.1 9.2	Example extracts from toc, lof and lot files	141 141
10.1	Example framed figure	170

圖	' 목ネ	₹}	xvi

10.2	Example framed figure and caption	170
10.3	Example ruled figure	171
10.4	Example ruled figure and caption	171
10.5	Example float with two illustrations	171
10.6	Graphic 1 in a float	172
10.7	Graphic 2 in same float	172
10.8	Left center aligned	173
10.9	Right figure. This has more text than the adjacent caption (10.8) so the	1.0
10.0	heights are unequal	173
10.10	Left top aligned	174
	Right figure. This has more text than the adjacent caption (10.10) so the	111
10.11	heights are unequal	174
10 19	Left bottom aligned	175
	Right figure. This has more text than the adjacent caption (10.12) so the	110
10.13	heights are unequal	175
10.14	Float and text page parameters	177
	Float parameters	178
	Long \bitwonumcaption	191
	Lang \bitwonumcaption	191
10.17	Long English \bionenumcaption	191
10 10	Lang Deutsch \bionenumcaption	191
	Short English \bicaption	192
	Figure with two subfigures	195
,	a) Subfigure 1	195
,	b) Subfigure 2	195
	A picture is worth a thousand words	201
10.21	A different kind of figure caption	202
11.1	Example of a regular tabular	220
11.2	Example tabularx and tabular* with widths of 250pt	221
11.3	Example tabularx and tabular* with widths of 300pt	221
11.4	Changing the width of a row ordered table	228
11.5	Changing the width of a column ordered table	228
11.0	changing the width of a column ordered table	220
12.1	Footnote layout parameters	233
12.2	Interpretation of the arguments to the \Xmargin commands for specifying	
	the side in which to place side note like material. X here equals marginpar,	
	sidepar, sidebar, or sidefoot.	237
12.3	Example endnote listing	244
17.1	Raw indexing: (left) index commands in the source text; (right) idx file entries	304
17.2	Processed index: (left) alphabeticized ind file; (right) typeset index	305
18.1	The four trimmark types	320
21.1	Example Archibald Smythe University title page	366
21.2	Example Archibald Smythe University approval page	367

	,	
xviii	圖 목차	

B.1	The default chapterstyle
B.2	The section chapterstyle
B.3	The hangnum chapterstyle
B.4	The companion chapterstyle
B.5	The article chapterstyle
B.6	The bianchi chapterstyle
B.7	The bringhurst chapterstyle
B.8	The brotherton chapterstyle
B.9	The chappell chapterstyle
B.10	The crosshead chapterstyle
B.11	The culver chapterstyle
B.12	The dash chapterstyle
B.13	The demo2 chapterstyle
B.14	The dowding chapterstyle
B.15	The ell chapterstyle
B.16	The ger chapterstyle
B.17	The komalike chapterstyle
B.18	The lyhne chapterstyle. This style requires the graphicx package
B.19	The madsen chapterstyle. This style requires the graphicx package
B.20	The ntglike chapterstyle
B.21	The southall chapterstyle
B.22	The tandh chapterstyle
B.23	The thatcher chapterstyle
B.24	The veelo chapterstyle. This style requires the graphicx package
B.25	The verville chapterstyle
B.26	The wilsondob chapterstyle
	* v
D.1	Some points in the cartesian coordinate system
D.2	Specification of a line or arrow
D.3	Sloping lines and arrows
D.4	Some measuring scales
D.5	Two Bezier curves

# 表 목차

1.1	미터법 표시 클래스 용지 옵션과 그 명령어	1
1.2	미국 용지의 클래스 옵션과 그 명령	2
1.3	영국 용지의 클래스 옵션과 그 명령	2
2.1	Arguments and results for \settrimmedsize and \settypeblocksize	12
2.2	Average characters per line	14
2.3	Lowercase alphabet lengths, in points, for various fonts	16
2.4	Arguments and results for \setlrmargins	18
2.5	Arguments and results for \setlrmarginsandblock	18
2.6	Arguments and results for \setulmargins	20
2.7	Arguments and results for \setulmarginsandblock	20
2.8	Arguments and results for \setheaderspaces	21
2.9	The class and LaTeX page layout parameters	24
2.10	Results from sample \textheight adjustments	25
3.1	Glyphs in the LaTeX supplied Palatino roman font	36
3.2	Glyphs in the LaTeX distributed Symbol font	37
3.3	Glyphs in the LaTeX distributed Zapf Dingbat font	38
3.4	Font categorisation and commands	41
3.5	Font declarations	42
3.6	Standard font size declarations	45
3.7	Standard font sizes	45
3.8	The memoir class font size declarations	45
3.9	The memoir class font sizes	46
6.1	Division levels	79
6.2	Default display sectioning layout parameter values	94
6.3	Default run-in sectioning layout parameter values	95
6.4	Values for S in section styling macro names	95
6.5	Default fonts for sectional headings	102
6.6	Fonts used by different headstyles	104
7.1	\thispagestyle의 사용	107
7.2	페이지 머릿글에 대한 표지 매크로	108
9.1	Indents and Numwidths	142

xx 表 名外

9.2 9.3	Values for X in macros for styling the titles of 'List of'	145 148
10.1	Float placement parameters	179
10.1		179
-	Float spacing parameters	
10.3	Redesigned table caption style	184
10.4	A multi-part table	186
10.5	Another table	186
Legen	dary table (toc 1)	187
	Legendary table (toc 2)	187
10.6	Permitted arguments for some sidecaption related commands	198
11.1	The array and tabular preamble options	206
11.2	Demonstrating the parts of a table	214
11.3	Two views of one table	215
11.4	Micawber's law	216
11.5	A narrow table split half and half	216
11.6	Example table with fills	219
11.7	Example automatic row ordered table	227
12.1	Some footnote text styles	234
17.1	MakeIndex configuration file input parameters	305
17.2	MakeIndex configuration file output parameters	308
18.1	Defined words and phrases	335
E.1	Some internal macros for numbers	433

## 조판 例 목록

3.1	Badly mixed fonts
3.2	Sometimes mixed fonts work
3.3	Emphasis upon emphasis
4.1	예시 \maketitle 표제
6.1	A variety of subhead styles
8.1	Setting the source of a quotation
8.2	Paragraph's line not too short
8.3	Rules for spaces
8.4	Ragged paragraphs
8.5	A sprung paragraph
8.6	Smallcap quote style description list
8.7	Changing space before and after lists
11.1	Tabular with narrow and wide headings
14.1	Phantom text in verse
14.2	Verse with regular quote marks
14.3	Verse with hanging left quote marks
16.1	Named references should be to titled elements
16.2	Current title
18.1	TeX's minimum number in words (English style)
18.2	TeX's maximum number in words (American style)
	Varieties of fractions in text
	Super- and subscripts in text
D.1	Picture: text
	Picture: text in boxes
D.3	Picture: positioning text
	Picture: dashed box
D.5	Picture: framing
D.6	Picture: stacking
	Picture: saved boxes
	Picture: circles
	Picture: ovals
	Picture: text in oval
	Picture: repetitions

### 프레훠스

개인적 경험으로나 comp. tex. tex 뉴스그룹을 보거나 LaTeX을 사용함에 있어 주요 문제는 문서 디자인에 관련된 것이다. 몇 년 전만 하더라도 crr에 질문이 올라오면 누군가 주어진 특정 문제를 해결하는 짧은 코드를 제공하여 답변했고 이 일이 반복되었다. 더 최근에는 답변이 '——— 패키지를 사용하라'는 것으로 바뀌었고 역시 이런 상황이 반복되고 있다.

너무 많은 패키지를 사용하다 보니 순서도 뒤죽박죽이 되고 각 패키지의 사용설명서를 어디에 두었는지 잊어버리게 되었는데, 심지어 내가 작성한 패키지마저 그런 상태가 되었다. memoir는 book 클래스에서의 디자인 관련 패키지를 통합하려는 시도이다. 필자는 report가 아니라 book 클래스를 문제삼았다. 이 둘은 사실상 거의 동일한 클래스이고 유일한 차이는 book에 abstract 환경이 없다는 점이다. 사실 abstract를 필요하다면 가져다 쓰는 것은 어려운 일도 아니다. 심지어 조금 손을 보면 book 클래스 문서를 article 문서처럼 보이게 하는 것도 가능하다. 이런 관점에서 여러 상황에 맞도록 손보는 일을 유념하여 memoir 클래스를 작성하였다.

memoir 클래스는 지금까지 외부 패키지로 하던 기능들을 하나로 모아 짜넣은 것이다. 이 클래스로 들여온 각 패키지의 코드는 대부분 새롭게 구현하였고 단지 필자 자신이 작성한 패 키지는 단순히 복사–붙여넣기하였다. LaTeX과 여러 패키지의 도움이 없었다면 이 클래스는 작성될 수 없었을 것이다.

직접 활용한 패키지 말고도 문헌목록에 열거된 다양한 패키지의 아이디어를 빌어온 것도 많다. 이 패키지의 저자들은 스스로 인식하지 못하더라도 어떤 식으로든 이 클래스에 기여한 바가 있다. comp.text.tex 뉴스그룹의 참여자들 또한 귀중한 기여를 하였는데, 한편으로는 LaTeX에 관한 뭔가를 질문함으로써, 다른 한편으로 답변을 제공함으로써 그리하였다. 매우 분위기좋고 공부가 되는 포럼이었다.

Peter Wilson Seattle, WA June 2001

### One

### 시작하기

다른 클래스와 마찬가지로, memoir 클래스는 \documentclass [ $\langle options \rangle$ ] {memoir}로 불러온다. 옵션 (option)으로는 종이 크기, 본문의 활자 크기, 원고의 종류, 수식 조판상의 지시 등을지정할 수 있다.

#### 1.1 용지 크기 옵션

기본 용지 크기 (stock size)는 인쇄기에 공급되는 종이 (paper) 한 장의 크기를 말한다. 옵션으로 지정할 수 있는 용지 크기가 Table 1.1부터 Table 1.3까지 나열되어 있으며, 여기에는 용지 크기나 종이 크기를 같은 단위 (dimension)로 설정하기 위한 명령도 들어 있다.

표에 잘 어울리지 않는 다음 두 가지 옵션을 따로 적는다.

ebook  $6 \times 9$  inches의 용지 크기로, 주로 컴퓨터 모니터용 '전자책'에 적합함.

landscape 가로와 세로의 길이를 맞바꾸어 (옆으로 긴 종이로) 조판하기 위함.

landscape를 제외한 모든 옵션은 상호배타적이므로 같이 쓰지 않는다. 용지와 종이 크기의 기본값(default)은 letterpaper이다.

만약 리스트에 없는 용지 크기를 사용하고 싶다면 그에 맞는 방법들이 있겠지만, 나중에 설명하기로 한다.

옵션	크기	용지 크기 명령	페이지 크기 명령
a6paper	$148\times105\:\mathrm{mm}$	\stockavi	\pageavi
a5paper	$210\times148~\mathrm{mm}$	\stockav	\pageav
a4paper	$297 \times 210 \; \mathrm{mm}$	\stockaiv	\pageaiv
a3paper	$420\times297~\mathrm{mm}$	\stockaiii	\pageaiii
b6paper	$176\times125\:\mathrm{mm}$	\stockbvi	\pagebvi
b5paper	$250\times176~\mathrm{mm}$	\stockbv	\pagebv
b4paper	$353\times250~\mathrm{mm}$	\stockbiv	\pagebiv
b3paper	$500\times353\:\mathrm{mm}$	\stockbiii	\pagebiii
mcrownvopaper	$186\times123\:\mathrm{mm}$	\stockmetriccrownvo	\pagemetriccrownvo
mlargecrownvopaper	$198\times129\:\mathrm{mm}$	\stockmlargecrownvo	\pagemlargecrownvo
mdemyvopaper	$216\times138~\mathrm{mm}$	\stockmdemyvo	\pagemdemyvo
msmallroyalvopaper	$234\times156~\mathrm{mm}$	\stockmsmallroyalvo	\pagemsmallroyalvo

Table 1.1: 미터법 표시 클래스 용지 옵션과 그 명령어

옵션	크기	용지 크기 명령	페이지 크기 명령
dbillpaper	$7 \times 3$ in	\stockdbill	\pagedbill
statementpaper	$8.5 \times 5.5 \text{ in}$	\stockstatement	\pagestatement
executivepaper	$10.5 \times 7.25 \text{ in}$	\stockexecutive	\pageexecutive
letterpaper	$11 \times 8.5 \text{ in}$	\stockletter	\pageletter
oldpaper	$12 \times 9 \text{ in}$	\stockold	\pageold
legalpaper	$14 \times 8.5 \text{ in}$	\stocklegal	\pagelegal
ledgerpaper	$17 \times 11 \text{ in}$	\stockledger	\pageledger
broadsheetpaper	$22 \times 17 \text{ in}$	\stockbroadsheet	\pagebroadsheet

Table 1.2: 미국 용지의 클래스 옵션과 그 명령

Table 1.3: 영국 용지의 클래스 옵션과 그 명령

옵션	크기	용지 크기 명령	페이지 크기 명령
pottvopaper	$6.25 \times 4 \text{ in}$	\stockpottvo	\pagepottvo
foolscapvopaper	$6.75 \times 4.25 \text{ in}$	\stockfoolscapvo	\pagefoolscapvo
crownvopaper	$7.5 \times 5 \text{ in}$	\stockcrownvo	\pagecrownvo
postvopaper	$8 \times 5$ in	\stockpostvo	\pagepostvo
largecrownvopaper	$8 \times 5.25  \mathrm{in}$	\stocklargecrownvo	\pagelargecrownvo
largepostvopaper	$8.25 \times 5.25 \text{ in}$	\stocklargepostvo	\pagelargepostvo
smalldemyvopaper	$8.5 \times 5.675  \mathrm{in}$	\stocksmalldemyvo	\pagesmalldemyvo
demyvopaper	$8.75 \times 5.675 \text{ in}$	\stockdemyvo	\pagedemyvo
mediumvopaper	$9 \times 5.75  \mathrm{in}$	\stockmediumvo	\pagemediumvo
smallroyalvopaper	$9.25 \times 6.175 \text{ in}$	\stocksmallroyalvo	\pagesmallroyalvo
royalvopaper	$10 \times 6.25 \text{ in}$	\stockroyalvo	\pageroyalvo
superroyalvopaper	$10.25 \times 6.75 \text{ in}$	\stocksuperroyalvo	\pagesuperroyalvo
imperialvopaper	$11 \times 7.5 \text{ in}$	\stockimperialvo	\pageimperialvo

#### 1.2 활자 크기 옵션

활자 크기 옵션은 문서내 글꼴(font) 크기의 기본값을 설정한다. 이 클래스는 다음과 같이 상당히 많은 활자 크기를 제공한다.

9pt 기본 글꼴 크기 9pt

10pt 기본 글꼴 크기 10pt

11pt 기본 글꼴 크기 11pt

12pt 기본 글꼴 크기 12pt

14pt 기본 글꼴 크기 14pt<sup>1</sup>

17pt 기본 글꼴 크기 17pt

20pt 기본 글꼴 크기 20pt

25pt 기본 글꼴 크기 25pt

<sup>114</sup>pt 옵션의 경우, extrafontsizes 옵션이 없으면, \huge, \Huge, \HUGE가 \LARGE와 같은 크기로 나타난다.

1.2. 활자 크기 옵션

30pt 기본 글꼴 크기 30pt

36pt 기본 글꼴 크기 36pt

48pt 기본 글꼴 크기 48pt

60pt 기본 글꼴 크기 60pt

\*pt 저자가 정하는 기본 글꼴 크기

extrafontsizes 25pt보다 큰 scalable 글꼴을 사용할 수 있음

이는, 14pt 옵션의 경우, \huge, \Huge와 \HUGE를 포함한다. 17pt이상의 경우에는 extrafontsizes를 함께 사용하지 않으면, 에러가 발생한다. 14pt에서는 에러를 내지 않지만, extrafontsizes 옵션이 없으면, \LARGE보다 큰 크기를 나타낼 수 없다.

3

extrafontsizes를 제외하면, 이 옵션들은 서로 배타적이다. 기본 활자 크기는 10pt이다.

17pt 혹은 20pt보다 큰 옵션들은 scalable fonts를 사용하지 않는 이상 그 사용은 제한적이다 — the regular Computer Modern bitmap fonts는 25pt까지만 지원된다. extrafontsizes 옵션을 지정함으로써 25pt를 초과하는 scalable fonts를 사용할 것이라고 선언하는 것이다. 이 옵션은 T1 인코딩 Latin Modern을 기본 글꼴로 설정한다. (일반적으로는 0T1 인코딩 Computer Modern이 기본 글꼴이다.)

#### 1.2.1 확장된 글꼴 크기

기본값을 변경하지 않는 한, extrafontsizes 옵션을 쓰면, 문서의 기본 글꼴은 T1 인코딩 Latin Modern이다. 이는 문서 전처리부(preamble)에 다음을 사용한 것과 같다(extrafontsizes 옵션을 쓰면 불필요한 작업임).

\usepackage{lmodern}\usepackage[T1]{fontenc}

내부적으로 이 클래스는 새로운 글꼴과 인코딩을 특정하기 위해 \memfontfamily와 \memfontenc를 사용하며, \memfontpack을 글꼴 적용을 위한 패키지의 이름으로 사용한다. 내부적으로는 다음과 같이 정의된다.

\providecommand\*{\memfontfamily}{lmr}
\providecommand\*{\memfontenc}{T1}
\providecommand\*{\memfontpack}{lmodern}

이것은 결과적으로 T1 인코딩 1mr(Latin Modern) 글꼴을 기본 글꼴로 설정하며, 이는 Imodern 패키지로 실행된다. 다른 기본 글꼴을 사용하고 싶은 경우, 예를 들어 T1 인코딩 New Century Schoolbook을 사용하고 싶다면 다음과 같은 방법으로 해결할 수 있다.

\newcommand\*{\memfontfamily}{pnc}
\newcommand\*{\memfontpack}{newcent}
\documentclass[...]{memoir}

위 코드에서 \newcommand\*들이 \documentclass *이전*에 위치해 있는데, 이는 이 클래스 코드 내부의 \provide... 정의를 덮어씌울 수 있도록 하기 위함이다.

만일  $^*$ pt 옵션을 사용하는 경우에는 선택한 글꼴의 크기와 스페이스(space)에 관련된 모든 값들을 포함하는 clo 파일을 연결해주어야 한다. 또한 memoir에 그 파일명을 알려주어야 한다. 이를 위해 \documentclass 0전에 다음과 같이 \anyptfilebase와 \anyptsize 두 매크로를 정의해야 한다.

memoir는 글꼴의 크기와 스페이싱(spacing)에 관한 정보가 필요할 때 사용자가 제공한 \anyptfilebase\anyptsize.clo라는 파일에서 찾아 넣는다. \anyptsize ⟨num⟩은 정수여야 한다.² 내부적으로 이 클래스는 다음과 같은 동작을 한다.

\providecommand\*{\anyptfilebase}{mem} \providecommand\*{\anyptsize}{10}

이로써 10pt 글꼴에 해당하는 mem10.clo를 기본값으로 지정하는 것이다. 만약, 예를 들어, 사용자가 18pt 크기의 글꼴 파일을 가지고 있어서 이를 쓰고 싶다면 다음과 같은 방법을 사용할 수 있다.

\newcommand\*{\anyptfilebase}{myfont}
\newcommand\*{\anyptsize}{18}

\documentclass[...\*pt...]{memoir}

이렇게 하면 LaTeX은 사용자가 제공하는 myfont18.clo 파일을 찾아 필요 정보를 입력한다. 사용자의 clo 파일에 어떤 값들이 들어있어야 하는지 알아보고 싶다면, 제공되는 mem10.clo 나 mem60.clo 등을 그 예로 참고하자.

#### 1.3 인쇄 옵션

인쇄 옵션은 다음과 같다.

twoside 종이 양면을 이용하여 인쇄할 문서를 작성한다.

oneside 단면 인쇄용 문서를 작성한다.

twoside와 oneside 옵션은 서로 배타적이다.

onecolumn 한 페이지에 텍스트를 1단으로 출력한다.

twocolumn 한 페이지에 너비(width)가 같은 2단의 텍스트가 출력된다.

onecolumn과 twocolumn 옵션은 서로 배타적이다.

openright 각 장 (chapter)이 펼침면의 오른쪽(홀수쪽)에서 시작된다.

openleft 각 장이 펼침면의 왼쪽(짝수쪽)에서 시작된다.

openany 각 장이 짝수·홀수쪽 어디에서나 시작하도록 한다.

openright, openleft, openany 옵션은 서로 배타적이다.

final 문서의 인쇄용 최종본을 만든다.

draft 행넘침 (overfull) 이 있을 때 검은 선을 그어주고, 몇 가지 변화점을 표시해준다. 어떤 패키지를 사용하면 다른 효과들을 나타낼 수도 있다.

ms 문서가 타자기로 친 것처럼 보이도록 한다. 일부 출판편집자는 정제되지 않은 듯 보이는 원고 상태를 선호하기도 한다.

final, draft, ms 옵션은 서로 배타적이다.

showtrims 인쇄기를 거친 최종본에서 잘라내야할 용지의 여백을 표시하는 트림 마크를 찍어 준다. 인쇄 옵션의 기본값은 twoside, onecolumn, openright, final이다.

 $<sup>^2</sup>$ 정수가 아니라면 TeX이 파일명을 헷갈릴 수 있는데 - 이는 파일명에 단 하나의 온점 (.)을 기대하기 때문이다.

1.4. 기타 옵션 5

#### 1.4 기타 옵션

그 밖에 남아 있는 옵션은 다음과 같다.

legno 수식 (equation) 번호가 왼쪽에 붙는다(기본값은 오른쪽).

fleqn 별행수식(displaystyle math)이 왼쪽 마진에서 \mathindent 만큼 들여쓰기 된다(기본 값은 중앙정렬).

openbib 참고문헌(bibliography) 항목이 새로운 줄에서 시작하며, 두 번째 줄부터 \bibindent만 큼 들여쓰기 된다(기본값은 들여쓰기 없이 계속 이어지는 것).

article article 클래스를 홍내내지만 \chapter 명령을 쓸 수 있다. 이 경우, 기본적으로, \chapter는 \section처럼 동작한다. 장(chapter)이 새 페이지에서 시작되지 않으며 chapter headings는 section headings처럼 식자된다. 그림 따위의 번호는 장이 바뀌어도 재설정되지 않고 연속적으로 매겨진다. 그러나 \part 명령은 여전히 그 자체로 한 페이지에 heading을 표시한다.

oldfontcommands 오래되어 더 이상 쓰지 않는 LaTeX version 2.09 폰트 명령을 사용할 수 있게 한다. old font 명령을 사용할 때마다 경고 메지시를 받게 된다.

fullptlayout 어떤 layout 값(예로, \textwidth)에서 소수점자리 버림을 허용하지 않는다. 기본값으로는 소수점 반올림을 하는데, 이 옵션은 이러한 기능을 무력화한다.

위 옵션은 모두 기본값이 아니다.

#### 1.5 첨언

어떤 옵션도 없이 이 클래스를 부르는 것은 다음과 동일하다.

\documentclass[letterpaper,10pt,twoside,onecolumn,openright,final]{memoir}

이 매뉴얼의 소스파일은 다음과 같이 시작한다.

\documentclass[letterpaper,10pt,extrafontsizes]{memoir}

이는 다소 과한 것인데, letterpaper와 10pt는 이미 기본값이기 때문이다.

문서는 document 환경내의 내용만 조판된다. \documentclass 명령에서 document 환경이전까지의 영역을 전처리부(preamble)라 한다. 필요하다면 이 곳에서 외부 패키지를 불러들이거나 자신의 매크로를 정의한다.

#### \flushbottom \raggedbottom

twoside나 twocolumn 옵션이 주어지면 \flushbottom, 그 외의 경우에는 \raggedbottom으로 조판된다.

\raggedbottom이 선언되면 LaTeX이 조판 영역(type block)의 높이를 일정하게 유지하려는 시도를 하지 않기 때문에 페이지가 조금 부족하게 채워질 수도 있다.

\flushbottom이 선언되면, 사용자가 페이지 나눔을 일부러 사용하지 않는 한, LaTeX은 때 페이지마다 조판 영역의 높이를 일정하게 유지하려고 한다. LaTeX이 페이지 높이를 일정하게 유지하기 위해 사용하는 방법은 수직 간격(예를 들면, 문단과 문단의 사이의 간격이나 headings, 떠다니는 개체(float), 수식과 같이 삽입된 내용의 전후 공간 등)을 늘리거나 줄이는 것이다. 이는 일부의 페이지 컬러에 악영향을 줄 수도 있다.

만약 너무 많은 페이지가 \flushbottom 때문에 늘어난 상태라면 전처리부에\raggedbottom을 사용하는 것이 좋다.

ebook 옵션을 쓸 때는 12pt와 oneside 옵션을 함께 사용하는 편이 좋다.

### Two

### Laying out the page

Up until this chapter the *headings* pagestyle has been used; pagestyles are described in §7.2. This, and later chapters, are typeset with the *ruled* pagestyle.

#### 2.1 Introduction

The class provides a default page layout, in which the page size is the same as the stock size and the typeblock is roughly in the middle of the page. This chapter describes the commands provided by the class to help you produce your own page layout if the default is inappropriate.

If you are happy with the default layout you may skip the rest of this chapter.

The pages of a book carry the text which is intended to educate, entertain and/or amuse the reader. The page must be designed to serve the purposes of the author and to ease the reader's task in assimilating the author's ideas. A good page design is one which the general reader does not notice. If the reader is constantly noticing the page layout, even unconsciously, it distracts from the purpose of the book. It is not the job of the designer to shout, or even to murmur, 'look at my work'.

There are three main parts to a page: the page itself, the typeblock, and the margins separating the typeblock from the edges of the page. Of slightly lesser importance are the running headers and footers, and possibly marginal notes. The art of page design is obtaining a harmonious balance or rhythm between all these.

Although the form is different, the facilities described in this chapter are similar to those provided by the geometry package [Ume99].

Note, if your paper choice matches one of the class paper options, then you can skip forward to section 2.4 – The typeblock, as you have already chosen your stock size and does not need trimming.

On the other hand, if, say, you are designing a document, that is to be printed on one type of paper (the stock), and then trimmed to another, please read on.

#### 2.2 Stock material

Printing is the act of laying symbols onto a piece of stock material. Some print on T-shirts by a process called silk screening, where the shapes of the symbols are made in a screen and then fluid is squeezed through the screen onto the stock material — in this case the fabric of the T shirt. Whether or not this is of general interest it is not the sort of printing or stock

material that is normally used in book production. Books, except for the very particular, are printed on paper.

In the desktop publishing world the stock paper is usually one from a range of standard sizes. In the USA it is typically letterpaper (11 by 8.5 inches) and in the rest of the world A4 paper (297 by 210 mm), with one page per piece of stock. In commercial printing the stock material is much larger with several pages being printed on each stock piece; the stock is then folded, cut and trimmed to form the final pages for binding. The class assumes that desktop publishing is the norm.

#### 2.3 The page

The class assumes that there will be only a single page on a side of each piece of stock; two sides means that there can be two pages, one on the front and the other on the back.

The parameters used by LaTeX itself to define the page layout are illustrated in Figure 2.1. LaTeX does not actually care about the physical size of a page — it assumes that, with respect to the top lefthand corner, the sheet of paper to be printed is infinitely wide and infinitely long. If you happen to have a typeblock that is too wide or too long for the sheet, LaTeX will merrily position text outside the physical boundaries.

The LaTeX parameters are often not particularly convenient if, say, the top of the text must be a certain distance below the top of the page and the fore-edge margin must be twice the spine margin. It is obviously possible to calculate the necessary values for the parameters, but it is not a pleasurable task.

The class provides various means of specifying the page layout, which are hopefully more convenient to use than the standard ones. Various adjustable parameters are used that define the stock size, page size, and so on. These differ in some respects from the parameters in the standard classes, although the parameters for marginal notes are the same in both cases. Figure 2.3 shows the main class layout parameters for a recto page. These may be changed individually by \setlength or by using the commands described below. Figure 2.2 illustrates the same parameters on a verso page.

The first step in designing the page layout is to decide on the page size and then pick an appropriate stock size. Selecting a standard stock size will be cheaper than having to order specially sized stock material.

```
\start
```

The class options provide for some common stock sizes. So if you have specified the class option a4paper and intend to print on A4, then you have no need for \setstocksize, proceed directly to section 2.4. Alternatively, the class provide settings for many other standard stock sizes, see Table 1.1 through Table 1.3.

If you have some other size that you want to use, the command \setstocksize can be used to specify that the stock size is  $\langle height \rangle$  by  $\langle width \rangle$ . For example the following specifies a stock of 9 by 4 inches:

\setstocksize{9in}{4in}

The size of the page must be no larger than the stock but may be smaller which means that after printing the stock must be trimmed down to the size of the page. The page may be positioned anywhere within the bounds of the stock.

The circle is at 1 inch from the top and left of the page. Dashed lines represent (\hoffset + 1 inch) and (\voffset + 1 inch) from the top and left of the page.



Figure 2.1: LaTeX page layout parameters for a recto page



Dashed lines represent the actual page size after trimming the stock.

Figure 2.2: The memoir class page layout parameters for a verso page



Dashed lines represent the actual page size after trimming the stock.

Figure 2.3: The memoir class page layout parameters for a recto page

Table 2.1: Arguments and	results for	\settrimmedsize and	\settypeblocksize

$\langle \text{height} \rangle$	$\langle \mathrm{width} \rangle$	$\langle { m ratio} \rangle$	Result
H	W	r	ambiguous
H	W	*	H,W
${ m H}$	*	$\mathbf{r}$	W = rH
${ m H}$	*	*	ambiguous
*	W	$\mathbf{r}$	H = rW
*	W	*	ambiguous
*	*	$\mathbf{r}$	ambiguous
*	*	*	ambiguous

Page layout should be conceived in terms of a double spread; when you open a book in the middle what you see is a double spread — a verso page on the left and a recto page on the right with the spine between them. Most books when closed are taller than they are wide; this makes them easier to hold when open for reading. A squarish page when opened out into a wide spread makes for discomfort unless the book is supported on a table.

```
\settrimmedsize{\langle height \rangle} {\langle width \rangle} {\langle ratio \rangle}
```

Initially the page size is made the same as the stock size, as set by the paper size option. The command \settrimmedsize can be used to specify the height and width of the page (after any trimming). The  $\langle ratio \rangle$  argument is the amount by which the  $\langle height \rangle$  or the  $\langle width \rangle$  must be multiplied by to give the width or the height. Only two out of the three possible arguments must be given values with the other (unvalued) argument given as \* (an asterisk). The lengths \paperheight and \paperwidth are calculated according to the given arguments. That is, the command enables the \paperheight and \paperwidth to be specified directly or as one being in a given ratio to the other. The potential combinations of arguments and the corresponding results are listed in Table 2.1.

If you have used \setstocksize to redefine the stock, then to get the same page size,

```
\settrimmedsize{\stockheight}{\stockwidth}{*}
```

or for the page dimensions to be 90% of the stock dimensions:

```
\settrimmedsize{0.9\stockheight}{0.9\stockwidth}{*}
```

The following are three different ways of defining an 8 by 5 inch page.

```
\label{eq:settrimmedsize} $$\left\{ \sin \right\} = 0.625 times 8 $$\left\{ \sin \right\} = 1.6 times 5
```

If you look at a well bound hardback book you can see that the sheets are folded so that they are continuous at the spine, where they are sewn together into the binding. The top of the pages should be smooth so that when the book is upright on a bookshelf dust has a harder job seeping between the pages than if the top was all raggedy. Thus, if the stock is trimmed it will be trimmed at the top. It will also have been cut at the fore-edges of the pages and at the bottom, otherwise the book would be unopenable and unreadable.

```
\strims{\langle top \rangle} {\langle foredge \rangle}
```

The command \settrims can be used to specify the amount intended to be removed from the top  $(\langle top \rangle)$  and fore-edge  $(\langle foredge \rangle)$  of the stock material to produce the top and fore-edge of a recto page. Note that the combination of \settrims and \settrimmedsize locate the page with respect to the stock. By default the top and edge trims are zero, which means that if any trimming is required it will be at the spine and bottom edges of the stock unless \settrims is used to alter this.

You can either do any trim calculation for youself or let LaTeX do it for you. For example, with an 8in by 5in page on 10in by 7in stock

```
\settrims{2in}{2in}
```

specifies trimming 2in from the top and fore-edge of the stock giving the desired page size. Taking a design where, say, the page is 90% of the stock size it's easy to get LaTeX to do the calculation:

which will set all the trimming to be at the top and fore-edge. If you wanted, say, equal trims at the top and bottom you could go on and specify

```
\settrims{0.5\trimtop}{\trimedge}
```

See also section 2.11 – *Predefined layouts*, where we present some extra commands. For example \setpagecc for placing a trimmed page centered on the stock.

After trimming the various methods to specify the typeblock, now refers to the *trimmed* page, not the stock material.

### 2.4 The typeblock

Like the page, the typeblock is normally rectangular with the height greater than the width. The lines of text must be laid out so that they are easy to read. Common practice, and more recently psychological testing, has shown that long lines of text are difficult to read. Thus, there is a physiological upper limit to the width of the typeblock. From a practical viewpoint, a line should not be too short because then there is difficulty in justifying the text.

### 2.4.1 A note on the width of the typeblock

Experiments have shown that the number of characters in a line of single column text on a page should be in the range 60 to 70 for ease of reading. The range may be as much as 45 to 75 characters but 66 characters is often considered to be the ideal number. Much shorter and the eye is dashing back and forth between each line. Much longer and it is hard to pick up the start of the next line if the eye has to jump back too far — the same line may be read twice or the following line may be inadvertently jumped over. For double column text the ideal number of characters is around 45, give or take 5 or so.

Bringhurst [Bri99] gives a method for determining the number of characters in a line for any font: measure the length of the lowercase alphabet and use a copyfitting table that shows for a given alphabet length and line length, the average number of characters in

Table 2.2: Average characters per line

Pts.		Line length in picas						
	10	14	18	22	26	30	35	40
80	40	56	72	88	104			
85	38	53	68	83	98	113		
90	36	50	64	79	86	107		
95	34	48	62	75	89	103		
100	33	46	59	73	86	99	116	
105	32	44	57	70	82	95	111	
110	30	43	55	67	79	92	107	
115	29	41	53	64	76	88	103	
120	28	39	50	62	73	84	98	112
125	27	38	48	59	70	81	94	108
130	26	36	47	57	67	78	91	104
135	25	35	45	55	65	75	88	100
140	24	34	44	53	63	73	85	97
145	23	33	42	51	61	70	82	94
150	23	32	41	51	60	69	81	92
155	22	31	39	49	58	67	79	90
160	22	30	39	48	56	65	76	87
165	21	30	38	46	55	63	74	84
170	21	29	37	45	53	62	72	82
175	20	28	36	44	52	60	70	80
180	20	27	35	43	51	59	68	78
185	19	27	34	42	49	57	67	76
190	19	26	33	41	48	56	65	74
195	18	25	32	40	47	54	63	72
200	18	25	32	39	46	53	62	70
220	16	22	29	35	41	48	56	64
240	15	20	26	32	38	44	51	58
260	14	19	24	30	35	41	48	54
280	13	18	23	28	33	38	44	50
300	12	17	21	26	31	35	41	47
320	11	16	20	25	29	34	39	45
340	10	15	19	23	27	32	37	42

that line. Table 2.2 is an abridged version of Bringhurt's copyfitting table. For example, it suggests that a font with a length of 130pt should be set on a measure of about 26pc for a single column or in an 18pc wide column if there are multiple columns.

Morten Høgholm has done some curve fitting to the data. He determined that the expressions

$$L_{65} = 2.042\alpha + 33.41 \tag{2.1}$$

and

$$L_{45} = 1.415\alpha + 23.03 \tag{2.2}$$

fitted aspects of the data, where  $\alpha$  is the length of the alphabet in points, and  $L_i$  is the suggested width in points, for a line with i characters (remember that 1pc = 12pt).

Table 2.3 gives the lowercase alphabet lengths for some typefaces over a range of font sizes; this may be used in conjunction with Table 2.2 on page 14 when deciding on an appropriate textwidth. I have grouped the listed typefaces into roman, sans-serif, and monospaced, and they are all available in a standard LaTeX system. The Computer Modern Roman, Concrete Roman, Computer Sans, and Typewriter typefaces were all designed by Donald Knuth using Metafont, specifically for use with TeX. The other font families are PostScript outline fonts and can be used in many document publishing systems. These particular fonts are available for use in LaTeX via the packages in the psnfss bundle. Be aware that the Knuthian fonts were designed to form a font family — that is, they were designed to work together and complement each other — while the listed PostScript fonts were designed by different people at different times and for different purposes. Bringhurst [Bri99, p. 96] memorably says 'Baskerville, Helvetica, Palatino and Times Roman, for example — which are four of the most widely available typefaces — are four faces with nothing to offer one another except public disagreement'.

The monospaced fonts, Courier and Typewriter have no place in high quality typesetting except when typesetting computer code or the like, or when trying to fake text written on a real typewriter. Ignoring these, a quick glance at the Table shows that Bookman is a broad font while Times is narrow as befits its original design intent for typesetting narrow columns in newspapers. Computer Modern tends towards the narrow end of the range.

```
\xlvchars \lxvchars
```

Based on Table 2.3, the two lengths \xlvchars and \lxvchars are initially set to approximately the lengths of a line of text with 45 or 65 characters, respectively, for Computer Modern Roman in the type size selected for the document.

If you are using a different font or size you can use something like the following to calculate and print out the length for you.

The \typeout macro prints its argument to the terminal and the log file. There is, however, an easier method.

```
\setxlvchars[\langle fontspec \rangle]
\setlxvchars[\langle fontspec \rangle]
```

The macros \setxlvchars and \setlxvchars, which were suggested by Morten Høgholm, set the lengths \xlvchars and \lxvchars respectively for the font  $\langle fontspec \rangle$ . The default for  $\langle fontspec \rangle$  is \normalfont. For example, the values of \lxvchars and \xlvchars after calling:

	8pt	9pt	10pt	11pt	12pt	14pt	17pt	20pt
Bookman	113	127	142	155	170	204	245	294
Charter	102	115	127	139	152	184	221	264
Computer Modern	108	118	127	139	149	180	202	242
Concrete Roman	109	119	128	140	154	185	222	266
New Century Schoolbook	108	122	136	149	162	194	234	281
Palatino	107	120	133	146	160	192	230	276
Times Roman	96	108	120	131	143	172	206	247
Utopia	107	120	134	146	161	193	232	277
Avant Garde Gothic	113	127	142	155	169	203	243	293
Computer Sans	102	110	120	131	140	168	193	233
Helvetica	102	114	127	139	152	184	220	264
Courier	125	140	156	170	187	224	270	324
Typewriter	110	122	137	149	161	192	232	277

Table 2.3: Lowercase alphabet lengths, in points, for various fonts

\setlxvchars \setxlvchars[\small\sffamily]

are: \lxvchars = 292.73834pt, and \xlvchars = 179.59335pt. Morten Høgholm<sup>1</sup> also commented:

 $\,^{\circ}$ I was defining some environments that had to have \parindent as their indentation. For some reason I just wrote 1.5em instead of \parindent because I 'knew' that was the value. What I had overlooked was that I had loaded the mathpazo package [Pug02], thus, among other things, altering \parindent. Conclusion: the environment would use 1.5em = 18.0pt, whereas the \parindent was only 17.607pt.

This, and other related situations can be avoided if one places \RequirePackage{\( font-package \) \\ \normalfont

before \documentclass.

Note that, in general, it is inadviseable to put any commands before \documentclass.

### 2.4.2 Specifying the typeblock size

The height of the typeblock should be equivalent to an integral number of lines. The class provides two different methods to specify the typeblock. \settypeblocksize (below) will set the size of the typeblock, other commands will then be used to place it on the paper. Alternatively the typeblock size can be determined by setting the margins around it. We present this later on in this section (see \setlrmarginsandblock and \setulmarginsandblock).

The command \settypeblocksize is similar to \settrimmedsize except that it sets the \textheight and \textwidth for the typeblock. The potential combinations of arguments

<sup>&</sup>lt;sup>1</sup>Private communication

and the corresponding results are listed in Table 2.1 on page 12. For instance, here are three ways of specifying a 6in by 3in typeblock:

```
\settypeblocksize{6in}{3in}{*}
\settypeblocksize{6in}{*}{0.5}
\settypeblocksize{*}{3in}{2}
```

The typeblock has to be located on the page. There is a relationship between the page, typeblock and margins. The sum of the spine, or inner, margin, the fore-edge, or outer, margin and the width of the typeblock must equal the width of the page. Similarly the sum of the upper margin, the lower margin and the height of the typeblock must equal the height of the page. The process of locating the typeblock with respect to the page can be viewed either as positioning the typeblock with respect to the edges of the page or as setting the margins between the page and the typeblock.

Remembering that the page layout should be defined in terms of the appearance as a spread, the spine margin is normally half the fore-edge margin, so that the white space is equally distributed around the sides of the text.

**Note.** One will often find that using \settypeblocksize without subsequent use of \setlrmargins and \setulmargins will result in errors as the relationships mentioned above are not met (the \textwidth has changed, but not the margins).

We may add auto adjustments to a future version of memoir.

There is more latitude in choosing the proportions of the upper and lower margins, though usually the upper margin is less than the lower margin so the typeblock is not vertically centered.

Two methods are provided for setting the horizontal dimensions on a page. One is where the width of the typeblock is fixed and the margins are adjustable. The other method is where the size of the margins determines the width of the typeblock.

```
\label{eq:setlemargins} $$ \operatorname{spine} {\langle edge \rangle} {\langle ratio \rangle} $$
```

The command \setlrmargins can be used to specify the side margins<sup>2</sup> with the width of the page and the typeblock being fixed.

Not more than one one argument value is required, with any unvalued arguments being denoted by an asterisk. There are several cases to consider and these are tabulated in Table 2.4.

In the Table, S is the calculated spine margin, E is the calculated fore-edge margin, and  $P_w$  and  $B_w$  are respectively the page and typeblock widths. The \setlrmargins command maintains the relationship

$$S + E = K_w = \text{constant} \ (= P_w - B_w).$$

The cases marked ambiguous in the Table are where the particular combination of argument values may make it impossible to guarantee the relationship.

Assuming that we have a 3in wide typeblock on a 5in wide page and we want the spine margin to be 0.8in and the fore-edge margin to be 1.2in (i.e., the fore-edge margin is half as big again as the spine margin) this can be accomplished in three ways (with the \paperwidth and \textwidth being previously specified and fixed):

<sup>&</sup>lt;sup>2</sup>Only the spine margin is noted in Figure 2.3 and 2.2; the fore-edge margin is at the opposite side of the typeblock.

Table 2.4: Arguments and results for \setlrmargins

$\langle \text{spine} \rangle$	$\langle \mathrm{edge} \rangle$	$\langle {\rm ratio} \rangle$	Result
S	E	r	ambiguous
$\mathbf{S}$	$\mathbf{E}$	*	ambiguous
$\mathbf{S}$	*	r	ambiguous
$\mathbf{S}$	*	*	$E = K_w - S$
*	$\mathbf{E}$	r	ambiguous
*	$\mathbf{E}$	*	$S = K_w - E$
*	*	r	$E + S = K_w, E = rS$
*	*	*	$E + S = K_w, E = S$

Table 2.5: Arguments and results for \setlrmarginsandblock

$\langle \text{spine} \rangle$	$\langle \mathrm{edge} \rangle$	$\langle {\rm ratio} \rangle$	Result
S	E	r	S, E
$\mathbf{S}$	$\mathbf{E}$	*	S, E
$\mathbf{S}$	*	r	E = rS
$\mathbf{S}$	*	*	E = S
*	$\mathbf{E}$	r	S = rE
*	${ m E}$	*	S = E
*	*	r	${ m ambiguous}$
*	*	*	${ m ambiguous}$

```
% specify spine margin
   \setlrmargins{0.8in}{*}{*}
% specify fore-edge margin
   \setlrmargins{*}{1.2in}{*}
% specify fore-edge/spine ratio
   \setlrmargins{*}{*}{1.5}
```

```
\structure{$\langle spine \rangle } {\langle edge \rangle } {\langle ratio \rangle }
```

The command \setlrmarginsandblock can be used to specify the spine and fore-edge margins, where the page width is fixed and the width of the typeblock depends on the margins. Results for this command are given in Table 2.5. The same notation is used, but in this case \setlrmarginsandblock maintains the relationship

$$S + B_w + E =$$
constant  $(= P_w)$ .

The width of the typeblock is calculated from  $B_w = P_w - S - E$ .

Assuming that we want a 3in wide typeblock on a 5in wide page and we want the spine margin to be 0.8in and the fore-edge margin to be 1.2in (i.e., the fore-edge margin is half as big again as the spine margin) this can be accomplished in the following ways (with the \textwidth being calculated from the previously specified \paperwidth and the specified margins):

```
% specify both margins
   \setlrmarginsandblock{0.8in}{1.2in}{*}
% specify spine & fore-edge/spine ratio
   \setlrmarginsandblock{0.8in}{*}{1.5}
% specify fore-edge & spine/fore-edge ratio
   \setlrmarginsandblock{*}{1.2in}{0.667}
```

If we wanted the margins to be both 1in instead then any of the following will do it:

```
% specify both margins
   \setlrmarginsandblock{1in}{1in}{*}
% specify spine & fore-edge/spine ratio
   \setlrmarginsandblock{1in}{*}{1}
% specify spine (fore-edge = spine)
   \setlrmarginsandblock{1in}{*}{*}
% specify fore-edge & spine/fore-edge ratio
   \setlrmarginsandblock{*}{1in}{1}
% specify fore-edge (spine = fore-edge)
   \setlrmarginsandblock{*}{1in}{*}
```

```
\setbinding\{\langle length \rangle\}
```

In some cases, for example when doing a Japanese stab binding, it may be desireable to add a small allowance to the spine margin for the binding. You can use the command \setbinding for this purpose. It decreases the effective page width by  $\langle length \rangle$  and later this length will be added on to the spine margin, thus restoring the page width to its original value. If you use \setbinding than it must be *after* setting the page width and *before* setting the spine and fore-edge margins.

That completes the methods for specifying the horizontal spacings. There are similar commands for setting the vertical spacings which are described below.

```
\operatorname{setulmargins}(\operatorname{upper}) + (\operatorname{lower}) + (\operatorname{ratio})
```

The command \setulmargins can be used to specify the upper and lower margins<sup>3</sup> where the heights of the page and the typeblock are fixed. This is similar to \setlrmargins. Using a slightly different notation this time, with U being the upper margin, L being the lower margin, and  $P_h$  and  $B_h$  being the height of the page and typeblock, respectively, the results are shown in Table 2.6. The \setulmargins command maintains the relationship

$$U + L = K_h = \text{constant} (= P_h - B_h).$$

Note that in terms of the traditional LaTeX parameters memoir's  $\displaystyle \frac{1}{\sqrt{1+|x|}}$  is  $\displaystyle \frac{1}{\sqrt{1+|x|}}$ .

```
\stulmarginsandblock{\langle upper \rangle} {\langle lower \rangle} {\langle ratio \rangle}
```

The command \setulmarginsandblock can be used to specify the upper and lower margins, where the page height is fixed and the height of the typeblock depends on the margins.

<sup>&</sup>lt;sup>3</sup>Only the upper margin is noted in Figure 2.3 and 2.2; the lower margin is the distance between the bottom of the typeblock and the bottom of the page.

Table 2.6: Arguments and results for \setulmargins

$\langle \text{upper} \rangle$	$\langle lower \rangle$	$\langle { m ratio} \rangle$	Result
U	L	r	ambiguous
U	${ m L}$	*	ambiguous
U	*	r	ambiguous
U	*	*	$L = K_h - U$
*	${ m L}$	r	ambiguous
*	${ m L}$	*	$U = K_h - L$
*	*	r	$L+U=K_h, L=rU$
*	*	*	$L+U=K_h, L=U$

Table 2.7: Arguments and results for \setulmarginsandblock

$\langle \mathrm{upper} \rangle$	$\langle lower \rangle$	$\langle {\rm ratio} \rangle$	Result
U	L	r	U, L
U	L	*	U, L
U	*	r	L = rU
U	*	*	L = U
*	L	r	U = rL
*	L	*	U = L
*	*	$\mathbf{r}$	${ m ambiguous}$
*	*	*	ambiguous

Results for this command are given in Table 2.7. The same notation is used, but in this case \setulmarginsandblock maintains the relationship

$$U + B_h + L = \text{constant } (P_h).$$

The height of the typeblock is calculated from  $B_h = P_h - U - L$ .

*Note.* Readers may find several folio designs in [Wil09d].

$$\verb|\setcolsepandrule{| \langle colsep \rangle}{ \langle thickness \rangle}|$$

For twocolumn text the width of the gutter between the columns must be specified. LaTeX also lets you draw a vertical rule in the middle of the gutter. The macro \setcolsepandrule sets the gutter width, \columnsep, to  $\langle colsep \rangle$  and the thickness of the rule, \columnseprule, to  $\langle thickness \rangle$ . A  $\langle thickness \rangle$  of 0pt means that the rule will be invisible. Visible rules usually have a thickness of about 0.4pt. The total width of the twocolumns of text and the gutter equals the width of the typeblock.

This completes the methods for specifying the layout of the main elements of the page — the page size, the size of the typeblock and the margins surrounding the typeblock.

$\overline{\langle \text{headdrop} \rangle}$	$\langle \text{headsep} \rangle$	(ratio)	Result
	$H_s$	r	ambiguous
D	$H_s$	*	ambiguous
D	*	r	ambiguous
D	*	*	$H_s = C_h - D$
*	$H_s$	r	ambiguous
*	$H_s$	*	$D = C_h - H_s$
*	*	r	$H_s + D = C_h, H_s = rD$
*	*	*	$H_s + D = C_h, H_s = D$

Table 2.8: Arguments and results for \setheaderspaces

### 2.5 Headers, footers and marginal notes

A page may have two additional items, and usually has at least one of these. They are the running header and running footer. If the page has a folio then it is located either in the header or in the footer. The word 'in' is used rather lightly here as the folio may not be actually *in* the header or footer but is always located at some constant relative position. A common position for the folio is towards the fore-edge of the page, either in the header or the footer. This makes it easy to spot when thumbing through the book. It may be placed at the center of the footer, but unless you want to really annoy the reader do not place it near the spine.

Often a page header contains the current chapter title, with perhaps a section title on the opposite header, as aids to the reader in navigating around the book. Some books put the book title into one of the headers, usually the verso one, but I see little point in that as presumably the reader knows which particular book he is reading, and the space would be better used providing more useful signposts.

$$\setheadfoot{\langle headheight \rangle}{\langle footskip \rangle}$$

The \setheadfoot macro sets the \headheight parameter to the value  $\langle headheight \rangle$  and the \footskip parameter to  $\langle footskip \rangle$ . It is usual to set the \headheight to at least the value of the \baselineskip of the normal body font.

$$\setheaderspaces{\langle headdrop \rangle} {\langle headsep \rangle} {\langle ratio \rangle}$$

The command \setheaderspaces is similar to \setulmargins. Using the notation U for the upper margin (as before),  $H_h$  for the \headheight,  $H_s$  for the \headsep and D for the \headdrop, where the \headdrop is the distance between the top of the trimmed page and the top of the header<sup>4</sup>, then the macro \setheaderspaces maintains the relationship

$$D + H_s = C_h = \text{constant} (= U - H_h).$$

The macro \setheaderspaces is for specifying the spacing above and below the page header. The possible arguments and results are shown in Table 2.8. Typically, the \headsep is of more interest than the \headdrop.

Finally, some works have marginal notes. These really come last in the design scheme, providing the margins have been made big enough to accommodate them. Figure 2.2 shows

<sup>&</sup>lt;sup>4</sup>The head drop is not shown in Figure 2.3 or 2.2.

the marginal note parameters on a verso page, and also illustrates that some parameters control different positions on the stock.

```
\setmarginnotes{\langle separation \rangle} {\langle width \rangle} {\langle push \rangle}
```

The command \setmarginnotes sets the parameters for marginal notes. The distance \marginparsep between the typeblock and any note is set to  $\langle separation \rangle$ , the maximum width for a note, \marginparwidth, is set to  $\langle width \rangle$  and the minimum vertical distance between notes, \marginparpush, is set to  $\langle push \rangle$ .

**Note.** As of memoir v3.6k, we have added an auto adjustment feature for \marginparwidth, such that unless \setmarginnotes have been used to make a specific choice, the \marginparwidth is chosen according to the algorithm described in Section G.1.1. The algorithm relies on \marginparmargin (if used) being set before \checkandfixthelayout.

We may add other auto adjusting features to future memoir releases.

### 2.6 Other

```
\stfootins{\langle length for normal \rangle}{\langle length for minipage \rangle}
```

When footnotes are added to the text block they are added \skip\footins below the text. Since this is a skip it usually needs special syntax to change it. Instead we have provided an interface to set it.<sup>5</sup> The default sizes are \bigskipamount.

### 2.7 Putting it together

The page layout parameters just discussed are not always the same as those that are expected by LaTeX, or by LaTeX packages. The parameters that LaTeX expects are shown in Figure 2.1. You can either use the class commands for changing the page layout or change the LaTeX parameters directly using either \setlength or \addtolength applied to the parameter(s) to be modified. If you choose the latter route, those class parameters which differ from the standard LaTeX parameters will *not* be modified.

The general process of setting up your own page layout is along these lines:

- Decide on the required finished page size, and pick a stock size that is at least as large as the page.
- If needed, use \setstocksize to set the values of \stockheight and \stockwidth, followed by \settrimmedsize to specify the values of \paperheight and \paperwidth.
  - If you use and print on, say, A4, the a4paper class option is enough, no  $\start$  needed.
- Decide on the location of the page with respect to the stock. If the page and stock sizes are the same, then call \settrims{0pt}{0pt}. If they are not the same size then decide on the appropriate values for \trimtop and \trimedge to position the page on the stock, and then set these through \settrims.
- Decide on the size of the typeblock and use \settypeblocksize to specify the values of \textheight and \textwidth.

 $<sup>^5</sup>$ This interface also sets the equivalent lengths used when  $\t\infty$  and friends are being used.

- If you need a binding allowance, now is the time for \setbinding.
- Pick the value for the spine margin, and use \setlrmargins to set the values for the \spinemargin and \foremargin.
  - An alternative sequence is to use \setlrmarginsandblock to set the \textwidth for a particular choice of side margins.
- Pick the value for the upper margin and use \setulmargins to set the values for the \uppermargin and \lowermargin.
  - An alternative sequence is to use \setulmarginsandblock to set the \textheight for a particular choice of upper and lower margins.
  - The typeblock is now located on the page.
- Pick the values for the \headheight and \footskip and use \setheadfoot to specify these.
- Pick your value for \headskip and use \setheaderspaces to set this and \headmargin.
- If you are going to have any marginal notes, use \setmarginnotes to specify the values for \marginparsep, \marginparwidth and \marginparpush.

You can plan and specify your layout in any order that is convenient to you. Each of the page layout commands is independent of the others; also if a value is set at one point, say the \textwidth, and is then later changed in some way, only the last of the settings is used as the actual value.

Comparing Figures 2.3 and 2.1 you can see the different layout parameters provided by the class and used by standard LaTeX. For convenience, and because the figures do not show all the parameters, the two sets of parameters are listed in Table 2.9.

Unless you are satisfied with the default page layout, after specifying the layout that you want you have to call the \checkandfixthelayout command to finally implement your specification.

The \checkandfixthelayout macro uses \checkthelayout to check the page layout specification you have given, and then calls \fixthelayout to finally implement it.

The \checkthelayout macro checks the class layout parameters to see whether they have 'sensible' values (e.g., the \textwidth is not negative) and, depending on the  $\langle algorithm \rangle$  argument, it may modify the \textheight. It does not actually implement the layout.

When using \flushbottom LaTeX expects that the \textheight is such that an integral number of text lines in the body font will fit exactly into the height. If not, then it issues 'underfull vbox' messages. More precisely, if b is the \baselineskip and t is the \topskip, N is an integer (the number of lines in the typeblock), and T is the \textheight then to avoid underfull vboxes the following relationship must hold

$$T = (N-1)b + t (2.3)$$

Table 2.9: The class and LaTeX page layout parameters

Class	LaTeX
\stockheight	
\trimtop	
\trimedge	
\stockwidth	
\paperheight	\paperheight
\paperwidth	\paperwidth
\textheight	\textheight
\textwidth	\textwidth
\columnsep	\columnsep
\columnseprule	\columnseprule
\spinemargin	
\foremargin	
	$\odsidemargin$
	\evensidemargin
\uppermargin	
$\headmargin$	
	$ ag{topmargin}$
$\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $	$\headheight$
\headsep	\headsep
\footskip	$\footskip$
$\mbox{\tt marginparsep}$	\marginparsep
$\mbox{\mbox{\tt marginparwidth}}$	$\mbox{\tt marginparwidth}$
\marginparpush	\marginparpush

By default \checkthelayout ensures that the final \textheight meets this criterion. The optional  $\langle algorithm \rangle$  argument lets you control just how it does this. In the following H is your requested value for the \textheight and the other symbols are as before, with T as the adjusted value, and using integer arithmetic. The permissible values for  $\langle algorithm \rangle$  are:

fixed The \textheight is not altered.

$$T = H (2.4)$$

If you use this option you may find that underfull vboxes are reported for \flushbottom pages.

classic This is the default and is the one used by the standard classes.

$$T = b|H/b| + t \tag{2.5}$$

The relationship (2.3) is maintained. This algorithm gets as close to H as possible from below.

<sup>&</sup>lt;sup>6</sup>In this context 'integer arithmetic' means that the result of a division will be rounded down. For example 99/10 in 'real arithmetic' results in 9.9, whereas with integer arithmetic the result is 9, not 10.

	Algorithm					
	fixed	classic	lines	nearest		
Requested height	ac	djusted height	in pts, (lines)			
10.0\baselineskip	120.0pt, (10)	130pt, (11)	118pt, (10)	118pt, (10)		
$10.2 \text{\colored}$	122.4pt, (10)	130pt, (11)	118pt, (10)	118pt, (10)		
$10.4 \$ baselineskip	124.8pt, (10)	130pt, (11)	118pt, (10)	130pt, (11)		
$10.6 \text{\colored}$	127.2pt, (10)	130pt, (11)	118pt, (10)	130pt, (11)		
$10.8$ \baselineskip	129.6pt, (10)	130pt, (11)	118pt, (10)	130pt, (11)		
$11.0 \$ baselineskip	132.0pt, (11)	142pt, (12)	130pt, (11)	130pt, (11)		

Table 2.10: Results from sample \textheight adjustments

lines This is similar to classic, but results in a smaller final value.

$$T = b|(H - b)/b| + t (2.6)$$

The relationship (2.3) is maintained.

nearest The calculated value is the nearest to the given value while still maintaining the relationship (2.3).

$$T = b\lfloor (H - t + b/2)/b \rfloor + t \tag{2.7}$$

In contrast to classic, nearest will get as close to H as possible even if this means that T ends up being slightly larger than H.

Table 2.10 shows the results from the various \textheight adjustment calculations where the \baselineskip is 12pt and the \topskip is 10pt, which are the normal values for a Computer Modern 10pt font. In all cases the fixed algorithm resulted in underfull vboxes. If you know the number of lines that you want, say 42, then requesting

% setting equivalent to \setlength{\textheight}{42\baselineskip} \checkandfixthelayout[lines]

will result in the most appropriate \textheight.

If you use the calc package you can use constructs like the following in a page layout specification:

\setlength{\textheight}{41\baselineskip + \topskip} \settypeblocksize{41\baselineskip + \topskip}{33pc}{\*}

The \fixthelayout macro finally implements the layout, making due adjustement for any binding allowance, and calculates the values for all the standard LaTeX layout parameters (such that packages can use these expected values). If you have used the class macros to change the layout in any way, you must call \checkandfixthelayout after you have made all the necessary changes. As an aid, the final layout parameter values are displayed on the terminal and written out to the log file.

 $<sup>^7</sup>$ For comparison the optimum heights from equation 2.3 for 10, 11 and 12 lines are respectively 118pt, 130pt and 142pt.

```
\t typeoutlayout \\ typeoutstandardlayout \\ settypeoutlayoutunit {\langle unit \rangle}
```

\typeoutlayout writes the current class layout parameter values to the terminal and the log file. It is called by \checkandfixthelayout but you can use it yourself at any time. The macro \typeoutstandardlayout writes the standard layout parameter values to the terminal and log file so that you can compare the two sets of parameter values.

By using the macro \settypeoutlayoutunit, the user can change the unit in which the layout list is shown. Very handy when designing in, say, centimeters. Supported units are pt, pc, mm, cm, in, bp, dd and cc, default being pt, see Table ?? for more information about the units.

### 2.8 Side margins

In twoside printing the spine margin is normally the same on both recto and verso pages and, unless the spine and fore-edge margins are the same, the typeblock is shifted side to side when printing the recto and verso pages. Additionally you can have different headers and footers for the recto and verso pages. However, in oneside printing the typeblock is not moved and the headers and footers are the same for both odd and even pages.

Some documents are designed to have, say, a very wide righthand margin in which to put illustrations; this leads to needing the spine margin on verso pages to be much larger than the spine margin on recto pages. This can be done with the oneside option. However, different headers and footers are required for the recto and verso pages, which can only be done with the twoside option. The way to get the desired effects is like this (twoside is the default class option):

```
\documentclass{memoir}
%%% set up the recto page layout
\checkandfixthelayout% or perhaps \checkandfixthelayout[lines]
\setlength{\evensidemargin}{\oddsidemargin}% after \checkandfix...
```

### 2.9 Emitting the page size

At the start of the document the class will automatically emit the chosen stocksize to the output format (DVI or PDF).<sup>8</sup>

### 2.10 Example

Suppose you want a page that will fit on both A4 and US letterpaper stock, wanting to do the least amount of trimming. The layout requirements are as follows.

• The width of the typeblock should be such that there are the optimum number of characters per line.

<sup>&</sup>lt;sup>8</sup>In earlier versions we had macros \fixpdflayout and \fixdvipslayout that held some of the code needed to emit this data. With the newer engines like LuaLaTeX, this became unfeasible thus they we discontinued (their use now emit a warning).

- The ratio of the height to the width of the typeblock should equal the golden section.
- The text has to start 50pt below the top of the page. We will use the default \headheight and \footskip.
- The ratio of the outer margin to the inner margin should equal the golden section, as should the space above and below the header.
- There is no interest at all in marginal notes, so we can ignore any settings for these.

We can either do the maths ourselves or get LaTeX to do it for us. Let's use LaTeX. First we will work out the size of the largest sheet that can be cut from A4 and letterpaper, whose sizes are  $297 \times 210$  mm and  $11 \times 8.5$  in; A4 is taller and narrower than letterpaper.

```
\settrimmedsize{11in}{210mm}{*}
```

The stocksize is defined by the class option, which could be either letterpaper or a4paper, but we have to work out the trims to reduce the stock to the page. To make life easier, we will only trim the fore-edge and the bottom of the stock, so the \trimtop is zero. The \trimtop and \trimedge are easily specified by

```
\setlength{\trimtop}{Opt}
\setlength{\trimedge}{\stockwidth}
\addtolength{\trimedge}{-\paperwidth}
```

Or if you are using the calc package, perhaps:

```
\settrims{Opt}{\stockwidth - \paperwidth}
```

Specification of the size of the typeblock is also easy

```
\settypeblocksize{*}{\lxvchars}{1.618}
```

and now the upper and lower margins are specified by

```
\setulmargins{50pt}{*}{*}
```

The spine and fore-edge margins are specified just by the value of the golden section, via

```
\setlrmargins{*}{*}{1.618}
```

The only remaining calculation to be done is the \headmargin and \headsep. Again this just involves using a ratio

```
\setheaderspaces{*}{*}{1.618}
```

To finish off we have to make sure that the layout is changed

```
\checkandfixthelayout
```

### 2.10.1 The page layout of this manual

The page layout for this manual is defined in the preamble as:

```
\settrimmedsize{11in}{210mm}{*}
\setlength{\trimtop}{0pt}
\setlength{\trimedge}{\stockwidth}
\addtolength{\trimedge}{-\paperwidth}
\settypeblocksize{7.75in}{33pc}{*}
\setulmargins{4cm}{*}{*}
\setlrmargins{1.25in}{*}{*}
```



Dashed lines represent the actual page size after trimming the stock.



Figure 2.4: The recto page layout for this manual

```
\setmarginnotes{17pt}{51pt}{\onelineskip}
\setheadfoot{\onelineskip}{2\onelineskip}
\setheaderspaces{*}{2\onelineskip}{*}
\checkandfixthelayout
```

An illustration of the layout is shown in Figure 2.4 which also lists the parameter values resulting from the code above, to the nearest point.

I initially used the layout defined in §2.10, which I thought looked reasonable, but then I decided to use the one above in order to save paper when anyone printed out the manual. The wider typeblock also makes it easier for TeX when dealing with lines that include unhyphenatable text, like the LaTeX code.

Andreas Mathias, via Rolf Niepraschk, has suggested that the following might be better for typesetting the manual on A4 paper.

```
\documentclass[a4paper]{memoir}
...
\settrimmedsize{297mm}{210mm}{*}
\setlength{\trimtop}{0pt}
\setlength{\trimedge}{\stockwidth}
\addtolength{\trimedge}{-\paperwidth}
\settypeblocksize{634pt}{448.13pt}{*}
\setulmargins{4cm}{*}{*}
\setlrmargins{*}{*}{1.5}
\setmarginnotes{17pt}{51pt}{\onelineskip}
\setheadfoot{\onelineskip}{2\onelineskip}
\setheaderspaces{*}{2\onelineskip}{*}
\checkandfixthelayout
```

However, the layout that I have provided will print on both letterpaper and A4 sized stock even if it might look better if it was designed for always being printed on one or the other.

### 2.11 Predefined layouts

The class, like the standard classes, will automatically produce working layouts for letter-paper and a4paper size options. They might be a bit problematic, though, when the page is much smaller, particularly with respect to the space alloted for marginal notes. You perhaps will find the layouts package [Wil03a] useful for checking the page layout.

Some non-default layouts are provided via the commands \medievalpage, \isopage and \semiisopage; these set the size and position of the typeblock with respect to the page. After using any of these commands you *must* call \checkandfixthelayout (and after having made any other changes to match the new layout).

```
\mbox{\em medievalpage}[\langle spine \rangle]
```

The \medievalpage command generates the position and size of the typeblock according to the principles of medieval scribes and the early printers, as discovered and described by Jan Tschichold [Tsc91]. The basic principle is that the spine, top, fore-edge and bottom margins

 $<sup>^9</sup>$ Email from niepraschk@ptb.de on 2002/02/05.



Figure 2.5: Default layout for letterpaper



Figure 2.7: Letterpaper layout: Left \isopage, Right \isopage[12]



Figure 2.8: Letterpaper layout: Left \semiisopage, Right \semiisopage[12]



Figure 2.9: Default layout for a4paper



Figure 2.10: A4paper layout: Left \medievalpage, Right \medievalpage[12]



Figure 2.11: A4paper layout: Left \isopage, Right \isopage[12]



Figure 2.12: A4paper layout: Left \semiisopage, Right \semiisopage[12]

around the typeblock are in the ratio 2:3:4:6. Typically the spine margin was 1/9 the width of the page, which is what \medievalpage assumes by default. This can be changed with the optional  $\langle spine \rangle$  argument. For example, to get narrower margins with the spine being 1/12 the page width:

```
\medievalpage[12]
```

Note that  $\langle spine \rangle$  must be an integer.

Robert Bringhurst [Bri99] presented a page layout that was especially suitable for ISO proportioned paper, although it can be applied to any page proportion. The \isopage command implements this design. The spine margin is normally 1/9 the page width and the top margin is 1/9 the page height, and the fore-edge and bottom margins are respectively twice the spine and top margins.

The \semiisopage layout is similar where the spine margin by default is 1/9 the page width, but the top margin is the same as the spine margin. Again the fore-edge and bottom margins are respectively twice the spine and top margins.

The size of the spine (and top) margins can be changed by using the optional  $\langle spine \rangle$  argument, which must be an integer. To set the spine margin to be, for example, 1/8 of the page width:

```
\semiisopage[8]% or \isopage[8]
```

The same factor applies to both the spine and top margins in the case of \isopage.

Spreads showing a variety of these layouts are in Figure 2.5 through 2.12. These were produced with the aid of the layouts package.

```
\label{eq:local_set_pagebl} $$ \left( \langle height \rangle \right) \left( \langle width \rangle
```

When your page is smaller than the stock it has to be positioned on the stock by specifying the trims to give the desired size and location. The macro \setpagebl, which takes the same arguments as \settrimmedsize (see Table 2.1 on page 12), calculates the trims so that the page is located at the bottom left of the stock. Similarly \setpageml and \setpagetl will locate the page at the middle left and the top left, respectively, of the stock. For instance, if you are using letterpaper stock but want the final trimmed page size to be A5, then this will put page area at the bottom left of the stock.

```
\pagebv % sets page height and width for A5 paper
\setpagebl{\paperheight}{\paperwidth}{*}
...
\checkandfixthelayout
```

The above macros position the page at the left of the stock because usually trimming of the stock is limited to the top, right, and bottom, the left being the spine when the pages are finally assembled. To reposition the page to the center of the stock the following will halve the top and edge trims.

```
\strims{0.5\trimtop}{0.5\trimedge}
```

\checkandfixthelayout

The commands \setpagetm, \setpagetr, \setpagemr, \setpagebr, \setpagebm, \setpagec are analogous to the earlier ones and they set a page at the top middle, top right, middle right, bottom right, bottom middle and centered with respect to the stock.

Remember that after you have finished defining the layout you want you have to call \checkandfixthelayout for all the changes to take effect.

In Figure 2.13 we show the effect of the nine \setpageXX. The code used for the examples is similar to the code below, and then the result is scaled down.

```
\documentclass[showtrims]{memoir}
\trimLmarks
\setstocksize{18cm}{15cm}
\makeatletter
\setpageb1{16cm}{12cm}{*}
\makeatother
\setlrmarginsandblock{15mm}{15mm}{*}
\setulmarginsandblock{15mm}{15mm}{*}
\setheadfoot{5mm}{5mm}
\checkandfixthelayout[fixed]
\pagestyle{empty}
\AtBeginDocument{\LARGE}
\begin{document}
\begin{vplace}
 \centering
 \cs{setpagebl}\marg{height}\marg{width}\marg{ratio}
\end{vplace}
\end{document}
```



Figure 2.13: Example of the nine  $\sectpageXX$  macros for placing a trimmed page on the stock. These are all odd pages, under twoside, even pages will have the left/right trims reversed.

## Three

## 텍스트와 글꼴

새로운 문서를 작성할때, 아마 조금이더라도 텍스트를 포함할 것이다. 이 챕터에서는 사용할 수 있는 글꼴의 종류와 페이지에 텍스트가 나타나는 방식에 대해 설명하겠다.

### 3.1 글꼴

(La)TeX에는 Donald Knuth가 디자인한 Computer Modern 글꼴 제품군으로 알려진 표준 글꼴 집합이 있다.

Knuthian 글꼴은 Metafont 프로그램 [Knu92, Knu87]을 통해 만들어졌으며 bitmaps 형식으로 되어 있다. (즉, 각 문자는 작은 점들로 표시된다). 이런 종류의 글꼴을 bitmap fonts. 이라고 한다.

표준 세트에는, 다른 많은 이들에 의해 추가되어진 광범위한 Metafont 도 있다. 보다 현대적인 디지털 글꼴, 예를 들어 PostScript 혹은 TrueType 글꼴은 문자를 개략적으로 나타내는 곡선으로 표현되며, 인쇄 기계가 외곽선을 채우는 역할을 한다 (작은 점들로 묶음). 이 유형의 글꼴을 윤곽선 글꼴(outline fonts)이라고 한다

Metafont fonts are designed for a particular display resolution and cannot reasonably be scaled to match an arbitrary display device, whereas outline fonts can be scaled before they are physically displayed.

There is an excessive number of PostScript and TrueType fonts available and these can all, with some amount of effort, be used with LaTeX. How to do that is outside the scope of this work; Alan Hoenig has written an excellent book on the subject [Hoe98] and there is the invaluable *The LaTeX Companion* [MG+04, Chapter 7 Fonts and Encodings]. The original *The LaTeX Graphics Companion* had chapters on PostScript fonts and tools but these were dropped in the second edition to keep it below an overwhelming size. This material has been updated and is available free from http://xml.cern.ch/lgc2; you can also get a 'work in progress' about XeTeX, Unicode, and Opentype fonts from the same source. There is less detailed, but also free, information available via CTAN, for example Philipp Lehman's *Font Installation Guide* [Leh04]; even if you are not interested in installing PostScript fonts this is well worth looking at just as an example of the kind of elegant document that can be achieved with LaTeX. If you choose one of the popular PostScript fonts, such as those built into PostScript printers, you may well find that the work has been done for you and it's just a question of using the appropriate package.

A standard LaTeX distribution includes some PostScript fonts and the packages to support them are in the psnfss bundle. Most of the fonts are for normal text work but two supply symbols rather than characters. Table 3.1, although it is specifically for Palatino, shows the glyphs typically available. Tables 3.2 and 3.3 show the glyphs in the two symbol fonts.

Table 3.1: Glyphs in the LaTeX supplied Palatino roman font

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	¿ 15
1 16	17	` 18	19	× 20	~ <sub>21</sub>	- 22	° 23
, 24	ß 25	æ 26	œ 27	Ø 28	Æ 29	Œ 30	Ø 31
32	! 33	" 34	# 35	\$ 36	% 37	<b>&amp;</b> 38	, 39
( 40	) 41	* 42	+ 43	, 44	<b>-</b> 45	. 46	/ 47
0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55
8 56	9 57	: 58	; 59	< 60	= 61	> 62	? 63
@ 64	A 65	B 66	C 67	D 68	E 69	F 70	G 71
H 72	I 73	J 74	K 75	L 76	M 77	N 78	O 79
P 80	Q 81	R 82	S 83	T 84	U 85	V 86	W 87
X 88	Y 89	Z 90	[ 91	\ 92	] 93	^ 94	_ 95
<b>'</b> 96	a 97	b 98	C 99	d 100	<b>e</b> 101	f 102	g 103
h 104	<b>i</b> 105	<b>j</b> 106	k 107	1 108	m 109	n 110	O 111
p 112	<b>q</b> 113	<b>r</b> 114	<b>S</b> 115	t 116	<b>u</b> 117	V 118	W 119
X 120	y 121	<b>Z</b> 122	{ 123	l <sub>124</sub>	} 125	~ 126	127
^ <sub>128</sub>	~ 129	Ç 130	Í 131	Î 132	<b>ã</b> 133	ë 134	è 135
<b>Š</b> 136	<b>Ž</b> 137	Ð 138	139	140	141	142	143
144	145	Š 146	147	148	149	150	151
Ÿ 152	153	Ž 154	155	156	157	158	159
160	161	<b>¢</b> 162	£ 163	/164	¥ 165	f 166	§ 167
¤ 168	© 169	170	« 171	₹ 172	→ 173	fi 174	fl 175
° 176	<b>—</b> 177	2 178	3 179	• 180	181	¶ 182	• 183
, 184	1 185	" <sub>186</sub>	» 187	1/4 188	1/2 189	3/4 190	191
À 192	Á 193	194	à 195	Ä 196	Å 197	198	199
È 200	É 201	Ê 202	Ë 203	Ì 204	″ <sub>205</sub>	, 206	Ϊ 207
208	$\tilde{ m N}$ 209	Ò 210	Ó 211	Ô 212	Õ 213	Ö 214	215
216	Ù 217	Ú 218	Û 219	Ü 220	Ý 221	Þ 222	223
à 224	á 225	â 226	a 227	ä 228	å 229	230	Ç 231
Ł 232	<b>é</b> 233	ê 234	o 235	Ì 236	<b>Í</b> 237	<b>î</b> 238	<b>ï</b> 239
240	<b>ñ</b> 241	Ò 242	Ó 243	Ô 244	Õ 245	Ö 246	247
ł 248	ù 249	<b>ú</b> 250	û 251	Ü 252	<b>ý</b> 253	þ 254	ÿ 255

Table 3.2: Glyphs in the LaTeX distributed Symbol font

		\ /			01	0	_
32	! 33	∀ 34	# 35	∃ 36	% 37	<b>&amp;</b> 38	€ 39
( 40	) 41	* 42	+ 43	, 44	<b>-</b> 45	. 46	/ 47
0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55
8 56	9 57	: 58	; 59	< 60	= 61	> 62	? 63
≅ 64	A 65	B 66	X 67	$\Delta$ 68	E 69	Ф 70	Γ 71
H 72	I 73	<b>v</b> 74	K 75	Λ 76	M 77	N 78	O 79
П 80	Θ 81	P 82	Σ 83	T 84	Y 85	ς 86	$\Omega$ 87
Ξ 88	Ψ 89	Z 90	[ 91	∴ 92	] 93	⊥ 94	_ 95
96	α 97	β 98	χ 99	δ 100	<b>E</b> 101	<b>ф</b> 102	γ 103
η 104	<b>l</b> 105	φ 106	<b>K</b> 107	λ 108	μ 109	ν 110	0 111
π 112	θ 113	ρ 114	σ 115	τ 116	<b>U</b> 117	<b>T</b> 118	<b>W</b> 119
ξ 120	Ψ 121	ζ 122	{ 123	124	} 125	~ 126	127
160	Υ 161	162	≤ 163	/164	∞ <sub>165</sub>	f 166	<b>♣</b> 167
<b>♦</b> 168	<b>V</b> 169	<b>1</b> 70	↔ 171	← 172	↑ <sub>173</sub>	$\rightarrow$ 174	<b>↓</b> 175
° 176	± 177	2 178	3 179	X 180	∝ <sub>181</sub>	ð 182	• 183
÷ 184	1 185	≡ 186	≈ 187	1/4 188	1/2 189	3/4 190	→ 191
<b>₹</b> 192	J 193	<b>R</b> 194	<b>(</b> ) 195	⊗ 196	<b>(</b> 197	Ø 198	∩ 199
U 200	⊃ 201	⊇ 202	⊄ 203	⊂ 204	⊆ 205	€ 206	∉ 207
∠ 208	V 209	® 210	© 211	TM <sub>212</sub>	∏ 213	√ <sub>214</sub>	• 215
¬ 216	∧ 217	V 218	⇔ 219	← 220	↑ <sub>221</sub>	⇒ 222	↓ 223
	<b>\( 225</b>	® 226	© 227	TM <sub>228</sub>	<b>∑</b> 229	230	231
232	T 233	234	L 235	236	{ 237	238	239
240	241	∫ <sub>242</sub>	ſ <sub>243</sub>	244	J <sub>245</sub>	246	247
)248	7249	250	] <sub>251</sub>	252	} <sub>253</sub>	J <sub>254</sub>	255

S<del>~</del> 33 **≫** 34 **≯** 35 **≈** 36 **☎** 37 © 38 **♣** 39 32 **●** 42 ₿ 44 **→** 40  $\boxtimes$  41 F 43 **L** 45 **46 48** ⇔ 49 **◆** 50 **√** 51 **✓** 52  $\times$  53 **X** 54 X 55 **X** 56 **+** 57 **+** 58 **+** 59 60 **†** 61 T 62 **f** 63  $\Leftrightarrow$  65 + 66 **4** 67 **%** 68 **4** 69 **♦** 70 ♦ 71 ₩ 64  $\stackrel{\wedge}{\bowtie} 73$ **☆** 78 **★** 72 **②** 74 **★** 77 **★** 75 **★** 76 **★** 79 **☆** 80 **≭** 82 **\*** 83 ₩ 85 **\*** 81 **\*** 84 **\*** 86 **\*** 87 **\*** 89 ₩ 90 **\*** 88 **\*** 91 **\*** 92 **\*** 93 **\$** 94 **\$** 95 **9** 97 98 **%** 99 攀 101 米 103 ₩ 96 ₩ 100 攀 102 **\*** 104 **\*** 105 O 109 110 **\*** 106 **\*** 107 108 1111 **V** 116 **♦** 117 **1**19 **1**112 113 118 **1**114 **1**15 **66** 125 1 120 121 122 6 <sub>123</sub> 9 <sub>124</sub> **99** 126 127 **1** 162 ¶ 161 **1**63 **♥** 164 **1**65 **¥** 166 **₹** 167 ¥ 170 2 173 3 174 168 ♦ 169 171 1 172 4 175 3 179 6 177 9 180 **10** 181 **1** 182 • 183 **5** 176 1 185 **6** 184 **6** 187 1/4 188 1/2 189 3/4 190 **6** 186 191 ①  $_{192}$ 6 197 ⑦ 198 2 193  $3_{194}$ 4 195 **⑤** 196 ® <sub>199</sub> 9 200 10 201 **1** 202 **2** 203 **3** 204 **4** 205 **6** 206 **6** 207 **7** 208 **3** 209 **9** 210  $\mathbf{0}_{211}$  $\rightarrow 212$  $\rightarrow$  213  $\leftrightarrow$  214 Ĵ 215 **1** 216 **→** 217 **≯** 218 **→** 219 **→** 220 → <sub>221</sub> **→** 222 **→** 223 ₩ 224 **→** 225 ➤ 226 → 227 **>** 228 → 229 **→** 230 231 **→** 232 **□>** 233 ➪ 234 ⇒ 236 \$ 237 ightharpoons 238 ⇒ 239  $\Rightarrow$  241  $\supset_{242}$ **⋙** 243 N 244 **>→** 245 **7** 246 **♣** 247 **▶** 248 **⇒** 254 **7** 249 **→** 251 **▶** 253 **→** 250 **→** 252 255

Table 3.3: Glyphs in the LaTeX distributed Zapf Dingbat font

These supplied PostScript fonts, their respective LaTeX fontfamily names, and running text examples of each, are:

ITC Avant Garde Gothic is a geometric sans type designed by Herb Lubalin and Tom Carnase and based on the logo of the Avant Garde magazine. The fontfamily name is pag.

It was a dark and stormy night. While all the good men were coming to the aid of the party, the quick brown dog had jumped over the fast red fox to its great surprise. The cattle had wound slowly o'er the lea and I was in the dark.

'But aren't Kafka's Schloß and Æsop's Œuvres often naïve vis-à-vis the dæmonic phœnix's official rôle in fluffy soufflés?'

Ångelå Beatrice Claire Diana Érica Françoise Ginette Hélène Iris Jackie Kāren Łaura María Nátalĭe Øctave Pauline Quêneau Roxanne Sabine Tãja Uršula Vivian Wendy

### Xanthippe Yvønne Zäzilie

ITC Bookman was originally sold in 1860 by the Miller & Richard foundry in Scotland; it was designed by Alexander Phemister. The ITC revival is by Ed Benguiat. The fontfamily name is pbk.

It was a dark and stormy night. While all the good men were coming to the aid of the party, the quick brown dog had jumped over the fast red fox to its great surprise. The cattle had wound slowly o'er the lea and I was in the dark.

'But aren't Kafka's Schloß and Æsop's Œuvres often naïve vis-à-vis the dæmonic phœnix's official rôle in fluffy soufflés?'

Ångelå Beatrice Claire Diana Érica Françoise Ginette Hélène Iris Jackie Kāren Łaura María Nấtalĭe Øctave Pauline Quêneau Roxanne Sabine Tãja Uršula Vivian Wendy Xanthippe Yvønne Zäzilie

Bitstream Charter was designed by Matthew Carter for display on low resolution devices, and is useful for many applications, including bookwork. The fontfamily name is bch.

It was a dark and stormy night. While all the good men were coming to the aid of the party, the quick brown dog had jumped over the fast red fox to its great surprise. The cattle had wound slowly o'er the lea and I was in the dark.

'But aren't Kafka's Schloß and Æsop's Œuvres often naïve vis-à-vis the dæmonic phœnix's official rôle in fluffy soufflés?'

Ångelå Beatrice Claire Diana Érica Françoise Ginette Hélène Iris Jackie Kāren Łaura María Nătalĭe Øctave Pauline Quêneau Roxanne Sabine Tãja Uršula Vivian Wendy Xanthippe Yvønne Zäzilie

Courier is a monospaced font that was originally designed by Howard Kettler at IBM and then later redrawn by Adrian Frutiger. The fontfamily name is pcr.

It was a dark and stormy night. While all the good men were coming to the aid of the party, the quick brown dog had jumped over the fast red fox to its great surprise. The cattle had wound slowly o'er the lea and I was in the dark.

'But aren't Kafka's Schloß and Æsop's Œuvres often naïve vis-à-vis the dæmonic phœnix's official rôle in fluffy soufflés?'

Ångelå Beatrice Claire Diana Érica Françoise Ginette Hélène Iris Jackie Kāren Łaura María Nătalĭe Øctave Pauline Quêneau Roxanne Sabine Tãja Uršula Vivian Wendy Xanthippe Yvønne Zäzilie

Helvetica was originally designed for the Haas foundry in Switzerland by Max Miedinger; it was later extended by the Stempel foundry and further refined by Linotype. The fontfamily name is phv.

It was a dark and stormy night. While all the good men were coming to the aid of the party, the quick brown dog had jumped over the fast red fox to its great surprise. The cattle had wound slowly o'er the lea and I was in the dark.

'But aren't Kafka's Schloß and Æsop's Œuvres often naïve vis-à-vis the dæmonic phœnix's official rôle in fluffy soufflés?'

Ångelå Beatrice Claire Diana Érica Françoise Ginette Hélène Iris Jackie Kāren Łaura María Nấtałĭe Øctave Pauline Quêneau Roxanne Sabine Tãja Uršula Vivian Wendy Xanthippe Yvønne Zäzilie

New Century Schoolbook was designed by Morris Benton for ATF (American Type Founders) in the early 20th century. As its name implies it was designed for maximum legibility in schoolbooks. The fontfamily name is pnc.

It was a dark and stormy night. While all the good men were coming to the aid of the party, the quick brown dog had jumped over the fast red fox to its great surprise. The cattle had wound slowly o'er the lea and I was in the dark.

'But aren't Kafka's Schloß and Æsop's Œuvres often naïve vis-à-vis the dæmonic phœnix's official rôle in fluffy soufflés?'

Ångelå Beatrice Claire Diana Érica Françoise Ginette Hélène Iris Jackie Kāren Łaura María Nấtalĭe Øctave Pauline Quêneau Roxanne Sabine Tãja Uršula Vivian Wendy Xanthippe Yvønne Zäzilie

Palatino was designed by Hermann Zapf and is one of the most popular typefaces today. The fontfamily name is ppl.

It was a dark and stormy night. While all the good men were coming to the aid of the party, the quick brown dog had jumped over the fast red fox to its great surprise. The cattle had wound slowly o'er the lea and I was in the dark.

'But aren't Kafka's Schloß and Æsop's Œuvres often naïve vis-à-vis the dæmonic phœnix's official rôle in fluffy soufflés?'

Ångelå Beatrice Claire Diana Érica Françoise Ginette Hélène Iris Jackie Kāren Łaura María Natalĭe Øctave Pauline Quêneau Roxanne Sabine Tãja Uršula Vivian Wendy Xanthippe Yvønne Zäzilie

Times Roman is Linotype's version of the Times New Roman face designed by Stanley Morison for the Monotype Corporation for printing The Times newspaper. The font-family name is ptm.

It was a dark and stormy night. While all the good men were coming to the aid of the party, the quick brown dog had jumped over the fast red fox to its great surprise. The cattle had wound slowly o'er the lea and I was in the dark.

'But aren't Kafka's Schloß and Æsop's Œuvres often naïve vis-à-vis the dæmonic phœnix's official rôle in fluffy soufflés?'

Ångelå Beatrice Claire Diana Érica Françoise Ginette Hélène Iris Jackie Kāren Łaura María Natalĭe Øctave Pauline Quêneau Roxanne Sabine Tāja Uršula Vivian Wendy Xanthippe Yvønne Zäzilie

Utopia was designed by Robert Slimbach and combines Transitional features and contemporary details. The fontfamily name is put.

It was a dark and stormy night. While all the good men were coming to the aid of the party, the quick brown dog had jumped over the fast red fox to its great surprise. The cattle had wound slowly o'er the lea and I was in the dark.

Table 3.4: Font categorisation and commands

	Shape			
Upright shape Italic shape Slanted shape Small Caps shape	<pre>\textup{Upright shape} \textit{Italic shape} \textsl{Slanted shape} \textsc{Small Caps shape}</pre>			
Series or weight				
Medium series Bold series	<pre>\textmd{Medium series} \textbf{Bold series}</pre>			
	Family			
Roman family Sans serif family Typewriter family	<pre>\textrm{Roman family} \textsf{Sans serif family} \texttt{Typewriter family}</pre>			

'But aren't Kafka's Schloß and Æsop's Œuvres often naïve vis-à-vis the dæmonic phœnix's official rôle in fluffy soufflés?'

Ångelå Beatrice Claire Diana Érica Françoise Ginette Hélène Iris Jackie Kāren Łaura María Nătalĭe Øctave Pauline Quêneau Roxanne Sabine Tãja Uršula Vivian Wendy Xanthippe Yvønne Zäzilie

ITC Zapf Chancery is a Script type fashioned after the chancery handwriting styles of the Italian Renaissance. It was created by Hermann Zapf. The fontfamily name is pzc.

It was a dark and stormy night. While all the good men were coming to the aid of the party, the quick brown dog had jumped over the fast red fox to its great surprise. The cattle had wound slowly o'er the lea and I was in the dark.

'But aren't Kafka's Schloß and Æsop's Œuvres often naïve vis-à-vis the dæmonic phœnix's official rôle in fluffy soufflés?'

Ångelå Beatrice Claire Diana Érica Françoise Ginette Hélène Iris Jackie Kāren Łaura María Nătalĭe Øctave Pauline Quêneau Roxanne Sabine Tãja Uršula Vivian Wendy Xanthippe Yvønne Zäzilie

Symbol contains various symbols and Greek letters for mathematical work; these are most easily accessible via the pifont package. The fontfamily name is psy.

The available glyphs are shown in Table 3.2.

Zapf Dingbats contains a variety of dingbats which, like the Symbol characters, are most easily accessible via the pifont package. The fontfamily name is pzd.

The available glyphs are shown in Table 3.3.

In LaTeX there are three characteristics that apply to a font. These are: (a) the shape, (b) the series (or weight), and (c) the family. Table 3.4 illustrates these and lists the relevant commands to access the different font categories.

The normal body font — the font used for the bulk of the text — is an upright, medium, roman font of a size specified by the font size option for the \documentclass.

Table 3.5: Font declarations

Shape						
Upright shape Italic shape Slanted shape Small Caps shape	<pre>{\upshape Upright shape} {\itshape Italic shape} {\slshape Slanted shape} {\scshape Small Caps shape}</pre>					
Series or weight						
Medium series Bold series	{\mdseries Medium series} {\bfseries Bold series}					
Family						
Roman family	{\rmfamily Roman family}					
Sans serif family	{\sffamily Sans serif family}					
Typewriter family {\ttfamily Typewriter fami						

### Typeset example 3.1: Badly mixed fonts

Mixing different series, **families** and **shapes**, especially in one sentence, is usually highly inadvisable!

### \normalfont

The declaration \normalfont sets the font to be the normal body font.

There is a set of font declarations, as shown in Table 3.5, that correspond to the commands listed in Table 3.4. The commands are most useful when changing the font for a word or two, while the declarations are more convenient when you want to typeset longer passages in a different font.

Do not go wild seeing how many different kinds of fonts you can cram into your work as in example 3.1.

```
Source for example 3.1

Mixing \textbf{different series, \textsf{families}} and
\textsl{\texttt{shapes,}} \textsc{especially in one sentence,}
is usually \emph{highly inadvisable!}
```

On the other hand there are occasions when several fonts may be used for a reasonable effect, as in example 3.2.

### Source for example 3.2

### Typeset example 3.2: Sometimes mixed fonts work

Des Dames du Temps Jadis Prince, n'enquerez de sepmaine

Ou elles sont, ne de cest an, Qu'a ce reffrain ne vous remaine: Mais ou sont les neiges d'antan?

Prince, do not ask in a week Or yet in a year where they are, I could only give you this refrain: But where are the snows of yesteryear?

François Villon [1431--1463?]

```
\begin{center}
\textsc{Des Dames du Temps Jadis}
\end{center}%
\settowidth{\versewidth}{Or yet in a year where they are}
\begin{verse}[\versewidth] \begin{itshape}
Prince, n'enquerez de sepmaine \\*
Ou elles sont, ne de cest an, \\*
Qu'a ce reffrain ne vous remaine: \\*
Mais ou sont les neiges d'antan?
\end{itshape}
Prince, do not ask in a week \\*
Or yet in a year where they are, \\*
I could only give you this refrain: \\*
But where are the snows of yesteryear?
\end{verse}
\begin{flushright}
{\bfseries Fran\c{c}ois Villon} [1431--1463?]
\end{flushright}
```

 $\left( \left\langle text\right\rangle \right)$ 

The \emph command is a font changing command that does not fit into the above scheme of things. What it does is to typeset its  $\langle text \rangle$  argument using a different font than the surrounding text. By default, \emph switches between an upright shape and an italic shape. The commands can be nested to produce effects like those in the next example.

### Typeset example 3.3: Emphasis upon emphasis

The \emph command is used to produce some text that should be emphasised for some reason and can be infrequently interspersed with some further emphasis just like in this sentence.

### Source for example 3.3

The \verb?\emph? command is used to produce some text that should be \emph{emphasised for some reason and can be \emph{infrequently interspersed} with some further emphasis} just like in this sentence.

### \eminnershape $\{\langle shape \rangle\}$

If the \emph command is used within italic text then the newly emphasized text will be typeset using the \eminnershape font shape. The default definition is:

\newcommand\*{\eminnershape}{\upshape}

which you can change if you wish.

### 3.2 Font sizes

The Computer Modern Metafont fonts come in a fixed number of sizes, with each size being subtly different in shape so that they blend harmoniously. Traditionally, characters were designed for each size to be cut, and Computer Modern follows the traditional type design. For example, the smaller the size the more likely that the characters will have a relatively larger width. Outline fonts can be scaled to any size, but as the scaling is typically linear, different sizes do not visually match quite as well.

Computer Modern fonts come in twelve sizes which, rounded to a point, are: 5, 6, 7, 8, 9, 10, 11, 12, 14, 17, 20 and 25pt. In LaTeX the size for a particular font is specifed by a macro name like \scriptsize and not by points; for example \scriptsize, not 7pt. The actual size of, say, \scriptsize characters, is not fixed but depends on the type size option given for the document.

Standard LaTeX provides ten declarations, illustrated in Table 3.6, for setting the type size, which means that two of the sizes are not easily accessible. Which two depend on the class and the selected point size option. However, for normal typesetting four different sizes should cover the majority of needs, so there is plenty of scope with a mere ten to choose from.

The \normalsize is the size that is set as the class option and is the size used for the body text. The \footnotesize is the size normally used for typesetting footnotes. The

<sup>&</sup>lt;sup>1</sup>It is possible to use points but that is outside the scope of this manual.

Table 3.6: Standard font size declarations

\tiny	tiny	\scriptsize	scriptsize
\footnotesize	footnotesize	\small	small
\normalsize	normal size	\large	large
\Large	Large	\LARGE	LARGE
\huge	huge	\Huge	Huge

Table 3.7: Standard font sizes

Class option	10pt	11pt	12pt
\tiny	$5\mathrm{pt}$	6pt	6pt
\scriptsize	$7\mathrm{pt}$	8pt	8pt
\footnotesize	8pt	9pt	10pt
\small	9pt	10pt	11pt
\normalsize	10pt	11pt	12pt
\large	12pt	12pt	14pt
\Large	14pt	14pt	17pt
\LARGE	17pt	17pt	20pt
\huge	20pt	20pt	25pt
\Huge	25pt	25pt	25pt

Table 3.8: The memoir class font size declarations

<pre>\miniscule \scriptsize \small</pre>	miniscule scriptsize small	\tiny \footnotesize \normalsize	tiny footnotesize normalsize
\large	large	\Large	Large
\LARGE	LARGE	\huge	huge
\Huge	Huge	\HUGE	HUGE

standard classes use the other sizes, usually the larger ones, for typesetting certain aspects of a document, for example sectional headings.

With respect to the standard classes, the memoir class provides a wider range of the document class type size options and adds two extra font size declarations, namely \miniscule and \HUGE, one at each end of the range.

The memoir class font size declarations names are given in Table 3.8 together with the name set in the specified size relative to the manual's \normalsize font. font. The corresponding actual sizes are given in Table 3.9.

Whereas the standard font sizes range from 5pt to 25pt, memoir provides for fonts rang-

Table 3.9: The memoir class font sizes

Class option	9pt	10pt	11pt	12pt	14pt	17pt	20pt	25pt	30pt	36pt	48pt	60pt
\miniscule	4pt	5pt	6pt	7pt	8pt	9pt	10pt	11pt	12pt	14pt	17pt	20pt
\tiny	5pt	6pt	$7\mathrm{pt}$	8pt	9pt	10pt	11pt	12pt	14pt	17pt	20pt	25pt
\scriptsize	6pt	$7\mathrm{pt}$	8pt	9pt	10pt	11pt	12pt	14pt	17pt	20pt	25pt	30pt
\footnotesize	$7\mathrm{pt}$	8pt	9pt	10pt	11pt	12pt	14pt	17pt	20pt	25pt	30pt	36pt
\small	8pt	9pt	10pt	11pt	12pt	14pt	17pt	20pt	25pt	30pt	36pt	48pt
\normalsize	9pt	10pt	11pt	12pt	14pt	17pt	20pt	25pt	30pt	36pt	48pt	60pt
\large	10pt	11pt	12pt	14pt	17pt	20pt	25pt	30pt	36pt	48pt	60pt	72pt
\Large	11pt	12pt	14pt	17pt	20pt	25pt	30pt	36pt	48pt	60pt	72pt	84pt
\LARGE	12pt	14pt	17pt	20pt	25pt	30pt	36pt	48pt	60pt	72pt	84pt	96pt
\huge	14pt	17pt	20pt	25pt	30pt	36pt	48pt	$60 \mathrm{pt}$	72pt	84pt	96pt	108pt
\Huge	17pt	20pt	25pt	30pt	36pt	48pt	60pt	72pt	84pt	96pt	108pt	120pt
\HUGE	20pt	25pt	30pt	36pt	48pt	$60 \mathrm{pt}$	72pt	84pt	96pt	108pt	120pt	132pt

ing from 4pt to 132pt. That is:

 $\begin{array}{l} {\rm from\ the\ 4pt\ size\ (the\ 9pt\ \ miniscule\ size)} \\ through\ the\ 9pt\ normal\ size \end{array}$ 

# through the 60pt normal size

## to the 132pt Size

# (the 60pt HUGE

# size).

This extended range, though, is only accessible if you are using outline fonts and the extrafontsizes class option. If you are using bitmap fonts then, for example, the \HUGE font will be automatically limited to 25pt, and the minimum size of a \miniscule font is 5pt.

### 3.3 Spaces

### 3.3.1 Paragraphs

In traditional typography the first line of a paragraph, unless it comes immediately after a chapter or section heading, is indented. Also, there is no extra space between paragraphs. Font designers go to great pains to ensure that they look good when set with the normal leading. Sometimes, such as when trying to meet a University's requirements for the layout of your thesis, you may be forced to ignore the experience of centuries.

If you like the idea of eliminating paragraph indentation and using extra interparagraph space to indicate where paragraphs start and end, consider how confused your reader will be if the last paragraph on the page ends with a full line; how will the reader know that a new paragraph starts at the top of the following page?

```
\parskip $$ \abnormalparskip {$\langle length \rangle$} $$ \nonzeroparskip $$ \traditionalparskip $$
```

In the input text the end of a paragraph is indicated either by leaving a blank line, or by the \par command. The length \parskip is the inter-paragraph spacing, and is normally 0pt. You can change this by saying, for example:

\setlength{\parskip}{2\baselineskip}

but you are likely to find that many things have changed that you did not expect, because LaTeX uses the \par command in many places that are not obvious.

If, in any event, you wish to do a disservice to your readers you can use \abnormalparskip to set the inter-paragraph spacing to a value of your own choosing. Using the \nonzeroparskip will set the spacing to what might be a reasonable non-zero value. Both these macros try and eliminate the worst of the side effects that occur if you just simply change \parskip directly.

 $<sup>^2{\</sup>rm Except}$  that all values except zero are unreasonable.

Following the \traditionalparskip declaration all will be returned to their traditional values.

I based the code for these functions upon the NTG classes [LEB04] which indicated some of the pitfalls in increasing the spacing. The difficulty is that \par, and hence \parskip, occurs in many places, some unexpected and others deeply buried in the overall code.

```
\parindent
```

The length \parindent is the indentation at the start of a paragraph's first line. This is usually of the order of 1 to  $1\frac{1}{2}$  em. To make the first line of a paragraph flushleft set this to zero:

\setlength{\parindent}{0pt}

### 3.3.2 Double spacing

Some of those that have control over the visual appearance of academic theses like them to be 'double spaced'. This, of course, will make the thesis harder to read<sup>3</sup> but perhaps that is the purpose, or maybe they have stock (shares) in papermills and lumber companies, as the theses were usually required to be printed single sided as well.

```
\baselineskip \onelineskip
```

The length \baselineskip is the space, or leading, between the baselines of adjacent text lines, and is constant throughout a paragraph. The value may change depending on the size of the current font. More precisely, the \baselineskip depends on the font being used at the *end* of the paragraph. The class also provides the length \onelineskip which is the default leading for the normal body font.<sup>4</sup> As far as the class is concerned this is a constant value; that is, unlike \baselineskip, it never alters \onelineskip. You can use (fractions) of \onelineskip to specify vertical spaces in terms of normal text lines.

The following is heavily based on the setspace package [Tob00], but the names have been changed to avoid any clashes. Like the nonzero \parskip, the \baselineskip rears its head in many places, and it is hard for a package to get at the internals of the overlying class and kernel code. This is not to say that all is well with trying to deal with it at the class level.

```
\OnehalfSpacing \OnehalfSpacing*
\DoubleSpacing \DoubleSpacing*
```

The declaration \OnehalfSpacing increases the spacing between lines so that they appear to be double spaced (especially to the thesis layout arbiters), while the declaration \DoubleSpacing really doubles the spacing between lines which really looks bad; but if you have to use it, it is there. The spacing in footnotes and floats (e.g., captions) is unaltered, which is usually required once the controllers see what a blanket double spacing brings. Sometimes it is also required to make page notes and floats (including captions)

 $<sup>^3</sup>$ I certainly found them so when I was having to read them before examining the candidates for their degrees. The writers of the regulations, which were invariably single spaced, seemed immune to any suggestions.

<sup>&</sup>lt;sup>4</sup>That is \onelineskip is set in the memX.clo file corresponding to the font size class option. For 10pt, the size is set via mem10.clo.

in 'double' spacing. The starred version of the two macros above takes care of this. Alternatively the spacing in page notes (i.e. footnotes and friends) or floats an be set explicitly using

\setFloatSpacing should go after say \OnehalfSpacing\* if used.

```
\label{eq:singleSpacing} $$ \SingleSpacing* $$ \setSingleSpace{$\langle factor \rangle$} $$
```

The \setSingleSpace command is meant to be used to adjust slightly the normal spacing betwen lines, perhaps because the font being used looks too crampled or loose. The effect is that the normal \baselineskip spacing will be multiplied by  $\langle factor \rangle$ , which should be close to 1.0. Using \setSingleSpace will also reset the float and page note spacings.

The declaration \SingleSpacing returns everthing to normal, or at least the setting from \setSingleSpace if it has been used. It will also reset float and page note spacings to the same value.

**Note.** \SingleSpacing will also issue a \vskip\baselineskip at the end (which is ignored if \SingleSpacing is used in the preamble). This skip makes sure that comming from \DoubleSpacing to \SingleSpacing still looks ok.

But in certain cases, this skip is unwanted. Therefore as of 2018 we added a \SingleSpacing\* that is equal to \SingleSpacing but does not add this skip.

```
\begin{SingleSpace} ...\end{SingleSpace}
\begin{Spacing}{\( factor \) \} ... \end{Spacing}
\begin{OnehalfSpace} ... \end{OnehalfSpace}
\begin{OnehalfSpace*} ... \end{OnehalfSpace*}
\begin{DoubleSpace*} ... \end{DoubleSpace}
\begin{DoubleSpace*} ... \end{DoubleSpace*}
```

These are the environments corresponding to the declarations presented earlier, for when you want to change the spacing locally.

If you have increased the interlinear space in the text you may wish, or be required, to increase it around displays (of maths). The declaration setDisplayskipStretch will increase the before and after displayskips by  $\langle fraction \rangle$ , which must be at least 0.0. More precisely, it defines memdskipstretch to be  $\langle fraction \rangle$ . The noDisplayskipStretch declaration sets the skips back to their normal values. It is equivalent to

```
\setDisplayskipStretch{0.0}
```

The skips are changed within the macro \memdskips which, in turn, is called by \everydisplay. If you find odd spacing around displays then redefine \memdskips to do nothing. Its orginal specification is:

```
\newcommand*{\memdskips}{%
    \advance\abovedisplayskip \memdskipstretch\abovedisplayskip
    \advance\belowdisplayskip \memdskipstretch\belowdisplayskip
    \advance\abovedisplayshortskip \memdskipstretch\abovedisplayshortskip
    \advance\belowdisplayshortskip \memdskipstretch\belowdisplayshortskip}
```

If you need to use a minipage as a stand-alone item in a widely spaced text then you may need to use the vminipage environment instead to get the before and after spacing correct.

### 3.4 Overfull lines

TeX tries very hard to keep text lines justified while keeping the interword spacing as constant as possible, but sometimes fails and complains about an overfull hbox.

```
\fussy\sloppy
\begin{sloppypar} ... \end{sloppypar}
\midsloppy
\begin{midsloppypar} ... \end{midsloppypar}
```

The default mode for LaTeX typesetting is \fussy where the (variation of) interword spacing in justified text is kept to a minimum. Following the \sloppy declaration there may be a much looser setting of justified text. The sloppypar environment is equivalent to:

```
{\par \sloppy ... \par}
```

Additionally the class provides the \midsloppy declaration (and the midsloppypar environment) which allows a setting somewhere between \fussy and \sloppy. Using \midsloppy you will get fewer overfull lines compared with \fussy and fewer obvious large interword spaces than with \sloppy. I have used \midsloppy for this manual; it hasn't prevented overfull lines or noticeably different interword spaces, but has markedly reduced them compared with \fussy and \sloppy respectively.

### 3.5 Sloppybottom

TeX does its best to avoid widow and orphan lines — a widow is where the last line of a paragraph ends up at the top of a page, and an orphan<sup>5</sup> is when the first line of a paragraph is at the bottom of a page.

The following is the generally suggested method of eliminating widows and orphans, but it may well result in some odd looking pages, especially if \raggedbottom is not used.

You can use \enlargethispage to add or subtract to the text height on a particular page to move a line forwards or backwards between two pages.

Here is one person's view on the matter:

<sup>&</sup>lt;sup>5</sup>Knuth uses the term 'club' instead of the normal typographers' terminology.

···in experimenting with raggedbottom, widowpenalty, and clubpenalty, I think that I have not found a solution that strikes me as particularly desirable. I think what I would really like is that widows (i.e., left-over single lines that begin on the following page) are resolved not by pushing one extra line from the same paragraph also onto the next page, but by stretching the textheight to allow this one extra at the bottom of the same page.

/iaw (from ctt, widow handling?, May 2006)

As so often happens, Donald Arseneau came up with a solution.

```
\sloppybottom
```

The declaration \sloppybottom lets TeX put an extra line at the bottom of a page to avoid a widow on the following page.

The \topskip must have been increased beforehand for this to work (a 60% increase is reasonable) and this will push the text lower on the page. Run \checkandfixthelayout after the change (which may reduce the number of lines per page). For example, in the preamble:

```
\setlength{\topskip}{1.6\topskip}
\checkandfixthelayout
\sloppybottom
```

The late Michael Downes provided the following (from CTT widow/orphan control package (for 2e)?, 1998/08/31):

For what it's worth here are the penalty values that I use when I don't [want] to *absolutely* prohibit widow/orphan break, but come about as close as TeX permits otherwise. This is copied straight out of some code that I had lying around. I guess I could wrap it into package form and post it to CTAN. Michael Downes

```
% set \clubpenalty, etc. to distinctive values for use
\% in tracing page breaks. These values are chosen so that
% no single penalty will absolutely prohibit a page break, but
\% certain combinations of two or more will.
\clubpenalt=9996
\widowpenalty=9999
\brokenpenalty=4991
% Reiterate the default value of \redisplaypenalty, for
% completeness.
% Set postdisplaypenalty to a fairly high value to discourage a
% page break between a display and a widow line at the end of a
% paragraph.
\predisplaypenalty=10000
\postdisplaypenalty=1549
% And then \displaywidowpenalty should be at least as high as
% \postdisplaypenalty, otherwise in a situation where two displays
% are separated by two lines, TeX will prefer to break between the
% two lines, rather than before the first line.
\displaywidowpenalty=1602
```

As you can see, perfect automatic widow/orphan control is problematic though typographers are typically more concerned about widows than orphans — a single line of a paragraph somehow looks worse at the top of a page than at the bottom. If all else fails, the solution is either to live with the odd line or to reword the text.

### 3.6 Text case

The standard kernel  $\mbox{MakeUppercase}\{\langle text \rangle\}$  and  $\mbox{MakeLowercase}\{\langle text \rangle\}$  basically upper or lower case everything it can get its hands on. This is not particularly nice if the  $\langle text \rangle$  contain, say, math.

In order to help with this we provide the  $\MakeTextUppercase$  and  $\MakeTextLowercase$  macros from the textcase package ([Car04]) by David Carlisle. The following is DCs own documentation of the provided code changed to match the typography we use.

\MakeTextUppercase and \MakeTextLowercase are versions of the standard \MakeUppercase and \MakeLowercase that do not change the case of any math sections in their arguments.

```
\MakeTextUppercase{abc\ae\ \( a = b \) and $\alpha \neq a$
or even \ensuremath{x=y} and $\ensuremath{x=y}$}
```

Should produce:

```
ABCÆ a = b AND \alpha \neq a OR EVEN x = y AND x = y
```

We incorporates some changes suggested by Donald Arseneau so that as well as math mode, the arguments of \cite, \label and \ref are also prevented from being uppercased. So you can now go

```
\MakeTextUppercase{%
```

```
Text in section~\ref{sec:text-case}, about \cite[pp 2--4]{textcase}}
```

which produces

```
TEXT IN SECTION 3.6, ABOUT [Car04, PP 2–4]
```

If, instead, the standard \MakeUppercase were used here, the keys 'sec:text-case' and 'textcase' would be uppercased and generate errors about undefined references to SEC:TEXT-CASE and TEXTCASE.

```
\NoCaseChange{\langle text \rangle}
```

Sometimes there may be a special section of text that should not be uppercased. This can be marked with \NoCaseChange, as follows.

```
\MakeTextUppercase{%
```

```
Text \NoCaseChange{More Text} yet more text}
```

which produces

TEXT More Text YET MORE TEXT

\NoCaseChange has other uses. If for some reason you need a tabular environment within an uppercased section, then you need to ensure that the name 'tabular' and the preamble (eg 'll') does not get uppercased:

### 3.6.1 Nested text

The commands defined here only skip math sections and \ref arguments if they are not 'hidden' inside a { } brace group. All text inside such a group will be made uppercase just as with the standard \MakeUppercase.

```
\MakeTextUppercase{a b {c $d$} $e$}
```

produces

Of course, this restriction does not apply to the arguments of the supported commands \ensuremath, \label, \ref, and \cite.

If you cannot arrange for your mathematics to be at the outer level of brace grouping, you should use the following basic technique (which works even with the standard \MakeUppercase command). Define a new command that expands to your math expression, and then use that command, with \protect, in the text to be uppercased. Note that if the text being uppercased is in a section title or other moving argument you may need to make the definition in the document preamble, rather than just before the section command, so that the command is defined when the table of contents file is read.

```
\label{thm:linear_case} $$ \text{MakeTextUppercase}$$ and $x=y$}$$ \end{\mathexprone} $$ a=b$ \end{\mathexprone} $$ \end{\mathexprone}
```

See also [Car04] for some information about upper casing citations using a non-nummeric style.

### Four

### 표제

표준 클래스들은 표지를 설정하는데 큰 도움이 되지 못하는데, \maketitle 명령은 테크니컬 논문집에서 논문의 제목을 생성하는 것이 주목적이기 때문이다. 이는 학위 논문, 보고서나 서적의 표지를 만들기에 불충분하다. 필자는 이를 위해 \maketitle을 무시하고, 직접 LaTeX 표준 명령어를 사용해 표지 레이아웃을 디자인할 것을 권장한다.

Ruari McLean [McL80, p. 148] 을 인용하자면, 그는 표지에 대해서 다음과 같이 말한다:

표지는 책의 실제 제목과 (있을 경우 부제목을 포함해) 저자의 이름, 출판사, 그리고 간혹 삽화의 개수를 포함하지만, 나아가 그것보다 많은 일을 해야 한다. 디자이너의 관점에서 표지는 책에서 가장 중요한 부분으로, 책의 양식을 정한다. 표지는 독자와의 소통을 시작한다…

만약 책에서 삽화가 큰 비중을 차지한다면, 표지는 이를 시각적으로 보일 수 있 거나 그래야 한다. 책에서, 예컨대 장의 시작에서라던지, 어떠한 형태로든 장식이 사용된다면 독자는 이것이 표지에서도 반복될 것을 기대할 것이다.

책의 형식이 어떻든지간에, 표지는 그것의 맛보기를 보여줘야 한다. 만약 책이 줄글로 구성되어 있다면, 표지는 적어도 그것과 조화를 이룰 수 있어야 한다. 표지 자체는 글 영역의 너비보다 넓어서는 안되며, 일반적으로는 더 좁을 것이다…

McLean의 표지를 모방한 것이 Figure 4.1에 나와 있다.

\maketitle 명령의 조판 형식은 LaTeX 표준 클래스에서 사실상 고정된 것으로 보아야한다. 이 클래스는 제목 정보, 즉 \title, \author, \date의 내용이 표시되는 형태를 수정할수 있는 일련의 형식화 명령들을 제공한다. 또, 이는 위 명령들의 값을 나중에 문서에서 다시사용할 수 있도록 유지한다.

나아가 이 클래스는 \maketitle 명령이 사용된 후에 사용된 명령 값이 자동으로 지워지는 것을 방지한다. 그러므로 하나의 문서 안에서 같거나 다른 표지가, 예를 들면 반표지와 전표 지에, 여러 번 나오게 할 수도 있다. \thanks 명령은 기능이 확장되어 감사의 말 주석의 표지 부호와 레이아웃을 다양하게 설정할 수 있게 되었다.

일반적으로 \maketitle 레이아웃에서 약간의 변화를 넘어선다면 \maketitle을 무시하고 레이아웃 전체를 여러분이 손수 만들어 여러분이 종의의 원하는 곳에 모든 것을 위치시키는 편이 낫다.

### 4.1 표지 양식화

표지를 조판하기 위해 제공되는 도구들은 국한되어 있는데, 이들은 본질적으로 테크니컬 논문집에 출판되는 논문의 제목 형식을 제공해주는 것이 전부이다. 이들은 보고서의 제목을 조판할때에도 빠르고 지저분한 방법을 제공해줄 수 있지만, 책이나 학위 논문과 같은 중요한 작업의표지에 대해서는 수작업이 필요하다. 예컨대 나의 학생인 Donal Sanderson은 그의 박사 학위

<sup>&</sup>lt;sup>1</sup>만약 여러분이 학위 논문을 작성 중이라면, 그것이 어떻게 생겨야 하는지 아마도 정해져 있을 것이다.

The Author The Big Book of CONUNDRUMS With 123 illustrations The Publisher  $\boxed{\mathcal{PL}}$ 

Figure 4.1: 타이포그래피에 관한 책의 표지 레이아웃

### CONUNDRUMS CONSIDERED AS PUZZLES FOR THE MIND

By

The Candidate

A Thesis Submitted to the Graduate
Faculty of The University
in Partial Fulfillment of the
Requirements for the Degree of
DEGREE

Major Subject: Logic

Approved by the
Examining Committee:

A Professor, Thesis Advisor

Another Professor, Thesis Advisor

A Faculty, Member

Another Faculty, Member

A Third Faculty, Member

The University
The Address

The Date

Figure 4.2: 박사 학위 논문의 지정 표지 양식의 예시

### THE NEW

### FAMILY RECEIPT BOOK

CONTAINING A LARGE COLLECTION OF

HIGHLY ESTIMATED RECEIPTS IN A VARIETY OF BRANCHES, NAMELY:

### BREWING,

MAKING AND PRESERVING BRITISH WINES,

### DYING,

RURAL AND DOMESTIC ECONOMY,

SELECTED FROM EXPERIENCED & APPROVED RECEIPTS,

FOR THE USE OF PUBLICANS

AND HOUSEKEEPERS IN GENERAL,

A GREAT MANY OF WHICH WERE NEVER BEFORE PUBLISHED.

### BY G. MILLSWOOD.

\_\_\_\_

PRICE ONE SHILLING

DERBY: PRINTED AND SOLD BY G. WILKINS AND SON,  $\label{eq:QUEEN STREET} \text{QUEEN STREET}.$ 

Figure 4.3: 빅토리아 시대의 표지 예시

# BIG BOOK OF CONUNDRUMS BY

FOREWORD BY AN OTHER

THE AUTHOR

THE PUBLISHER

Figure 4.4: 책 디자인에 관한 책의 표지 레이아웃

### Some Conundrums Puzzles for the Mind THE AUTHOR The Publisher

Figure 4.5: 책들에 관한 책의 표지 레이아웃

### Typeset example 4.1: 예시 \maketitle 표제

### **MEANDERINGS**

T. H. E. River and A. Wanderer\*
Dun Roamin Institute, NY

1 April 1993<sup>†</sup>

:

논문을 조판하기 위해 LaTeX 을 사용했고, Figure 4.2에 Rensselear Polytechnic Institute에서 1994년에 지정된 표지 양식이 나와 있다. 이외의 다양한 표지의 예시와, 이들을 만드는데 사용된 코드가 [Wil07a]에 있다.

[Wil07a]에 나오는 또 다른 수작업된 표지가 Figure 4.3에 나와 있다. 이것은 필자가 19세기 말 즈음에 출판된 옛 소책자를 기반으로 한 것이며, 다양한 활자를 표시할 수 있는 빅토리아시대의 프린터에 대한 애정을 확인할 수 있다. 여기서 괘선이 표지의 핵심적인 부분이라고 할 수 있다. 본 매뉴얼을 위해 필자는 LaTeX 표준 배포판에 포함된 New Century Schoolbook 글꼴을 사용했는데, 본래는 SoftMaker/ATF 라이브러리에서 라이선스를 받은, 그리고 Christoper League의 공로[Lea03]로 LaTeX 을 지원하는 FontSite 글꼴 중 하나인 Century Old Style을 선택했었다.

Figure 4.4의 표지는 *The Design of Books* [Wil93]의 양식을 따르며, 페이지는 Nicholas Basbanes의 *A Gentle Madness: Bibliophiles, Bibliomanes, and the Eternal Passion for Books*와 유사한 것이 Figure 4.5에 나와 있다. 이들 모두는 [Wil07a]에서 가져온 것이며 수작업된 것들이다.

반면에 다음 코드는 \maketitle의 표준 레이아웃을 생성한다.

본 클래스의 이 부분은 titling 패키지[Wil01g]를 재구현한 것이다.

<sup>\*</sup> Supported by a grant from the R. Ambler's Fund

<sup>&</sup>lt;sup>†</sup> First drafted on 29 February 1992

이 클래스는 설정 가능한 \maketitle 명령을 제공한다. \maketitle 명령은 이 클래스에서 본질적으로

```
\newcommand{\maketitle}{%
  \vspace*{\droptitle}
  \maketitlehooka
  {\pretitle \title \posttitle}
  \maketitlehookb
  {\preauthor \author \postauthor}
  \maketitlehookc
  {\predate \date \postdate}
  \maketitlehookd
  \thispagestyle{title}
}
```

와 같이 정의되는데, 이때 *title* 페이지 양식은 처음에는 *plain* 페이지 양식과 동일하다. \maketitle 내에서 사용되는 각종 매크로는 아래에 설명되어 있다.

```
\label{eq:local_precision} $$ \operatorname{d}_{\det \{\langle text \rangle\}} \operatorname{d}_{\det \{\langle text \rangle\}} $$ \operatorname{d}_{\det \{\langle text \rangle\}} \operatorname{d}_{\det \{\langle text \rangle\}} $$
```

위 여섯 개의 명령들은 각자 하나의 인자  $\langle text \rangle$ 를 가지며, 이는 문서에서 \maketitle 명령의 표준 요소의 조판을 통제한다. \title은 본질적으로 \pretitle과 \posttitle 사이에서 처리되는데, 다음과 같이

{\pretitle \title \posttitle}

되며 \author과 \date 명령에 대해서도 유사하다. 이 명령들은 report 클래스의 일반적인 \maketitle의 조판 결과를 모방하도록 초기화되어 있다. 즉, 명령들의 기본 정의는 다음과 같다.

```
\pretitle{\begin{center}\LARGE}
\posttitle{\par\end{center}\vskip 0.5em}
\preauthor{\begin{center}
          \large \lineskip 0.5em%
          \begin{tabular}[t]{c}}
\postauthor{\end{tabular}\par\end{center}}
\predate{\begin{center}\large}
\postdate{\par\end{center}}
```

이들은 다른 효과를 내기 위해 바꿀 수 있다. 예를 들어 우측 정렬된 sans-serif 제목과 좌측 정렬된 small caps 날짜를 얻으려면 다음과 같이 하라.

```
\pretitle{\begin{flushright}\LARGE\sffamily}
\posttitle{\par\end{flushright}\vskip 0.5em}
\predate{\begin{flushleft}\large\scshape}
\postdate{\par\end{flushleft}}
```

```
\droptitle
```

\maketitle 명령은 제목을 페이지의 특정 높이에 놓는다. 여러분은 제목의 세로 위치를 \droptitle 길이를 통해 변경할 수 있다. 여기에 양수를 대입하면 제목을 낮출 것이며, 음수를 대입하면 높일 것이다. 기본 정의는 다음과 같다.

\setlength{\droptitle}{0pt}

```
\maketitlehooka \maketitlehookb
\maketitlehookc \maketitlehookd
```

이 네 후크 명령은 \maketitle에 추가적인 요소를 넣을 수 있도록 제공된다. 이들은 기본적으로 아무것도 하지 않도록 정의되어 있지만, 재정의할 수 있다. 예를 들어, 일부 출판물에서는 논문이 출판된 곳에 대한 문장을 실제 제목 문구 직전에 삽입하는 것을 요구한다. 다음은 \published 명령을 정의하여 출판 정보를 담을 수 있도록 하며, \maketitle에 의해 자동적으로 출력된다.

```
\newcommand{\published}[1]{%
    \gdef\puB{#1}}
\newcommand{\puB}{}
\renewcommand{\maketitlehooka}{%
    \par\noindent \puB}

O|

\brace{\Phi}
\published{Originally published in
    \textit{The Journal of ...}\thanks{Reprinted with permission}}
...
\maketitle
```

와 같이 하여 출판 정보와 일반적인 제목 정보를 모두 출력할 수 있다. 새로운 \published 명령과 함께 \thanks 명령어를 사용하기 위해서 추가적인 조치가 필요 없다는 것에 주의하라.

```
\begin{titlingpage} text \end{titlingpage} 
\begin{titlingpage*} text \end{titlingpage*} 
\titlingpageend{\langle twoside code \rangle \} {\langle oneside code \rangle \}
```

표준 클래스들과 titlepage 옵션이 함께 사용되면, \maketitle은 제목 요소를 숫자가 붙지 않은 페이지에 넣고 새로운 페이지를 페이지 번호 1로 시작한다. 표준 클래스는 titlepage 환경도 제공하여, 번호가 붙지 않은 새로운 장을 시작하고 이후 다시 페이지 번호 1부터 새로운 페이지를 시작한다. 이 표지에 어떤 내용을 넣고 어디에 놓을지는 전적으로 여러분의 책임에 달려 있다. 만약 \maketitle이 titlepage 환경 안에서 사용된다면 이는 또 다른 페이지를 시작할 것이다.

이 클래스는 titlingpage 옵션이나 titlepage 환경 둘 중 어느 것도 제공하지 않는다. 대신이는 titlepage 옵션과 titlepage 명령 중간쯤 되는 titlingpage 환경을 제공한다. 여러분은 titlingpage 환경에서 \maketitle을 포함한 명령들을 사용할 수 있다. titlingpage 페이지 양식이 사용되며, 끝에는 번호 1이 붙은 일반적인 페이지를 시작한다 (\begin{titlingpage\*}는 페이지 번호를 재설정하지 않는다). titlingpage 페이지 양식은 empty 페이지 양식과 같게 초기 설정되어 있다.

titlingpage의 끝에는 지우기 코드가 실행되는데, 이를 통해 다음 페이지 혹은 다음 우츨페이지로 이동할 수 있다. \titlingpageend{\\(\lambda\) twoside code\\}-{\(\lambda\) tolearpage이다. 2 그러나 그냥 \clearforchapter을 따르도록 하는 것이 더 나은 선택이 될 수 있다.

\titlingpageend{\clearforchapter}{\clearforchapter}

<sup>&</sup>lt;sup>2</sup>따라서 이 수정은 기존의 문서를 바꾸지 않을 것이다, LM, 2018/03/06.

이 값을 사용하면, titlingpage는 openany와 함께 예상대로 작동할 예를 들어, plain 페이지 양식으로 제목과 요약을 같은 표지에 두고 싶다면 다음과 같이 한다.

```
\begin{document}
\begin{titlingpage}
\aliaspagestyle{titlingpage}{plain}
\setlength{\droptitle}{30pt} lower the title
\maketitle
\begin{abstract}...\end{abstract}
\end{titlingpage}
```

그러나 \begin{titlingpage}를 사용해서 표지를 만드는 것이 필수는 아니므로, 여러분은 특수한 환경 없이 일반적인 LaTeX 조판을 사용할 수 있다. 그 방법은 다음과 같을 것이다.

```
\pagestyle{empty}
%%% Title, author, publisher, etc., here
\cleardoublepage
```

기본적으로, 제목 정보는 조판 영역 너비를 기준으로 가운데 정렬된다. 간혹 누군가 comp.text.tex 뉴스 그룹에 표지의 제목 정보를 실물 페이지 기준으로 가운데 정렬하는 방법을 묻고는 한다. 만약 조판 영역이 실제 페이지 기준으로 가운데에 있다면 기본 가운데 정렬로 충분할 것이다. 만약 조판 영역이 실제로 가운데가 아니라면, 제목 정보를 가로 방향으로 이동시키던가, \maketitle에게 조판 영역이 가로 방향으로 이동되었다고 믿게할 수 있다. 가장 간단한 해결책은 \calccentering 명령과 adjustwidth\* 환경을 사용하는 것이다. 예를 들면 다음과 같다.

```
\begin{titlingpage}
  \calccentering{\unitlength}
  \begin{adjustwidth*}{\unitlength}{-\unitlength}
  \maketitle
  \end{adjustwidth*}
\end{titlingpage}
```

```
\label{title} $$ \left( \left\langle text \right\rangle \right) $$ the title $$ \author \left( \left\langle text \right\rangle \right) $$ the author $$ \date \left( \left\langle text \right\rangle \right) $$ the date $$
```

일반적인 문서 클래스에서 \maketitle을 위해 사용되는 \title, \author과 date 매크로의 내용들  $(\langle text \rangle)$  은 \maketitle 사용 이후 접근할 수 없다. 본 클래스는 \thetitle, \theauthor과 \thedate 명령을 제공하여 원한다면 이후 문서에서 제목 요소를 출력하는데 사용할 수 있도록 한다.

```
\and \andnext
```

매크로 \and는 \author 명령어의 인자 안에 쓰이며, 저자들의 이름 사이에 추가 공백을 넣어준다. 이 클래스의 \andnext 매크로는 공백 대신에 새로운 줄을 넣어준다. \theauthor 매크로 안에서 \and와 \andnext는 둘 다 쉼표로 치환된다.

본 클래스는 표준 클래스처럼 \maketitle 사용 이후에 표지 명령을 자동으로 끄는 관행을 따르지 않는다. 여러분은 원한다면 문서에 여러 개의 \title, \author, date, 그리고 \maketitle 명령들을 넣을 수 있다. 예를 들어, 일부 보고서는 표지로 시작해서 뒤이어 요약문이 따르고, 다르게 표현될 수도 있는 또다른 제목을 본문 앞에 가진다. 이는 다음과 같이 구현할 수 있다.

```
\title{Cover title}
...
\begin{titlingpage}
\maketitle
\end{titlingpage}
...
\title{Body title}
\maketitle
...
```

\killtitle \keepthetitle \emptythanks

\killtitle 매크로 사용 후에 \thetitle 등을 포함한 모든 표지 관련 명령을 사용하지 못하도록 한다 (이 명령을 사용해 \thetitle 등과 같은 명령이 필요하지 않을 경우 매크로 공간을 확보할 수 있다). 이 명령은 본 클래스에서 표준 클래스의 자동 종료 수행의 수동 버전이다. \keepthetitle 명령은 비슷한 기능을 하지만, 다른 모든 기능은 끄면서 \thetitle, \theauthor과 \thedate 명령을 유지한다는 차이가 있다.

\emptythanks 명령은 기존 \thanks의 모든 문구를 지운다. 이 명령은 \maketitle이 여러 번 사용될 경우 유용하다. \thanks 명령은 매 사용마다 문구를 쌓아올리므로, \maketitle을 매번 사용할 때 기존의 모든 \thanks 문구가 새 문구와 함께 출력될 것이다. 이를 방지하기 위해서는 \maketitle을 사용하기 전에 매번 \emptythanks를 넣으면 된다.

### 4.2 감사의 말 양식화하기

본 클래스는 설정 가능한 \thanks 명령을 제공한다.

```
\thanksmarkseries \{\langle format \rangle\} \\ \tsymbol thanksmark
```

모든 \thanks는 표지나 각주에 기호로 표시된다. 명령 \thanksmarkseries는 이러한 표지 양식을 바꾸는데 사용될 수 있다. 〈format〉 인자는 카운터를 출력하는 형식들 중 하나의 이름이다. 이름은 카운터 형식과 같지만 백슬래시를 포함하지 않는다. \thanks가 기호 대신에 소문자로 표시되기 위해서는 다음과 같이 하라.

### \thanksmarkseries{alph}

편의를 위해서 \symbolthanksmark는 목록을 각주 기호로 지정한다. 이 클래스를 사용하면 ⟨format⟩에는 arabic, roman, Roman, alph, Alph 그리고 fnsymbol이 이름으로 올 수 있다.

```
\continuousmarks
```

\thanks 명령어는 footnote 카운터를 사용하며, 표제 이후에는 각주 표지를 1부터 시작하기 위해 영으로 초기화된다. 만약에 카운터를 영으로 만들고 싶지 않다면 \continuousmarks를 명시하라. 이는 감사의 말 표지가 수로 표기된다면 필요할 수 있다.

\thanksheadextra 명령어는  $\langle pre \rangle$ 와  $\langle post \rangle$ 를 각각 표제 영역에 있는 감사의 말 전후에 삽입할 것이다. 기본적으로  $\langle pre \rangle$ 와  $\langle post \rangle$ 는 비어있다. 예를 들어, 표제 표지를 감싸는 괄호를 넣기 위해서는 다음과 같이 한다.

\thanksheadextra{(){)}

### 

간혹 여러 저자에게 동일한 감사의 말 문구를 적용해야할 경우가 있는데, 예를 들어 여섯 명 중 첫 세 명과 마지막의 네 저자에게 적용해야한다고 하자. \thanksmark $\{\langle n \rangle\}$  명령어는 \thanks 명령어의  $\langle n \rangle$ 번째 표지와 같은 표지를 출력한다는 점에서 \footnotemark $[\langle n \rangle]$ 와 유사하다. 각주의 감사의 말에는 어떠한 변경도 생기지 않는다. 예를 들어, 다음의 Alpha와 Omega 저자는 동일한 표지를 가질 것이다.

\title{The work\thanks{Draft}}

\author{Alpha\thanks{ABC},

Beta\thanks{XYZ} and

Omega\thanksmark{2}}

\maketitle

 $\t$  \thanksmarkstyle{(defn)}

기본적으로 각주의 표지는 윗첨자로 조판된다. 본 클래스에서 이는

\thanksmarkstyle{\textsuperscript{#1}}

와 같이 지정할 수 있으며, 이때 #1는 감사의 말 표지 기호로 치환된다. 여러분이 원한다면 표지 양식을 바꿀 수 있다. 예를 들어, 표지를 괄호로 감싸고 베이스라인에 normal 크기로 조판하고 싶다면 다음과 같이 한다.

\thanksmarkstyle{(#1)}

### \thanksmarkwidth

각주의 감사의 말 표지는 너비 \thanksmarkwidth의 박스에 오른쪽 정렬되어 조판된다. 감사의 말 문구의 첫 줄은 이 박스 뒤에 시작된다. 초기값은

\setlength{\thanksmarkwidth}{1.8em}

으로 기본 위치가 설정되어 있다.

### \thanksmarksep

길이값 \thanksmarksep는 감사의 말의 두 번째 및 그 이후 줄의 들여쓰기를 표지 박스 끝에 맞추어 조절한다. 예를 들어서

\setlength{\thanksmarksep}{0em}

는 여러 줄에 걸친 감사의 말을 왼쪽 끝에 따라 정렬할 것이며

\setlength{\thanksmarksep}{-\thanksmarkwidth}

는 두 번째와 그 이후 줄을 왼쪽 정렬 할 것이다. 이 마지막 설정이 초기값으로, 기본 설정이다.

### \thanksfootmark

각주 위치의 감사의 말 표지는 \thanksfootmark로 식자된다. 이에 대한 코드는 대략 다음과 같다.

 $\verb|\newcommand{\thanksfootmark}{%}|$ 

\hbox to\thanksmarkwidth{\hfil\normalfont%

\thanksscript{\thefootnote}}}

여러분이 \thanksfootmark의 정의를 바꿀 필요는 없지만, 여기에 사용되는 몇몇 매크로들의 기본 정의를 바꿀 수도 있다.

이는 기본적으로

\newcommand{\thanksscript}[1]{\textsuperscript{#1}}

와 같이 정의되어 있어서, 감사의 말의 기본 설정으로 \thanksscript가 인자를 윗첨자로 식자 하도록 한다.

\makethanksmark \makethanksmarkhook

매크로 \makethanksmark는 (\thanksfootmark를 통해서) 감사의 말 표지와 감사의 말 문구를 둘 다 식자한다. 여러분은 아마도 이의 기본 정의를 바꿀 필요는 없을 것이다. 그래도 혹시 모르니, \makethanksmark는 식자하기 전에 먼저 \makethanksmarkhook를 부른다. 이의 기본 정의는

\newcommand{\makethanksmarkhook}{}

으로 아무것도 하지 않는다.

여러분은 \makethanksmarkhook가 유용한 일을 하도록 재정의할 수 있다. 예를 들어, 조금 더 큰 베이스라인 간격이 필요하다면

\renewcommand{\makethanksmarkhook}{\fontsize{8}{11}\selectfont}

와 같이 할 수 있고, 이때 8과 11는 각각 폰트의 포인트 크기와 베이스라인 간격을 나타낸다. 이 예시에서 8pt는 10pt 문서의 보통 footnotesize이며, 11pt는 11pt 문서에서 베이스라인 간격이다 (10pt 문서에서 \footnotesize의 보통 베이스라인 간격은 9.5pt이다). 이들 값을 워하는대로 조절하라.

\thanksrule \usethanksrule \cancelthanksrule

기본적으로 titlingpage 환경에서 \thanks 문구 위에는 선이 없다. 만약 해당 환경에서 선을 그리고 싶다면 \usethanksrule을 \maketitle 명령 전에 넣어 \thanksrule의 현재 정의에 맞게 선을 출력하도록 할 수 있다. \thanksrule은 preamble 끝에 정의된 \footnoterule의 사본으로 초기 설정되어 있다. \thanksrule의 정의는 \begin{document} 이후에 변경할 수 있다. 만약 \thanksrule의 정의가 변경된 후 \usethanksrule이 사용되었다면, 각주에 대해서도 동일한 선이 사용될 수 있다. \cancelthanksrule을 사용하면 이후로 보통 \footnoterule 정의가 사용될 것이다. 나중에 다시 바꾸고 싶다면 \usethanksrule을 이후에 다시 쓰면 된다.

각주와 감사의 말 주석의 세로 위치에 대한 매개변수와 기본 \footnoterule은 같다 (page 233의 Figure 12.1을 보라.). 각주와 감사의 말 주석의 세로 간격이 달라야 한다면 이들 중 일부를 바꿔야 한다.

### Five

### **Abstracts**

Abstracts do not normally appear in books but they are often an essential part of an article in a technical journal. Reports may or may not include an abstract, but if so it will often be called a 'Summary'. There may be an even shorter abstract as well, often called an 'Executive Summary', for those who feel that details are irrelevant.

In the standard classes appearance of the abstract environment is fixed. The class provides a set of controls adjusting the appearance of an abstract environment.

Questions about how to have a one-column abstract in a two-column paper seem to pop up fairly regularly on the comp.text.tex newsgroup. While an answer based on responses on crt is provided in the FAQ, the class provides a more author-friendly means of accomplishing this.

### 5.1 Styling

Much of this part of the class is a reimplementation of the abstract package [Wil01a].

The typeset format of an abstract in a report or article class document depends on the class options.<sup>1</sup> The formats are:

- titlepage class option: The abstract heading (i.e., value of \abstractname) is typeset centered in a bold font; the text is set in the normal font and to the normal width.
- twocolumn class option: The abstract heading is typeset like an unnumbered section; the text is set in the normal font and to the normal width (of a single column).
- Default (neither of the above class options): The abstract heading is typeset centered in a small bold font; the text is set in a small font and indented like the quotation environment.

This class provides an abstract environment and handles to modify the typesetting of an abstract.

```
\begin{abstract} text \end{abstract}
```

There is nothing special about using the abstract environment. Formatting is controlled by the macros described below.

\abstractcol
\abstractintoc
\abstractnum
\abstractrunin

<sup>&</sup>lt;sup>1</sup>The abstract environment is not available for the book class.

The normal format for an abstract is with a centered, bold title and the text in a small font, inset from the margins.

The \abstractcol declaration specifies that an abstract in a twocolumn class option document should be typeset like a normal, unnumbered chapter. The \abstractintoc specifies that the abstract title should be added to the ToC. The declaration \abstractnum specifies that the abstract should be typeset like a numbered chapter and \abstractrunin specifies that the title of the abstract should look like a run-in heading; these two declarations are mutually exclusive. Note that the \abstractnum declaration has no effect if the abstract is in the \frontmatter.

```
\abstractname
\abstractnamefont
\abstracttextfont
```

\abstractname (default 'Abstract') is used as the title for the abstract environment and is set using the \abstractnamefont. The body of the abstract is typeset using the \abstracttextfont. These two commands can be redefined to change the fonts if you wish. The default definitions are

\newcommand{\abstractnamefont}{\normalfont\small\bfseries}
\newcommand{\abstracttextfont}{\normalfont\small}

```
\absleftindent \absrightindent \absparindent \absparsep
```

This version of abstract uses a list environment for typesetting the text. These four lengths can be changed (via \setlength or \addtolength) to adjust the left and right margins, the paragraph indentation, and the vertical skip between paragraphs in this environment. The default values depend on whether or not the twocolumn class option is used. The general layout parameters for lists are illustrated in Figure 8.2.

```
\abslabeldelim{\langle text \rangle}
```

If the \abstractrunin declaration has been given, the heading is typeset as a run-in heading. That is, it is the first piece of text on the first line of the abstract text. The  $\langle text \rangle$  argument of \abslabeldelim is typeset immediately after the heading. By default it is defined to do nothing, but if you wanted, for example, the \abstractname to be followed by a colon and some extra space you could specify

\abslabeldelim{:\quad}

```
\absnamepos
```

If the \abstractrunin declaration is not used then the heading is typeset in its own environment, specified by \absnamepos. The default definition is

\newcommand{\absnamepos}{center}

It can be defined to be one of flushleft, center, or flushright to give a left, centered or right aligned heading; or to any other appropriate environment which is supported by a used package.

```
\abstitleskip
```

With the \abstractrunin declaration a horizontal space of length \abstitleskip is typeset before the heading. For example, if \absparindent is non-zero, then:

```
\setlength{\abstitleskip}{-\absparindent}
```

will typeset the heading flush left.

Without the \abstractrunin declaration, \abstitleskip is aditional vertical space (either positive or negative) that is inserted between the abstract name and the text of the abstract.

### 5.2 One column abstracts

The usual advice [FAQ] about creating a one-column abstract in a twocolumn document is to write code like this:

The class provides a onecolabstract environment that you can use for a one column abstract in a twocolumn document, and it is used like this:

The command \saythanks ensures that any \thanks texts from an earlier \maketitle are printed out as normal.

If you want, you can use the onecolabstract environment in place of the abstract environment — it doesn't have to be within the optional argument of the \twocolumn command. In fact, onecolabstract internally uses abstract for the typesetting.

### 6

### Document divisions

For this chapter the *pedersen* chapterstyle has been used in order to demonstrate how it appears.

In this chapter I first discuss the various kinds of divisions within a book and the commands for typesetting these.

After that I describe the class methods for modifying the appearance of the chapter and other sectional titles (subheads). The facilities described here provide roughly the same as you would get if you used the titlesec [Bez99] and sectsty [McD98] packages together; the commands are different, though.

### 6.1 Logical divisions

As described earlier there are three main logical divisions to a book; the front matter, main matter and back matter. There are three LaTeX commands that correspond to these, namely \frontmatter, \mainmatter and \backmatter.

```
\frontmatter\frontmatter*
```

The \frontmatter declaration sets the folios to be printed in lowercase roman numerals, starts the page numbering from i, and prohibits any numbering of sectional divisions. Caption, equations, etc., will be numbered continuously. The starred version of the command, \frontmatter\*, is similar to the unstarred version except that it makes no changes to the page numbering or the print style for the folios. Even though \chapter and other divisions will not be numbered their titles will be added to the ToC.

If it is to be used at all, the \frontmatter declaration should come before any text is set, otherwise the pagination scheme will be in disarray (in books pagination starts on the first page).

```
\mainmatter \mainmatter*
```

The \mainmatter declaration, which is the default at the start of a document, sets the folios to be printed in arabic numerals, starts the page numbering from 1, and sections and above will be numbered. Float captions, equations, etc., will be numbered per chapter. The starred version of the command, \mainmatter\*, is similar to the unstarred version except that it makes no changes to the page numbering or the print style for the folios.

Please note that \mainmatter will not only change the folio numbers to arabic and restart it at 1, it will also make sure it starts at the next coming recto page. (Even when running under the openany option).

```
\backmatter
```

The \backmatter declaration makes no change to the pagination or folios but does prohibit sectional division numbering, and captions, etc., will be numbered continuously.

If you have other types of floats that might be used in the front-main- or backmatter, then you can change some internals to add these to be numbered in the same manner as we do with figures and tables. They are defined as

\newcommand\@memfront@floats{%
 \counterwithout{figure}{chapter}

```
\counterwithout{table}{chapter}}
\newcommand\@memmain@floats{%
  \counterwithin{figure}{chapter}
  \counterwithin{table}{chapter}}
\newcommand\@memback@floats{%
  \counterwithout{figure}{chapter}
  \counterwithout{table}{chapter}
  \setcounter{figure}{0}
  \setcounter{table}{0}}
```

The macros can also be changed in case you want to have consecutive figure numbering throughout, i.e.,

```
\makeatletter
\counterwithout{figure}{chapter}
\counterwithout{table}{chapter}
\renewcommand\@memfront@floats{}
\renewcommand\@memmain@floats{}
\newcommand\@memback@floats{}
\makeatother
```

in the preamble.

### 6.2 Sectional divisions

The memoir class lets you divide a document up into eight levels of named divisions. They range from book, part through chapter and down to sub-paragraph. A particular sectional division is specified by one of the commands \book, \part, \chapter, \section, \subsection, which is probably as deep as you want to go. If you really need finer divisions, they are \subsubsection, \paragraph and lastly \subparagraph. The sectional commands, except for \book and \part, have the same form, so rather than describing each one in turn I will use \section as model for all but the two exceptions.

```
\label{eq:local_continuous} $$\operatorname{(title)} (\theta) = (\theta) + (\theta)
```

There are two forms of the command; the starred version is simpler, so I'll describe its effects first—it just typesets  $\langle title \rangle$  in the document in the format for that particular sectional division. Like the starred version, the plain version also typesets  $\langle title \rangle$  in the document, but it may be numbered. Diferent forms of the division title are available for the Table of Contents (ToC) and a running header, as follows:

- No optional argument: \(\lambda title \rangle\) is used for the division title, the ToC title and a page header title.
- One optional argument:  $\langle title \rangle$  is used for the division title;  $\langle toc\text{-}title \rangle$  is used for the ToC title and a page header title.
- Two optional arguments:  $\langle title \rangle$  is used for the division title;  $\langle toc\text{-}title \rangle$  is used for the ToC title;  $\langle head\text{-}title \rangle$  is used for a page header title.

A \section command restarts the numbering of any \subsections from one. For most of the divisions the  $\langle title \rangle$  is put on the page where the command was issued. The \book, \part and \chapter commands behave a little differently.

The \book and \part commands are simpler and both behave in the same way.

```
\book{\langle title \rangle}
\part{\langle title \rangle}
```

The  $\book{\langle title \rangle}$  command puts the book name (default Book), number and  $\t title \rangle$  on a page by itself. The numbering of books has no effect on the numbering of  $\arrowvert$  or  $\arrowvert$  chapters. Similarly the  $\arrowvert$  command puts the part name (default Part), number and  $\arrowvert$  on a page by itself. The numbering of parts has no effect on the numbering of  $\arrowvert$  chapters.

Later I'll give a list of LaTeX's default names, like Part.

```
\chapter[\langle toc\text{-}title \rangle] [\langle head\text{-}title \rangle] \{\langle title \rangle\} \\ \chapter*[\langle head\text{-}title \rangle] \{\langle title \rangle\}
```

The \chapter command starts a new page and puts the chapter name (default Chapter), number and  $\langle title \rangle$  at the top of the page. It restarts the numbering of any \sections from one. If no optional arguments are specified,  $\langle title \rangle$  is used as the ToC entry and for any page headings. If one optional argument is specified this is  $\langle toc\text{-}title \rangle$  and is used for the ToC entry and for page headings. If both optional arguments are specified the  $\langle head\text{-}title \rangle$  is used for page headings.

The \chapter\* command starts a new page and puts  $\langle title \rangle$  at the top of the page. It makes no ToC entry, changes no numbers and by default changes no page headings. If the optional  $\langle head\text{-}title \rangle$  argument is given, this is used for page headings. Use of the optional argument has the side-effect that the secnumdepth counter is set to maxsecnumdepth (see below for an explanation of these).

When the article option is in effect, however, things are slightly different. New chapters do not necessarily start on a new page. The \mainmatter command just turns on sectional numbering and starts arabic page numbering; the \backmatter command just turns off sectional numbering. The \tableofcontents command and friends, as well as any other commands created via \newlistof, always¹ call thispagestyle{chapter}. If you are using the article option you will probably want to ensure that the chapter pagestyle is the same as you normally use for the document.

Unlike the standard classes the  $\langle title \rangle$  is typeset ragged right. This means that if you need to force a linebreak in the  $\langle title \rangle$  you have to use \newline instead of the more usual \\. For instance

\section{A broken\newline title}

In the standard classes a \section or other subhead that is too close to the bottom of a page is moved to the top of the following page. If this happens and \flushbottom is in effect, the contents of the short page are stretched to make the last line flush with the bottom of the typeblock.

```
\raggedbottomsection
\normalbottomsection
\bottomsectionskip
\bottomsectionpenalty
```

<sup>&</sup>lt;sup>1</sup>This is a consequence of the internal timing of macro calls.

The \raggedbottomsection declaration will typeset any pages that are short because of a moved subhead as though \raggedbottom was in effect for the short page; other pages are not affected. The length \bottomsectionskip controls the amount of stretch on the short page. Setting it to zero allows the last line to be flush with the bottom of the typeblock. The default setting of 10mm appears to remove any stretch. \bottomsectionpenalty control the penalty it costs to make a page break at this point. The detault is zero as the stretch is usually enough, by setting it to a negative integer one can be a bit more incouraging regarding a possible page break.

The declaration \normalbottomsection, which is the default, cancels any previous \raggedbottomsection declaration.

### 6.2.1 Appendices

Appendices normally come after the main text and are often considered to be part of the \mainmatter as they are normally numbered (the \backmatter declaration turns off all sectional numbering).

```
\appendix
\appendixname
```

The \appendix declaration changes the numbering of chapters to an alphabetic form and also changes the names of chapters from \chaptername (default Chapter) to the value of \appendixname (default Appendix). Thus, the first and any subsequent \chapters after the \appendix command will be 'Appendix A ...', 'Appendix B ...', and so on. That is as far as the standard classes go but this class provides more ways of dealing with appendices.

```
\appendixpage
\appendixpage*
\appendixpagename
```

The \appendixpage command generates a part-like page (but no name or number) with the title given by the value of \appendixpagename (default Appendices). It also makes an entry in the ToC using \addappheadtotoc (see below). The starred version generates the appendix page but makes no ToC entry.

```
\addappheadtotoc
\appendixtocname
```

The command \addappheadtotoc adds an entry to the ToC. The title is given by the value of \appendixtocname (default Appendices).

```
\begin{appendices} text \end{appendices}
```

The appendices environment acts like the \appendix command in that it resets the numbering and naming of chapters. However, at the end of the environment, chapters are restored to their original condition and any chapter numbers continue in sequence as though the appendices environment had never been there.

```
\begin{subappendices} text \end{subappendices}
\namedsubappendices \unnamedsubappendices
```

The subappendices environment can be used to put appendices at the end of a chapter. Within the environment \section starts a new sub-appendix. You may put \addappheadtotoc at the start of the environment if you want a heading entry in the ToC.

Table 6.1: Division levels

Division	Level
\book	-2
\part	-1
\chapter	0
\section	1
\subsection	2
\subsubsection	3
\paragraph	4
\subparagraph	5

If you put the declaration \namedsubappendices before the subappendices environment, the sub-appendix number in the body of the document will be preceded by the value of \appendixname. The \unnamedsubappendices declaration, which is the default, may be used to switch off this behaviour.

**Caveat:** The implementation of the named subappendices make use of \setsecnumformat, thus if you have used this command to change the formating of the section number you will need to re-do this in a special manner inside the subappendices environemt. Something like this (where a user wanted to use old style numerials for sectioning numbers)

```
\begin{subappendices}
\setsecnumformat{\sectionname\
   \oldstylenums{\csname the#1\endcsname\quad}}
```

The macro sectionname is a special macro that only lives inside the subappendices environment and is only available when \namedsubappendices is applied.

### 6.3 Numbering

Each type of sectional division has an associated *level* as shown in Table 6.1. Divisions are numbered if the value of the secnumdepth counter is equal to or greater than their level. For example, with

\setcounter{secnumdepth}{2}

then subsections up to book will be numbered.

```
\label{eq:local_second} $$\operatorname{\colorer{second}} $$ \mbox{\colorer{second}} $$ \mbox{\colorer{second}} $$
```

Instead of having to remember the levels if you want to change what gets numbered you can use the \setsecnumdepth command. It sets secnumdepth so that divisions \( \secname \) and above will be numbered. The argument \( \secname \) is the name of a sectional division without the backslash. For example, to have subsections and above numbered:

\setsecnumdepth{subsection}

You can also use all or none for  $\langle secname \rangle$  which will either turn on numbering for all levels, or turn off numbering altogether.

When used in the preamble \setsecnumdepth also calls \maxsecnumdepth, which is the numbering level used once \mainmatter is called. You can use \setsecnumdepth anywhere in the \mainmatter to (temporarily) change the numbering level.

By default, the class sets:

```
\setsecnumdepth{section} \maxsecnumdepth{section}
```

The \frontmatter commands sets the numbering level to none. The commands \mainmatter and \mainmatter\* set the numbering level to the value specified by \maxsecnumdepth.

The number setting commands come from the tocvsec2 package [Wil99b].

### 6.4 Book and part headings

Book and part headings *always* start on a new page with the *book* and *part* pagestyles, respectively. The typical book and part heading consists of the name (e.g., 'Book' or 'Part') followed by a number represented as an uppercase Roman numeral. There is a vertical space after which the title is printed. Finally a new page is started.

Several aspects of the typesetting of the \book and \part title are configurable. Ignoring details, such as the optional argument, the code for printing \part headings looks like this:

```
\newcommand{\part}[1]{%
                          % THIS IS A VERY SIMPLIFIED VERSION
                          % start a new recto page
 \cleardoublepage
 \thispagestyle{part}
                          % set the page style
 \beforepartskip
                          % space before Name and Number
 \printpartname\partnamenum\printpartnum
 \midpartskip
                          % space after Name and Number
 \printparttitle{#1}
                          % print the title
                          % finish off
 \partpageend}
\newcommand{\partpageend}{% THIS IS SIMPLIFIED
 \afterpartskip
 % ifblankpage then blank next page and restore twocolumn if necessary
```

The code for \book headings is similar.

The general layout for \book, \part and \chapter headings is similar and you may wish to refer to Figure 6.1 which, although it shows the vertical layout for a chapter head, is also applicable to \book and \part heads with appropriate changes in the names of the commands.

```
\beforebookskip \afterbookskip \beforepartskip \afterpartskip
```

These commands effectively control the spacing before and after the book and part titles. Their default definitions are:

```
\newcommand*{\beforebookskip}{\null\vfil}
\newcommand*{\afterbookskip}{\vfil\newpage}
```

```
\newcommand*{\beforepartskip}{\null\vfil}
\newcommand*{\afterpartskip}{\vfil\newpage}
```

Together, these vertically center any typesetting on the page, and then start a new page. To move the \part title upwards on the page, for example, you could do:

```
\renewcommand*{\beforepartskip}{\null\vskip Opt plus 0.3fil}
\renewcommand*{\afterpartskip}{\vskip Opt plus 0.7fil \newpage}
```

```
\midbookskip
\midpartskip
```

The macros \midbookskip and \midpartskip are the spacings between the number lines and the titles. The default definitions are:

```
\newcommand{\midbookskip}{\par\vspace 2\onelineskip}
\newcommand{\midpartskip}{\par\vspace 2\onelineskip}
```

and they can be changed.

```
\printbookname\booknamefont
\booknamenum
\printbooknum\booknumfont
\printpartname\partnamefont
\partnamenum
\printpartnum\partnumfont
```

The macro \printbookname typesets the book name (the value of \bookname) using the font specified by \booknamefont. The default is the \bfseries font in the \huge size. Likewise the book number is typeset by \printbooknum using the font specified by \booknumfont, which has the same default as \booknamefont. The macro \booknamenum, which is defined to be a space, is called between printing the book name and the number. All these can be changed to obtain different effects.

Similarly, the macro \printpartname typesets the part name (the value of \partname) using the font specified by \partnamefont. The default is the \bfseries font in the \huge size. Likewise the part number is typeset by \printpartnum using the font specified by \partnumfont, which has the same default as \partnamefont. The macro \partnamenum, which is defined to be a space, is called between printing the part name and the number.

For example, to set a \part in a large sans font with the part name flush left:

or to only print the part number in the default font:

```
\renewcommand{\printpartname}{}
\renewcommand{\partnamenum}{}
```

```
\label{localization} $$  \printbooktitle{$\langle title\rangle$} \printparttitle{$\langle title\rangle$} \printparttitle{$\langle title\rangle$} $$
```

A book's title is typeset by \printbooktitle using the font specified by \booktitlefont. By default this is a \bfseries font in the \Huge size. This can be changed to have, say, the title set raggedleft in a small caps font by

\renewcommand{\booktitlefont}{\normalfont\Huge\scshape\raggedleft}

Similarly a part's title is typeset by \printparttitle using the font specified by \parttitlefont. By default this is a \bfseries font in the \Huge size.

The \parttitlefont font is also used by \appendixpage, or its starred version, when typesetting an appendix page.

```
\label{local_problem} $$ \ \partmark{\langle title \rangle}$
```

The \book code includes \bookpagemark{ $\langle title \rangle$ } for capturing the  $\langle title \rangle$  of the book division if it is going to be used, for example, in page headers. Its definition is simply:

```
\newcommand*{\bookpagemark}[1]{}
```

There is the corresponding \partmark for the title of \part divisions.

```
\bookpageend \bookblankpage \nobookblankpage \partpageend \partblankpage \nopartblankpage
```

The macro \bookpageend finishes off a book title page. It first calls \afterbookskip. If the \nobookblankpage is in effect it does nothing more. If the declaration \bookblankpage (the default) is in effect then it finishes the current page, outputs a blank page and then, if twocolumn typesetting was in effect before \book then it restores twocolumn typesetting. The macro \partpageend performs similar functions for \part pages.

So to add something on the back side of a  $\part page$  (assuming twoside) use something similar to

```
\nopartblankpage
\part{Title of the Part}
\thispagestyle{simple}
Text on the following (normally blank page)
\clearpage
```

Alternatively you can redefine \partpageend.

If you use the declaration \nopartblankpage (or \nobookblankpage) then you are responsible for setting everything correctly to end off the \part (or \book) page. This is the default definition of \partpageend (that for \bookpageend is similar):

```
\newcommand{\partpageend}{%
  \afterpartskip
  \ifm@mnopartnewpage% set by \(no)partblankpage
  \else% default finish off
  \if@twoside
  \if@openright% output blank page
  \null
  \thispagestyle{afterpart}%
  \newpage
  \fi
  \fi
  \fi
```

```
\if@tempswa% true if twocolumn was being used
  \twocolumn
\fi}
```

Here with the default definitions, \afterpartskip ends off the \part page, and then the rest of the code in \partpageend takes care of typesetting the blank back side of the \part page (or send us back to twocolumn mode).

If on the other hand we actually want to write something below the part title on the \part page, then we need a different route. The 'air' above and below the part title is by default defined as

```
\newcommand*{\beforepartskip}{\null\vfil}
\newcommand*{\afterpartskip}{\vfil\newpage}
```

Thus we need to redefined this such that it does not change the page and such that it add useful spacing above and below the part titling. Something like this may do the trick

```
\makeatletter
\newcommand*{\beforepartskip}{\null\vskip4cm}
\newcommand*{\afterpartskip}{\par\vskip1cm%
```

\@afterindentfalse\@afterheading} \makeatother

### 6.4.1 Leadpage

```
\label{lem:leadpage} $$ \operatorname{page-style} {\langle cmdname \rangle} {\langle title \rangle} $$ \renewleadpage $[\langle page-style \rangle] {\langle cmdname \rangle} {\langle title \rangle}$$
```

The \newleadpage<sup>2</sup> command defines a macro \cmdname that when called will typeset an Appendixpage-like page (see §6.2.1) with a title  $\langle title \rangle$  using the  $\langle page\text{-}style \rangle$  as the pagestyle for the page. The default is the *empty* pagestyle. The macro \renewleadpage redefines an existing leadpage command. \cmdname will add an entry to the TOC, the similarly defined \cmdname\* will not.

As an example:

```
\newleadpage{plates}{Picture Gallery}
```

creates the new command \plates which when called generates an unnumbered part-like page with the title **Picture Gallery**.

```
\leadpagetoclevel
```

When  $\(re)$  newleadpage is used the resulting command adds  $\langle title \rangle$  to the ToC as though it was an unnumbered  $\ledown$  entry, whose definition is

```
\newcommand*{\leadpagetoclevel}{chapter}
```

If you wished them to be entered like a \part header then simply:

```
\renewcommand*{\leadpagetoclevel}{part}
```

The layout of the page matches that for unnumbered \part pages, and internally the resulting commands use \partmark in case you wish to capture the  $\langle title \rangle$  to use in running headers.

 $<sup>^2{\</sup>rm The}$  suggestions for this came from Danie Els and Lars Madsen.

### 6.5 Chapter headings

The chapter headings are configurable in much the same way as book or part headings, but in addition there are some built in chapter styles that you may wish to try, or define your own.

Chapters, except when the article class option is used, *always* start on a new page with the *chapter* pagestyle. The particular page, recto or verso, that they start on is mainly controlled by the class options. If the oneside option is used they start on the next new page, but if the twoside option is used the starting page may differ, as follows.

openright The chapter heading is typeset on the next recto page, which may leave a blank verso leaf.

openleft The chapter heading is typeset on the next verso page, which may leave a blank recto leaf.

openany The chapter heading is typeset on the next page and there will be no blank leaf.

```
\openright \openleft \openany
```

These three declarations have the same effect as the options of the same name. They can be used anywhere in the document to change the chapter opening page.

Ignoring many details, like the optional arguments, the code for printing a \chapter heading is similar to that for \book and \part (the \chapterhead command below is *not* part of the class).

The general layout is shown in Figure 6.1.

```
\clearforchapter
```

The actual macro that sets the opening page for a chapter is \clearforchapter. The class options openright, openleft and openany (or their macro equivalents \openright, \openleft and \openany) define \clearforchapter to be respectively (see §18.13) \cleartorecto, \cleartoverso and \clearpage. You can obviously change \clearforchapter to do more than just start a new page.

```
\memendofchapterhook
```

Where \clearforchapter comes at the very beginning, \memendofchapterhook comes at the very end of the \chapter command. It does nothing by default, but could be redefined to insert, say, \clearpage:

```
\makeatletter
\renewcommand\memendofchapterhook{%
   \clearpage\m@mindentafterchapter\@afterheading}
```

```
This is the start of the after-heading text which continues on ... second line of text following the heading ...
```

Figure 6.1: Class layout parameters for chapter titles. Working with \beforechapskip need a little thought, see the text.

#### \makeatother

Some books have the chapter headings on a verso page, with perhaps an illustration or epigraph, and then the text starts on the opposite recto page. The effect can be achieved like:

```
\chapterheadstart \beforechapskip
\afterchapternum \midchapskip
\afterchaptertitle \afterchapskip
```

The macro \chapterheadstart is called just before printing a chapter name and number. By default it inserts \beforechapskip space (default 50pt).

The macro \afterchapternum is called just after printing a chapter number. By default it inserts \midchapskip space (default 20pt).

The macro \afterchaptertitle is called just after printing a chapter title. By default it inserts \afterchapskip space (default 40pt).

The lengths \beforechapskip, \midchapskip and \afterchapskip can all be changed by \setlength or \addtolength. Though as mentioned in Figure 6.1 they need some explanation:

#### \beforechapskip

See Figure 6.1. The actual distance between the first baseline of the chapter stuff to the top of the text block is \beforechapskip + \topskip + \baselineskip. But because the implementation of \chapter (via \chapterheadstart) make use of \vspace\*, getting rid of \beforechapskip a strange endeavour. If you want to avoid any space before the first text in the chapter heading, use

\setlength\beforechapskip{-\baselineskip}

or redefine \chapterheadstart to do nothing.

\midchapskip

\afterchapskip

for both, one has to add \baselineskip to get the distance baseline to baseline.

```
\printchaptername \chapnamefont
\chapternamenum
\printchapternum \chapnumfont
```

The macro \printchaptername typesets the chapter name (default Chapter or Appendix) using the font specified by \chapnamefont. The default is the \bfseries font in the \huge size. Likewise the chapter number is typeset by \printchapternum using the font specified by \chapnumfont, which has the same default as \chapnamefont. The macro \chapternamenum, which is defined to be a space, is called between printing the chapter name and the number.

```
\printchaptertitle{\langle title \rangle} \chaptitlefont
```

The title is typeset by \printchaptertitle using the font specified by \chaptitlefont. By default this is a \bfseries font in the \Huge size.

```
\printchapternonum
```

If a chapter is unnumbered, perhaps because it is in the \frontmatter or because \chapter\* is used, then when printing the command \printchapternonum is called instead of printing the name and number, as illustrated below:

By default the first paragraph following a \chapter is *not* indented, this can be controlled by

```
\indentafterchapter
\noindentafterchapter
```

The default is not to indent the first paragraph following a \chapter.

```
\insertchapterspace
```

By default a \chapter inserts a small amount of vertical space into the List of Figures and List of Tables. It calls \insertchapterspace to do this. The default definition is:

```
\newcommand{\insertchapterspace}{%
  \addtocontents{lof}{\protect\addvspace{10pt}}%
  \addtocontents{lot}{\protect\addvspace{10pt}}%
}
```

If you would prefer no inserted spaces then

\renewcommand{\insertchapterspace}{}

will do the job. Different spacing can be inserted by changing the value of the length arguments to \addvspace.

By making suitable changes to the above macros you can make some simple modifications to the layout.

#### 6.5.1 Defining a chapter style

The class provides many ways in which you can implement your designs for chapter headings.

```
\chapterstyle{\langle style \rangle}
```

The macro \chapterstyle is rather like the \pagestyle command in that it sets the style of any subsequent chapter headings to be  $\langle style \rangle$ .

The class provides some predefined chapter styles, including the *default* style which is the familiar LaTeX book class chapter headings style. To use the chapterstyle *fred* just issue the command

```
\chapterstyle{fred}
```

Different styles can be used in the same document.

The simpler predefined styles include:

default The normal LaTeX book class chapter styling; shown in Figure B.1.

section The heading is typeset like a section; that is, there is just the number and the title on one line. This is illustrated in Figure B.2.

*hangnum* Like the *section* style except that the chapter number is put in the margin, as shown in Figure B.3.

*companion* This produces chapter headings like those of the *LaTeX Companion* series of books. An example is in Figure B.4.

article The heading is typeset like a \section heading in the article class. This is similar to the section style but different fonts and spacings are used, as shown in Figure B.5.

reparticle When the article class option is used the default chapter and section styles are close, but not identical, to the corresponding division heads in the article class. The reparticle chapterstyle makes \chapter replicate the appearance of \section in the article class.

If you use only the predefined chapterstyles there is no need to plough through the rest of this section, except to look at the illustrations of the remaining predefined chapterstyles shown a little later.

The various macros shown in the \chapterhead code above are initially set up as (the code is known as the *default* style):

```
\newcommand{\chapterheadstart}{\vspace*{\beforechapskip}}
\newcommand{\printchaptername}{\chapnamefont \@chapapp}
\newcommand{\chapternamenum}{\space}
\newcommand{\printchapternum}{\chapnumfont \thechapter}
\newcommand{\afterchapternum}{\par\nobreak\vskip \midchapskip}
\newcommand{\printchapternonum}{}
\newcommand{\printchaptertitle}[1]{\chaptitlefont #1}
\newcommand{\afterchaptertitle}{\par\nobreak\vskip \afterchapskip}\
\newcommand{\chapnamefont}{\normalfont\huge\bfseries}
\newcommand{\chapnumfont}{\normalfont\huge\bfseries}
\newcommand{\chaptitlefont}{\normalfont\huge\bfseries}
\setlength{\beforechapskip}{50pt}
\setlength{\midchapskip}{20pt}
\setlength{\afterchapskip}{40pt}
```

(The mysterious \@chapapp is the internal macro that LaTeX uses to store normally the chapter name.<sup>3</sup> It will normally have different values, set automatically, when typesetting a chapter in the main body (e.g., Chapter) or in the appendices where it would usually be set to Appendix, but you can specify these names yourself.)

A new style is specified by changing the definitions of this last set of macros and/or the various font and skip specifications.

```
\verb|\makechapterstyle| \{ \langle style \rangle \} \{ \langle text \rangle \}
```

Chapter styles are defined via the \makechapterstyle command, where  $\langle style \rangle$  is the style being defined and  $\langle text \rangle$  is the LaTeX code defining the style. Please note that the  $\langle text \rangle$  always start by resetting macros and lengths to the values shown above. That is what you actually get is is the same as

To start things off, the *default* chapter style, which mimics the chapter heads in the standard book and report classes, as it appears in memoir.cls:

```
\makechapterstyle{default}{}
\chapterstyle{default}
```

- since the *default* style *is* the initial value of \makechapterstyle. The actual code is seen above.

 $<sup>^3</sup>$ Remember, if you use a macro that has an @ in its name it must be in a place where @ is treated as a letter.

As an example of setting up a simple chapterstyle, here is the code for defining the *section* chapterstyle. In this case it is principally a question of eliminating most of the printing and zeroing some spacing.

```
\makechapterstyle{section}{%
  \renewcommand*{\printchaptername}{}
  \renewcommand*{\chapternamenum}{}
  \renewcommand*{\chapnumfont}{\chaptitlefont}
  \renewcommand*{\printchapternum}{\chapnumfont \thechapter\space}
  \renewcommand*{\afterchapternum}{}
}
```

In this style, \printchaptername is vacuous, so the normal 'Chapter' is never typeset. The same font is used for the number and the title, and the number is typeset with a space after it. The macro \afterchapternum is vacuous, so the chapter title will be typeset immediately after the number.

In the standard classes the title of an unnumbered chapter is typeset at the same position on the page as the word 'Chapter' for numbered chapters. The macro \printchapternonum is called just before an unnumbered chapter title text is typeset. By default this does nothing but you can use \renewcommand to change this. For example, if you wished the title text for both numbered and unnumbered chapters to be at the same height on the page then you could redefine \printchapternonum to insert the amount of vertical space taken by any 'Chapter N' line. For example, as \printchapternonum is vaucuous in the default chapterstyle the vertical position of a title depends on whether or not it is numbered.

The *hangnum* style, which is like *section* except that it puts the number in the margin, is defined as follows:

```
\makechapterstyle{hangnum}{%
  \renewcommand*{\chapnumfont}{\chaptitlefont}
  % allow for 99 chapters!
  \settowidth{\chapindent}{\chapnumfont 999}
  \renewcommand*{\printchaptername}{}
  \renewcommand*{\chapturnamenum}{}
  \renewcommand*{\chapturnamenum}{%
  \renewcommand*{\printchapternum}{%
  \noindent\llap{\makebox[\chapindent][1]{%
  \chapnumfont \thechapter}}
  \renewcommand*{\afterchapternum}{}
}
```

The chapter number is put at the left of a box wide enough for three digits. The box is put into the margin, via \llap, for typesetting. The chapter title is then typeset, starting at the left margin.

```
\chapindent
```

The length \chapindent is provided for use in specifying chapterstyles, but you could use it for any other purposes.

The definition of the *companion* chapterstyle is more complicated.

```
\makechapterstyle{companion}{%
```

```
\renewcommand*{\chapnamefont}{\normalfont\LARGE\scshape}
\renewcommand*{\chapnumfont}{\normalfont\Huge}
\renewcommand*{\printchaptername}{%
   \raggedleft\chapnamefont \@chapapp}
\setlength{\chapindent}{\marginparsep}
\addtolength{\chapindent}{\marginparwidth}
\renewcommand{\printchaptertitle}[1]{%
   \begin{adjustwidth}{}-\chapindent}
   \raggedleft \chaptitlefont ##1\par\nobreak
   \end{adjustwidth}}
}
```

As shown in Figure B.4 the chapter name is in small caps and set flushright. The title is also set flushright aligned with the outermost part of the marginal notes. This is achieved by use of the adjustwidth environment<sup>4</sup> to make LaTeX think that the typeblock is locally wider than it actually is.

## 6.5.2 Further chapterstyles

The class provides more chapterstyles, which are listed here. Some are mine and others are from postings to crr by memoir users. I have modified some of the posted ones to cater for things like appendices, multiline titles, and unnumbered chapters which were not considered in the originals. The code for some of them is given later to help you see how they are done. Separately, Lars Madsen has collected a wide variety of styles [Mad06] and shows how they were created.

If you want to try several chapterstyles in one document, request the *default* style before each of the others to ensure that a previous style's changes are not passed on to a following one.

*bianchi* This style was created by Stefano Bianchi<sup>5</sup> and is a two line centered arrangement with rules above and below the large bold sanserif title line. The chapter number line is in a smaller italic font. An example is in Figure B.6.

bringhurst The bringhurst chapterstyle described in the manual and illustrated in Figure B.7.

brotherton A very simple style designed by William Adams<sup>6</sup> for the science fiction novel *Star Dragon* by Mike Brotherton. The novel is freely downloadable from Brotherton's web site. The style is the same as the *default* except that the number is spelt out in words. It is demonstrated in Figure B.8. In the novel the chapters are actually untitled i.e., via \chapter{}.

chappell The name and number are centered above a rule and the title in italics is below, again centered. It is illustrated in Figure B.9.

*crosshead* The number and title are centered and set with a large bold font. It is illustrated in Figure B.10.

culver A chapter style I created for Christopher Culver<sup>7</sup> based on the format of 'ancient' texts. It is one line, centered, bold and with the number printed as Roman numerals, as shown in Figure B.11.

<sup>&</sup>lt;sup>4</sup>See §8.5.

<sup>&</sup>lt;sup>5</sup>ctt, New chapter style: chapter vs chapter\*, 2003/12/09

 $<sup>^6</sup>$ ctt, An example of a novel?, 2006/12/09

 $<sup>^7</sup>$ ctt, "Biblical" formatting, how?, 2004/03/29

He also wanted sections to just start with the number and the text to immediately follow on the same line. That can be accomplished like this:

```
\renewcommand*{\thesection}{\arabic{section}}
\renewcommand*{\section}[1]{%
 \refstepcounter{section}%
 \par\noindent
 \textbf{\thesection.}%
 \space\nolinebreak}
```

- dash A simple two line centered chapterstyle. There is a short dash on either side of the number and a slightly larger version of the regular font is used for both the number and the title. This style is shown in Figure B.12.
- *demo*2 A two line chapterstyle with a large sanserif title; the number is above, centered and written (e.g., Six instead of 6) in a bold font. There are rules above and below the title. An example is shown in Figure B.13.
- *demo*<sup>3</sup> The chapterstyle used in this document. It is a modified version of the *demo*<sup>2</sup> chapterstyle, using an italic rather than bold font for the number.
- dowding A centered style where the name and number are set in a bold font, with the number spelled out. The title is below in a large italic font. The style is based on the design used in Dowding's *Finer Points* [Dow96]. It is illustrated in Figure B.14.
  - ell A raggedleft sanserif chapterstyle. The number line is separated from the title by rules like an 'L' on its side and the number is placed in the margin, as shown in Figure B.15. I will probably use this in my next book.
  - *ger* This style was created by Gerardo Garcia<sup>8</sup> and is a two line, raggedright, large bold style with rules above and below. It is demonstrated in Figure B.16.
- *komalike* A section-like style set with a sans serif type. It is like that in the scrbook class (part of the KOMA bundle). It is illustrated in Figure B.17.
  - *lyhne* A style created by Anders Lyhne<sup>9</sup> and shown in Figure B.18 where the raggedleft sanserif title is between two rules, with the name and number above. I modified the original to cater for unnumbered chapters.

**Caveat:** The *lyhne* style requires the graphicx package.

*madsen* This was created by Lars Madsen<sup>10</sup> and is shown in Figure B.19. It is a large sanserif raggedleft style with the number in the margin and a rule between the number and title lines.

**Caveat:** The *madsen* style requires the graphicx package.

- *ntglike* A smaller version of the standard chapterstyle; it is like that in the NTG classes (boek class) developed by the Dutch TeX User Group. It is illustrated in Figure B.20.
- pedersen This was created by Troels Pedersen<sup>11</sup> and requires the graphicx package, and, to get the full effect, the color package as well. The title is raggedright in large italics while

 $<sup>^8\</sup>mathrm{ctt},\,\mathrm{Fancy}$  Headings, Chapter Headings, 2002/04/12

<sup>&</sup>lt;sup>9</sup>ctt, Glossary, 2006/02/09

 $<sup>^{10}</sup>$ ctt, New chapter style: chapter vs chapter\*, 2003/12/09

 $<sup>^{11}</sup>$ ctt, Chapter style, 2006/01/31

the number is much larger and placed in the righthand margin (I changed the means of placing the number). The head of this chapter is set with the *pedersen* style, because it cannot be adequately demonstrated in an illustration.

**Caveat:** The *pedersen* style requires the graphicx package.

- southall This style was created by Thomas Dye. It is a simple numbered heading with the title set as a block paragraph, and with a horizontal rule underneath. It is illustrated in Figure B.21.
  - tandh A simple section-like style in a bold font. It is based on the design used in the Thames & Hudson *Manual of Typography* [McL80] and is illustrated in Figure B.22.
- thatcher A style created by Scott Thatcher<sup>12</sup> which has the chapter name and number centered with the title below, also centered, and all set in small caps. There is a short rule between the number line and the title, as shown in Figure B.23. I have modified the original to cater for multiline titles, unnumbered chapters, and appendices.
  - *veelo* This style created by Bastiaan Veelo is shown in Figure B.24 and is raggedleft, large, bold and with a black square in the margin by the number line.

**Caveat:** The *veelo* style requires the graphicx package.

- *verville* A chapterstyle I created for Guy Verville<sup>13</sup>. It is a single line, large centered style with rules above and below, as in Figure B.25. Unlike my posted version, this one properly caters for unnumbered chapters.
- wilsondob A one line flushright (raggedleft) section-like style in a large italic font. It is based on the design used in Adrian Wilson's *The Design of Books* [Wil93] and is illustrated in Figure B.26.

The code for some of these styles is given in section B.1 within the Showcase Appendix. For details of how the other chapter styles are defined, look at the documented class code. This should give you ideas if you want to define your own style.

Note that it is not necessary to define a new chapterstyle if you want to change the chapter headings — you can just change the individual macros without putting them into a style.

# 6.5.3 Chapter precis

Some old style novels, and even some modern text books, <sup>14</sup> include a short synopsis of the contents of the chapter either immediately after the chapter heading or in the ToC, or in both places.

```
\cline{Constraints}
```

The command \chapterprecis prints its argument both at the point in the document where it is called, and also adds it to the .toc file. For example:

```
...
\chapter{}% first chapter
\chapterprecis{Our hero is introduced; family tree; early days.}
```

 $<sup>^{12}\</sup>mathrm{ctt},$  memoir: chapter headings capitalize math symbols, 2006/01/18

 $<sup>^{13}</sup>$ ctt, Headers and special formatting of sections, 2005/01/18

<sup>&</sup>lt;sup>14</sup>For example, Robert Sedgewick, Algorithms, Addison-Wesley, 1983.

. . .

Now for the details.

```
\prechapterprecisshift
```

The length \prechapterprecisshift controls the vertical spacing before a \chapterprecis. If the precis immediately follows a \chapter then a different space is required depending on whether or not the article class option is used. The class sets:

The \chapterprecis command calls these two commands to print the  $\langle text \rangle$  in the document (the \chapterprecishere command) and to put it into the ToC (the \chapterprecistoc command). These can be used individually if required.

```
\precisfont
\prechapterprecis \postchapterprecis
```

The \chapterprecishere macro is intended for use immediately after a \chapter. The  $\langle text \rangle$  argument is typeset in the \precisfont font in a quote environment. The macro's definition is:

```
\newcommand{\chapterprecishere}[1]{%
  \prechapterprecis #1\postchapterprecis}
```

where \prechapterprecis, \postchapterprecis and \precisfont are defined as:

```
\newcommand{\prechapterprecis}{%
  \vspace*{\prechapterprecisshift}%
  \begin{quote}\precisfont}
\newcommand{\postchapterprecis}{\end{quote}}
\newcommand*{\precisfont}{\normalfont\itshape}
```

Any or all of these can be changed if another style of typesetting is required.

Next the following macros control the formatting of the precis text in the ToC.

```
\precistoctext{\langle text\rangle} \precistocfont \precistocformat
```

The \chapterprecistoc macro puts \precistoctext{ $\langle text \rangle$ } into the toc file. The default definition similar to (but not exactly 15)

```
\DeclareRobustCommand{\precistoctext}[1]{%
    {\nopagebreak\leftskip \cftchapterindent\relax
    \advance\leftskip \cftchapternumwidth\relax
    \rightskip \@tocrmarg\relax
    \precistocformat\precistocfont #1\par}}
```

<sup>&</sup>lt;sup>15</sup>Internally we use a different name for \leftskip and \rightskip to make it easier to do right to left documents with the class and the bidi package.

Figure 6.2: Displayed sectional headings

Table 6.2:	Default	display	sectioning	layout	parameter	values

	section	subsection	subsubsection
beforeskip (-ex)	3.5 + 12	$3.25{+}1{-}.2$	3.25 + 12
indent	0	0	0
afterskip (ex)	$2.3{+}.2$	1.5+.2	1.5+.2
font	\Large\bfseries	\large\bfseries	\bfseries

Effectively, in the ToC \precistoctext typesets its argument like a chapter title using the \precistocfort (default \itshape), and \precistocformat (default \noindent).

### 6.6 Lower level headings

The lower level heads — sections down to subparagraphs — are also configurable, but there is nothing corresponding to chapter styles.

There are essentially three things that may be adjusted for these heads: (a) the vertical distance between the baseline of the text above the heading to the baseline of the title text, (b) the indentation of the heading from the left hand margin, and (c) the style (font) used for the heading title. Additionally, a heading may be run-in to the text or as a display before the following text; in the latter case the vertical distance between the heading and the following text may also be adjusted. Figure 6.2 shows the parameters controlling a displayed sectional heading and Figure 6.3 shows the parameters for a run-in heading. The default values of the parameters for the different heads are in Table 6.2 for the display heads and Table 6.3 for the run-in heads.

In the following I will use S to stand for one of sec, subsec, subsubsec, para or subpara, which are in turn shorthand for section through to subparagraph, as summarised in Table 6.4.

Figure 6.3: Run-in sectional headings

Table 6.3: Default run-in sectioning layout parameter values

	paragraph	$\operatorname{subparagraph}$
beforeskip (+ex)	$3.25{+}1{-}.2$	$3.25{+}1\text{-}.2$
indent	0	\parindent
afterskip	-1em	-1em
font	\bfseries	\bfseries

Table 6.4: Values for S in section styling macro names.

S	sec	subsec	subsubsec	para	subpara
	section	subsection	subsubsection	paragraph	subparagraph

```
\setbeforeSskip{\langle skip \rangle}
```

The absolute value of the  $\langle skip \rangle$  length argument is the space to leave above the heading. If the actual value is negative then the first line after the heading will not be indented. The default  $\langle skip \rangle$  depends on the particular level of heading, but for a \section (i.e., when S = sec) it is

```
-3.5ex plus -1ex minus -.2ex
```

where the plus and minus values are the allowable stretch and shrink; note that all the values are negative so that there is no indentation of the following text. If you wanted indentation then you could do

\setbeforesecskip{3.5ex plus 1ex minus .2ex}

```
\st Sindent {\langle length \rangle}
```

The value of the  $\langle length \rangle$  length argument is the indentation of the heading (number and title) from the lefthand margin. This is normally 0pt.

```
\stSheadstyle{\langle font \rangle}
```

This macro specifies the style (font) for the sectional number and title. As before, the default value of the  $\langle font \rangle$  argument depends on the level of the heading. For a \subsection

(i.e., S=subsec) it is \large\bfseries\raggedright, to typeset in the \bfseries font in the \large size; the title will also be set ragged right (i.e., there will be no hyphenation in a multiline title).

Note that the very last element in the  $\langle font \rangle$  argument may be a macro that takes one argument (the number and title of the heading). So, if for some reason you wanted \subsubsection titles to be all uppercase, centered, and in the normal font, you can do

\setsubsubsecheadstyle{\normalfont\centering\MakeUppercase}

As another example, although I don't recommend this, you can draw a horizontal line under section titles via:

```
\newcommand{\ruledsec}[1]{%
  \Large\bfseries\raggedright #1 \rule{\textwidth}{0.4pt}}
\setsecheadstyle{\ruledsec}
```

```
\starterSskip{\langle skip \rangle}
```

If the value of the  $\langle skip \rangle$  length argument is positive it is the space to leave between the display heading and the following text. If it is negative, then the heading will be run-in and the value is the horizontal space between the end of the heading and the following text. The default  $\langle skip \rangle$  depends on the particular level of heading, but for a \section (i.e., when S = sec) it is 2.3ex plus .2ex, and for a \subparagraph (i.e., S = subpara), which is a run-in heading, it is -1em.

```
\label{eq:code} $$ \ensuremath{\operatorname{Code}} $$ \sethangfrom{\langle code \rangle}$
```

Internally all the titling macros use a macro called \@hangfrom which by default makes multiline titles look like a hanging paragraph. The default definition of \@hangfrom (in file ltsect.dtx) is effectively:

```
\newcommand{\@hangfrom}[1]{\setbox\@tempboxa\hbox{{#1}}%
\hangindent \wd\@tempboxa\noindent\box\@tempboxa}
```

The argument is put into a box and its width is measured, then a hanging paragraph is started with the argument as the first thing and second and later lines indented by the argument's width.

The \sethangfrom macro redefines \@hangfrom to be  $\langle code \rangle$ . For example, to have the titles set as block paragraphs instead of hanging paragraphs, simply do:

```
\sethangfrom{\noindent #1}
```

Note that you have to use #1 at the position in the replacement code for \@hangfrom where the argument to \@hangfrom is to be located.

```
\label{eq:code} $$ \ensuremat{\langle code \rangle}$ $$ \ensuremat{\langle code \rangle}$
```

Internally all the titling macros use a kernel macro called \@seccntformat which defines the formatting of sectional numbers in a title. Its default definition (in file ltsect.dtx) is effectively:

which formats the sectional numbers as  $\t$  with a space afterwards. The command  $\s$  because of the section o

\setsecnumformat{\csname the#1\endcsname:\quad}

Note that you have to use #1 where you want the argument (sectional number) of \@seccntformat to go.

Note that \setsecnumformat applies to all \section, \subsection, etc. If you want to change it only for, say, \subsection, use the \setsubsechook described below.

\*

```
\hangsecnum
\defaultsecnum
```

The macro \hangsecnum is a declaration that makes sectional numbers hang in the margin. The macro \defaultsecnum is a declaration that reverses the effect of \hangsecnum, that is, sectional numbers will be typeset in their familiar places.

```
\Shook
\setShook{\langle text \rangle}
```

The macro \Shook is called immediately before the typesetting of the title; its default definition does nothing. The macro \setShook redefines \Shook to be  $\langle text \rangle$ . You can use this hook, for example, to insert a \sethangfrom or \setsecnumformat command into the definition of a particular section division command. In that case, remember that if you want to refer to the #1 argument, in the argument for \setsecnumformat, then you have to double the #, i.e. use ##1, see the example below.

Here are some example lower level heads and the code used to produce them.

```
Source for example 6.1
```

```
\setsubsubsecheadstyle{\bfseries\raggedright}
   \subsubsection*{Bold raggedright}
\setsubsubsecheadstyle{\scshape\raggedright}
    \subsubsection*{Small caps raggedright}
\setsubsubsecheadstyle{\itshape\raggedright}
   \subsubsection*{Italic raggedright}
\setsubsubsecheadstyle{\Large\centering}
    \subsubsection*{Large centered}
\setsubsubsecheadstyle{\large\centering\MakeUppercase}
    \subsubsection*{large centered uppercase}
\setsubsubsecheadstyle{\bfseries\centering}
    \subsubsection*{Bold centered}
\setsubsubsecheadstyle{\scshape\centering}
   \subsubsection*{Small caps centered}
\setsubsubsecindent{2\parindent}
\setsubsubsecheadstyle{\scshape\raggedright}
```

```
\subsubsection*{Small caps indented}
\setsubsubsecindent{0pt}
\setsubsubsecheadstyle{\itshape\raggedleft}
\subsubsection*{Italic flushright}
\newcommand*{\shortcenter}[1]{%
\sethangfrom{\noindent ##1}%
\normalfont\boldmath\bfseries
\centering
\parbox{3in}{\centering #1}\par}
\setsubsubsecheadstyle{\shortcenter}
\subsubsection*{Bold centered but taking up no more than 3 inches

if a long title}
```

Hang the A less traditional style is to put the whole heading into the margin. I have done this here whole for a \paragraph heading (which is not otherwise used in this manual). The code is:

heading in the margin

```
\newcommand{\marginbox}[1]{%
  \parbox[t][0pt]{6em}{\itshape\raggedleft\leavevmode #1}}
\newcommand{\marginhead}[1]{%
  {\llap{\marginbox{#1}\kern0.5em}}}
\setparaindent{0em}
\setafterparaskip{0em}
\setparaheadstyle{\marginhead}
\setparahook{\setsecnumformat{\csname the##1\endcsname\ }}
\paragraph{Hang the whole heading in the margin}%
```

## Typeset example 6.1: A variety of subhead styles

Bold raggedright Small caps raggedright Italic raggedright

Large centered

LARGE CENTERED UPPERCASE

Bold centered Small caps centered

Small caps indented

Italic flushright

Bold centered but taking up no more than 3 inches if a long title

The macro \marginbox puts its argument, raggedleft, into a zero height \parbox of width 6em, aligned at the top. The \marginhead macro puts its argument into a \marginbox and puts the \marginbox 0.5em to the left. The \paragraph head style is then set to use \marginhead to typeset the heading. The format for the number is reset via \setparahook and \setsecnumformat.

\*

A different approach is to create new macros, each named by the type of sectional macro it formats, and then make the number format call these macros. In this example we will provide separate formatting for \section and \subsection.

```
\setsecnumformat{\csname #1secnumformat\endcsname}
\newcommand\sectionsecnumformat{\thesection:\quad}
\newcommand\subsectionsecnumformat{\fbox{\enspace\thesubsection\enspace}\enspace}
```

Since the macro is only called in the proper context, we can use \thesection directly in the code for \section.

### 6.7 Fancy anonymous breaks

Often, in novels, there is a need to break up the text to indicate that there is a major break in the story, but not enough to warrant starting a new chapter. I have called these *anonymous divisions* as there is neither number nor title associated with them.

The \plainbreak is an anonymous division. It puts  $\langle num \rangle$  blank lines into the typescript and the first line of the following paragraph is not indented. Another anonymous division is \fancybreak which puts  $\langle text \rangle$  centered into the typescript and the initial line of the following paragraph is not indented. For example:

```
\fancybreak{{*}\\{* * *}\\{*}}
```

typesets a little diamond made of asterisks.

The starred versions of the commands indent the first line of the following paragraph.

```
\plainfancybreak{\langle space \rangle} {\langle num \rangle} {\langle text \rangle} \\ plainfancybreak*{\langle space \rangle} {\langle num \rangle} {\langle text \rangle} \\
```

If a plain break comes at the top or bottom of a page then it is very difficult for a reader to discern that there is a break at all. If there is text on the page and enough space left to put some text after a break the \plainfancybreak command will use a \plainbreak with  $\langle num \rangle$  lines, otherwise (the break would come at the top or bottom of the page) it will use a \fancybreak with  $\langle text \rangle$ . The  $\langle space \rangle$  argument is a length specifying the space needed for the  $\langle num \rangle$  blank lines and some number of text lines for after the plain break. The starred version of the command uses the starred versions of the \plainbreak and \fancybreak commands.

Unfortunately there is an interaction between the requested, plain, and fancy break spaces. Let P be the space (in lines) required for the plain break, F the space (in lines) required for the fancy break, and S the  $\langle space \rangle$  argument (in lines). From some experiments it appears that the condition for the plain break to avoid the top and bottom of the

page is that S-P>1. Also, the condition for the fancy break to avoid being put in the middle of a page (i.e., not at the top or bottom) is that S-F<3. For example, if the plain and fancy breaks take the same vertical space then S=P+2 is the only value that matches the conditions. In general, if F=P+n then the condition is 1< S-P<3+n, which means that for the \plainfancybreak command the fancy break must always take at least as much space as the plain break.

\* \* \*

The \plainfancybreak macro inserts a plain break in the middle of a page or if the break would come at the bottom or top of a page it inserts a fancy break instead.

```
\pfbreak \pfbreak* \\ pfbreakskip \\ pfbreakdisplay {\langle text \rangle}
```

The \pfbreak macro is an alternate for \plainfancybreak that may be more convenient to use. The gap for the plain break is given by the length \pfbreakskip which is initialised to produce two blank lines. The fancy break, which takes the same vertical space, is given by the  $\langle text \rangle$  argument of \pfbreakdisplay. The default definition:

```
\newcommand*{\pfbreakdisplay}{*\quad*\quad*}
```

typesets three asterisks, as shown a few lines before this.



You can change the definition of \pfbreakdisplay for a different style if you wish. The fancy break just before this was produced via:

```
\renewcommand{\pfbreakdisplay}{%
  \ensuremath{\clubsuit\quad\diamondsuit\quad\clubsuit}}
\fancybreak{\pfbreakdisplay}
```

I used \fancybreak as I'm not sure where the break will come on the page and the simple \pfbreak macro might just have produced a couple of blank lines instead of the fancy display.

The paragraph following \pfbreak is not indented. If you want it indented use the \pfbreak\* starred version.

```
es es es
```

The fancy break using fleurons just before this paragraph was produced by:

where the \ding command is from the pifont package.

The fancy break made with fleurons was simple to specify. There are many other symbols that you can use in LaTeX and these can be combined in potentially attractive ways to produce a fancy break like the one just above.

The following idea was originally suggested by Christina Thiele [Thi98], and can be used to string together mathematical symbols. It works following the same principles as the dot leaders in the Table of Contents.

Define a macro called with the syntax  $\mathtt{motif}\{\langle shape \rangle\}$ , where  $\langle shape \rangle$  is a symbol or other shape to be repeated in a chain,

\newcommand{\motif}[1]{\cleaders\hbox{#1}\hfil}

The definition of \motif is basically taken from TeX, and is part of the code for making things like dot leaders.  $\hbox{\langle stuff\rangle}$  puts  $\langle stuff\rangle$  into a horizontal box, and  $\cleaders\langle box\rangle$  fills a specified amount of space using whatever number of copies of the  $\langle box\rangle$  as is needed; if there is too much space to be filled by a whole number of boxes, the spare space is spread around equally.  $\hfill$  is stretchy space. The  $\motif$  macro essentially says, fill up a space with with copies of  $\langle shape\rangle$ .

We also need another macro,  $\chain{\langle shape \rangle} {\langle length \rangle}$ , where  ${\langle shape \rangle}$  is a shape to be repeated as many times as it takes to fill up a distance  $\langle length \rangle$ .

The \leavevmode makes sure that we are typesetting horizontally, and \hbox to <length>{stuff} puts  $\langle stuff \rangle$  into a horizontal box with the fixed length of  $\langle length \rangle$ . Roughly, what \chain and \motif do together is typeset enough copies of  $\langle shape \rangle$  to make up a distance  $\langle length \rangle$ .

That is what we have been aiming for. All that remains is to decide on what shape we might want to use. Here is one consisting of diamonds.

```
\makeatletter
\newcommand{\diamonds}{\m@th$\mkern-.6mu \diamond \mkern-.6mu$}
\makeatother
```

The \diamond symbol can only be used in math mode, hence it is surrounded by the short-hand \\$...\\$. TeX usually leaves a little space around maths but the \m@th command stops that. \mkern adjusts space in math mode, and in this case we are eliminating the spaces that would normally be on either side of the diamond symbol. The whole effect gives us a diamond symbol with zero space around it.

The fancy break at the start of this discussion was typeset by

```
% define \motif, \chain, \diamonds and then
\fancybreak{\chain{\diamonds}{0.25\textwidth}}}
```

The code is not part of the memoir class; I defined it just as indicated in the body of the book. It would more naturally go into the preamble or a package. You might like to try specifying your own pattern, say \clubs, using the \club math symbol but leaving some space between them.

## 6.8 Footnotes in division headings

With the sectioning commands the text of the required argument  $\langle title \rangle$  is used as the text for the section title in the body of the document.

When the optional argument  $\langle toc\text{-}title \rangle$  is used in a sectioning command it is moving and any fragile commands must be  $\langle toc\text{-}title \rangle$  argument is fixed. The  $\langle toc\text{-}title \rangle$  also serves double duty:

- 1. It is used as the text of the title in the ToC;
- 2. It is used as the text in page headers.

<sup>&</sup>lt;sup>16</sup>It is usually a matter for experiment to find the right values for the kerning.

\booknamefont	\huge\bfseries	huge
\booknumfont	\huge\bfseries	huge
\booktitlefont	\Huge\bfseries	Huge
\partnamefont	\huge\bfseries	huge
\partnumfont	\huge\bfseries	huge
\parttitlefont	\Huge\bfseries	Huge
\chapnamefont	\huge\bfseries	huge
\chapnumfont	\huge\bfseries	huge
\chaptitlefont	\Huge\bfseries	Huge
\secheadstyle	\Large\bfseries	Large
\subsecheadstyle	\large\bfseries	Large
\subsubsecheadstyle	\normalsize\bfseries	normal
\paraheadstyle	\normalsize\bfseries	normal
\subparaheadstyle	\normalsize\bfseries	normal

Table 6.5: Default fonts for sectional headings

If the optional argument is not present, then the  $\langle title \rangle$  is moving and serves the triple duty of providing the text for the body and ToC titles and for page headers.

Some folk feel an urge to add a footnote to a sectioning title, which should be resisted. If their flesh is weak, then the optional argument must be used and the \footnote attached to the required argument only. If the optional argument is not used then the footnote mark and text is likely to be scattered all over the place, on the section page, in the ToC, on any page that includes  $\langle title \rangle$  in its headers. This is unacceptable to any reader. So, a footnoted title should look like this:

\chapter[Title]{Title\footnote{Do you really have to do this?}}

## 6.9 Predefined heading styles

All LaTeX classes for typesetting books and reports provide a particular style for sectional headings. The memoir class is unusual in that it provides several sets of heading styles. Each set has different spacing around the division heads, and different fonts in different sizes. As a reference, Table 6.5 lists the default fonts used for the sectional headings. These fonts are all bold but in different sizes depending on the division level.

```
\label{local_local_local_local} $$\max_{e} {\langle name \rangle} {\langle code \rangle}$$ $$ \headstyles{\langle name \rangle}$
```

The default sectional division head styles provided by memoir form the *default* headstyles and give the same appearance as the standard book and report classes. The set is created via the \makeheadstyles macro and called for via the headstyles declaration.

```
\makeheadstyles{default}{%
```

\renewcommand\*{\booknamefont}{\normalfont\huge\bfseries}

```
\renewcommand*{\subparaheadstyle}{\normalfont\normalsize\bfseries}
 }
 \headstyles{default}
   A somewhat different set of headstyles is used for this manual.
                                                                        When using
\makeheadstyles you only need to specify things that differ from the default. Within the
class the memman set of headstyles is defined as:
 \newcommand*{\addperiod}[1]{#1.}
 \makeheadstyles{memman}{%
 % book changes
   \renewcommand*{\booknamefont}{\normalfont\huge\sffamily}
   \renewcommand*{\booknumfont}{\normalfont\huge\sffamily}
   \renewcommand*{\booktitlefont}{\normalfont\Huge\sffamily}
   \renewcommand*{\midbookskip}{\par\vskip 2\onelineskip}%
 % part changes
   \renewcommand*{\partnamefont}{\normalfont\huge\sffamily}
   \renewcommand*{\partnumfont}{\normalfont\huge\sffamily}
   \renewcommand*{\parttitlefont}{\normalfont\Huge\sffamily}
   \renewcommand*{\midpartskip}{\par\vskip 2\onelineskip}%
 % chapter
   \chapterstyle{demo3}
 % section
   \setbeforesecskip{-1.333\onelineskip
                      \@plus -0.5\onelineskip \@minus -.5\onelineskip}%
   \setaftersecskip{0.667\onelineskip \@plus 0.1\onelineskip}%
   \setsecheadstyle{\normalfont\scshape\raggedright}%
 % subsection
   \setbeforesubsecskip{-0.667\onelineskip
                        \@plus -0.25\onelineskip \@minus -0.25\onelineskip}%
   \setaftersubsecskip{0.333\onelineskip \@plus 0.1\onelineskip}%
   \setsubsecheadstyle{\normalfont\bfseries\raggedright}%
 % subsubsection
   \setbeforesubsubsecskip{-0.667\onelineskip
                           \@plus -0.25\onelineskip \@minus -0.25\onelineskip}%
   \setaftersubsubsecskip{0.333\onelineskip \@plus 0.1\onelineskip}%
   \setsubsubsecheadstyle{\normalfont\normalsize\itshape\raggedright}%
 % paragraph
   \setbeforeparaskip{1.0\onelineskip
                      \@plus 0.5\onelineskip \@minus 0.2\onelineskip}%
   \setafterparaskip{-1em}%
   \setparaheadstyle{\normalfont\normalsize\itshape\addperiod}%
 % subparagraph
   \setsubparaindent{\parindent}%
   \setbeforesubparaskip{1.0\onelineskip
                         \@plus 0.5\onelineskip \@minus 0.2\onelineskip}%
   \setaftersubparaskip{-1em}%
```

% and so on down to subparagraph specification

chapter	section	$\operatorname{subsec}$	$\operatorname{subsubsec}$	para	$\operatorname{subpara}$
CAPS	s. caps	Italic	s. caps	Italic	Italic
Bold	CAPS	Bold	s. caps	Italic	s. caps
Bold	Bold	Bold	Bold	Bold	Bold
Italic	CAPS	s. caps	Italic	Italic	Italic
Sans	Sans	Sans	Sans	Sans	Sans
Sans	s. caps	Bold	Italic	Italic	Italic
Bold	Bold	Bold	Slanted	Slanted	Slanted
Bold	CAPS	Italic	Bold	Italic	Italic
Italic	CAPS	Italic	s. caps	Italic	Italic
	CAPS Bold Bold Italic Sans Sans Bold Bold	CAPS s. caps Bold CAPS Bold Bold Italic CAPS Sans Sans Sans s. caps Bold Bold Bold CAPS	CAPS s. caps Italic Bold CAPS Bold Bold Bold Bold Italic CAPS s. caps Sans Sans Sans Sans s. caps Bold Bold Bold Bold Bold CAPS Italic	CAPS s. caps Italic s. caps Bold CAPS Bold s. caps Bold Bold Bold Bold Italic CAPS s. caps Italic Sans Sans Sans Sans Sans s. caps Bold Italic Bold Bold Bold Slanted Bold CAPS Italic Bold	CAPS s. caps Italic s. caps Italic Bold CAPS Bold s. caps Italic Bold Bold Bold Bold Bold Italic CAPS s. caps Italic Italic Sans Sans Sans Sans Sans Sans s. caps Bold Italic Italic Bold Bold Bold Slanted Bold CAPS Italic Bold Italic

Table 6.6: Fonts used by different headstyles

\setsubparaheadstyle{\normalfont\normalsize\itshape\addperiod}}

You can see the effect throughout this document. This chapter is slightly different in that I have used the *pedersen* chapterstyle instead of the *demo3* chapterstyle that I have normally used.

Several other sets of headstyles are provided as well and the full list is below. The different fonts used are given in Table 6.6 and generally speaking they start off being large for chapter heads but are normal size by the time subsubsection heads are reached, or before.

bringhurst A set based on Bringhurst's *Elements of Typographic Style* [Bri99]. It uses the *bringhurst* chapterstyle (Figure B.7).

*crosshead* This set uses the *crosshead* chapterstyle and the lower level division titles are set as crossheads.

*default* The default set corresponding the LaTeX book class.

dowding A set based on Dowding's Finer Points [Dow96]. It uses the dowding chapterstyle (Figure B.14).

*komalike* A set based on the kind of headings used in the KOMA scrbook class, where there are all in a bold sans serif font. It uses the *komalike* chapterstyle (Figure B.17).

memman The set used in this document, including the demo3 chapterstyle.

ntglike A set based on the kind of headings used in the NTG (Dutch TUG) book class. It uses the ntglike chapterstyle (Figure B.20) and the headings are quiter than the default.

tandh A set based the heads used in Thames & Hudson Manual of Typography [McL80]. It uses the tandh chapterstyle (Figure B.22)

wilsondob A set based on those used in Adrian Wilson's *Design of Books* [Wil93]. It uses the wilsondob chapterstyle (Figure B.26).

# Seven

# 페이지 매김과 머릿글

본 단원은 페이지에 이정표를 표시함으로써 독자가 문서를 더 수월하게 찾을 수 있도록 하는데 집중한다.

## 7.1 페이지 매김과 폴리오

LaTeX 문서의 모든 페이지는 페이지 매김된다. 즉, 각 페이지에는 번호가 할당되며 이것이 page 카운터의 값이다. 이 값은 \setcounter나 \addtocounter을 통해 언제든지 바꿀 수 있다.

```
\pagenumbering \{\langle rep \rangle\} \\ \pagenumbering * \{\langle rep \rangle\} \\
```

매크로 \pagenumbering과 \pagenumbering\*은 폴리오(면 번호)가 카운터 표시  $\langle rep \rangle$ 를 통해 페이지 번호를 출력하도록 할 수 있는데,  $\langle rep \rangle$ 는 Alph, alph, arabic, Roman, 혹은 roman을 사용해 각각 알파벳 대문자와 소문자, 아라비아 숫자, 대문자와 소문자 로마 숫자 중 하나가 될수 있다. 알파벳에는 26개의 문자만 있기 때문에 Alph 혹은 alph는 한정된 수의 페이지에만 사용될 수 있다. 사실상 이 매크로들은 \thepage가 \rep{page}가 되도록 재정의한다.

추가적으로, \pagenumbering 명령은 page 카운터를 1로 초기화하며, 별표 붙은 것은 카운터를 변경하지 않는다. 양식이 바뀔 때마다 페이지 번호를 다시 1로 초기화하는 것이 일반적이지만, 간혹 보여지는 형식과는 상관없이 일련의 숫자를 가지도록 해야할 때에는 별표 붙은 버전이 유용하다.

```
\savepagenumber
\restorepagenumber
```

매크로 \savepagenumber는 현재 페이지 번호를 저장하며, 매크로 \restorepagenumber는 페이지 번호를 저장된 것으로 설정한다. 이 한 쌍의 명령은 페이지 매김을 외관상 중단할 때 사용될 수 있다. 예를 들어, 전면 삽화가 전자 문서에 끼워 넣어질 수 있는데 이 때는 페이지 매김이 불필요하다. 이는 다음과 같이 할 수 있다.

여러분이 이런 비슷한 작업을 한다면, 복원된 페이지 번호를 1 정도 조절해야 할 수도 있다.

\restorepagenumber

% perhaps \addtocounter{page}{1} or \addtocounter{page}{-1}

\...pagenumber 명령의 실행 시점과 페이지 나눔에 대한 TeX의 결정에 따라 이런 조정이 필요할 수도, 그렇지 않을 수도 있다.

7.2 페이지 양식

본 클래스는 여러분이 쓸 수 있는 페이지 양식 모음을 제공하며, 여러분 마음에 들지 않는다면 직접 정의할 수도 있다.

이 기능들은 fancyhdr 패키지 [Oos96]에 영향을 받았으나, 명령어들은 다르다.

표준 클래스들은 홀수와 짝수 페이지에 대한 바닥글과 머릿글을 제공한다. 따라서 페이지 양식에 대한 네 요소를 지정할 수 있다. 이 클래스는 머릿글과 바닥글을 왼쪽, 가운데, 그리고 오른쪽 부분으로 나누어 페이지 양식을 위해 총 12개의 요소를 지정해야 한다. 그러나 내장된 페이지 양식이 충분할 경우에는 이 모든 설정에 대해서 걱정하지 않아도 된다.

\pagestyle $\{\langle style \rangle\}$ \thispagestyle $\{\langle style \rangle\}$ 

\pagestyle은 현재 페이지 양식을 \langle style \rangle 로 정하는데, \langle style \rangle 은 문자로만 구성된 단어이다. 특정 페이지에서는 \thispagestyle을 사용해 현재 페이지 양식을 해당 쪽만 덮어쓸 수 있다. 클래스의 일부 명령들은 자동으로 \thispagestyle을 부른다. 예를 들어,

• titlingpage 환경은

 $\verb|\thispagestyle{titlingpagestyle}|$ 

을 부른다.

• 만약 \cleardoublepage 이 빈 왼쪽 면을 만든다면

\thispagestyle{cleared}

를 해당 빈 페이지에서 부른다.

참고로, 전체 목록은 Table 7.1에 주어져 있다.

본 클래스에 의해 제공되는 페이지 양식은 다음과 같다.

empty 머릿글과 바닥글이 비어 있다.

plain 머릿글이 비어 있고 폴리오 (페이지 번호)는 페이지 하단 중앙에 놓인다.

headings 바닥글이 비어 있다. 머릿글은 폴리오를 페이지 바깥쪽에 둔다. 왼쪽 페이지에는 장 이름, 번호와 제목이 기울어진 대문자로 책 등 여백에 놓이고, 오른쪽 페이지에는 절 이름과 대문자 제목이 책 등 여백에 놓인다.

myheadings *heading* 양식처럼 바닥글이 비어 있다. 여러분이 머릿글에 들어갈 내용을 지정해 야 한다.

simple 바닥글이 비어 있고 머릿글은 폴리오 (페이지 번호)를 페이지 바깥쪽에 놓는다. 이는 headings 양식과 비슷하지만 제목 문구가 없다.

ruled 바닥글이 폴리오를 바깥쪽에 놓는다. 왼쪽 페이지의 머릿글과 제목은 바깥쪽에 small caps로, 오른족 페이지에서는 절 제목이 normal 폰트로 바깥쪽에 식자된다. 머릿글 밑에 선이 그려진다.

Ruled 이는 머릿글과 바닥글이 재단 여백으로 확장된다는 것을 제외하고는 ruled 양식과 같다.

Table 7.1: \thispagestyle의 사용

사용된 곳	양식
\book	book
\chapter	chapter
\cleardoublepage	cleared
\cleartorecto	cleared
\cleartoverso	cleared
\epigraphhead	epigraph
\listoffigures	chapter
\listoftables	chapter
\maketitle	title
\part	part
\tableofcontents	chapter
thebibliography	chapter
theindex	chapter
titlingpage	titlingpage

companion 이는 *Companion* 시리즈의 페이지 양식 사본이다 (예시로는 [MG+04]을 보라). 이는 머릿글이 여백 주석의 바깥쪽 모서리까지 이어지는 선을 가지고 있다는 점에서 *Ruled*와 비슷하다. 폴리오는 왼쪽 페이지 머릿글에 볼트체 오른쪽 정렬되어 있으며, 절이름과 제목이 오른쪽 머릿글에 다시 볼드체로 왼쪽 정렬되어 있다. 바닥글은 없다.

book 이는 plain 페이지 양식과 같다.

chapter 이는 plain 페이지 양식과 같다.

cleared 이는 empty 페이지 양식과 같다.

part 이는 plain 페이지 양식과 같다.

title 이는 plain 페이지 양식과 같다.

titlingpage 이는 empty 페이지 양식과 같다.

## \uppercaseheads \nouppercaseheads

headings에서 \nouppercaseheads 정의에 뒤따르는 제목은 자동으로 대문자로 바뀌지 않을 것다. 기본은 \uppercaseheads로, 제목은 자동으로 대문자 처리되도록 지정하는 것이다.

2012 수정: uppercaseheads에 사용되는 대문자화 매크로는 MakeTextUppercase로 변경되어 수식, 참조나 인용을 건드리지 않게 되었다.

위의 myheadings을 위해서, 여러분은 머릿글에 들어갈 제목을 직접 정의해야 한다. \sec 과 같은 장절 명령은 각각 \secmark라는 매크로를 부른다. 페이지 양식은 보통 이 명령어를 정의해 \sec의 제목이나 번호를 추출한다. 페이지 양식은 이후 이 정보를 용도에 맞게 사용할수 있다.

 $\label{left} $$ \mathbf{\langle} left \rangle + (\langle right \rangle) $$ \markright + (\langle right \rangle) $$$ 

\markboth는 이 명령이 불리운 자리에서 두  $\overline{x}$ 의 값을  $\langle left \rangle$ 와  $\langle right \rangle$ 에 설정한다.

# \leftmark \rightmark

주 매크로	기본 표지 정의
\book(*)	\newcommand*{\bookpagemark}[1]{}
\part(*)	<pre>\newcommand*{\partmark}[1]{}</pre>
\chapter(*)	<pre>\newcommand*{\chaptermark}[1]{}</pre>
\section(*)	<pre>\newcommand*{\sectionmark}[1]{}</pre>
\subsection(*)	<pre>\newcommand*{\subsectionmark}[1]{}</pre>
\subsubsection(*)	<pre>\newcommand*{\subsubsectionmark}[1]{}</pre>
\paragraph(*)	<pre>\newcommand*{\paragraphmark}[1]{}</pre>
\subparagraph(*)	<pre>\newcommand*{\subparagraphmark}[1]{}</pre>
\tableofcontents(*)	<pre>\newcommand*{\tocmark}[1]{}</pre>
\listoffigures(*)	<pre>\newcommand*{\lofmark}[1]{}</pre>
\listoftables(*)	<pre>\newcommand*{\lotmark}[1]{}</pre>
\thebibliography	<pre>\newcommand*{\bibmark}{}</pre>
\theindex	<pre>\newcommand*{\indexmark}{}</pre>
$ ag{theglossary}$	<pre>\newcommand*{\glossarymark}{}</pre>
\PoemTitle	<pre>\newcommand*{\poemtitlemark}[1]{}</pre>
\PoemTitle*	<pre>\newcommand*{\poemtitlestarmark}[1]{}</pre>

Table 7.2: 페이지 머릿글에 대한 표지 매크로

매크로 \leftmark는 쪽에서 *마지막* \markboth 명령의  $\langle left \rangle$  인자의 값을 담는다. 매크로 \rightmark는 페이지의 처음 \markboth나 \markright의  $\langle right \rangle$  인자의 값을 담거나, 해당 명령이 없다면 가장 최근 것의  $\langle right \rangle$  인자의 값을 담는다.

페이지 양식은 \markboth와 \markright으로 secmark 명령어를 정의할 수 있고, \leftmark와(혹은) \rightmark를 머릿글이나 바닥글에서 쓴다. 이것이 어떻게 되는 것인지 나중에 예시를 보일 것이며, 이는 myheadings 양식이 보통 구현되는 방식이다.

모든 장절 구획 명령은 해당 머릿글과 관련해 여러분이 정의할 수 있는 매크로를 포함하고 있다. 다른 명령어들도 여러분이 재정의할 수 있는 표지 설정을 포함한다.

\...mark 명령어들이 Table 7.2에 나열되어 있다. 이들이 관련된 주 매크로에 의해 불리우면, 'title'을 인자의 값으로 취한다. 예를 들어, \chapter 매크로는 \chaptermark이 머릿글이되도록 제목의 값을 지정하여 부른다.

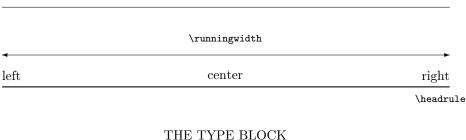
부디 Table 7.2에 나열된 매크로들이 '제공자' 매크임을 잊지 말아달라. 즉, 이들은 여러분이 \leftmark와 \rightmark의 정보를 나중에 쓸 수 있도록 제공해 준다. 절 제목에 접근하기 위해서 \sectionmark를 머릿글이나 바닥글에 사용하면 안된다. 매크로가 정보를 제공해주지만, 여러분은 \sectionmark를 어떻게 정의했는지에 따라서 \leftmark 혹은 \rightmark를 사용해 접근해야 한다.

### 7.3 머릿글과 바닥글 만들기

언급하였듯이, 이 클래스는 짝수와 홀수 머릿글과 바닥글의 왼쪽, 가운데, 그리고 오른쪽 영역을 제공한다. 본 절은 여러분이 이 12개의 영역을 사용해 직접 페이지 양식을 만드는 방법을 설명한다. 페이지의 6개 영역은 Figure 7.1에 도표로 나타나 있다.

이 클래스 자체도 이 절의 명령들을 사용한다. 예를 들어, plain 페이지 양식은 다음과 같이 정의되어 있다.

\makepagestyle{plain}



		\footrule
left	center	right

Figure 7.1: 머릿글과 바닥글 영역

\makeevenfoot{plain}{}{\thepage}{} \makeoddfoot{plain}{}{\thepage}{}

이는 페이지 번호를 페이지 하단에 중앙 정렬한다.

```
\mbox{makepagestyle} \langle style \rangle
\aliaspagestyle{\langle alias\rangle}{\langle original\rangle}
\operatorname{copypagestyle}(\langle copy \rangle) \{\langle original \rangle\}
```

\makepagestyle 명령은 ⟨style⟩ 페이지 양식을 지정하는데, 이는 초기에 empty 페이지 양식 과 동일하다. 반면, \aliaspagestyle은 ⟨alias⟩ 페이지 양식을 ⟨original⟩ 페이지 양식과 같도록 정의한다. 후자의 예시로 본 클래스가 다음 코드를 포함한다.

```
\aliaspagestyle{part}{plain}
\aliaspagestyle{chapter}{plain}
\aliaspagestyle{cleared}{empty}
```

\copypagestyle 명령은 〈original〉 페이지 양식 설정으로 〈copy〉라는 새로운 페이지 양식을 생성한다.

만약 참조(alias)와 사본(copy) 페이지 양식이 〈original〉을 토대로 생성된 후, 〈original〉이 수정되었다면, 참조와 사본은 다르게 행동한다. 참조 페이지 양식은 수정된 (original)과 여전히 일치하지만 사본 페이지 양식은 (original)의 어떠한 변경에도 영향을 받지 않는다. 여러분은 참조 페이지 양식을 수정할 수 없지만 사본 페이지 양식은 수정할 수 있다.

```
\label{lem:lemmad} $$ \mathbf{\langle style \rangle} {\langle left \rangle} {\langle center \rangle} {\langle right \rangle} $$
   \mbox{\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\
   \mbox{\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\
```

매크로 \makeevenhead는 짝수 페이지에 대해 〈left〉, 〈center〉, 그리고 〈right〉 영역의 〈style〉 페이지 양식 머릿글을 정의한다. 비슷하게 \makeoddhead, \makeevenfoot, 그리고 \makeoddfoot은 홀수 페이지의 〈left〉, 〈center〉, 〈right〉 영역 〈style〉 머릿글을 정의하고, 홀수와 짝수 페이지의 바닥글을 정의한다. 이  $\langle style \rangle$  명령들은 대응되는  $\langle style \rangle$ 의 \makepagestyle 이후에 사용되어야 한다.

매크로 \makerunningwidth는  $\langle style \rangle$  페이지 양식의 머릿글과 바닥글의 너비를 지정한다. 머릿글의 너비는  $\langle headwidth \rangle$ 로 설정된다. 만약  $\langle footwidth \rangle$ 가 선택적으로 존재한다면, 바닥글 너비는 이 값으로 설정될 것이며 그렇지 않을 경우에는  $\langle headwidth \rangle$ 로 설정된다. 머릿글 너비는 길이  $\langle style \rangle$ headrunwidth로, 바닥글 너비는  $\langle style \rangle$ footrunwidth에 저장된다.

\makepagestyle은 너비를 본문 너비로 설정하므로, 이 매크로는 다른 너비 값이 필요할 때만 사용된다. 길이 \headwidth는 머릿글이나 바닥글, 혹은 다른 목적으로 사용될 수 있도록 (임시) 길이로 제공된다.

머릿글은 조판 영역의 위와 자신 사이에 선이 그려질 수 있으며, 비슷하게 조판 영역 아래와 바닥글 위에 선이 그려질 수 있다. \makeheadrule 매크로는  $\langle style \rangle$  페이지 양식 머릿글 아래 선의  $\langle width \rangle$ 와  $\langle thickness \rangle$ 를 지정한다. 추가적 인자  $\langle skip \rangle$ 은 바닥 선의 세로 위치를 지정하는 길이이다 (\footruleskip을 보라). \makepagestyle 매크로는  $\langle width \rangle$ 를 \textwidth로 초기화하고  $\langle thickness \rangle$ 를 0pt로 초기화해서, 기본적으로는 선이 보이지 않는다. 매크로 \makeheadfootruleprefix는 머리/바닥 선의 대체 색상을 추가할 수 있도록 하는데, 예를 들어 다음과 같다.

\makeheadfootruleprefix{mystyle}{\color{red}}{\color{blue}}

```
\normalrulethickness
```

\normalrulethickness는 보이는 선의 기본 두께로, 0.4pt이다. 이는 \setlength를 통해 바꿀 수 있지만, 필자는 여러분이 적어도 14pt 클래스 옵션을 사용하지 않는 이상 그렇게 하지 않기를 권장한다.

```
\footruleheight
\footruleskip
```

매크로 \footruleheight은 바닥글 위의 기본 선의 높이이다 (기본으로 0). \footruleskip은 바닥 선이 조판 영역의 하단과 바닥글 사이에 놓일 것을 보장하는 충분한 길이이다. 길이로 보이기는 하지만, 만약 이 값을 바꾸고 싶다면 \setlength이 아닌 \renewcommand를 사용하라.

makeheadposition 매크로는 짝수와 홀수 머릿글과 바닥글의 가로 위치를  $\langle style \rangle$  페이지 양식에 맞게 지정한다.  $\langle ...pos \rangle$  인자 각각은 그 뜻이 분명한 flushleft, center, 혹은 flushright 이 될 수 있다. 비어있거나 알 수 없는 값은 center과 동일하다. 이 매크로는 머릿글/바닥글의 너비가 textwidth와 같지 않을 때에만 사용된다.

```
\mbox{\mbox{makepsmarks}} \langle style \rangle \} \{ \langle code \rangle \}
```

\pagestyle{ $\langle style \rangle$ }이 하는 마지막 일은  $\langle style \rangle$ 에 대한 \makepsmarks 매크로의  $\langle code \rangle$  인자를 부르는 것이다. 이는 특정 페이지 양식에 대한 비표준 코드(이전의 매크로들로는 지정할 수 없는 코드)를 지정할 때 보통 사용된다. 이 코드는 보통 머릿글과 바닥글에서 사용될 표지를 정의한다.

```
\mbox{\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\
```

머릿글과 바닥글은 세 개의 다른 것들로 구성되어 있다. 처음과 끝에는 〈style〉과 관련된 특별한 strut가 삽입되어 있다. 기본적으로 (strut가 비어 있는 empty 양식을 제외하고) \makepagestyle은 이들을 \strut에 초기화할 것이다. 이 strut들을 다른 것으로 바꿀 때는 위의 매크로를 사용할 수 있다.

## 7.3.1 예시 페이지 양식

여러분이 초고를 준비할 때 각 페이지에 이것이 초고 문서라는 주석을 남기고 싶을 수도 있다. 여러분이 *headings* 페이지 양식과 장 시작에 기본 *plain* 페이지 양식을 사용하고 있다고 가정 한다면, 다음을 preamble에 정의할 수 있다(\ifdraftdoc는 클래스에 의해 제공되고, draft 옵션이 사용된다면 true로 설정된다).

#### \ifdraftdoc

```
\makeevenfoot{plain}{\thepage}{\textit{Draft: \today}}
\makeoddfoot{plain}{\textit{Draft: \today}}{\thepage}{}
\makeevenfoot{headings}{}{\textit{Draft: \today}}
\makeoddfoot{headings}{\textit{Draft: \today}}{}}
\fi
```

이제 draft 옵션이 쓰이면 단어 'Draft:'와 함께 현재 날짜가 이탤릭체로 매 페이지 하단의 책등 여백 쪽에 식자될 것이다. 만약 *empty* 페이지도 표시되어야 한다면, 해당 양식에도 비슷한 바닥글을 지정하라.

여기에 여러 LaTeX 내부 명령을 사용하는 book 클래스 *headings* 페이지 양식의 표준 정의의 일부가 나와 있다. 그러나 memoir은 이를 사용하지 않는다는 것에 유의하라.

```
\let\@oddfoot\@empty\let\@evenfoot\@empty
\def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
\def\chaptermark##1{%
 \markboth{\MakeUppercase{%
   \ifnum\c@secnumdepth > \m@ne
     \if@mainmatter
       \@chapapp\ \thechapter. \ %
     \fi
   \fi
   ##1}}{}}%
\def\sectionmark##1{%
 \markright{\MakeUppercase{%
   \ifnum\c@secnumdepth > \z@
     \thesection. \ %
   \fi
   ##1}}}
```

이를 이해할 필요는 없지만, 개략적으로 첫 세 줄은 바닥글과 머릿글의 내용을 지정하고, 나머지 코드는 머릿글에 사용될 표지를 지정한다. \leftmark는 'chapter'라는 단어 뒤에, \mainmatter에 있으며 secnumdepth가 해당 장에 번호가 붙도록 설정되어 있다면 번호가 붙고, 뒤에 장의 제목이 붙도록 지정된다. 이 모두는 (\MakeUppercase 매크로에 의해) 대문자가된다. 비슷하게 다른 표지인 \rightmark도 있다면 절 번호와 절의 제목이 모두 대문자로 된다.

```
이 코드를 memoir의 코딩 양식으로 의역한다면 다음과 같다.
   \makepagestyle{headings}
   \makeevenhead{headings}{\thepage}{}{\slshape\leftmark}
   \makeoddhead{headings}{\slshape\rightmark}{}{\thepage}
   \makepsmarks{headings}{%
       \def\chaptermark##1{%
           \markboth{\MakeUppercase{%
               \ifnum\c@secnumdepth > \m@ne
                   \if@mainmatter
                        \@chapapp\ \thechapter. \ %
                   \fi
                \fi
                ##1}}{}}%
       \def\sectionmark##1{%
           \markright{\MakeUppercase{%
                \ifnum\c@secnumdepth > \z@
                   \thesection. \ %
                \fi
               ##1}}}
       \def\tocmark{\markboth{\MakeUppercase{\contentsname}}{}}
       \def\lofmark{\markboth{\MakeUppercase{\listfigurename}}{}}
       \def\lotmark{\markboth{\MakeUppercase{\listtablename}}{}}
       \def\bibmark{\markboth{\MakeUppercase{\bibname}}{}}
       \def\indexmark{\markboth{\MakeUppercase{\indexname}}{}}
       \def\glossarymark{\markboth{\MakeUppercase{\glossaryname}}}}}
여러분이 볼 수 있듯, 페이지 양식의 표지를 정의하는 것이 그렇게 간단하지만은 않은 일이다.
그러나 Lars Madsen의 도움으로 일이 간단해졌다.
         \createplainmark{\langle type \rangle}{\langle marks \rangle}{\langle text \rangle}
         \mbox{\em UChead}\{\langle text \rangle\}
         \uppercaseheads \nouppercaseheads
         \createmark{\langle sec \rangle}{\langle marks \rangle}{\langle show \rangle}{\langle prefix \rangle}{\langle postfix \rangle}
       매크로 \createplainmark는 \<type>mark를 정의하는데, ⟨type⟩은 번호 붙지 않은 toc,
lof, index와 같은 영역의 머릿글로 〈text〉를 표지 값으로 사용하고, 〈marks〉는 left, both,
혹은 right이다. 예를 들어
   \createplainmark{toc}{left}{\contentsname}
   \createplainmark{lot}{right}{\listtablename}
   \createplainmark{bib}{both}{\bibname}
은 다음과 같다.
   \def\tocmark{\markboth{\memUChead{\contentsname}}{}}
   \def\lotmark{\markright{\memUChead{\listtablename}}}
   \label{lofmark} $$ \def \ \mem \ \m
      \uppercaseheads의 정의에 뒤따르는 \memUChead 명령은 \MakeUppercase와 동일
하지만 \nouppercaseheads에 뒤따를 때는 (아무것도 하지 않는) \relax와 동일하다.
\createplainmark 매크로는 \memUChead를 생성된 \mark(both/right) 매크로의 \text\ 인자
```

양 옆에 감싼다. \(no)uppercaseheads 선언을 사용해 여러분은 대문자나 표지 문구를 조정할 수 있다. 기본은 \uppercaseheads이다.

만약 여러분이 사전 정의된 페이지 양식을 쓰고 싶은데 자동 대문자를 사용하고 싶지 않다면, \nouppercaseheads를 쓴 후 페이지 양식을 다시 불러야 한다. 예를 들어, memoir의 기본 페이지 양식을 쓴다면 다음과 같이 하라.

\nouppercaseheads
\pagestyle{headings}

매크로 \createmark{\sec\}-{\marks\}-{\show\}-{\show\}-{\show\}-\show\}-\show\perceived part, chapter, section등과 같은 절 구획에 해당하고, \show\perceived (shownumber 혹은 nonumber)가 \mainmatter에 구획 번호가 표시될지 결정할 때, \sec>mark를 정의한다. \show\perceived 인자는 left, both 혹은 right이고, \show\perceived postfix\perceived 구획 번호 앞과 뒤에 첨부된다. 예를 들어

```
\createmark{section}{left}{nonumber}{}{}
\createmark{section}{both}{nonumber}{}{}
\createmark{section}{right}{nonumber}{}{}
```

는 각각 다음과 같다.

```
\def\sectionmark#1{\markboth{#1}{}}
\def\sectionmark#1{\markboth{#1}{#1}}
\def\sectionmark#1{\markight{#1}}
```

\createmark와 \createplainmark의 차이는, 전자는 인자를 취하는 매크로를 만들고 \createplainmark는 그렇지 않다는 것이다.

이 명령들을 사용해서 memoir의 \makepsmarks{headings}의 현재 정의는 훨씬 간결하다 (이는 또한 toc 등이 \leftmark만이 아니라 \leftmark와 \rightmark 둘 다 표시하기 때문에 약간 다른 결과를 만든다).

\makepsmarks{headings}{%

```
\createmark{chapter}{left}{shownumber}{\@chapapp\}{. \ }
\createmark{section}{right}{shownumber}{\{. \ }
\createplainmark{toc}{both}{\contentsname}
\createplainmark{lof}{both}{\listfigurename}
\createplainmark{lot}{both}{\listtablename}
\createplainmark{bib}{both}{\bibname}
\createplainmark{index}{both}{\indexname}
\createplainmark{glossary}{both}{\glossaryname}}
```

memoir이 페이지 양식의 표지 부분을 행할 때, 이는 오래된 표지를 없애지 않는다. 즉, 만약예전 \sectionmark가 존재한다면 우리가 바꾸지 않더라도 이는 여전히 존재한다. 이는 좋을수도, 나쁠 수도 있는 일이다. 사용자가 이 표지들이 아무것도 하지 않게 재정의할 수 있도록우리는 다음을 제공한다.

```
\label{eq:clearplainmark} $$ \clearplainmark{\langle type \rangle}$ $$ \clearmark{\langle type \rangle}$
```

사용되는 인자 종류는 \createplainmark 및 \createmark와 같다.

## 문서 제목이 있는 머릿글

앞서 언급하였듯, 일부 출판사는 책의 제목이 머릿글에 제목이 들어가기를 선호한다. 아마도 이는 기술적인 출판물이 아닐 것이기에 간단한 머릿글이면 충분할 것이다. 여기에 *myheading*의 사용례가 있다.

\makevenhead{myheadings}{\thepage}{}{DOCUMENT TITLE}
\makeoddhead{myheadings}{Chapter~\thechapter}{}{\thepage}

## 머릿글의 부(部) 와 장(章)

일부 문서는 부와 장 구획을 둘 다 가지고 있고, 이런 경우에는 머릿글에 현재 부와 장 제목이 있으면 독자들에게 유용할 수 있다. headings 페이지 양식은 부와 장 표지를 재설정함으로써 이를 쉽게 구현할 수 있다.

## \makepsmarks{headings}{%

```
\createmark{part}{left}{shownumber}{\partname\ }{. \ }
\createmark{chapter}{right}{shownumber}{\@chapapp\ }{. \ }
\createplainmark{toc}{both}{\contentsname}
\createplainmark{lof}{both}{\listfigurename}
\createplainmark{lot}{both}{\listtablename}
\createplainmark{bib}{both}{\bibname}
\createplainmark{index}{both}{\indexname}
\createplainmark{glossary}{both}{\glossaryname}}
```

## Companion 페이지 양식

이 예시는 대부분의 페이지 양식 명령을 보여준다. LaTeX Companion 시리즈  $[MG^+04, GM^+07, GR99]$ 에는 머릿글이 조판 영역보다 넓어 바깥 여백으로 삐져 나와 있고, 아래에 선이 있다. 페이지 번호는 볼드체로 머릿글 바깥쪽에 있다. 장 제목은 왼쪽 머릿글에 있고 절 제목은 오른쪽 머릿글에 둘 다 볼드체로 안쪽 여백에 있다. 바닥글은 비어 있다.

이 양식을 구현하기 위해서 해야할 첫 번째 일은 여백 주석을 덮는 머릿글의 너비를 계산하는 것이다.

\setlength{\headwidth}{\textwidth}

\addtolength{\headwidth}{\marginparsep}
\addtolength{\headwidth}{\marginparwidth}

이제 우리는 빈 *companion* 페이지 양식을 만들고 새로운 머릿글과 바닥글 너비를 지정하는 것으로 시작하면 된다.

\makepagestyle{companion}

\makerunningwidth{companion}{\headwidth}

그리고 머릿글 선의 너비와 두께를 설정해야 보일 것이다.

\makeheadrule{companion}{\headwidth}{\normalrulethickness}

머릿글이 재단 여백으로 삐져 나오려면 왼쪽 머릿글은 우측 정렬되어야 하고, 오른쪽 머릿글은 좌측 정렬되어야 한다. 바닥글은 비어 있으므로, 이 영역은 중요하지 않다.

\makeheadposition{companion}{flushright}{flushleft}{}}

현재 장과 절 제목은 \leftmark와 \rightmark 매크로에서부터 얻는데, 이들은 \chaptermark와 \sectionmark 매크로로 정의된다. \leftmark는 페이지의 마지막 〈left〉 표지이고 \rightmark는 첫 〈right〉 표지임을 기억하자.

장 번호는 머릿글에 넣지 않지만 절 번호는 있다면 머릿글에 들어간다. 우리는 이들과 다른 비슷한 요소 $ToC^1$ 에 올바른 정의가 사용되었는지 주의를 기울여야 하고, 이때  $\mbox{\ }$ makepsmarks 매크로를 사용하게 된다.

¹ToC와 관련 내용은 Chapter 9에 자세히 설명되어 있다.

```
\makepsmarks{companion}{%
   \nouppercaseheads
   \createmark{chapter}{both}{nonumber}{}{} . \space}
   \createmark{section}{right}{shownumber}{}{. \space}
   \createplainmark{toc}{both}{\contentsname}
   \createplainmark{lof}{both}{\listfigurename}
   \createplainmark{lot}{both}{\listtablename}
   \createplainmark{bib}{both}{\bibname}
   \createplainmark{index}{both}{\lindexname}
   \createplainmark{glossary}{both}{\glossaryname}
```

준비 작업은 모두 완료되었고, 이제 각 머릿글과 바닥글 영역에 어떤 내용이 들어갈지만 지정하면 된다 (그러나 바닥글은 비어 있다).

```
\makeevenhead{companion}%
```

```
{\normalfont\bfseries\thepage}{}{%
```

\normalfont\bfseries\leftmark}

\makeoddhead{companion}%

{\normalfont\bfseries\rightmark}{}{%

\normalfont\bfseries\thepage}

이제 \pagestyle{companion} 명령을 주면 companion 페이지 양식 머릿글을 가진 페이지 가 조판될 것이다. 이 페이지 양식은 본 클래스에 포함되어 있다.

```
\addtopsmarks{\langle pagestyle \rangle}{\langle prepend \rangle}{\langle append \rangle}
```

\addtopsmarks{\pagestyle\}.{\prepend\}.{\append\}.는 이 보조 매크로 모음의 마지막 매크로 이다. 이는 \prepend\와 \append\= \pagestyle\에 대한 \makepsmarks의 현 정의 전과 후에 삽입한다. 예를 들어, 만약 여러분이 companion 페이지 양식의 페이지 머릿글에 \subsection 제목이 나타나기를 원한다면 다음과 같이 할 수 있다.

```
\addtopsmarks{companion}{}{%
```

\createmark{subsection}{right}{shownumber}{}{. \space}}

## ruled 페이지 양식

필자는 실용적인 이유로 장 제목이 머릿글에 적어도 페이지 중앙에, 기술적인 저술에 대해서는 재단 여백 쪽에 있는 페이지 양식을 선호한다. 필자는 또한 페이지 번호가 바깥쪽 모서리에 위치하는 것을 선호한다. 책을 집어 들어 훑어볼 때, 안에 어떤 내용이 있는지 알아보거나 좀 더구체적인 내용을 찾을 때, 필자는 한 손으로 책 등을 잡고 다른 손으로 페이지를 넘긴다. 이때 책이 반쯤 접혀 있기 때문에 책등 가까이 있는 것보다는 재단 여백 쪽에 위치한 것들이 더 눈에잘 띈다. ruled 페이지 양식은 이와 같다. 일반적인 내용은 다음과 같다.

```
\makepagestyle{ruled}
\makeevenfoot {ruled}{\thepage}{}{} % page numbers at the outside
\makeoddfoot {ruled}{}{\thepage}
\makeheadrule {ruled}{\textwidth}{\normalrulethickness}
\makeevenhead {ruled}{\scshape\leftmark}{}{} % small caps
\makeoddhead {ruled}{}{\rightmark}
```

The other part of the specification has to ensure that the \chapter and \section commands make the appropriate marks for the headers. I wanted the numbers to appear in the head-

ers, but not those for sections. The following code sets these up, as well as the marks for the other document elements.

```
\makepsmarks{ruled}{%
  \nouppercaseheads
 \createmark{chapter}{left}{shownumber}{}{. \space}
  \createmark{section}{right}{nonumber}{}{}
 \createplainmark{toc}{both}{\contentsname}
  \createplainmark{lof}{both}{\listfigurename}
 \createplainmark{lot}{both}{\listtablename}
  \createplainmark{bib}{both}{\bibname}
 \createplainmark{index}{both}{\indexname}
  \createplainmark{glossary}{both}{\glossaryname}
}
```

#### 7.3.2 Index headers

If you look at the Index you will see that the header shows the first and last entries on the page. A main entry in the index looks like:

```
\item \idxmark{entry}, page number(s)
and in the preamble to this book \idxmark is defined as
```

This typesets the entry and also uses the entry as markers so that the first entry on a page is held in \rightmark and the last is in \leftmark.

As index entries are usually very short, the Index is set in two columns. Unfortunately LaTeX's marking mechanism can be very fragile on twocolumn pages.<sup>2</sup>

The index itself is called by

```
\clearpage
\pagestyle{index}
\renewcommand{\preindexhook}{%
The first page number is usually, but not always,
the primary reference to
the indexed topic.\vskip\onelineskip}
\printindex
```

The *index* pagestyle, which is the crux of this example, is defined here as:

```
\makepagestyle{index}
 \makeheadrule{index}{\textwidth}{\normalrulethickness}
 \makeevenhead{index}{\rightmark}{}{\leftmark}
 \makeoddhead{index}{\rightmark}{}{\leftmark}
 \makeevenfoot{index}{\thepage}{}{}
 \makeoddfoot{index}{}{}{\thepage}
```

This, as you can hopefully see, puts the first and last index entries on the page into the header at the left and right, with the folios in the footers at the outer margin.

<sup>&</sup>lt;sup>2</sup>This was fixed in the LATEX kernel, but including the functionality from the fixltx2e package.

## 7.3.3 Float pages

```
\left\langle \left\langle yes\right\rangle \right\rangle \left\langle \left\langle no\right\rangle \right\rangle
```

There are occasions when it is desirable to have different headers on pages that only contain figures or tables. If the command \ifonlyfloats is issued on a page that contains no text and only floats then the  $\langle yes \rangle$  argument is processed, otherwise on a normal page the  $\langle no \rangle$  argument is processed. The command is most useful when defining a pagestyle that should be different on a float-only page.

For example, assume that the *companion* pagestyle is to be generally used, but on floatonly pages all that is required is a pagestyle similar to *plain*. Borrowing some code from the *companion* specification this can be accomplished like:

```
\makepagestyle{floatcomp}
% \headwidth has already been defined for the companion style
\makeheadrule{floatcomp}{\headwidth}%
    {\ifonlyfloats{0pt}{\normalrulethickness}}
\makeheadposition{floatcomp}{flushright}{flushleft}{}}
\makepsmarks{floatcomp}{\companionpshook}
\makeevenhead{floatcomp}%
          {\ifonlyfloats{}{\normalfont\bfseries\thepage}}%
          {\ifonlyfloats{}{\normalfont\bfseries\rightmark}}
\makeoddhead{floatcomp}%
          {\ifonlyfloats{}{\normalfont\bfseries\rightmark}}%
          {\ifonlyfloats{}{\normalfont\bfseries\rightmark}}%
          {\ifonlyfloats{}{\normalfont\bfseries\thepage}}}
\makeevenfoot{floatcomp}{}{\ifonlyfloats{\thepage}{}}{}}
\makeoddfoot{floatcomp}{}{\ifonlyfloats{\thepage}{}}}{}
```

The code above for the *floatcomp* style should be compared with that for the earlier *companion* style.

The headrule is invisible on float pages by giving it zero thickness, otherwise it has the \normalrulethickness. The head position is identical for both pagestyles. However, the headers are empty for *floatcomp* and the footers have centered page numbers on float pages; on ordinary pages the footers are empty while the headers are the same as the *companion* headers.

The code includes one 'trick'. The macro \makepsmarks{X}{code} is equivalent to \newcommand{\Xpshook}{code}

I have used this knowledge in the line:

```
\makepsmarks{floatcomp}{\companionpshook}
```

which avoids retyping the code from \makepsmarks{companion}{...}, and ensures that the code is actually the same for the two pagestyles.

If you have two pre-existing pagestyles, one that will be used for text pages and the other that can be used for float pages, then the \mergepagefloatstyle command provides a simpler means of combining them than the above example code for *floatcomp*. The argument

 $\langle style \rangle$  is the name of the pagestyle being defined. The argument  $\langle textstyle \rangle$  is the name of the pagestyle for text pages and  $\langle floatstyle \rangle$  is the name of the pagestyle for float-only pages. Both of these must have been defined before calling \mergepagefloatstyle. So, instead of the long winded, and possibly tricky, code I could have simply said:

```
\mergepagefloatstyle{floatcomp}{companion}{plain}
```

One author thought it would be nice to be able to have different page headings according to whether the page was a floatpage, or there was a float at the top of the page, or a float at the bottom of a page or there was text at the top and bottom.

This, I think, is not a common requirement and, further, that to provide this involves changing parts of the LaTeX output routine — something only to be tackled by the bravest of the brave. If it were to be done then were best done in a package that could be easily ignored. The following is an outline of what might be done; I do not recommend it and if you try this and all your work dissappears then on your own head be it.

```
% notefloat.stv
\newif\iffloatattop
  \floatattopfalse
\newif\iffloatatbot
  \floatatbotfalse
\renewcommand*{\@addtotoporbot}{%
 \@getfpsbit \tw@
 \ifodd \@tempcnta
    \@flsetnum \@topnum
    \ifnum \@topnum>\z@
      \@tempswafalse
      \@flcheckspace \@toproom \@toplist
      \if@tempswa
        \@bitor\@currtype{\@midlist\@botlist}%
        \if@test
        \else
          \@flupdates \@topnum \@toproom \@toplist
          \@inserttrue
  \global\floatattoptrue
        \fi
      \fi
    \fi
 \fi
 \if@insert
  \else
    \@addtobot
  \fi}
\renewcommand*{\@addtobot}{%
 \@getfpsbit 4\relax
 \ifodd \@tempcnta
    \@flsetnum \@botnum
```

```
\ifnum \@botnum>\z@
      \@tempswafalse
     \@flcheckspace \@botroom \@botlist
     \if@tempswa
        \global \maxdepth \z@
        \@flupdates \@botnum \@botroom \@botlist
        \@inserttrue
 \global\floatatbottrue
      \fi
    \fi
 \fi}
\let\p@wold@output\@outputpage
\renewcommand*{\@outputpage}{%
 \p@wold@output
 \global\floatattopfalse
 \global\floatatbotfalse}
\endinput
```

\floatattop is probably set true if there is a float at the top of the page and \floatatbot is probably set true if there is a float at the bottom of the page.

### 7.4 The showloss pagestyle

The *showlocs* pagestyle is somewhat special as it is meant to be used as an aid when designing a page layout. Lines are drawn showing the vertical positions of the headers and footers and a box is drawn around the textblock. It is implemented using two zero-sized pictures.<sup>3</sup>

```
\framepichead
\framepictextfoot
\framepichook
\showheadfootlocoff
\showtextblockoff
```

The macro \framepichead creates a zero-sized picture that draws a line at the header location, and the macro \framepictextfoot creates a zero-sized picture that draws a line at the footer location and also draws a box around the typeblock. Following the declaration \showheadfootlocoff the macros \framepichead and \framepictextfoot do not draw lines showing the header and footer locations. The declaration \showtextblockoff prevents \framepictextfoot from drawing a box around the textblock.

In case you want to change the color of the *showlocs*, simply do

```
\renewcommand\framepichook{\color{red}}
```

If you generally want a box around the textblock you may want to create your own pagestyle using \framepictextfoot and the *showlocs* code as a starting point, see memoir. cls for details.

<sup>&</sup>lt;sup>3</sup>A zero-sized picture starts off with begin{picture}(0,0)....

# 7.5 Other things to do with page styles

Back on page 111 we presented a way of adding some draft information. Here is a more advanced example of this.

One interesting use for page styles is to provide extra information below the footer. This might be some kind of copyright information. Or if your document is under version control with a system like Subversion, and you have all your chapter laying in seperate files, then why not add information at the start of very chapter, specifying who did the last change to this chapter at which time. See the svn-multi package ([Sch09]) and the PracTEX Journal article [Sch07] by the same author. Then this information can be added to the start of every chapter using something like:

```
\usepackage[filehooks]{svn-multi}
\makeatletter
% remember to define a darkgray color
\newcommand\addRevisionData{%
  \begin{picture}(0,0)%
    \put(0,-10){%
      \tiny%
      \expandafter\@ifmtarg\expandafter{\svnfiledate}{}{%
        \textcolor{darkgray}{Chapter last updated
          \svnfileyear/\svnfilemonth/\svnfileday
         \enspace \svnfilehour:\svnfileminute\ (revision \svnfilerev)}
     }%
    }%
  \end{picture}%
}
\makeatother
% chapter is normally an alias to the plain style, we want to change
% it, so make it a real pagestyle
\makepagestyle{chapter}
\makeoddfoot{chapter}{\addRevisionData}{\thepage}{}
\makeevenfoot{chapter}{\addRevisionData}{\thepage}{}
```

# Eight

## 문단과 리스트

절 구분 내에서 보통 내용은 문단들로 나뉘어진다. 이따금 인용문이나 리스트와 같이 일반적인 문단과는 다르게 나열되는 내용이 있을 수 있다.

## 8.1 문단

기본적으로, 보통 문단의 형태를 조절하는 매개변수가 두 개 있다.

```
\parindent \parskip
```

길이 \parindent는 문단 첫 줄의 들여쓰기이며, 길이 \parskip은 문단 사이의 세로 간격로, Figure 8.1에 나타나 있다. \parskip의 값은 보통 0pt이고, \parindent는 보통 em 단위로정의되어서 사용되는 글꼴에 따라 들여쓰기가 달라지도록 한다. 만약 \parindent가 음의길이로 설정이 되며, 문단의 첫 줄은 왼쪽 여백으로 '내어쓰기'가 될 것이다.

#### 8.1.1 블록 문단

블록 문단은 \parindent를 0em으로 설정하여 얻을 수 있다. \parskip은 어떤 양의 값으로 설정하여 문단들이 구분될 수 있도록 해야 한다. 대부분의 인쇄기술자는 블럭 문단을 진심으로 싫어하는데, 이는 단지 심미적인 측면이 아니라 실용적인 이유에서도 그러하다. 블럭 문단의 마지막 줄이 꽉 차 있으면서도 그 페이지의 마지막 줄인 상황을 생각해보자. 이어지는 블럭 문단은 다음 페이지의 맨 위에서 시작하지만, 이를 구분할 수 있는 문단 사이의 간격은 없다.

It is important to know that LaTeX typesets paragraph by paragraph. For example, the \baselineskip that is used for a paragraph is the value that is in effect at the end of the paragraph, and the font size used for a paragraph is according to the size declaration (e.g., \large or \normalsize or \small) at the end of the paragraph, and the raggedness or

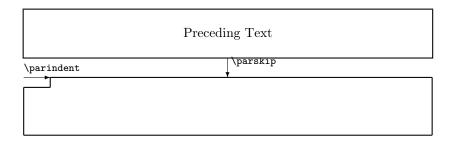


Figure 8.1: 문단의 매개변수

otherwise of the whole paragraph depends on the declaration (e.g., \centering) in effect at the end of the paragraph. If a pagebreak occurs in the middle of a paragraph TeX will not reset the part of the paragraph that goes onto the following page, even if the textwidths on the two pages are different. LaTeX은 문단 단위로 조판한다는 사실을 알아두는 것이 중요하다. 예를 들어, \baselineskip은

## 8.1.2 Hanging paragraphs

A hanging paragraph is one where the length of the first few lines differs from the length of the remaining lines. (A normal indented paragraph may be considered to be a special case of a hanging paragraph where 'few = one').

Using  $\n$  at the start of a paragraph will cause the paragraph to be hung. If the length  $\langle indent \rangle$  is positive the lefthand end of the lines will be indented but if it is negative the righthand ends will be indented by the specified amount. If the number  $\langle num \rangle$ , say N, is negative the first N lines of the paragraph will be indented while if N is positive the N+1 th lines onwards will be indented. This paragraph was set with  $\n$  angpara (3em) {-3}. There should be no space between the  $\n$  and the start of the paragraph.

The hangparas environment is like the \hangpara command except that every paragraph in the environment will be hung.

The code implementing the hanging paragraphs is the same as for the hanging package [Wil01f]. Examples of some uses can be found in [Thi99].

As noted eleswhere the sectioning commands use the internal macro \@hangfrom as part of the formatting of the titles.

The \hangfrom macro is provided as an author's version of the internal \@hangfrom macro used, among other things, in typesetting section titles.

Simple hung paragraphs (like this one) can be specified using the \hangfrom macro. The macro puts \langle text \rangle in a box and then makes a hanging paragraph of the following material. This paragraph commenced with \hangfrom{Simple hung paragraphs }(like ... and you are now reading the result.

The commands for hanging paragraphs do not quite work as might be expected when they are used in a list environment, for example inside an enumerate. If you wish for a hanging paragraph inside such an environment you will have to define your own commands for this. If you feel capable of doing so then, with my congratulations, move on to the next section. If you are not so confident you could try using the following non-guaranteed code, which is based on an idea by Patrik Nyman which he posted on CTT in January 2004.

```
%\makeatletter
% A version of \hangpara for use in a list
% \listhanging{indent}{num} text text text ...
\def\listhanging#1#2#3\par{%
  \@tempdima\textwidth \advance\@tempdima -\leftmargin
```

```
\parbox{\@tempdima}{\hangpara{#1}{#2}#3}\par}
% A version of \hangfrom for use in a list
% \listhangfrom{stuff} text text text ...
\def\listhangfrom#1#2\par{%
  \@tempdima\textwidth \advance\@tempdima -\leftmargin
  \parbox{\@tempdima}{\@hangfrom{#1}#2}\par}
%\makeatother
```

## 8.2 Flush and ragged

Flushleft text has the lefthand end of the lines aligned vertically at the lefthand margin and flushright text has the righthand end of the lines aligned vertically at the righthand margin. The opposites of these are raggedleft text where the lefthand ends are not aligned and raggedright where the righthand end of lines are not aligned. LaTeX normally typesets flushleft and flushright.

```
\begin{flushleft} text \end{flushleft}
\begin{flushright} text \end{flushright}
\begin{center} text \end{center}
```

Text in a flushleft environment is typeset flushleft and raggedright, while in a flushright environment is typeset raggedleft and flushright. In a center environment the text is set raggedleft and raggedright, and each line is centered. A small vertical space is put before and after each of these environments.

```
\raggedleft \raggedright \centering
```

The \raggedleft declaration can be used to have text typeset raggedleft and flushright, and similary the declaration \raggedright causes typesetting to be flushleft and raggedright. The declaration \centering typesets raggedleft and raggedright with each line centered. Unlike the environments, no additional space is added.

```
\centerfloat
```

The contents of floats like tables or figures are usually centered and \centering should be used for this, not the center environment which adds extra, usually undesired, vertical space. For example:

```
\begin{figure}
\centering
...
\caption{...}
\end{figure}
```

However, if the float is wider than the textblock then it is aligned with the left margin and extends into the right margin. The command \centerfloat is a special version of \center that when used in a wide float will center it with respect to the textblock, i.e., it will extend equally into both margins. Note that \centerfloat needs to be applied where there is a known width; if applied to a regular text paragraph it will center the paragraph but put all the text on one line.

```
\raggedyright[\langle space \raggedyright[\langle space \raggedyright[\langle space \rangle space \r
```

## Typeset example 8.1: Setting the source of a quotation

This quotation has a short last line so there there is enough space for the source to be set at the end of the line.

I. M. Short

The last line of this quotation turns out to be too long for the source to be set at the end, so it is automatically set flushright on the following line.

N. O. Space

When using \raggedright in narrow columns the right hand edge tends to be too ragged, and paragraphs are not indented. Text set \raggedyright usually fills more of the available width and paragraphs are indented by \raggraphiant, which is initially set to \parindent. The optional  $\langle space \rangle$  argument, whose default is 2em, can be used to adjust the amount of raggedness. As examples:

Remember that LaTeX typesets on a per-paragraph basis, so that putting the sequence of \centering, \raggedleft declarations in the same paragraph will cause the entire paragraph to be typeset raggedleft and flushright — the \centering declaration is not the one in effect at the end of the paragraph.

#### 8.3 Quotations

LaTeX provides two environments that are typically used for typesetting quotations.

```
\begin{quote} text \end{quote} \begin{quotation} text \end{quotation}
```

In both of these environments the text is set flushleft and flushright in a measure that is smaller than the normal textwidth. The only difference between the two environments is that paragraphs are not indented in the quote environment but normal paragraphing is used in the quotation environment.

```
\scalebox{ } sourceatright[\langle length \rangle] {\langle text \rangle}
```

Some quotations are completed by giving the source or author. Using \sourceatright at the end of the quotation will typeset  $\langle text \rangle$  flushright at the end of the line if there is enough space, otherwise it typesets it flushright on the next line. A space  $\langle length \rangle$  (default 2em) is left between the end of the quote and  $\langle text \rangle$ .

```
Source for example 8.1 \begin{quotation} This quotation has a short last line so there there is enough space
```

```
for the source to be set at
the end of the line.\sourceatright{I. M. Short}
\end{quotation}

\begin{quotation}

The last line of this quotation turns out to be too long for
the source to be set at the end, so it is automatically
set flushright on the following line.\sourceatright{N. O. Space}
\end{quotation}
```

#### 8.4 Some less common paragraph shapes

The paragraph shapes described in this section are based on a series that I presented in my *Glisterings* column [Wil07e, Wil08b]. Like the earlier \centering, etc., paragraph style declarations, the style that applies is the one in effect at the *end* of the paragraph. Thus the general usage is:

```
\bgroup% a group to keep changes local % or could be { or \begin...
\paragraphstyle
.... text
\par% ensure the end of a paragraph
\egroup% end the group % or could be } or \end...
```

If you use one of these paragraph shapes then using \\ to break a line may give a surprising result. If so, the following may help.

```
\atcentercr
\break
\memorigdbs
\memorigpar
```

You could try \atcentcr, which is user level version of an internal LaTeX command used in some paragraph settings for line breaking, or \break, which is a TeX command to end a line.

In some cases the paragraph shaping commands change the definitions of \\ or \par. Just in case you need to restore them, copies of the original definitions are in \memorigdbs (for \\) and \memorigpar (for \par).

```
\flushleftright
```

If you use one of the shapes listed later in this section and things go wrong, the declaration \flushleftright returns all paragraphing parameters¹ to their normal values, thus producing paragraphs as normal — justified left and right with the last line flushleft and raggedright.

## 8.4.1 Last line not short

On occasion a paragraph may end with a single short word as the last line.

 $<sup>^{1}\</sup>mathrm{Except}$  for the **\parindent**, which it leaves at its current value.

## Typeset example 8.2: Paragraph's line not too short

The last line of this paragraph will be no shorter than a particular length. a b c d e f g h i

The last line of this paragraph will be no shorter than a particular length. a b c d e f g h i j k

## $\label{linear} \$ linenottooshort [ $\langle length \rangle$ ]

length. a b c d e f g h i % j k l m n

Following the \linenottooshort declaration paragraphs will be set as normal, except that the last line will not be shorter than  $\langle length \rangle$  (default 2em).

## Source for example 8.2

```
\linenottooshort[4em]
The last line of this paragraph will be no shorter than a particular
```

The last line of this paragraph will be no shorter than a particular length. a b c d e f g h i j k % l m n

#### 8.4.2 Russian typography

Apparently in the Russian typographic tradition the last line of a multiline paragraph must either be at least as long as the \parindent and have at least \parindent at the end, or it must fill the whole line (i.e., flushleft and flushright).

```
\russianpar
```

Ending a paragraph with \russianpar causes it to be set following Russian typographic rules.

If you have many such paragraphs it may be more convenient to do it like:

```
\let\par\russianpar
... many paragraphs
\let\par\memorigpar

or as:
  \begingroup% start a group
\let\par\russianpar
... many paragraphs
\endgroup% end the group
```

## 8.4.3 Fill with rules

In some legal documents there must be no space at the end of the lines in order to prevent anyone inserting something at a later date. Typically it is only the last line in a paragraph that needs this treatment.

## Typeset example 8.3: Rules for spaces

The last line of this paragraph will be be set by ending it with a rule to fill up any space. —

## Typeset example 8.4: Ragged paragraphs

Paragraphs following the \justlastraggedleft declaration, as this one does, have their lines justified except for the last which is set raggedleft. The demonstration works best if there are three or more lines.

This paragraph is set following the \raggedrightthenleft declaration. The first line is set raggedright and all the remaining lines are set raggedleft. The demonstration is better if there are three or more lines.

This paragraph is set following the **\leftcenterright** declaration. We really need three,

or four may be better,

lines to show the effect of this.

\lastlinerulefill \lastlineparrule

## Source for example 8.3

The last line of this paragraph will be be set by ending it with a rule to fill up any space.\lastlinerulefill

Using \lastlinerulefill to end a paragraph will cause any spaces at the ends of the lines to be filled with the \lastlineparrule rule. If you have many paragraphs of this kind then try:

\let\par\lastlinerulefill
.... many paragraphs
\let\par\memorigpar

Remember that LaTeX treats many constructs (like section headings or captions) as paragraphs, so you may have to alternate between filled text paragraphs and regular paragraphing.

#### 8.4.4 Some ragged paragraphs

A few paragraph shapes with unusual ragged lines are available.

```
\justlastraggedleft
\raggedrightthenleft
\leftcenterright
```

Following the \justlastraggedleft declaration paragraphs will be set justified except the last line will be set raggedleft.

Following the declaration \raggedrightthenleft paragraphs will be set with the first line raggedright and the remainder set raggedleft.

Following the declaration \leftcenteright paragraphs will be set with the first line flushleft (and raggedright) and the last line flushright (and raggedleft) and those in the middle will be centered. This declaration should be used within a group; also \everypar{} should be called at the end.

```
Source for example 8.4
\justlastraggedleft
Paragraphs following the \verb?\justlastraggedleft? declaration, as
this one does, have their lines justified except for the last which
is set raggedleft. The demonstration works best if there are three
or more lines.
\raggedrightthenleft
This paragraph is set following the \verb?\raggedrightthenleft?
declaration. The first line is set raggedright and all the remaining
lines are set raggedleft. The demonstration is better if there are three or
more lines.
\leftcenterright
This paragraph is set following the \verb?\leftcenterright?
declaration. We really need three, \\ or four may be better, \\
lines to show the effect of this.
\everypar{}
```

## 8.4.5 Left spring right

Typically the lines of a paragraph are both flushleft and flushright and filled with text, but sometimes filling is not desired.

```
\label{leftspringright} $$ \left(\frac{lfrac}{(frac)}, \frac{(ltext)}{(rtext)}\right) $$
```

The \leftspringright macro sets  $\langle ltext \rangle$  flushleft and raggedright in a column whose width is  $\langle lfrac \rangle$  of the textwidth and, in parallel, it also sets  $\langle rtext \rangle$  raggedleft and flushright in a column that is  $\langle rfrac \rangle$  of the textwidth; the effect is as though there are springs between the lines of the two texts. The sum of  $\langle lfrac \rangle$  and  $\langle rfac \rangle$  must be less than one.

```
Source for example 8.5
```

\leftspringright{0.3}{0.6}%

## Typeset example 8.5: A sprung paragraph

Text at the left is set flushleft and raggedright.

But the text at the right is set ragged left and flushright. It's as though there was a spring pushing the lines apart.

```
{Text at the left is set flushleft and raggedright.}
{But the text at the right is set raggedleft and flushright.
It's as though there was a spring pushing the lines apart.}
```

#### 8.5 Changing the textwidth

The quote and quotation environments both locally change the textwidth, or more precisely, they temporarily increase the left and right margins by equal amounts. Generally speaking it is not a good idea to change the textwidth but sometimes it may be called for.

The commands and environment described below are similar to those in the originally found in the chngpage package, but do differ in some respects.

The adjustwidth environment temporarily adds the length  $\langle left \rangle$  to the lefthand margin and  $\langle right \rangle$  to the righthand margin. That is, a positive length value increases the margin and hence reduces the textwidth, and a negative value reduces the margin and increases the textwidth. The quotation environment is roughly equivalent to

```
\begin{adjustwidth}{2.5em}{2.5em}
```

The starred version of the environment, adjustwidth\*, is really only useful if the left and right margin adjustments are different. The starred version checks the page number and if it is odd then adjusts the left (spine) and right (outer) margins by  $\langle left \rangle$  and  $\langle right \rangle$  respectively; if the page number is even (a verso page) it adjusts the left (outer) and right (spine) margins by  $\langle right \rangle$  and  $\langle left \rangle$  respectively.

```
\strictpagecheck \easypagecheck
```

Odd/even page checking may be either strict (\strictpagecheck) or easy (or one might call it lazy) (\easypagecheck). Easy checking works most of the time but if it fails at any point then the strict checking should be used.

As an example, if a figure is wider than the textwidth it will stick out into the righthand margin. It may be desireable to have any wide figure stick out into the outer margin where there is usually more room than at the spine margin. This can be accomplished by

```
\begin{figure}
\centering
```

```
\strictpagecheck
\begin{adjustwidth*}{0em}{-3em}
% the illustration
\caption{...}
\end{adjustwidth*}
\end{figure}
```

A real example in this manual is Table 10.1 on page 179, which is wider than the typeblock. In that case I just centered it by using adjustwidth to decrease each margin equally. In brief, like

```
\begin{table}
\begin{adjustwidth}{-1cm}{-1cm}
\centering
...
\end{adjustwidth}
\end{table}
```

Note that the adjustwidth environment applies to complete paragraphs; you can't change the width of part of a paragraph except for hanging paragraphs or more esoterically via \parshape. Further, if the adjusted paragraph crosses a page boundary the margin changes are constant; a paragraph that is, say, wider at the right on the first page will also be wider at the right as it continues onto the following page.

The center environment horizontally centers its contents with respect to the typeblock. Sometimes you may wish to horizontally center some text with respect to the physical page, for example when typesetting a colophon which may look odd centered with respect to the (unseen) typeblock.

The calculation of the necessary changes to the spine and fore-edge margins is simple. Using the same symbols as earlier in §2.4 ( $P_w$  and  $B_w$  are the width of the trimmed page and the typeblock, respectively; S and E are the spine and fore-edge margins, respectively) then the amount M to be added to the spine margin and subtracted from the fore-edge margin is calculated as:

$$M = 1/2(P_w - B_w) - S$$

For example, assume that the \textwidth is 5 inches and the \spinemargin is 1 inch. On US letterpaper (\paperwidth is 8.5 inches) the fore-edge margin is then 2.5 inches, and 0.75 inches<sup>2</sup> must be added to the spine margin and subtracted from the fore-edge to center the typeblock. The adjustwidth environment can be used to make the (temporary) change.

\begin{adjustwidth\*}{0.75in}{-0.75in} ...

If you don't want to do the above calculations by hand, \calccentering will do it for you. The  $\langle length \rangle$  argument must be the name of a pre-existing length command, including the backslash. After calling \calccentering,  $\langle length \rangle$  is the amount to be added to the spine margin and subtracted from the foredge margin to center the typeblock. An example usage is

<sup>&</sup>lt;sup>2</sup>On A4 paper the result would be different.

```
\calccentering{\mylength}
\begin{adjustwidth*}{\mylength}{-\mylength}
text horizontally centered on the physical page
\end{adjustwidth*}
```

You do not necessarily have to define a new length for the purposes of \calccentering. Any existing length will do, such as \unitlength, provided it will be otherwise unused between performing the calculation and changing the margins (and that you can, if necessary reset it to its original value — the default value for \unitlength is 1pt).

8.6 Lists

Standard LaTeX provides four kinds of lists. There is a general list environment which you can use to define your own particular kind of list, and the description, itemize and enumerate lists (which are internally defined in terms of the general list environment<sup>3</sup>).

This class provides the normal description list, plus a couple of others of the same kind, but the itemize and enumerate lists are extended versions of the normal ones.

In a description list an  $\land$ item's  $\langle label \rangle$  is typeset by descriptionlabel. The default definition is

which gives a bold label. To have, for example, a sans label instead, do

The only noticeable difference between a description list and a blockdescription list is that the latter is set as indented block paragraphs; invisibly, it also has its own \blockdescriptionlabel.

The labelled environment is like the description environment except that you can specify the label format via the  $\langle name \rangle$  argument where \name is the formatting macro. For example, if you wanted the item labels set in italics, then

```
\newcommand*{\itlabel}[1]{\hspace\labelsep \normalfont\itshape #1}
\begin{labelled}{itlabel}
\item[First] ...
```

 $<sup>^3</sup>$ The quote and quotation environments are also defined in terms of the general list environment. You may be surprised where it crops up.

## Typeset example 8.6: Smallcap quote style description list

This example shows how the flexlabelled list can be used to change the formatting of a description-like list.

First The labels should be typeset using smallcaps and the first paragraph should be set as block paragraph.

Further paragraphs completing an \item's descriptive text will be set with the normal paragraph indent.

Second The list should be indented from each margin like the quote and quotation environments.

More major changes to a description-like list will probably involve writing the code for a new environment.

. . .

The flexlabelled environment adds additional controls to the labelled one. The  $\langle name \rangle$  argument is the same as that for labelled and the remainder are lengths that correspond to the dimensions shown in Figure 8.2. If you want any of the dimensions to retain their current values, use \* instead of a length as the value for that particular argument.

## Source for example 8.6

This example shows how the \texttt{flexlabelled} list can be used to change the formatting of a description-like list.
\newcommand\*{\sclabel}[1]{\normalfont\scshape #1}
\begin{flexlabelled}{sclabel}{0pt}{0.5em}{0.5em}{\*}{\leftmargin}
\item[First] The labels should be typeset using smallcaps and the first paragraph should be set as block paragraph.

Further paragraphs completing an \cs{item}'s descriptive text will be set with the normal paragraph indent.

\item[Second] The list should be indented from each margin like the \texttt{quote} and \texttt{quotation} environments.

\end{flexlabelled}

More major changes to a description-like list will probably involve writing the code for a new environment.

The itemize and enumerate environments below are based on the enumerate package [Car98c].

The normal markers for \items in an itemize list are:

bullet (●\textbullet),

- bold en-dash (-\bfseries\textendash),
- 3. centered asterisk (\*\textasteriskcentered), and
- 4. centered dot ( · \textperiodcentered).

The optional  $\langle marker \rangle$  argument can be used to specify the marker for the list items in a particular list. If for some reason you wanted to use a pilcrow symbol as the item marker for a particular list you could do

The normal markers for, say, the third item in an enumerate list are: 3., c., iii., and C. The optional  $\langle style \rangle$  argument can be used to specify the style used to typeset the item counter. An occurrence of one of the special characters A, a, I, i or 1 in  $\langle style \rangle$  specifies that the counter will be typeset using uppercase letters (A), lowercase letters (a), uppercase Roman numerals (I), lowercase Roman numerals (i), or arabic numerals (1). These characters may be surrounded by any LaTeX commands or characters, but if so the special characters must be put inside braces (e.g., {a}) if they are to be considered as ordinary characters instead of as special styling characters. For example, to have the counter typeset as a lowercase Roman numeral followed by a single parenthesis

```
\begin{enumerate}[i)]
...
```

#### Recommended alternative

memoir does not provide high level interfaces to configure the appearance. We provide some simple tools to adjust vertical spacing, see below.

Users seeking more control can have a look at the excellent enumitem package by Javier Bezos. If loaded as

```
\usepackage[shortlabels]{enumitem}
then our
\begin{enumerate}[i)]
\item \label{item:tst} ...
syntax will work out of the box.
```

One key difference: In memoir \ref{item:tst} will give you 'i', whereas, if enumitem is loaded the full formatting is returned from the cross reference, i.e., 'i)'. This is fully configurable in enumitem.

Note that, \tightlists, \defaultlists, \firmlists, \firmlists\* presented below, are not supported by enumitem, it provides a highlevel key based interface instead.

Another feature from enumitem that I (LM) uses a lot is to combine the \setlist with \AtBeginEnvironment from the etoolbox package to specifically adjust enumerate used inside certain theorem constructions. That wway we can control the apparence of the lists from the preamble and does not need to use say

\begin{enumerate}[(a)] every single time.

```
\tightlists \defaultlists \firmlists \firmlists*
```

The normal LaTeX description, itemize and enumerate lists have an open look about them when they are typeset as there is significant vertical space between the items in the lists. After the declaration \tightlists is issued, the extra vertical spacing between the list items is deleted. The open list appearance is used after the \defaultlists declaration is issued. These declarations, if used, must come before the relevant list environment(s). The class initially sets \defaultlists. This manual, though, uses \tightlists. The spacing following the \firmlists declaration is intermediate between \defaultlists and \tightlists. The starred version, \firmlists\*, allows sligthly less space around the lists when they are preceded by a blank line than does the unstarred \firmlists.

Caveat. Due to the manner in which \small and \footnotesize are implemented, \tightlists and \firmlists will have no effect on lists typeset under \small or \footnotesize.

A comprehensible solution can be done via the enumitem package via

```
% \tightlists equivalent
\usepackage[shortlabels]{enumitem}
\setlist{ noitemsep }
```

```
\firmlist \tightlist
```

The command \firmlist or \tightlist can be used immediately after the start of a list environment to reduce the vertical space within that list. The \tightlist removes all the spaces while the \firmlist produces a list that still has some space but not as much as in an ordinary list.

```
\begin{list}{\langle default-label\rangle}{\langle code\rangle}\ items \end{list}
```

LaTeX's list environments are defined in terms of a general list environment; some other environments, such as the quote, quotation and adjustwidth are also defined in terms of a list. Figure 8.2 shows the parameters controlling the layout of the list environment.

The list environment takes two arguments. The  $\langle \textit{default-label} \rangle$  argument is the code that should be used when the \item macro is used without its optional  $\langle \textit{label} \rangle$  argument. For lists like enumerate this is specified but often it is left empty, such as for the adjustwidth environment.

The  $\langle code \rangle$  argument is typically used for setting the particular values of the list layout parameters. When defining your own types of lists it is advisable to set each of the parameters unless you know that the default values are suitable for your purposes. These parameters can all be modified with either the \setlength or \addtolength commands.

As an example, here is the specification for a description-like list that uses an italic rather than bold font for the items, and is somewhat tighter than the normal description list.

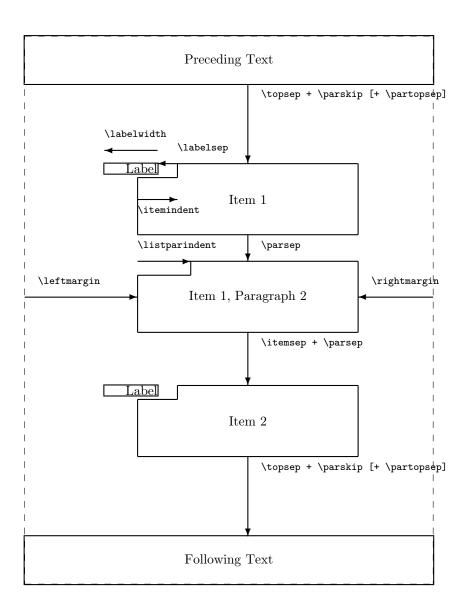


Figure 8.2: The layout parameters for general lists

```
\newcommand{\itlabel}[1]{\hspace\labelsep\normalfont\itshape #1}
 \newenvironment{itdesc}{%
   \list{}{%
     \setlength{\labelsep}{0.5em}
     \setlength{\itemindent}{Opt}
     \setlength{\leftmargin}{\parindent}
     \setlength{\labelwidth}{\leftmargin}
     \addtolength{\labelwidth}{-\labelsep}
     \setlength{\listparindent}{\parindent}
     \setlength{\parsep}{\parskip}
     \setlength{\itemsep}{0.5\onelineskip}
     \let\makelabel\itlabel}}{\endlist}
   This gets used like any other list:
 \begin{itdesc}
 \item[label] ....
 \end{itdesc}
   Here is another kind of list called symbols that might be used for a list of symbols or
other similar kind of listing.
 % Symbol list
 \newenvironment{symbols}%
                 {\list{}% empty label
                              {\setlength{\topsep}{\baselineskip}
                               \setlength{\partopsep}{0pt}
                                \setlength{\itemsep}{0.5\baselineskip}
                                \setlength{\parsep}{0pt}
                                \setlength{\leftmargin}{2em}
                                \setlength{\rightmargin}{0em}
                                \setlength{\listparindent}{1em}
                                \setlength{\itemindent}{0em}
                                \setlength{\labelwidth}{0em}
                                \setlength{\labelsep}{2em}}}%
                 {\endlist}
 \newcommand{\symb}[1]{\item[#1]\mbox{}\\nopagebreak}
In this case it gets used like this
 \begin{symbols}
 \symb{SYMBOL 1} definition
 \symb{SYMBOL 2} ...
 \end{symbols}
```

%%%%% An italic and tighter description environment

In the code for the symbols list I used the command forms (i.e., \list and \endlist) for specifying the start and end of a list. It is a matter of taste whether you use the command or  $\end{\dots}$  and  $\end{\dots}$  forms, but the latter does make it more obvious that an environment is being dealt with.

Several LaTeX environments are defined in terms of a very simple list, called a trivlist. Such a list has little internal structure but like the list environment the vertical space before

## Typeset example 8.7: Changing space before and after lists

This example shows that the space around the

#### CENTER AND OTHER LIST ENVIRONMENTS

can be minimised by using the

\zerotrivseps declaration.

The normal spacing can be restored by using the

\restoretrivseps command.

An alternative is to use the \centering macro.

and after a trivlist (or any list based on it) is set by \topsep and \partopsep, as shown in Figure 8.2.

```
\zerotrivseps \savetrivseps \restoretrivseps
```

The center environment is one of several that is based on a trivlist, and so has space before and after it. If you don't want this the \zerotrivseps declaration eliminates those spaces. You can think of it as being defined as:

```
\newcommand*{\zerotrivseps}{%
  \setlength{\topsep}{0pt}%
  \setlength{\partopsep}{0pt}}
```

Before doing this, though, you might consider calling \savetrivseps which stores the current values of \topsep and \partopsep; it is initially defined to store the default values. The command \restoretrivseps sets the current values of these lengths to the ones saved by \savetrivseps.

## Source for example 8.7

This example shows that the space around the \begin{center}
CENTER AND OTHER LIST ENVIRONMENTS
\end{center}
can be minimised by using the \zerotrivseps
\begin{center}
\verb?\zerotrivseps? declaration.
\end{center}
The normal spacing can be restored by using the \restoretrivseps
\begin{center}
\verb?\restoretrivseps? command.
\end{center}
An alternative is to use the \verb?\centering? macro. Among the environments defined in terms of a trivlist are: flushleft, center, flushright, verbatim, and others. The example (8.7) shows how it is possible to change the spacing around the center environment, but it applies equally to the other environments.

## Nine

## Contents lists

This chapter describes how to change the appearance of the Table of Contents (ToC) and similar lists like the List of Figures (LoF). In the standard classes the typographic design of these is virtually fixed as it is buried within the class definitions.

As well as allowing these lists to appear multiple times in a document, the memoir class gives handles to easily manipulate the design elements. The class also provides means for you to define your own new kinds of "List of $\cdots$ ".

The functionality described is equivalent to the combination of the tocloft and tocbibind packages [Wil01i, Wil01h].

```
\tableofcontents \tableofcontents*
\listoffigures \listoffigures*
\listoftables \listoftables*
```

The commands \tableofcontents, \listoffigures and \listoftables typeset, repectively, the Table of Contents (ToC), List of Figures (LoF) and List of Tables (LoT). In memoir, unlike the standard classes, the unstarred versions add their respective titles to the ToC. The starred versions act like the standard classes' unstarred versions as they don't add their titles to the ToC.

This chapter explains the inner workings behind the ToC and friends, how to change their appearance and the appearance of the entries, and how to create new 'List of $\cdots$ '. If you don't need any of these then you can skip the remainder of the chapter.

#### 9.1 General ToC methods

In §9.2 we will provide the class configuration interface for the various parts of the ToC.

In order to understand how these macros are used, we start by providing some background information this is a general description of how the standard LaTeX classes process a Table of Contents (ToC). As the processing of List of Figures (LoF) and List of Tables (LoT) is similar I will just discuss the ToC. You may wish to skip this section on your first reading.

The basic process is that each sectioning command writes out information about itself — its number, title, and page — to the toc file. The \tableofcontents command reads this file and typesets the contents.

First of all, remember that each sectional division has an associated level as listed in in Table 6.1 on page 79. LaTeX will not typeset an entry in the ToC unless the value of the tocdepth counter is equal to or greater than the level of the entry. The value of the tocdepth counter can be changed by using \setcounter or \settocdepth.

```
\verb| \addcontentsline{| \langle file \rangle | {\langle kind \rangle | {\langle text \rangle | }}|}
```

LaTeX generates a toc file if the document contains a \tableofcontents command. The sectioning commands put entries into the toc file by calling the \addcontentsline command, where  $\langle file \rangle$  is the file extension (e.g., toc),  $\langle kind \rangle$  is the kind of entry (e.g., section or subsection), and  $\langle text \rangle$  is the (numbered) title text. In the cases where there is a number, the  $\langle text \rangle$  argument is given in the form {\numberline{number}title text}.

```
\contentsline{\langle kind \rangle} {\langle text \rangle} {\langle page \rangle}
```

The \addcontentsline command writes an entry to the given file in the form: \contentsline $\{\langle kind \rangle\} \{\langle text \rangle\} \{\langle page \rangle\}$ 

where  $\langle page \rangle$  is the page number.

For example, if \section{Head text} was typeset as '3.4 Head text' on page 27, then there would be the following entry in the toc file:

\contentsline{section}{\numberline{3.4} Head text}{27}

Extracts from toc, lof and lot files are shown in Figure 9.1.

For each  $\langle kind \rangle$  that might appear in a toc (lof, lot) file, LaTeX provides a command:  $\log \{\langle title \rangle\} \{\langle page \rangle\}$ 

which performs the actual typesetting of the \contentsline entry.

The general layout of a typeset entry is illustrated in Figure 9.2. There are three internal La-TeX commands that are used in the typesetting. The page number is typeset flushright in a box of width \@pnumwidth, and the box is at the righthand margin. If the page number is too long to fit into the box it will stick out into the righthand margin. The title text is indented from the righthand margin by an amount given by \@tocrmarg. Note that \@tocrmarg should be greater than \@pnumwidth. Some entries are typeset with a dotted leader between the end of the title text and the righthand margin indentation. The distance, in math units² between the dots in the leader is given by the value of \@dotsep. In the standard classes the same values are used for the ToC, LoF and the LoT.

The standard values for these internal commands are:

- \@pnumwidth = 1.55em
- \@tocrmarg = 2.55em
- $\dotsep = 4.5$

The values can be changed by using \renewcommand, in spite of the fact that the first two appear to be lengths.

Dotted leaders are not available for Part and Chapter ToC entries.

```
\nmberline{\langle number\rangle}
```

Each \l@kind macro is responsible for setting the general *indent* from the lefthand margin, and the *numwidth*. The \numberline macro is responsible for typesetting the number flushleft in a box of width *numwidth*. If the number is too long for the box then it will protrude

<sup>&</sup>lt;sup>1</sup>For figures and tables it is the \caption command that populates the lof and lot files.

 $<sup>^2</sup>$ There are 18mu to 1em.

```
Parts of a toc file:
\contentsline{section}{\numberline{10.1}The spread}{77}
\contentsline{section}{\numberline{10.2}Typeblock}{89}
\contentsline{subsection}{\numberline{10.2.1}Color}{77}
\contentsline{chapter}{Index}{226}
                                 Part of a lof file:
\contentsline{figure}{\numberline{8.6}Measuring scales}{56}
\addvspace{10pt}
\addvspace{10pt}
\verb|\contentsline{figure}| {\tt line{10.1}Two subfigures}{\{62\}}|
\contentsline{subfigure}{\numberline{(a)}Subfigure 1}{62}
\contentsline{subfigure}{\numberline{(b)}Subfigure 2}{62}
                                 Part of a lot file:
\contentsline{table}{\numberline{1.7}Font declarations}{11}
\contentsline{table}{\numberline{1.8}Font sizes}{13}
\addvspace
\contentsline{table}{\numberline{3.1}Division levels}{21}
```

Figure 9.1: Example extracts from toc, lof and lot files

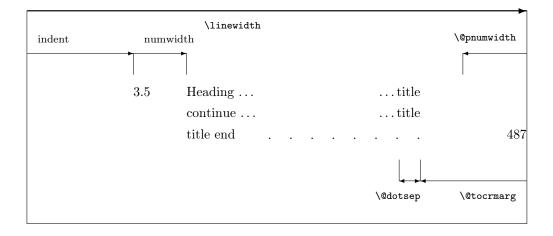


Figure 9.2: Layout of a ToC (LoF, LoT) entry

Entry	Level	Sta	andard	mem	oir class
		indent	${\bf numwidth}$	indent	$\operatorname{numwidth}$
book	-2	_	_	0	_
part	-1	0		0	1.5
chapter	0	0	1.5	0	1.5
section	1	1.5	2.3	1.5	2.3
subsection	2	3.8	3.2	3.8	3.2
subsubsection	3	7.0	4.1	7.0	4.1
paragraph	4	10.0	5.0	10.0	5.0
subparagraph	5	12.0	6.0	12.0	6.0
figure/table	(1)	1.5	2.3	0	1.5
subfigure/table	(2)	_	_	1.5	2.3

Table 9.1: Indents and Numwidths (in ems)

into the title text. The title text is indented by (*indent* + *numwidth*) from the lefthand margin. That is, the title text is typeset in a block of width (\linewidth - indent - numwidth - \Otocrmarg).

Table 9.1 lists the standard values for the *indent* and *numwidth*. There is no explicit *numwidth* for a part; instead a gap of 1em is put between the number and the title text. Note that for a sectioning command the values depend on whether or not the document class provides the \chapter command; the listed values are for the book and report classes — in the article class a \section is treated like a \chapter, and so on. Also, which somewhat surprises me, the table and figure entries are all indented.

```
\verb|\dottedtocline|{||level||} + ||level|| + ||level||
```

Most of the \l@kind commands are defined in terms of the \@dottedtocline command. This command takes three arguments: the  $\langle level \rangle$  argument is the level as shown in Table 9.1, and  $\langle indent \rangle$  and  $\langle numwidth \rangle$  are the *indent* and *numwidth* as illustrated in Figure 9.2. For example, one definition of the \l@section command is:

 $\label{loss} $$\operatorname{\command}{\comman$ 

If it is necessary to change the default typesetting of the entries, then it is usually necessary to change these definitions, but memoir gives you handles to easily alter things without having to know the LaTeX internals.

You can use the  $\addcontentsline$  command to add  $\contentsline$  commands to a file.

```
\addtocontents{\langle file \rangle}{\langle text \rangle}
```

LaTeX also provides the \addtocontents command that will insert  $\langle text \rangle$  into  $\langle file \rangle$ . You can use this for adding extra text and/or macros into the file, for processing when the file is typeset by \tableofcontents (or whatever other command is used for  $\langle file \rangle$  processing, such as \listoftables for a lot file).

As \addcontentsline and \addtocontents write their arguments to a file, any fragile commands used in their arguments must be \protected.

You can make certain adjustments to the ToC, etc., layout by modifying some of the above macros. Some examples are:

If your page numbers stick out into the righthand margin

```
\renewcommand{\@pnumwidth}{3em}
\renewcommand{\@tocrmarg}{4em}
```

but using lengths appropriate to your document.

• To have the (sectional) titles in the ToC, etc., typeset ragged right with no hyphenation

```
\renewcommand{\@tocrmarg}{2.55em plus1fil}
```

where the value 2.55em can be changed for whatever margin space you want.

• The dots in the leaders can be eliminated by increasing \@dotsep to a large value:

```
\renewcommand{\@dotsep}{10000}
```

• To have dotted leaders in your ToC and LoF but not in your LoT:

```
...
\tableofcontents
\makeatletter \renewcommand{\@dotsep}{10000} \makeatother
\listoftables
\makeatletter \renewcommand{\@dotsep}{4.5} \makeatother
\listoffigures
...
```

• To add a horizontal line across the whole width of the ToC below an entry for a Part:

```
\part{Part title}
\addtocontents{toc}{\protect\mbox{}\protect\hrulefill\par}
```

As said earlier any fragile commands in the arguments to \addtocontents and \addcontentsline must be protected by preceding each fragile command with \protect. The result of the example above would be the following two lines in the .toc file (assuming that it is the second Part and is on page 34):

If the \protects were not used, then the second line would instead be:

```
\unhbox \voidb@x \hbox {}\unhbox \voidb@x \leaders \hrule \hfill
    \kern \z@ \par
```

which would cause LaTeX to stop and complain because of the commands that included the @ (see §E.4). If you are modifying any command that includes an @ sign then this must be done in either a .sty file or if in the document itself it must be surrounded by  $\mbox{makeatletter}$  and  $\mbox{makeatletter}$ . For example, if you want to modify  $\mbox{@dotsep}$  in the preamble to your document you have to do it like this:

```
\makeatletter
\renewcommand{\@dotsep}{9.0}
\makeatother
```

• To change the level of entries printed in the ToC (for example when normally subsections are listed in the ToC but for appendices only the main title is required)

```
\appendix
\addtocontents{toc}{\protect\setcounter{tocdepth}{0}}
\chapter{First appendix}
```

#### 9.2 The class ToC methods

The class provides various means of changing the look of the ToC, etc., without having to go through some of the above.

```
\tableofcontents \tableofcontents*
\listoffigures \listoffigures*
\listoftables \listoftables*
```

The ToC, LoF, and LoT are printed at the point in the document where these commands are called, as per normal LaTeX. You can use \tableofcontents, \listoffigures, etc., more than once in a memoir class document.

However, there are two differences between the standard LaTeX behaviour and the behaviour with this class. In the standard LaTeX classes that have \chapter headings, the ToC, LoF and LoT each appear on a new page. With this class they do not necessarily start new pages; if you want them to be on new pages you may have to specifically issue an appropriate command beforehand. For example:

```
\clearpage
\tableofcontents
\clearpage
\listoftables
```

Also, the unstarred versions of the commands put their headings into the ToC, while the starred versions do not.

```
\begin{KeepFromToc} \listof... \end{KeepFromToc}
```

There is at least one package that uses \tableofcontents for its own 'List of...'. When used with the class this will put the package's 'List of...' title into the ToC, and the package doesn't seem to know about \tableofcontents\*. The heading of any \listof... command that is in the KeepFromToc environment will not be added to the ToC. For example:

```
\begin{KeepFromToc}
\listoffigures
\end{KeepFromToc}

is equivalent to \listoffigures*.

\onecoltocetc
\twocoltocetc
\doccoltocetc
```

Table 9.2: Values for X in macros for styling the titles of 'List of...'

```
toc lof lot ...
```

In the standard classes the ToC, etc., are set in one column even if the document as a whole is set in two columns. This limitation is removed. Following the \one coltocetc declaration, which is the default, the ToC and friends will be set in one column but after the \twocoltocetc declaration they will be set in two columns. Following the \doccoltocetc declaration they will be set in either one or two columns to match the document class one column or twocolumn option.

The class \maxtocdepth command sets the tocdepth counter. It is currently not used in the memoir class.

The memoir class command \settocdepth is somewhat analogous to the \setsecnumdepth command described in §6.3. It sets the value of the tocdepth counter and puts it into the ToC to (temporarily) modify what will appear. The \settocdepth and \maxtocdepth macros are from the tocvsec2 package [Wil99b].

```
\phantomsection
```

NOTE: The hyperref package [Rahtz02] appears to dislike authors using \addcontentsline. To get it to work properly with hyperref you normally have to put \phantomsection (a macro defined within this class and the hyperref package) immediately before \addcontentsline.

## 9.2.1 Changing the titles

Commands are provided for controlling the appearance of the ToC, LoF and LoT titles.

```
\contentsname \listfigurename \listtablename
```

Following LaTeX custom, the title texts are the values of the \contentsname, \listfigurename and \listtablename commands.

The commands for controlling the typesetting of the ToC, LoF and LoT titles all follow a similar pattern, so for convenience (certainly mine, and hopefully yours) in the following descriptions I will use X, as listed in Table 9.2, to stand for the file extension for the appropriate 'List of ···'. That is, any of the following:

- tocor
- lof or
- lot.

For example, \Xmark stands for \tocmark or \lofmark or \lotmark.

The code for typesetting the ToC title looks like:

```
\tocheadstart
\printtoctitle{\contentsname}
\tocmark
\thispagestyle{chapter}
\aftertoctitle
```

where the macros are described below.

```
\Xheadstart
```

This macro is called before the title is actually printed. Its default definition is

\newcommand{\Xheadstart}{\chapterheadstart}

```
\printXtitle{\langle title \rangle}
```

The title is typeset via \printXtitle, which defaults to using \printchaptertitle for the actual typesetting.

```
\Xmark
```

These macros sets the marks for use by the running heads on the ToC, LoF, and LoT pages. The default definition is equivalent to:

```
\newcommand{\Xmark}{\markboth{\...name}{\...name}}
```

where \...name is \contentsname or \listfigurename or \listtablename as appropriate. You probably don't need to change these, and in any case they may well be changed by the particular \pagestyle in use.

```
\afterXtitle
```

This macro is called after the title is typeset and by default it is defined to be \afterchaptertitle.

Essentially, the ToC, LoF and LoT titles use the same format as the chapter titles, and will be typeset according to the current chapterstyle. You can modify their appearance by either using a different chapterstyle for them than for the actual chapters, or by changing some of the macros. As examples:

Doing

```
\renewcommand{\printXtitle}[1]{\hfill\Large\itshape #1}
```

will print the title right justified in a Large italic font.

• For a Large bold centered title you can do

```
\renewcommand{\printXtitle}[1]{\centering\Large\bfseries #1}
```

Writing

will result in the first page of the listing using the *empty* pagestyle instead of the default *chapter* pagestyle.

Doing

```
\renewcommand{\afterXtitle}{%
  \par\nobreak \mbox{}\hfill{\normalfont Page}\par\nobreak}
```

will put the word 'Page' flushright on the line following the title.

## 9.2.2 Typesetting the entries

Commands are also provided to enable finer control over the typesetting of the different kinds of entries. The parameters defining the default layout of the entries are illustrated as part of the layouts package [Wil03a] or in [MG $^+$ 04, p. 51], and are repeated in Figure 9.2.

Most of the commands in this section start as \cft..., where cft is intended as a mnemonic for *Table of Contents*, *List of Figures*, *List of Tables*.

```
\cftdot
```

In the default ToC typesetting only the more minor entries have dotted leader lines between the sectioning title and the page number. The class provides for general leaders for all entries. The 'dot' in a leader is given by the value of \cftdot. Its default definition is \newcommand{\cftdot}{.} which gives the default dotted leader. By changing \cftdot you can use symbols other than a period in the leader. For example

\renewcommand{\cftdot}{\ensuremath{\ast}}

will result in a dotted leader using asterisks as the symbol.

```
\cftdotsep
\cftnodots
```

Each kind of entry can control the separation between the dots in its leader (see below). For consistency though, all dotted leaders should use the same spacing. The macro \cftdotsep specifies the default spacing. However, if the separation is too large then no dots will be actually typeset. The macro \cftnodots is a separation value that is 'too large'.

The page numbers are typeset in a fixed width box. The command \setpnumwidth can be used to change the width of the box (LaTeX 's internal \@pnumwidth). The title texts will end before reaching the righthand margin. \setrmarg can be used to set this distance (LaTeX 's internal \@tocrmarg). Note that the length used in \setrmarg should be greater than the length set in \setpnumwidth. These values should remain constant in any given document.

This manual requires more space for the page numbers than the default, so the following was set in the preamble:

```
\setpnumwidth{2.55em}
\setrmarg{3.55em}
```

```
\cftparskip
```

Normally the \parskip in the ToC, etc., is zero. This may be changed by changing the length \cftparskip. Note that the current value of \cftparskip is used for the ToC, LoF and LoT, but you can change the value before calling \tableofcontents or \listoffigures or \listoffables if one or other of these should have different values (which is not a good idea).

Again for convenience, in the following I will use K to stand for the *kind* of entry, as listed in Table 9.3; that is, any of the following:

- book for \book titles.
- part for \part titles

Table 9.3: Value of K in macros for styling entries in a 'List of...'

K	Kind of entry	K	Kind of entry
book	\book title	subparagraph	\subparagraph title
part	\part title	figure	figure caption
chapter	\chapter title	subfigure	subfigure caption
section	\section title	table	table caption
subsection	\subsection title	subtable	subtable caption
subsubsection	\subsubsection title		

- chapter for \chapter titles
- section for \section titles
- subsection for \subsection titles
- subsubsection for \subsubsection titles
- paragraph for \paragraph titles
- subparagraph for \subparagraph titles
- figure for figure \caption titles
- subfigure for subfigure \caption titles
- table for table \caption titles
- subtable for subtable \caption titles

\cftbookbreak \cftpartbreak \cftchapterbreak

When \look starts to typeset a \book entry in the ToC the first thing it does is to call the macro \cftbookbreak. This is defined as:

which encourages a page break before rather than after the entry. As usual, you can change \cftbookbreak to do other things that you feel might be useful. The macros \cftpartbreak and \cftchapterbreak apply to \part and \chapter entries, respectively, and have the same default definitions as \cftbookbreak.

```
\cftbeforeKskip
```

This length controls the vertical space before an entry. It can be changed by using \setlength.

```
\cftKindent
```

This length controls the indentation of an entry from the left margin (*indent* in Figure 9.2). It can be changed using \setlength.

\cftKnumwidth

This length controls the space allowed for typesetting title numbers (*numwidth* in Figure 9.2). It can be changed using \setlength. Second and subsequent lines of a multiline title will be indented by this amount.

The remaining commands are related to the specifics of typesetting an entry. This is a simplified pseudo-code version for the typesetting of numbered and unnumbered entries.

{\cftKfont TITLE}{\cftKleader}{\cftKformatpnum{PAGE}}\cftKafterpnum\par

where SNUM is the section number, TITLE is the title text and PAGE is the page number. In the numbered entry the pseudo-code

{\cftKpresnum SNUM\cftKaftersnum\hfil}

is typeset within a box of width \cftKnumwidth, see the \...numberlinebox macros later on.

```
\cftKfont
```

This controls the appearance of the title (and its preceding number, if any). It may be changed using \renewcommand.

\cftKfont takes no arguments as such, but the number and title is presented to it as an argument. Thus one may end \cftKfont with a macro taking one argument, say \MakeUppercase, and which then readjust the text as needed.

*Caveat.* Please read the section entitled About upper or lower casing TOC entries on page 156 if you consider using upper/lower cased TOC entries and especially if you are also using the hyperref package.

```
\cftKname
```

The first element typeset in an entry is \cftKname.<sup>3</sup> Its default definition is

\newcommand\*{\cftKname}{}

so it does nothing. However, to put the word 'Chapter' before each chapter number in a ToC and 'Fig.' before each figure number in a LoF do:

```
\renewcommand*{\cftchaptername}{Chapter\space}
\renewcommand*{\cftfigurename}{Fig.\space}
```

```
\cftKpresnum \cftKaftersnum \cftKaftersnumb
```

The section number is typeset within a box of width \cftKnumwidth. Within the box the macro \cftKpresnum is first called, then the number is typeset, and the \cftKaftersnum macro is called after the number is typeset. The last command within the box is \hfil to make the box contents flushleft. After the box is typeset the \cftKaftersnumb macro is called before typesetting the title text. All three of these can be changed by \renewcommand. By default they are defined to do nothing.

<sup>&</sup>lt;sup>3</sup>Suggested by Danie Els.

```
\label{line} $$ \operatorname{num} = {\langle num \rangle} $$ \operatorname{partnumberline} {\langle num \rangle} $$ \operatorname{chapternumberline} {\langle num \rangle} $$
```

In the ToC, the macros \booknumberline, \partnumberline and \chapternumberline are responsible respectively for typesetting the \book, \part and \chapter numbers, whereas \numberline does the same for the sectional siblings. Internally they use \cftKpresnum, \cftKaftersnum and \cftKaftersnumb as above. If you do not want, say, the \chapter number to appear you can do:

\renewcommand{\chapternumberline}[1]{}

```
\label{eq:linehook} $$ \cftwhatismyname $$ booknumberlinehook{$\langle num\rangle$} $$ partnumberlinehook{$\langle num\rangle$} $$  \chapternumberlinehook{$\langle num\rangle$} $$  \numberlinebox{$\langle length\rangle$} {$\langle code\rangle$} $$  \booknumberlinebox{$\langle length\rangle$} {$\langle code\rangle$} $$  \chapternumberlinebox{$\langle length\rangle$} {$\langle code\rangle$} $$  \chapternumberlinebox{$\langle length\rangle$} {$\langle code\rangle$} $$
```

Inside the four \...numberline macros, the first thing we do is to give the \...numberline argument to a hook. By default this hook does nothing. But, with the right tools, they can be used to record the widths of the sectional number. Which then can be used to automatically adjust the various  $\langle numwidth \rangle$  and  $\langle indent \rangle$  within the \cftsetindents macro. In order to tell the section types apart (they all use \numberline), the value of the \cftwhatismyname macro will locally reflect the current type.

As mentioned earlier, the \book, \part and \chapter numbers are typeset inside a box of certain fixed widths. Sometimes it can be handy *not* having this box around. For this you can redefine one of the four \...numberlinebox macros listed above. For example via

```
\renewcommand\chapternumberlinebox[2]{#2}
```

The first argument is the width of the box to be made. All four macros are defined similar to this (where #1 is a length)

```
\newcommand\chapternumberlinebox[2]{%
  \hb@xt@#1{#2\hfil}}
```

```
\cftKleader
\cftKdotsep
```

\cftKleader defines the leader between the title and the page number; it can be changed by \renewcommand. The spacing between any dots in the leader is controlled by \cftKdotsep (\@dotsep in Figure 9.2). It can be changed by \renewcommand and its value must be either a number (e.g., 6.6 or \cftdotsep) or \cftnodots (to disable the dots). The spacing is in terms of math units where there are 18mu to 1em.

The default leaders macro is similar to

<sup>&</sup>lt;sup>4</sup>Which we do not currently supply..., but have a look at Sniplet C.6 on page 399.

Note that the spacing of the dots is affected by the font size (as the math unit is affected by the font size). Also note that the \cftchapterleader is bold by default.

The macro \cftKformatpnum typesets an entry's page number, using the \cftKpagefont.<sup>5</sup> The default definition is essentially:

```
\newcommand*{\cftKformatpnum}[1]{%
  \cftKformatpnumhook{#1}%
  \box to \@pnumwidth{\hfil{\cftKpagefont #1}}}
```

which sets the number right justified in a box \@pnumwidth wide. To have, say, a \part page number left justified in its box, do:

```
\renewcommand*{\cftpartformatpnum}[1]{%
  \cftpartformatpnumhook{#1}%
  \box to \@pnumwidth{{\cftpartpagefont #1}}}
```

The \cftKformatpnumhook does nothing by default (other than eating the argument), but could be redefined to record the widest page number and report it back, even reusing it to auto adjust on the next run to set \@pnumwidth (see \setpnumwidth).

```
\cftKafterpnum
```

This macro is called after the page number has been typeset. Its default is to do nothing. It can be changed by \renewcommand.

```
\verb|\cftsetindents{|\langle kind \rangle|}{|\langle indent \rangle|}{|\langle numwidth \rangle|}
```

The command  $\cftsetindents$  sets the  $\langle kind \rangle$  entries indent to the length  $\langle indent \rangle$  and its numwidth to the length  $\langle numwidth \rangle$ . The  $\langle kind \rangle$  argument is the name of one of the standard entries (e.g., subsection) or the name of entry that has been defined within the document. For example

```
\cftsetindents{figure}{0em}{1.5em}
```

will make figure entries left justified.

This manual requires more space for section numbers in the ToC than the default (which allows for three digits). Consequently the preamble contains the following:

```
\cftsetindents{section}{1.5em}{3.0em} \cftsetindents{subsection}{4.5em}{3.9em} \cftsetindents{subsubsection}{8.4em}{4.8em} \cftsetindents{paragraph}{10.7em}{5.7em} \cftsetindents{subparagraph}{12.7em}{6.7em}
```

Note that changing the indents at one level implies that any lower level indents should be changed as well.

 $<sup>^5{\</sup>rm This}$  addition to the class was suggested by Dan Luecking, ctt Re: setting numbers in toc in their natural width box, 2007/08/15.

Various effects can be achieved by changing the definitions of \cftKfont, \cftKaftersnum, \cftKaftersnumb, \cftKleader and \cftKafterpnum, either singly or in combination. For the sake of some examples, assume that we have the following initial definitions

```
\newcommand*{\cftKfont}{}
\newcommand*{\cftKaftersnum}{}
\newcommand*{\cftKaftersnumb}{}
\newcommand*{\cftKleader}{\cftdotfill{\cftKdotsep}}
\newcommand*{\cftKdotsep}{\cftdotsep}
\newcommand*{\cftKpagefont}{}
\newcommand*{\cftKafterpnum}{}
```

Note that the same font should be used for the title, leader and page number to provide a coherent appearance.

• To eliminate the dots in the leader:

```
\renewcommand*{\cftKdotsep}{\cftnodots}
```

• To put something (e.g., a name) before the title (number):

```
\renewcommand*{\cftKname}{SOMETHING }
```

• To add a colon after the section number:

```
\renewcommand*{\cftKaftersnum}{:}
```

• To put something before the title number, add a double colon after the title number, set everything in bold font, and start the title text on the following line:

```
\renewcommand*{\cftKfont}{\bfseries}
\renewcommand*{\cftKleader}{\bfseries\cftdotfill{\cftKdotsep}}
\renewcommand*{\cftKpagefont}{\bfseries}
\renewcommand*{\cftKname}{SOMETHING}
\renewcommand{\cftKaftersnum}{::}
\renewcommand{\cftKaftersnumb}{\\}
```

If you are adding text in the number box in addition to the number, then you will probably have to increase the width of the box so that multiline titles have a neat vertical alignment; changing box widths usually implies that the indents will require modification as well. One possible method of adjusting the box width for the above example is:

• To set the chapter number and title as just 'NUM · TITLE', i.e. un-boxed number plus a symbolic separator, use

```
\renewcommand\cftchapteraftersnumb{\enspace\textperiodcentered\enspace}
\renewcommand\chapternumberlinebox[2]{#2}
```

– of couse, it works best, only if the TITLE is a single line.

Make chapter titles lower case small caps

\renewcommand\cftchapterfont{\scshape\MakeTextLowercase}

- here we do not touch the case of any math.
- To set the section numbers flushright:

```
\setlength{\mylen}{0.5em}  % extra space at end of number
\renewcommand{\cftKpresnum}{\hfill} % note the double '1'
\renewcommand{\cftKaftersnum}{\hspace*{\mylen}}
\addtolength{\cftKnumwidth}{\mylen}
```

In the above, the added initial \hfill in the box overrides the final \hfil in the box, thus shifting everything to the right hand end of the box. The extra space is so that the number is not typeset immediately at the left of the title text.

• To set the entry ragged left (but this only looks good for single line titles):

```
\renewcommand{\cftKfont}{\hfill\bfseries}
\renewcommand{\cftKleader}{}
```

• To set the titles ragged right instead of the usual flushright. Assuming that there are more than 100 pages in the document (otherwise adjust the length):

```
\setrmarg{3.55em plus 1fil}
```

where the last four characters before the closing brace are: digit 1, lowercase F, lowercase I, and lowercase L.

• To set the page number immediately after the entry text instead of at the righthand margin:

```
\renewcommand{\cftKleader}{}
\renewcommand{\cftKafterpnum}{\cftparfillskip}
```

```
\cftparfillskip
```

By default the \parfillskip value is locally set to fill up the last line of a paragraph. Just changing \cftKleader as in the above example puts horrible interword spaces into the last line of the title. The \cftparfillskip command is provided just so that the above effect can be achieved.

```
\cftpagenumbersoff{$\langle kind \rangle$} \\ cftpagenumberson{$\langle kind \rangle$}
```

The command \cftpagenumbersoff will eliminate the page numbers for  $\langle kind \rangle$  entries in the listing, where  $\langle kind \rangle$  is the name of one of the standard kinds of entries (e.g., subsection, or figure) or the name of a new entry defined in the document.

The command  $\cftpagenumberson$  reverses the effect of a corresponding  $\cftpagenumbersoff$  for  $\langle kind \rangle$ .

For example, to eliminate page numbers for appendices in the ToC:

```
...
\appendix
\addtocontents{toc}{\cftpagenumbersoff{chapter}}
\chapter{First appendix}
```

If there are other chapter type headings to go into the ToC after the appendices (perhaps a bibliography or an index), then it will be necessary to do a similar

```
\addtocontents{toc}{\cftpagenumberson{chapter}}
```

after the appendices to restore the page numbering in the ToC.

Sometimes it may be desirable to make a change to the global parameters for an individual entry. For example, a figure might be placed on the end paper of a book (the inside of the front or back cover), and this needs to be placed in a LoF with the page number set as, say, 'inside front cover'. If 'inside front cover' is typeset as an ordinary page number it will stick out into the margin. Therefore, the parameters for this particular entry need to be changed.

```
\label{localchange} $$ \cftlocalchange{\langle ext \rangle} = \langle pnumwidth \rangle = \langle tocrmarg \rangle $$
```

```
\label{line} $$ \operatorname{cftaddtitleline}_{\langle ext\rangle}_{\langle kind\rangle}_{\langle title\rangle}_{\langle page\rangle}_{\langle title\rangle}_{\langle title\rangle}_
```

The command  $\cftaddtitleline$  will write a  $\cftaddtitleline$  entry into  $\cdot entry$  for a  $\cdot kind$  entry with title  $\cdot title$  and page number  $\cdot page$ . Any fragile commands used in the arguments must be protected. That is, an entry is made of the form:

```
\contentsline{kind}{title}{page}
```

The command  $\cftaddnumtitleline$  is similar to  $\cftaddtitleline$  except that it also includes (num) as the argument to  $\cftaddtitleline$ . That is, an entry is made of the form

```
\contentsline{kind}{\numberline{num} title}{page}
```

As an example of the use of these commands, noting that the default LaTeX values for \@pnumwidth and \@tocrmarg are 1.55em and 2.55em respectively, one might do the following for a figure on the frontispiece page.

```
this is the frontispiece page with no number draw or import the picture (with no \caption) \cftlocalchange{lof}{4em}{5em} % make pnumwidth big enough for % frontispiece and change margin \cftaddtitleline{lof}{figure}{The title}{frontispiece} \cftlocalchange{lof}{1.55em}{2.55em} % return to normal settings \clearpage ...
```

Recall that a \caption command will put an entry in the lof file, which is not wanted here. If a caption is required, then you can either craft one youself or, assuming that your general captions are not too exotic, use the \legend command (see later). If the illustration is numbered, use \cftaddnumtitleline instead of \cftaddtitleline.

Inserting stuff into the content lists

The next functions were suggested by Lars Madsen who found them useful if, for example, you had two versions of the ToC and you needed some aspects to be formatted differently.

```
\label{localization} $$ \cftinsertcode{\langle name \rangle} {\langle code \rangle} $$ \cftinserthook{\langle file \rangle} {\langle name \rangle} $$
```

The \cftinserthook is somewhat like \addtocontents in that it enables you to insert a code hook into the ToC, etc., where  $\langle file \rangle$  is the (toc, lof,  $\cdots$ ) file and  $\langle name \rangle$  is the 'name' of the hook. The  $\langle code \rangle$  for the hook is specified via \cftinsertcode where  $\langle name \rangle$  is the name you give to the hook. These can be used to make alterations to a 'List of...' on the fly. For example:

```
\cftinsertcode{A}{%
  \renewcommand*{\cftchapterfont}{\normalfont\scshape}
  ... }% code for ToC
\frontmatter
\tableofcontents
\cftinsertcode{G}{...}\% code for LoF
\cftinsertcode{F}{...}% code for LoF
\listoffigures
\cftinserthook{lof}{G}
\chapter{...}
. . .
\mainmatter
\cftinserthook{toc}{A}
\cftinserthook{lof}{F}
\chapter{...}
. . .
```

If you do not use  $\cftinsertcode$  before calling the command to type the 'List of...' that it is intended for then nothing will happen. No harm will come if a matching  $\cftinserthook$  is never used. No harm occurs either if you call  $\cftinserthook$  and there is no prior matching  $\cftinsertcode$ .

One use of these ToC hooks is reusing the ToC data to, say, create chapter ToC's. The code for this is shown in Sniplet C.7 on page 401. In the sniplet we use the following two hooks that are executed right before and right after \chapter, \part, \book, \appendixpage writes to the ToC. By default they do nothing.<sup>6</sup>

```
\mempreaddchaptertotochook
\mempostaddchaptertotochook
\mempreaddparttotochook
\mempostaddparttotochook
\mempreaddbooktotochook
\mempostaddbooktotochook
\mempreaddapppagetotochook
\mempostaddapppagetotochook
```

<sup>&</sup>lt;sup>6</sup>More hooks may be added in later releases.

Extra chapter material in the ToC

```
\precistoctext{\langle text \rangle} \precistocfont \precistocformat
```

The \chapterprecistoc macro puts \precistoctext{ $\langle text \rangle$ } into the toc file. Further information as to the definition of this macro can be found in section 6.5.3.

About upper or lower casing TOC entries

Some designs call for upper (or lower casing) TOC entries. This *is* possible but the solution depends on whether the hyperref package is used or not.

Without hyperref one can simply end the CftKfont with say MakeTextUppercase and the K-type entry will be upper cased.

With hyperref the possibilities are limited. Explanation: The upper/lower casing macros are not that robust, and need the content to be simple. When hyperref is used, the hyperlink is wrapped around the entry before \cftKfont gains access to it, and is thus generally too complicated for, say, \MakeTextUppercase to handle. The follow workaround draw inspiration from http://tex.stackexchange.com/q/11892/3929.

```
\settocpreprocessor\{\langle type \rangle\} \{\langle code \rangle\}
```

Here  $\langle type \rangle$  is one of chapter, part or book.<sup>8</sup> And  $\langle code \rangle$  can be something like this example:

```
\makeatletter
\settocpreprocessor{chapter}{%
  \let\tempf@rtoc\f@rtoc%
  \def\f@rtoc{%
   \texorpdfstring{\MakeTextUppercase{\tempf@rtoc}}{\tempf@rtoc}}%
}
\makeatother
```

Where \f@rtoc is a placeholder inside \chapter, \part and \book, holding the material to be written to the actual TOC file before hyperref accesses it. This way the upper casing is sneaked into the TOC file, and the bookmark part of hyperref will not complain about the \MakeTextUppercase in the data. Of course, you will not have upper cased titles in the bookmark list.

## 9.2.3 Example: No section number

There are at least two ways of listing section titles in the ToC without displaying their numbers and both involve the \numberline command which typesets the number in a box.

The first method redefines \numberline so it throws away the argument. We do this by modifying the \cftKfont macro which is called before \numberline and the \cftKafterpnum which is called after the page number has been typeset.

```
\let\oldcftsf\cftsectionfont% save definition of \cftsectionfont
\let\oldcftspn\cftsectionafterpnum% and of \cftsectionafterpnum
\renewcommand*{\cftsectionfont}{%
  \let\oldnl\numberline% save definition of \numberline
  \renewcommand*{\numberline}[1]{}% change it
```

<sup>&</sup>lt;sup>7</sup>For some definition of simple.

 $<sup>^{8}</sup>$ If needed we will attempt to add a similar feature to the rest of the sectional types, please write the maintainer.

Probing a little deeper, the \numberline macro is called to typeset section numbers and is defined as:

```
\renewcommand*{\numberline}[1]{%
  \hb@xt@\@tempdima{\@cftbsnum #1\@cftasnum\hfil}\@cftasnumb}
```

Each kind of heading \lets the \@cftbsnum macro to \cftKpresnum, and the \@cftasnum macro to \cftKaftersnum, and the \@cftasnumb macro to \cftKaftersnumb as appropriate for the heading. The second method for killing the number uses a TeX method for defining a macro with a delimited argument.

```
\def\cftsectionpresnum #1\@cftasnum{}
```

The interpretation of this is left as an exercise for anyone who might be interested.

# 9.2.4 Example: Multicolumn entries

If the subsection entries, say, in the ToC are going to be very short it might be worth setting them in multiple columns. Here is one way of doing that which depends on using the multicol package [Mit18]. This assumes that subsections will be the lowest heading in the ToC.

```
\newcounter{toccols}
\setcounter{toccols}{3}
\newenvironment{mysection}[1]{%
  \section{#1}%
  \addtocontents{toc}{\protect\begin{multicols}{\value{toccols}}}}%
  {\addtocontents{toc}{\protect\end{multocols}}}
```

The counter toccols controls the number of columns to be used. For each section where you want subsections to be typeset in multiple columns in the ToC, use the mysection environment instead of \section, like:

```
\begin{mysection}{Columns}
...
\subsection{Fat}
...
\subsection{Thin}
...
\end{mysection}
```

Any ToC entries generated from within the environment will be enclosed in a multicols environment in the ToC. The \protects have to be used because environment \begin and \end commands are fragile.

# 9.2.5 Example: Multiple contents

It is easy to have two ToCs, one short and one long, when they are of the same style, like this:

```
...
\renewcommand*{\contentsname}{Short contents}
\setcounter{tocdepth}{0}% chapters and above
\tableofcontents
% \clearpage
\renewcommand*{\contentsname}{Contents}
\setcounter{tocdepth}{2}% subsections and above
\tableofcontents
```

(Note that you can't use \settocdepth in this case as that writes the change into the ToC, so that the second use would override the first.)

This book has both a short and a long ToC, neither of which look like those typically associated with LaTeX. This is how they were done.

The general style for the ToC, etc., is specified in the memsty package file.

```
%%% need more space for ToC page numbers
\setpnumwidth{2.55em}
\setrmarg{3.55em}
%%% need more space for ToC section numbers
\cftsetindents{section}{1.5em}{3.0em}
\cftsetindents{subsection}{4.5em}{3.9em}
\cftsetindents{subsubsection}{8.4em}{4.8em}
\cftsetindents{paragraph}{10.7em}{5.7em}
\cftsetindents{subparagraph}{12.7em}{6.7em}
%%% need more space for LoF & LoT numbers
\cftsetindents{figure}{0em}{3.0em}
\cftsetindents{table}{0em}{3.0em}
%%% remove the dotted leaders
\renewcommand{\cftsectiondotsep}{\cftnodots}
\renewcommand{\cftsubsectiondotsep}{\cftnodots}
\renewcommand{\cftsubsubsectiondotsep}{\cftnodots}
\renewcommand{\cftparagraphdotsep}{\cftnodots}
\renewcommand{\cftsubparagraphdotsep}{\cftnodots}
\renewcommand{\cftfiguredotsep}{\cftnodots}
\renewcommand{\cfttabledotsep}{\cftnodots}
```

Three macros are defined to control the appearance of the short and the long ToC. First, the macro \setupshorttoc for the short version. The first few lines ensure that only chapter or part titles will be set, and any chapter precis text or tocdepth changes will be ignored. The rest of the code specifies how the chapter titles are to be typeset, and finally the part and book titles.

```
\newcommand*{\setupshorttoc}{%
  \renewcommand*{\contentsname}{Short contents}
  \let\oldchangetocdepth\changetocdepth
  \renewcommand*{\changetocdepth}[1]{}
  \let\oldprecistoctext\precistoctext
  \renewcommand{\precistoctext}[1]{}
  \let\oldcftchapterfillnum\cftchapterfillnum
```

```
\setcounter{tocdepth}{0}% chapters and above
  \renewcommand*{\cftchapterfont}{\hfill\sffamily}
  \renewcommand*{\cftchapterleader}{ \textperiodcentered\space}
  \renewcommand*{\cftchapterafterpnum}{\cftparfillskip}
%% \setpnumwidth{0em}
%% \setpnumwidth{1.5em}
  \renewcommand*{\cftchapterfillnum}[1]{%
    {\cftchapterleader}\nobreak
    \hbox to 1.5em{\cftchapterpagefont ##1\hfil}\cftchapterafterpnum\par}
  \setrmarg{0.3\textwidth}
  \setlength{\unitlength}{\@tocrmarg}
  \addtolength{\unitlength}{1.5em}
  \let\oldcftpartformatpnum\cftpartformatpnum
  \renewcommand*{\cftpartformatpnum}[1]{%
    \hbox to\unitlength{{\cftpartpagefont ##1}}}}
  \let\oldcftbookformatpnum\cftbookformatpnum
  \renewcommand*{\cftbookformatpnum}[1]{%
    \hbox to\unitlength{{\cftbookpagefont ##1}}}}
```

You can do many things using the \cft... macros to change the appearance of a ToC but they can't be entirely coerced into specifying the paragraphing of the \subsection titles. The \setupparasubsecs also went in the preamble.

```
\newcommand*{\setupparasubsecs}{%
 \let\oldnumberline\numberline
 \renewcommand*{\cftsubsectionfont}{\itshape}
 \renewcommand*{\cftsubsectionpagefont}{\itshape}
 \renewcommand{\l@subsection}[2]{%
   \def\numberline###1{\textit{####1}~}%
   \leftskip=\cftsubsectionindent
   \rightskip=\@tocrmarg
%% \advance\rightskip Opt plus \hsize % uncomment this for raggedright
\parfillskip=\fill
   \ifhmode ,\ \else\noindent\fi
   \ignorespaces
   {\cftsubsectionfont ##1}~{\cftsubsectionpagefont##2}%
   \let\numberline\oldnumberline\ignorespaces}
}
\AtEndDocument{\addtocontents{toc}{\par}
```

The above code changes the appearance of subsection titles in the ToC, setting each group as a single paragraph (each is normally set with a paragraph to itself). By uncommenting or commenting the noted lines in the code you can change the layout a little.

Caveat. We have an interesting caveat regarding \setupparasubsecs if you are using hyperref and you have subsubsections, that are not shown in the ToC. You may see some inline subsection entries showing up as '··· text 15, ···', that is a strange space appears before the comma.

This is an artifact due to the way hyperref wraps itself around the ToC entries, even the ones that are not typeset, and thus an end of line space survives. We fix it using \endlinechar:

```
\begingroup
\endlinechar=-1
\tableofcontents
\endgroup
```

Note again that it only happen if you have subsubsections with an inline subsection entry list, and you are using hyperref.

Normally, section titles (and below) are set as individual paragraphs. Effectively the first thing that is done is to end any previous paragraph, and also the last thing is to end the current paragraph. Notice that the main code above neither starts nor finishes a paragraph. If the group of subsections is followed by a section title, that supplies the paragraph end. The last line above ensures that the last entry in the toc file is \par as this might be needed to finish off a group of subsections if these are the last entries.

And thirdly for the main ToC, the macro \setupmaintoc reverts everything back to normal.

```
\newcommand*{\setupmaintoc}{%
 \renewcommand{\contentsname}{Contents}
 \let\changetocdepth\oldchangetocdepth
 \let\precistoctext\oldprecistoctext
 \let\cftchapterfillnum\oldcftchapterfillnum
 \addtodef{\cftchapterbreak}{\par}{}
 \renewcommand*{\cftchapterfont}{\normalfont\sffamily}
 \renewcommand*{\cftchapterleader}{%
               \sffamily\cftdotfill{\cftchapterdotsep}}
 \renewcommand*{\cftchapterafterpnum}{}
 \setpnumwidth{2.55em}
 \setrmarg{3.55em}
 \setcounter{tocdepth}{2}}
 \let\cftpartformatpnum\oldcftpartformatpnum
   \addtodef{\cftpartbreak}{\par}{}
 \let\cftbookformatpnum\oldcftbookformatpnum
   \addtodef{\cftbookbreak}{\par}{}
```

The first few lines restore some macros to their original definitions.

```
\addtodef{\cftchapterbreak}{\par}{}
```

ensures that a chapter entry starts off with a \par; this is needed when the previous entry is a group of subsections and their paragraph has to be ended. The remaining code lines simply set the appearance of the chapter titles and restore that for parts and books, as well as ensuring that they start off new paragraphs.

In the document itself, \tableofcontents was called twice, after the appropriate setups:

```
...
\setupshorttoc
```

```
\tableofcontents
\clearpage
\setupparasubsecs
\setupmaintoc
\tableofcontents
\setlength{\unitlength}{1pt}
```

After all this note that I ensured that \unitlength was set to its default value (it had been used as a scratch length in the code for \setupparasubsecs).

9.3 New 'List of...' and entries

```
\label{listofcom} $$\operatorname{listofcom}}{\langle listofcom\rangle}{\langle listofname\rangle}$
```

The command \newlistof creates a new 'List of...', and assorted commands to go along with it. The first argument,  $\langle listofcom \rangle$  is used to define a new command called \listofcom which can then be used like \listoffigures to typeset the 'List of...'. The  $\langle ext \rangle$  argument is the file extension to be used for the new listing. The last argument,  $\langle listofname \rangle$  is the title for the 'List of...'. Unstarred and starred versions of \listofcom are created. The unstarred version, \listofcom, will add  $\langle listofname \rangle$  to the ToC, while the starred version, \listofcom\*, makes no entry in the ToC.

As an example:

```
\newcommand{\listanswername}{List of Answers}
\newlistof{listofanswers}{ans}{\listanswername}
```

will create a new \listofanswers command that can be used to typeset a listing of answers under the title \listanswername, where the answer titles are in an ans file. It is up to the author of the document to specify the 'answer' code for the answers in the document. For example:

```
\newcounter{answer}[chapter]
\renewcommand{\theanswer}{\arabic{answer}}
\newcommand{\answer}[1]{
   \refstepcounter{answer}
   \par\noindent\textbf{Answer \theanswer. #1}
   \addcontentsline{ans}{answer}{\protect\numberline{\theanswer}#1}\par}
which, when used like:
   \answer{Hard} The \ldots
will print as:
```

```
Answer 1. Hard
The ···
```

As mentioned above, the \newlistof command creates several new commands in addition to \listofcom, most of which you should now be familiar with. For convenience, assume that  $\newlistof{...}{X}{...}$  has been issued so that X is the new file extension and corresponds to the X in §9.2.1. Then in addition to \listofcom the following new commands will be made available.

The four commands, \Xmark, \Xheadstart, \printXtitle, and \afterXtitle, are analagous to the commands of the same names described in §9.2.1 (internally the class uses the \newlistof macro to define the ToC, LoF and LoT). In particular the default definition of \Xmark is equivalent to:

\newcommand{\Xmark}{\markboth{listofname}}{listofname}}

However, this may well be altered by the particular \pagestyle in use.

```
Xdepth
```

The counter Xdepth is analogous to the standard tocdepth counter, in that it specifies that entries in the new listing should not be typeset if their numbering level is greater than Xdepth. The default definition is equivalent to

\setcounter{Xdepth}{1}

Remember that the \chapter command uses \insertchapterspace to insert vertical spaces into the LoF and LoT. If you want similar spaces added to your new listing then you have to modify \insertchapterspace. The easiest way to do this is via the \addtodef macro, like:

```
\addtodef{\insertchapterspace}{}%
{\addtocontents{ans}{\protect\addvspace{10pt}}}
```

The \addtodef macro is described later in §18.10.

The other part of creating a new 'List of...', is to specify the formatting of the entries, i.e., define an appropriate \l@kind macro.

```
\label{lem:listentry} $$ \operatorname{listentry} [\langle within \rangle] {\langle cntr \rangle} {\langle ext \rangle} {\langle level-1 \rangle} $$
```

The command \newlistentry creates the commands necessary for typesetting an entry in a 'List of···'. The first required argument,  $\langle cntr \rangle$  is used to define a new counter called cntr, unless cntr is already defined. The optional  $\langle within \rangle$  argument can be used so that cntr gets reset to one every time the counter called within is changed. That is, the first two arguments when cntr is not already defined, are equivalent to calling \newcounter{ $\langle cntr \rangle}[\langle within \rangle]$ . If cntr is already defined, \newcounter is not called. cntr is used for the number that goes along with the title of the entry.

The second required argument,  $\langle ext \rangle$ , is the file extension for the entry listing. The last argument,  $\langle level-1 \rangle$ , is a number specifying the numbering level minus one, of the entry in a listing.

Calling \newlistentry creates several new commands used to configure the entry. So in order to configure the list look of our previous answer example we would add

```
\newlistentry{answer}{ans}{0}
```

Assuming that  $\newlistentry$  is called as  $\newlistentry[within]{K}{X}{N}$ , where K and X are similar to the previous uses of them (e.g., K is the kind of entry X is the file extension), and N is an integer number, then the following commands are made available.

The set of commands \cftbeforeKskip, \cftKfont, \cftKpresnum, \cftKaftersnum, \cftKaftersnum, \cftKaftersnum, are

analagous to the commands of the same names described in §9.2.2. Their default values are also as described earlier.

The default values of \cftKindent and \cftKnumwidth are set according to the value of the  $\langle level-1 \rangle$  argument (i.e., N in this example). For N=0 the settings correspond to those for figures and tables, as listed in Table 9.1 for the memoir class. For N=1 the settings correspond to subfigures, and so on. For values of N less than zero or greater than four, or for non-default values, use the \cftsetindents command to set the values.

\local{ICK} is an internal command that typesets an entry in the list, and is defined in terms of the above \cft\*K\* commands. It will not typeset an entry if Xdepth is N or less, where X is the listing's file extension.

The command \thek prints the value of the K counter. It is initially defined so that it prints arabic numerals. If the optional  $\langle within \rangle$  argument is used, \thek is defined as

```
\renewcommand{\theK}{\thewithin.\arabic{K}}
```

otherwise as

```
\renewcommand{\theK}{\arabic{K}}}
```

As an example of the independent use of \newlistentry, the following will set up for sub-answers.

```
\newlistentry[answer]{subanswer}{ans}{1}
\renewcommand{\thesubanswer}{\theanswer.\alph{subanswer}}
\newcommand{\subanswer}[1]{
    \refstepcounter{subanswer}
    \par\textbf{\thesubanswer) #1}
    \addcontentsline{ans}{subanswer}{\protect\numberline{\thesubanswer}#1}
\setcounter{ansdepth}{2}
And then:
```

\answer{Harder} The \ldots \subanswer{Reformulate the problem} It assists \ldots

will be typeset as:

```
Answer 2. Harder
The ···

2.a) Reformulate the problem It assists ···
```

By default the answer entries will appear in the List of Answers listing (typeset by the \listofanswers command). In order to get the subanswers to appear, the \setcounter{ansdepth}{2} command was used above.

To turn off page numbering for the subanswers, do

```
\cftpagenumbersoff{subanswer}
```

As another example of \newlistentry, suppose that an extra sectioning division below subparagraph is required, called subsubpara. The \subsubpara command itself can be defined via the LaTeX kernel \@startsection command. Also it is necessary to define a \subsubparamark macro, a new subsubpara counter, a \thesubsubpara macro and a \l@subsubpara macro. Using \newlistentry takes care of most of these as shown below; remember the caveats about commands with @ signs in them (see §E.4).

Each 'List of...' uses a file to store the list entries, and these files must remain open for writing throughout the document processing. TeX has only a limited number of files that it can keep open, and this puts a limit on the number of listings that can be used. For a document that includes a ToC but no other extra ancilliary files (e.g., no index or bibliography output files) the maximum number of LoX's, including a LoF and LoT, is no more than about eleven. If you try and create too many new listings LaTeX will respond with the error message:

```
No room for a new write
```

If you get such a message the only recourse is to redesign your document.

# 9.3.1 Example: plates

As has been mentioned earlier, some illustrations may be tipped in to a book. Often, these are called *plates* if they are on glossy paper and the rest of the book is on ordinary paper. We can define a new kind of Listing for these.

```
\newcommand{\listplatename}{Plates}
\newlistof{listofplates}{lop}{\listplatename}
\newlistentry{plate}{lop}{0}
\cftpagenumbersoff{plate}
```

This code defines the \listofplates command to start the listing which will be titled 'Plates' from the \listplatename macro. The entry name is plate and the file extension is lop. As plate pages typically do not have printed folios, the \cftpagenumbersoff command has been used to prohibit page number printing in the listing.

If pages are tipped in, then they are put between a verso and a recto page. The afterpage package [Car95] lets you specify something that should happen after the current page is finished. The next piece of code uses the package and its \afterpage macro to define two macros which let you specify something that is to be done after the next verso (\afternextverso) or recto (\afternextverso) page has been completed.

```
\newcommand{\afternextverso}[1]{%
  \afterpage{\ifodd\c@page #1\else\afterpage{#1}\fi}}
\newcommand{\afternextrecto}[1]{%
  \afterpage{\ifodd\c@page\afterpage{#1}\else #1\fi}}
```

The  $\partial page fall (labelid)$  command typesets the page number corresponding to the location in the document where  $\partial page fall (labelid)$  is specified. The following code defines

two macros<sup>9</sup> that print the page number before (\priorpageref) or after (\nextpageref) that given by \pageref.

```
\newcounter{mempref}
\newcommand{\priorpageref}[1]{%
  \setcounter{mempref}{\pageref{#1}}\addtocounter{mempref}{-1}\themempref}
\newcommand{\nextpageref}[1]{%
  \setcounter{mempref}{\pageref{#1}}\addtocounter{mempref}{1}\themempref}
```

With these preliminaries out of the way, we can use code like the following for handling a set of physically tipped in plates.

```
\afternextverso{\label{tip}
  \addtocontents{lop}{%
    Between pages \priorpageref{tip} and \pageref{tip}
  \par\vspace*{\baselineskip}}
  \addcontentsline{lop}{plate}{First plate}
  \addcontentsline{lop}{plate}{Second plate}
  ...
  \addcontentsline{lop}{plate}{Nth plate}
}
```

This starts off by waiting until the next recto page is started, which will be the page immediately after the plates, and then inserts the label tip. The \addtocontents macro puts its argument into the plate list lop file, indicating the page numbers before and after the set of plates. With the plates being physically added to the document it is not possible to use \caption, instead the \addcontentsline macros are used to add the plate titles to the lop file.

With a few modifications the code above can also form the basis for listing plates that are electronically tipped in but do not have printed folios or \captions.

# 9.4 Chapter precis

See section 6.5.3 on page 92.

# 9.5 Contents lists and bookmarks

With the hyperref package, the table of contents is often added as a list of bookmarks thus providing a nice navigation for the user. There is one slight problem though: when using, say, parts in the document, all chapters in that part ends up as a child of this part bookmark—including the index and bibliography. A simple fix to this is to add

```
\makeatletter
\renewcommand*{\toclevel@chapter}{-1}
\makeatother
```

just before the material you would like to pull out of the part tree.

A better solution is the bookmark package, add it to the preamble, and add

\bookmarksetup{startatroot}

before the stuff you want to have moved out of, say, a part.

<sup>&</sup>lt;sup>9</sup>These only work for arabic page numbers.

# Ten

# Floats and captions

A float environment is a particular kind of box — one that LaTeX decides where it should go although you can provide hints as to where it should be placed; all other boxes are put at the point where they are defined. Within reason you can put what you like within a float but it is unreasonable, for example, to put a float inside another float. The standard classes provide two kinds of float environments, namely figure and table. The only difference between these is the naming and numbering of any caption within the environments — a \caption in a figure environment uses \figure and tables are numbered sequentially but the two numbering schemes are independent of each other.

The class provides means of defining new kinds of floats. It also provides additional forms of captions for use both within and outside float environments together with handles for changing the style of captions.

# 10.1 New float environments

It is often forgotten that the LaTeX float environments come in both starred and unstarred forms. The unstarred form typesets the float contents in one column, which is the most usual form for a book. The starred form typesets the contents of the float across the top of both columns in a twocolumn document. In a onecolumn document there is no difference between the starred and unstarred forms.

```
\label{lost} $$\operatorname{lost}[\langle within \rangle] {\langle fenv \rangle} {\langle ext \rangle} {\langle capname \rangle}$
```

The \newfloat command creates two new floating environments called  $\langle fenv \rangle$  and  $\langle fenv^* \rangle$ . If there is not already a counter defined for  $\langle fenv \rangle$  a new one will be created to be restarted by the counter  $\langle within \rangle$ , if that is specified. A caption within the environment will be written out to a file with extension  $\langle ext \rangle$ . The caption, if present, will start with  $\langle capname \rangle$ . For example, the figure float for the class is defined as:

```
\newfloat[chapter]{figure}{lof}{\figurename}
\renewcommand{\thefigure}{%
  \ifnum\c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
```

The last bit of the definition is internal code to make sure that if a figure is in the document before chapter numbering starts, then the figure number will not be preceded by a non-existent chapter number.

The captioning style for floats defined with \newfloat is the same as for the figures and tables.

The \newfloat command generates several new commands, some of which are internal LaTeX commands. For convenience, assume that the command was called as

```
\newfloat{F}{X}{capname}
```

so F is the name of the float environment and also the name of the counter for the caption, and X is the file extension. The following float environment and related commands are then created.

```
\begin{F} float material \end{F}
\begin{F*} float material \end{F*}
```

The new float environment is called F, and can be used as either  $\left\{F\right\}$  or  $\left\{F\right\}$ , with the matching  $\left\{F\right\}$  or  $\left\{F\right\}$ . It is given the standard default position specification of  $\left\{f\right\}$ .

```
Xdepth
```

The Xdepth counter is analogous to the standard tocdepth counter in that it specifies that entries in a listing should not be typeset if their numbering level is greater than Xdepth. The default definition is

```
\setcounter{Xdepth}{1}
```

To have a subfloat of X appear in the listing do

```
\setcounter{Xdepth}{2}
```

As an example, suppose you wanted both figures (which come with the class), and diagrams. You could then do something like the following.

```
\newcommand{\diagramname}{Diagram}
\newcommand{\listdiagramname}{List of Diagrams}
\newlistof{listofdiagrams}{dgm}{\listdiagramname}
\newfloat{diagram}{dgm}{{\diagramname}}
\newlistentry{diagram}{dgm}{0}
\begin{document}
...
\listoffigures
\listfofdiagrams
...
\begin{diagram}
\caption{A diagram} \label{diag1}
...
\end{diagram}
As diagram~\ref{diag1} shows ...
\setfloatadjustment{\floatname\}}{\code\}}
```

Often it is useful to add some global configuration to a given type of float such that one will not have to add this to each and every float. For example to have all (floating) figures and tables automatically centered plus have all (floating) tables typeset in \small use

```
\setfloatadjustment{figure}{\centering}
\setfloatadjustment{table}{\small\centering}
```

# 10.1.1 Margin floats

We also provide two environments to insert an image or table into the margin (using \marginpar). The construction is inspired by the Tufte LATEX collection.

```
\begin{marginfigure} [\langle len \rangle] float material \end{marginfigure} \begin{margintable} [\langle len \rangle] float material \end{margintable}
```

Because this is inserted differently than the ordinary figure or table floats, one might get into the situation where a figure float inserted before a margin float, might float *past* the margin float and thus have different caption numbering. For this reason the margin float contain a float blocking device such that any unplaced floats are forced to be placed before we start typesetting a margin figure.

The marginfigure and margintable environments can of course be adjusted using \setfloatadjustment, default

```
\setfloatadjustment{marginfigure}{\centering}
\setfloatadjustment{margintable}{\centering}
```

It may be useful to adjust the captioning separately, for this we have added

```
\star{ginfloat}captionadjustment\{\langle float \rangle\} \{\langle code \rangle\}
```

where  $\langle float \rangle$  is figure or table. The intent is to enable the user to choose a different captioning style (or similar) within a margin float, for example typesetting the caption ragged left/right depending on the page.

This *left/right depending on the page* is a little hard to do, so for the \marginpar (which the margin float use internally) we provide the following two macros

```
\starting{\at left of textblock} {\at right of textblock} } \at right of textblock} \\
```

Basically \mpjustification execute  $\langle at \ left \ of \ textblock \rangle$  when it is executed at the left of the text block and vice versa. For it to work the margin into which the marginpar should do, has to be specified using marginparmargin. The default is

```
\setmpjustification{\raggedleft}{\raggedright}
```

To have both a margin figure and its caption typeset ragged against the text block, use

```
\setfloatadjustment{marginfigure}{\mpjustification}
\setmarginfloatcaptionadjustment{figure}{\captionstyle{\mpjustification}}
```

It may be useful to allow hyphenation within the raggedness, which can be done using the ragged2e package and

```
\verb|\color= {\color= largedLeft} {\color= largedLeft} \\
```

#### 10.2 Setting off a float

Sometimes it is desireable to set off a float, more probably an illustration than a tabular, from its surroundings. The framed environment, described later in Chapter 15, might come in handy for this.

The following code produces the example Figures 10.1 and 10.2.

#### FRAMED FIGURE

Figure 10.1: Example framed figure

# FRAMED FIGURE AND CAPTION

Figure 10.2: Example framed figure and caption

```
\begin{figure}
\centering
\begin{framed}\centering
FRAMED FIGURE
\end{framed}
\caption{Example framed figure}\label{fig:framef}
\end{figure}
\begin{figure}
\begin{figure}
\begin{framed}\centering
FRAMED FIGURE AND CAPTION
\caption{Example framed figure and caption}\label{fig:framefcap}
\end{framed}
\end{figure}
```

If framing seems overkill then you can use rules instead, as in the example code below which produces Figures 10.3 and 10.4.

```
\begin{figure}
\centering
\hrule\vspace{\onelineskip}
RULED FIGURE
\vspace{\onelineskip}\hrule
\vspace{\onelineskip}
\caption{Example ruled figure}\label{fig:rulef}
\end{figure}
\begin{figure}
\centering
\hrule\vspace{\onelineskip}
RULED FIGURE AND CAPTION
\vspace{\onelineskip}\hrule
\vspace{0.2pt}\hrule
\vspace{\onelineskip}
\caption{Example ruled figure and caption}\label{fig:rulefcap}
```

# RULED FIGURE

Figure 10.3: Example ruled figure

# RULED FIGURE AND CAPTION

Figure 10.4: Example ruled figure and caption

# ILLUSTRATION 1

# ILLUSTRATION 2

Figure 10.5: Example float with two illustrations

\hrule
\end{figure}

# 10.3 Multiple floats

You can effectively put what you like inside a float box. Normally there is just a single picture or tabular in a float but you can include as many of these as will fit inside the box.

Three typical cases of multiple figures/tables in a single float come to mind:

- Multiple illustrations/tabulars with a single caption.
- Multiple illustrations/tabulars each individually captioned.
- Multiple illustrations/tabulars with one main caption and individual subcaptions.

Figure 10.5 is an example of multiple illustrations in a single float with a single caption.

The figure was produced by the following code.

```
\begin{figure}
\centering
\hspace*{\fill}
   {ILLUSTRATION 1} \hfill {ILLUSTRATION 2}
\hspace*{\fill}
\caption{Example float with two illustrations} \label{fig:mult1}
\end{figure}
```

# GRAPHIC 1 GRAPHIC 2

Figure 10.6: Graphic 1 in a float

Figure 10.7: Graphic 2 in same float

The \hspace\*{\fill} and \hfill commands were used to space the two illustrations equally. Of course \includegraphics or tabular environments could just as well be used instead of the {ILLUSTRATION N} text.

The following code produces Figures 10.6 and 10.7 which are examples of two separately captioned illustrations in one float.

```
\begin{figure}
\centering
\begin{minipage}{0.4\textwidth}
\centering
GRAPHIC 1
\caption{Graphic 1 in a float} \label{fig:mult2}
\end{minipage}
\hfill
\begin{minipage}{0.4\textwidth}
\centering
GRAPHIC 2
\caption{Graphic 2 in same float} \label{fig:mult3}
\end{minipage}
\end{figure}
```

In this case the illustrations (or graphics or tabulars) are put into separate minipage environments within the float, and the captions are also put within the minipages. Note that any required \label must also be inside the minipage. If you wished, you could add yet another (main) caption after the end of the two minipages.

It is slightly more complex if you want to put, say, both a tabulation captioned as a table and a graph, captioned as a figure, which illustrates the tabulation, as a float only permits one kind of caption. The class solves this problem by letting you define 'fixed' captions which are independent of the particular kind of the float. These are described in detail later.

Things do get a little trickier, though, if the bodies and/or the captions in a float are different heights (as in Figures 10.6 and 10.7) and you want to align them horizontally. Here are some examples.

This code produces Figures 10.8 and 10.9. The new \hhrule macro produces a rule twice as thick as \hrule does.

```
\newcommand*{\hhrule}{\hrule height 0.8pt}% double thickness
\begin{figure}
\hhrule \vspace{\onelineskip}
\null\hfill\parbox{0.45\linewidth}{%
  \centering
```

Aligned to the center of the right figure	This is the right figure which is taller than the first one (the one at the left)
Figure 10.8: Left center aligned	Figure 10.9: Right figure. This has more text than the adjacent caption (10.8) so the heights are unequal

```
Aligned to the center of the right figure
}\hfill
\parbox{0.45\linewidth}{%}
 \centering
  This is the right figure which is taller
  than the first one (the one at the left)
}\hfill\null
\vspace{\onelineskip}\hrule
\caption{Left figure}\label{fig:left1}%
}\hfill
\parbox[t]{0.4}linewidth}{%}
 \caption{Right figure. This has more text than the adjacent
          caption (ref{fig:left1}) so the heights are unequal}%
          \label{fig:right1}%
}\hfill\null
\hhrule
\end{figure}
 The following code produces Figures 10.10 and 10.11.
\begin{figure}
\hhrule \vspace{0.5\onelineskip}
\centering
 Aligned to the top of the right figure
}\hfill
\parbox[t]{0.45}\parbox[t]{%}
 \centering
  This is the right figure which is taller
  than the first one (the one at the left)
}\hfill\null
\vspace{0.5\onelineskip}\hrule
\null\hfill\parbox[t]{0.4\linewidth}{%
 \caption{Left top aligned}\label{fig:left2}%
}\hfill
```

Aligned to the top of the right figure	This is the right figure which is taller than the first one (the one at the left)
Figure 10.10: Left top aligned	Figure 10.11: Right figure. This has more text than the adjacent caption (10.10) so the heights are unequal

```
\parbox[t]{0.4\linewidth}{%
  \caption{Right figure. This has more text than the adjacent
           caption (\ref{fig:left2}) so the heights are unequal}%
           \label{fig:right2}%
}\hfill\null
\hhrule
\end{figure}
 The next code produces Figures 10.12 and 10.13.
\begin{figure}
\hhrule \vspace{0.5\onelineskip}
\\displaystyle \hfill\parbox[b]{0.45}linewidth}{%}
  \centering
  Aligned to the bottom of the right figure
}\hfill
\parbox[b]{0.45}\parbox[k]{%}
  \centering
   This is the right figure which is taller
   than the first one (the one at the left)
}\hfill\null
\vspace{0.5\onelineskip}\hrule
\null\hfill\parbox[t]{0.4\linewidth}{%
  \caption{Left bottom aligned}\label{fig:left3}%
}\hfill
\parbox[t]{0.4\linewidth}{%
  \caption{Right figure. This has more text than the adjacent
           caption (\ref{fig:left3}) so the heights are unequal}%
           \label{fig:right3}%
}\hfill\null
\hhrule
\end{figure}
  \newsubfloat{\langle float\rangle}
```

The \newsubfloat command creates subcaptions (\subcaption, \subtop and \subbottom) for use within the float environment  $\langle fenv \rangle$  previously defined via

Aligned to the bottom of the right figure	This is the right figure which is taller than the first one (the one at the left)
Figure 10.12: Left bottom aligned	Figure 10.13: Right figure. This has more text than the adjacent caption (10.12) so the heights are unequal

 $\mbox{\ensuremath{\mbox{newfloat}} $\{\langle ... \rangle\}$ } {\langle ... \rangle}.$  Subcaptions are discussed below in §10.9.

10.4 Where LaTeX puts floats

The general format for a float environment is:

 $\left[\langle loc \rangle\right]$  ...  $\left[\langle loc \rangle\right]$  ...  $\left[\langle loc \rangle\right]$  ...

 $\ensuremath{\mbox{begin{float*}}[\langle loc \rangle] ... \end{float*}}$ 

where the optional argument  $\langle loc \rangle$ , consisting of one or more characters, specifies a location where the float may be placed. Note that the multicol package only supports the starred floats and it will not let you have a single column float. The possible  $\langle loc \rangle$  values are one or more of the following:

- b *bottom*: at the bottom of a page. This does not apply to double column floats as they may only be placed at the top of a page.
- h *here*: if possible exactly where the float environment is defined. It does not apply to double column floats.
- p page: on a separate page containing only floats (no text); this is called a *float page*.
- t top: at the top of a page.
- ! make an extra effort to place the float at the earliest place specified by the rest of the argument.

The default for  $\langle loc \rangle$  is tbp, so the float may be placed at the top, or bottom, or on a float page; the default works well 95% of the time. Floats of the same kind are output in definition order, except that a double column float may be output before a later single column float of the same kind, or  $vice\text{-}versa^1$ . A float is never put on an earlier page than its definition but may be put on the same or later page of its definition. If a float cannot be placed, all succeeding floats will be held up, and LaTeX can store no more than 16 held up floats. A float cannot be placed if it would cause an overfull page, or it otherwise cannot be fitted according the float placement parameters. A \clearpage or \cleardoublepage or \end{document} flushes out all unprocessed floats, irrespective of the  $\langle loc \rangle$  and float parameters, putting them on float-only pages.

```
\stfloatlocations{\langle float \rangle}{\langle locs \rangle}
```

You can set the location for all floats of type  $\langle float \rangle$  to  $\langle locs \rangle$  with the \setfloatlocations declaration. The class initialises these using:

<sup>&</sup>lt;sup>1</sup>As of 2015 this has been fixed in the LAT<sub>F</sub>X kernel.

\setfloatlocations{figure}{tbp} \setfloatlocations{table}{tbp}

```
\suppressfloats[\langle pos \rangle]
```

You can use the command \suppressfloats to suppress floats at a given  $\langle pos \rangle$  on the current page. \suppressfloats[t] prevents any floats at the top of the page and \suppressfloats[b] prevents any floats at the bottom of the page. The simple \suppressfloats prevents both top and bottom floats.

```
\FloatBlock
\FloatBlockAllowAbove
\FloatBlockAllowBelow
```

Contrary to \suppressfloats \FloatBlock² will block floats from passing this point, i.e. it demands LaTeX to place any unprocessed floats before proceeding. It is similar to \clearpage but it does not necessarily introduce a page break before proceeding.

\FloatBlockAllowAbove lessens the restriction a little, in a situation like this

```
\FloatBlock some float here
```

\FloatBlockAllowAbove will allow the float to be placed at the top of the same page as \FloatBlock. \FloatBlockAllowBelow is the reverse situation.

It may be beneficial to be able to add \FloatBlock to sectional commands. This can be done via

```
\stFloatBlockFor{\langle sectional name \rangle}
```

where  $\langle sectional \ name \rangle$  is withput the  $\backslash$ , i.e.

\setFloatBlockFor{section}

The flafter package, which should have come with your LaTeX distribution, provides a means of preventing floats from moving backwards from their definition position in the text. This can be useful to ensure, for example, that a float early in a \section{...} is not typeset before the section heading.

Figures 10.14 and 10.15 illustrate the many float parameters and Table 10.1 lists the float parameters and the typical standard default values. The lengths controlling the spaces surroundind floats are listed in Table 10.2; typical values are shown as they depend on both the class and the size option.

Given the displayed defaults, the height of a top float must be less than 70% of the textheight and there can be no more than 2 top floats on a text page. Similarly, the height of a bottom float must not exceed 30% of the textheight and there can be no more than 1 bottom float on a text page. There can be no more than 3 floats (top, bottom and here) on the page. At least 20% of a text page with floats must be text. On a float page (one that has no text, only floats) the sum of the heights of the floats must be at least 50% of the textheight. The floats on a float page should be vertically centered.

<sup>&</sup>lt;sup>2</sup>Yes, it is the same as \FloatBarrier from the placeins package, kudos to Donald Arseneau. For various reasons we cannot emulate the placeins package and its options, thus we have verbatimly copied and renamed it instead.

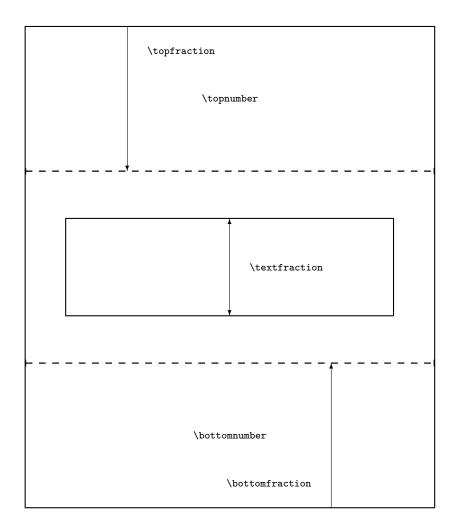


Figure 10.14: Float and text page parameters

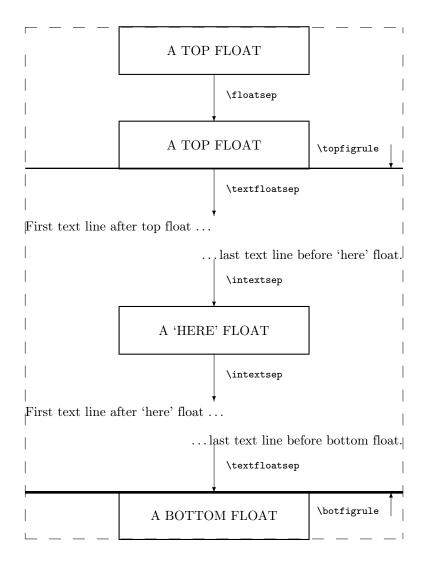


Figure 10.15: Float parameters

Table 10.1: Float placement parameters

Parameter	Controls	Default		
Counters — change with \setcounter				
topnumber	max number of floats at top of a page	2		
bottomnumber	max number of floats at bottom of a page	1		
totalnumber	max number of floats on a text page	3		
dbltopnumber	like topnumber for double column floats	2		
Commands — change with \renewcommand				
\topfraction	max fraction of page reserved for top floats	0.7		
\bottomfraction	max fraction of page reserved for bottom	0.3		
	floats			
\textfraction	min fraction of page that must have text	0.2		
\dbltopfraction	like \topfraction for double column floats	0.7		
-	floats			
\floatpagefraction	min fraction of a float page that must have	0.5		
	float(s)			
\dblfloatpagefraction	like \floatpagefraction for double column	0.5		
	floats			

Table 10.2: Float spacing parameters

Parameter	Controls	Default	
Text page lengths — change with \setlength			
\floatsep	vertical space between floats	12pt	
\textfloatsep	vertical space between a top (bottom) float	$20\mathrm{pt}$	
	and suceeding (preceeding) text		
\intextsep	vertical space above and below an h float	12pt	
\dblfloatsep	like \floatsep for double column floats	12pt	
$\d$	like \textfloatsep for double column floats	20pt	
Float page lengths — change with \setlength			
\@fptop	space at the top of the page	Opt plus 1fil	
\@fpsep	space between floats	8pt plus 2fil	
\@fpbot	space at the bottom of the page	Opt plus 1fil	
\@dblfptop	like \@fptop for double column floats	Opt plus 1fil	
\@dblfpsep	like \@fpsep for double column floats	8pt plus 2fil	
\@dblfpbot	like \@fpbot for double column floats	Opt plus 1fil	

Under certain extreme and unlikely conditions and with the defaults LaTeX might have trouble finding a place for a float. Consider what will happen if a float is specified as a bottom float and its height is 40% of the textheight and this is followed by a float whose height is 90% of the textheight. The first is too large to actually go at the bottom of a text page but too small to go on a float page by itself. The second has to go on a float page but it is too large to share the float page with the first float. LaTeX is stuck!

At this point it is worthwhile to be precise about the effect of a one character  $\langle loc \rangle$  argument:

- b means: 'put the float at the bottom of a page with some text above it, and nowhere else'. The float must fit into the \bottomfraction space otherwise it and subsequent floats will be held up.
- h means: 'put the float at this point and nowhere else'. The float must fit into the space left on the page otherwise it and subsequent floats will be held up.
- p means: 'put the float on a page that has no text but may have other floats on it'. There must be at least '\floatpagefraction' worth of floats to go on a float only page before the float will be output.
- t means: 'put the float at the top of a page with some text below it, and nowhere else'. The float must fit into the \topfraction space otherwise it and subsequent floats will be held up.
- !... means: 'ignore the \...fraction values for this float'.

You must try and pick a combination from these that will let LaTeX find a place to put your floats. However, you can also change the float parameters to make it easier to find places to put floats. Some examples are:

- Decrease \textfraction to get more 'float' on a text page, but the sum of \textfraction and \topfraction and the sum of \textfraction and \bottomfraction should not exceed 1.0, otherwise the placement algorithm falls apart. A minimum value for \textfraction is about 0.10 a page with less than 10% text looks better with no text at all, just floats.
- Both \topfraction and \bottomfraction can be increased, and it does not matter if their sum exceeds 1.0. A good typographic style is that floats are encouraged to go at the top of a page, and a better balance is achieved if the float space on a page is larger at the top than the bottom.
- Making \floatpagefraction too small might have the effect of a float page just having one small float. However, to make sure that a float page never has more than one float on it, do:

```
\renewcommand{\floatpagefraction}{0.01}
\setlength{\@fpsep}{\textheight}
```

- Setting \@fptop and \@dblftop to Opt, \@fpsep to 8pt, and \@fpbot and \@dblfpbot to Opt plus 1fil will force floats on a float page to start at the top of the page.
- Setting \@fpbot and \@dblfpbot to Opt, \@fpsep to 8pt, and \@fptop and \@dblfptop to Opt plus 1fil will force floats on a float page to the bottom of the page.

If you are experimenting, a reasonable starting position is:

```
\setcounter{topnumber}{3}
\setcounter{bottomnumber}{2}
\setcounter{totalnumber}{4}
\renewcommand{\topfraction}{0.85}
\renewcommand{\bottomfraction}{0.5}
\renewcommand{\textfraction}{0.15}
\renewcommand{\floatpagefraction}{0.7}
```

and similarly for double column floats if you will have any. Actually, there is no need to try these settings as they are the default for this class.

One of LaTeX's little quirks is that on a text page, the 'height' of a float is its actual height plus \textfloatsep or \floatsep, while on a float page the 'height' is the actual height. This means that when using the default  $\langle loc \rangle$  of [tbp] at least one of the text page float fractions (\texttt{topfraction} and/or \bottomfraction) must be larger than the \floatpagefraction by an amount sufficient to take account of the maximum text page separation value.

#### 10.5 Captions

Some publishers require, and some authors prefer, captioning styles other than the one style provided by standard LaTeX. Further, some demand that documents that include multipart tables use a *continuation caption* on all but the first part of the multi-part table. For the times where such a table is specified by the author as a set of tables, the class provides a simple 'continuation' caption command to meet this requirement. It also provides a facility for an 'anonymous' caption which can be used in any float environment. Captions can be defined that are suitable for use in non-float environments, such as placing a picture in a minipage and captioning it just as though it had been put into a normal figure environment.

The commands described below are very similar to those supplied by the ccaption package [Wil01d].

10.6 Caption styling

Just as a reminder, the default appearance of a caption for, say, a table looks like this:

Table 11.7: Title for the table

That is, it is typeset in the normal body font, with a colon after the number. The class uses the following to specify the standard LaTeX caption style:

```
\captionnamefont{}
\captiontitlefont{}
\captionstyle{}
\captionwidth{\linewidth}
\normalcaptionwidth
\normalcaption
\captiondelim{: }
```

These macros are explained in detail below.

```
\colon \colon
```

The default captioning style is to put a delimeter in the form of a colon between the caption number and the caption title. The command \captiondelim can be used to change the delimeter. For example, to have an en-dash instead of the colon, \captiondelim{--} \wilder will do the trick. Notice that no space is put between the delimeter and the title unless it is specified in the \( \lambda e \limb \) parameter. The class initially specifies \captiondelim{:} \text{ to give the normal delimeter.}

```
\colon=0
```

The \(\fontspec\)\ specified by \captionnamefont is used for typesetting the caption name; that is, the first part of the caption up to and including the delimeter (e.g., the portion 'Table 3:'). \(\fontspec\)\ can be any kind of font specification and/or command and/or text. This first part of the caption is treated like:

```
{\captionnamefont Table 3: }
```

so font declarations, not font text-style commands, are needed for  $\langle fontspec \rangle$ . For instance,

```
\captionnamefont{\Large\sffamily}
```

to specify a large sans-serif font. The class initially specifies \captionnamefont{} to give the normal font.

Similarly, the  $\langle fontspec \rangle$  specified by \captiontitlefont is used for typesetting the title text of a caption. For example, \captiontitlefont{\\itshape} for an italic title text. The class initially specifies \captiontitlefont{\} to give the normal font.

```
\label{lem:captionstyle} $$ \operatorname{captionstyle} (\langle short \rangle) = (\langle short \rangle)
```

By default the name and title of a caption are typeset as a block (non-indented) paragraph. \captionstyle can be used to alter this. Sensible values for  $\langle style \rangle$  are: \centering, \raggedleft or \raggedright for styles corresponding to these declarations. The \centerlastline style gives a block paragraph but with the last line centered. The class initially specifies \captionstyle{} to give the normal block paragraph style.

If a caption is less than one line in length it may look odd if the  $\langle style \rangle$  is \raggedright, say, as it will be left justified. The optional  $\langle short \rangle$  argument to \captionstyle can be used to specify the style for such short captions if it should differ from that for multiline captions. For example, I think that short captions look better centered:

\captionstyle[\centering]{\raggedright}

The declaration \hangcaption causes captions to be typeset with the second and later lines of a multiline caption title indented by the width of the caption name. The declaration \indentcaption will indent title lines after the first by  $\langle length \rangle$ . These commands are independent of the \captionstyle{...} and have no effect on short captions. Note that a caption will not be simultaneously hung and indented. The \normalcaption declaration undoes any previous \hangcaption or \indentcaption declaration. The class initially specifies \normalcaption to give the normal non-indented paragraph style.

Issuing the declaration \changecaptionwidth causes the captions to be typeset within a total width  $\langle length \rangle$  as specified by \captionwidth. Issuing the declaration \normalcaptionwidth causes captions to be typeset as normal full width captions. The class initially specifies

```
\normalcaptionwidth
\captionwidth{\linewidth}
```

to give the normal width. If a caption is being set within the side captioned environments from the sidecap package  $\lceil NG98 \rceil$  then it must be a \normalcaptionwidth caption.

```
\label{eq:local_precaption} $$ \operatorname{captiontitlefinal}_{\langle text\rangle} $$ \operatorname{costcaption}_{\langle posttext\rangle} $$
```

The commands \precaption and \postcaption specify  $\langle pretext \rangle$  and  $\langle posttext \rangle$  that will be processed at the start and end of a caption. For example

```
\precaption{\rule{\linewidth}{0.4pt}\par}
\postcaption{\rule{\linewidth}{0.4pt}}
```

will draw a horizontal line above and below the captions. The class initially specifies

```
\precaption{}
\postcaption{}
```

to give the normal appearance.

The argument to \captiontitlefinal is put immediately after the title text but will not appear in the LoF or LoT. The default is

```
\captiontitlefinal{}
```

but it could be used instead as, say

```
\captiontitlefinal{.}
```

to put a period (full stop) after the title.

If any of the above commands are used in a float, or other, environment their effect is limited to the environment. If they are used in the preamble or the main text, their effect persists until replaced by a similar command with a different parameter value. The commands do not affect the appearance of the title in any 'List of $\cdots$ '.

The normal LaTeX command  $\setminus$  can be used within the caption text to start a new line. Remember that  $\setminus$  is a fragile command, so if it is used within text that will be added to a 'List of...' it must be protected. As examples:

```
\caption{Title with a \protect\\ new line in
        both the body and List of}
\caption[List of entry with no new line]%
        {Title with a \\ new line}
```

Table 10.3 Redesigned table caption style

three III five V eight VIII

Replacing the above commands by their defaults leads to the simple format: {NAME NUMBER: }{THE TITLE}

As well as using the styling commands to make simple changes to the captioning style, more noticeable modifications can also be made. To change the captioning style so that the name and title are typeset in a sans font it is sufficient to do:

```
\captionnamefont{\sffamily}
\captiontitlefont{\sffamily}
A more obvious change in styling is shown in Table 10.3, which was coded as:
\begin{table}
\centering
\captionnamefont{\sffamily}
\captiondelim{}
\captionstyle{\\}
\captionstyle{\\}
\captiontitlefont{\scshape}
\setlength{\belowcaptionskip}{10pt}
\caption{Redesigned table caption style} \label{tab:style}
\begin{tabular}{lr} \toprule
...
\end{table}
```

This leads to the approximate caption format (processed within \centering):

```
{\sffamily NAME NUMBER}{\\ \scshape THE TITLE}
```

Note that the newline command (\\) cannot be put in the first part of the format (i.e., the {\sffamily NAME NUMBER}); it has to go into the second part, which is why it is specified via \captionstyle{\\} and not \captiondelim{\\}.

If a mixture of captioning styles will be used you may want to define a special caption command for each non-standard style. For example for the style of the caption in Table 10.3:

```
\newcommand{\mycaption}[2][\@empty]{
  \captionnamefont{\sffamily\hfill}
  \captiondelim{\hfill}
  \captionstyle{\centerlastline\\}
  \captiontitlefont{\scshape}
  \setlength{\belowcaptionskip}{10pt}
  \ifx \@empty#1 \caption{#2}\else \caption[#1]{#2}\fi}
```

Remember that any code that involves the @ sign must be either in a package (sty) file or enclosed between a \makeatletter \www.\makeatother pairing (see §E.4).

The code for the Table 10.3 example can now be written as:

```
\begin{table}
\centering
\mycaption{Redesigned table caption style} \label{tab:style}
\begin{tabular}{lr} \toprule
...
\end{table}
```

Note that in the code for \mycaption I have added two \hfill commands and \centerlastline compared with the original specification. It turned out that the original definitions worked for a single line caption but not for a multiline caption. The additional commands makes it work in both cases, forcing the name to be centered as well as the last line of a multiline title, thus giving a balanced appearence.

10.7 Continuation captions and legends

```
\contcaption{\langle text \rangle}
```

The \contcaption command can be used to put a 'continued' or 'concluded' caption into a float environment. It neither increments the float number nor makes any entry into a float listing, but it does repeat the numbering of the previous \caption command.

Table 10.4 illustrates the use of the \contcaption command. The table was produced from the following code.

```
\begin{table}
\centering
\caption{A multi-part table} \label{tab:m}
\begin{tabular}{lc} \toprule
  just a single line & 1 \\ \bottomrule
\end{tabular}
\end{table}

\begin{table}
\centering
\contcaption{Continued}
\begin{tabular}{lc} \toprule
  just a single line & 2 \\ \bottomrule
\end{tabular}
\end{tabular}
\end{tabular}
\end{table}
```

# Table 10.4: A multi-part table

just a single line

Table 10.4: Continued

just a single line

Table 10.4: Concluded

just a single line

```
\begin{table}
\centering
\contcaption{Concluded}
\begin{tabular}{lc} \toprule
  just a single line & 3 \\ \bottomrule
\end{tabular}
\end{tabular}
\legend{\leftabular}
```

The \legend command is intended to be used to put an anonymous caption, or legend into a float environment, but may be used anywhere.

For example, the following code was used to produce the two-line Table 10.5. The  $\ensuremath{\texttt{legend}}$  command can be used within a float independently of any  $\ensuremath{\texttt{caption}}$  command.

```
\begin{table}
  \centering
  \caption{Another table} \label{tab:legend}
  \begin{tabular}{lc} \toprule
```

```
Table 10.5: Another table

A legendary table 5
with two lines 6

The legend
```

# Legendary table An anonymous table 5 with two lines 6

```
A legendary table & 5 \\
with two lines & 6 \\ \bottomrule \end{tabular}
\legend{The legend}
\end{table}
```

Captioned floats are usually thought of in terms of the table and figure environments. There can be other kinds of float. As perhaps a more interesting example, the following code produces the titled marginal note which should be displayed near here.

```
\marginpar{\legend{LEGEND}

This is a marginal note with a legend.}
```

If you want the legend text to be included in the 'List of  $\cdots$ ' you can do it like this with the  $\addcontentsline$  macro.

```
\legend{Legend title}
% left justified
\addcontentsline{lot}{table}{Legend title} % or
% indented
\addcontentsline{lot}{table}{\protect\numberline{}Legend title}
```

The first of these forms will align the first line of the legend text under the normal table numbers. The second form will align the first line of the legend text under the normal table titles. In either case, second and later lines of a multi-line text will be aligned under the normal title lines.

As an example, the Legendary table is produced by the following code:

Look at the List of Tables to see how the two forms of \addcontentsline are typeset.

```
\verb|\namedlegend[|\langle short-title\rangle]| = \{\langle long-title\rangle\}|
```

This is a marginal note with a legend.

```
Table: Named legendary table

| seven VII | eight VIII |
```

As a convenience, the \namedlegend command is like the \caption command except that it does not number the caption and, by default, puts no entry into a 'List of...' file. Like the \caption command, it picks up the name to be prepended to the title text from the float environment in which it is called (e.g., it will use \tablename if called within a table environment). The following code is the source of the Named legendary table.

```
\begin{table}
\centering
\captionnamefont{\sffamily}
\captiontitlefont{\itshape}
\namedlegend{\Named legendary table}
\begin{tabular}{lr} \toprule
seven & VII \\
eight & VIII \\ bottomrule
\end{tabular}
\end{table}

\flegfloat{\(\lamb{name}\rangle\)}
\flegtlogtfloat{\(\lamb{title}\rangle\)}
\flegtlogtfloat{\(\lamb{title}\rangle\)}
\flegtlogtfloat{\(\lamb{title}\rangle\)}
\flegtlogtfloat{\(\lamb{title}\rangle\)}
\flegtlogtfloat{\(\lamb{title}\rangle\)}
\flegtlogtfloat{\(\lamb{title}\rangle\)}
\flegtlogtfloat{\(\lamb{title}\rangle\)}
\flegtlogtfloat{\(\lamb{title}\rangle\)}
\flegtloat{\(\lamb{title}\rangle\)}
\flegtloat{\(\lamb{title}\rangle\rangle\)}
\flegtloat{\(\lamb{title}\rangle\rangle\)}
\flegtloat{\(\lamb{title}\rangle\rangle\rangle\)}
\flegtloat{\(\lamb{title}\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle\rangle
```

The macro flegfloat, where float is the name of a float environment (e.g., figure) is called by the namedlegend macro. It is provided as a hook that defines the name to be used as the name in namedlegend. Two defaults are provided, flegtable and flegfigure defined as:

```
\newcommand{\flegtable}{\tablename}
\newcommand{\flegfigure}{\figurename}
```

which may be altered via \renewcommand if desired.

The macro \flegtocfloat, where again float is the name of a float environment (e.g., table) is also called by the \namedlegend macro. It is provided as a hook that can be used to add  $\langle title \rangle$  to the 'List of...'. Two examplars are provided, \flegtocfigure and \flegtoctable. By default they are defined to do nothing, and can be changed via \renewcommand. For instance, one could be changed for tables as:

```
\renewcommand{\flegtoctable}[1]{
  \addcontentsline{lot}{table}{#1}}
```

The \legend command produces a plain, unnumbered heading. It can also be useful sometimes to have named and numbered captions outside a floating environment, perhaps in a minipage, if you want the table or picture to appear at a precise location in your document.

The \newfixedcaption command, and its friends, can be used to create or modify a captioning  $\langle command \rangle$  that may be used outside the float environment  $\langle float \rangle$ . Both the environment  $\langle float \rangle$  and a captioning command,  $\langle capcommand \rangle$ , for that environment must have been defined before calling \newfixedcaption. Note that \namedlegend can be used as  $\langle capcommand \rangle$ .

For example, to define a new \figcaption command for captioning pictures outside the figure environment, do

```
\newfixedcaption{\figcaption}{figure}
```

The optional *(capcommand)* argument is the name of the float captioning command that is being aliased. It defaults to *\caption*. As an example of where the optional argument is required, if you want to create a new continuation caption command for non-floating tables, say *\ctabcaption*, then do

```
\newfixedcaption[\contcaption]{\ctabcaption}{table}
```

Captioning commands created by  $\mbox{\sc hewfixedcaption}$  will be named and numbered in the same style as the original  $\mbox{\sc capcommand}$ , can be given a  $\mbox{\sc habel}$ , and will appear in the appropriate 'List of...'. They can also be used within floating environments, but will not use the environment name as a guide to the caption name or entry into the 'List of...'. For example, using  $\mbox{\sc habeline}$  in a figure environment will still produce a **Table**... named caption.

Sometimes captions are required on the opposite page to a figure, and a fixed caption can be useful in this context. For example, if figure captions should be placed on an otherwise empty page immediately before the actual figure, then this can be accomplished by the following hack:

```
\newfixedcaption{\figcaption}{figure}
...
\afterpage{ % fill current page then flush pending floats
  \clearpage
  \begin{midpage} % vertically center the caption
  \figcaption{The caption} % the caption
  \end{midpage}
  \clearpage
  \begin{figure}THE FIGURE, NO CAPTION HERE\end{figure}
  \clearpage
} % end of \afterpage
```

Note that the afterpage package [Car95] is needed, which is part of the required tools bundle. The midpage package supplies the midpage environment, which can be simply defined as:

```
\newenvironment{midpage}{\vspace*{\fill}}{\vspace*{\fill}}}
```

The code, in particular the use of \clearpage, might need adjusting to meet your particular requirements.

• \clearpage gets you to the next page, which may be odd or even.

- \cleardoublepage gets you to the next odd-numbered page.
- \cleartoevenpage ensures that you get to the next even-numbered page.

As a word of warning, if you mix both floats and fixed environments with the same kind of caption you have to ensure that they get printed in the correct order in the final document. If you do not do this, then the 'List of...' captions will come out in the wrong order (the lists are ordered according the page number in the typeset document, *not* your source input order).

# 10.8 Bilingual captions

Some documents require bilingual (or more) captions. The class provides a set of commands for bilingual captions. Extensions to the set, perhaps to support trilingual captioning, are left as an exercise for the document author. Essentially, the bilingual commands call the \caption command twice, once for each language.

Several commands for bilingual captions are provided. They all produce the same appearance in the text but differ in what they put into the 'List of $\cdots$ '.

```
\label{long1} $$ \left(\frac{label}{\langle short1\rangle} {\langle long1\rangle} \% \right) $$ \left(\frac{NAME}{\langle short2\rangle} {\langle long2\rangle} \right) $$ \left(\frac{label}{\langle short1\rangle} {\langle long1\rangle} \% \right) $$ \left(\frac{NAME}{\langle short2\rangle} {\langle long2\rangle} \right) $$
```

Bilingual captions can be typeset by the \bitwonumcaption command which has six arguments. The first, optional argument  $\langle label \rangle$ , is the name of a label, if required.  $\langle short1 \rangle$  and  $\langle long1 \rangle$  are the short (i.e., equivalent to the optional argument to the \caption command) and long caption texts for the main language of the document. The value of the  $\langle NAME \rangle$  argument is used as the caption name for the second language caption, while  $\langle short2 \rangle$  and  $\langle long2 \rangle$  are the short and long caption texts for the second language. For example, if the main and secondary languages are English and German and a figure is being captioned:

```
\verb|\bitwonumcaption{Short}{Long}{Bild}{Kurz}{Lang}|
```

If the short title text(s) is not required, then leave the appropriate argument(s) either empty or as one or more spaces, like:

```
\bitwonumcaption[fig:bi1]{}{Long}{Bild}{ }{Lang}
```

Both language texts are entered into the appropriate 'List of...', and both texts are numbered.

Figure 10.16, typeset from the following code, is an example.

```
\begin{figure}
\centering
EXAMPLE FIGURE WITH BITWONUMCAPTION
\bitwonumcaption[fig:bi1]%
   {}{Long \cs{bitwonumcaption}}%
   {Bild}{ }{Lang \cs{bitwonumcaption}}
\end{figure}
```

Both \bionenumcaption and \bitwonumcaption take the same arguments. The difference between the two commands is that \bionenumcaption does not number the second

# EXAMPLE FIGURE WITH BITWONUMCAPTION

Figure 10.16: Long \bitwonumcaption
Bild 10.16: Lang \bitwonumcaption

# EXAMPLE FIGURE WITH BIONENUMCAPTION

Figure 10.17: Long English \bionenumcaption
Bild 10.17: Lang Deutsch \bionenumcaption

language text in the 'List of $\cdots$ '. Figure 10.17, typeset from the following, is an example of this.

When bilingual captions are typeset via the \bicaption command the second language text is not put into the 'List of...'. The command takes 5 arguments. The optional  $\langle label \rangle$  is for a label if required.  $\langle short1 \rangle$  and  $\langle long1 \rangle$  are the short and long caption texts for the main language of the document. The value of the  $\langle NAME \rangle$  argument is used as the caption name for the second language caption. The last argument,  $\langle long2 \rangle$ , is the caption text for the second language (which is not put into the 'List of...').

For example, if the main and secondary languages are English and German:

```
\bicaption{Short}{Long}{Bild}{Langlauf}
```

If the short title text is not required, then leave the appropriate argument either empty or as one or more spaces.

Figure 10.18 is an example of using \bicaption and was produced by the following code:

# EXAMPLE FIGURE WITH A RULED BICAPTION

Figure 10.18: Longingly Bild 10.18: Langlauf

Bilingual continuation captions can be typeset via the \bicontcaption command. In this case, neither language text is put into the 'List of...'. The command takes 3 arguments.  $\langle long1 \rangle$  is the caption text for the main language of the document. The value of the  $\langle NAME \rangle$  argument is used as the caption name for the second language caption. The last argument,  $\langle long2 \rangle$ , is the caption text for the second language. For example, if the main and secondary languages are again English and German:

\bicontcaption{Continued}{Bild}{Fortgefahren}

```
\mbox{\mbox{midbicaption}} \langle text \rangle \}
```

The bilingual captions are implemented by calling \caption twice, once for each language. The command \midbicaption, which is similar to the \precaption and \postcaption commands, is executed just before calling the second \caption. Among other things, this can be used to modify the style of the second caption with respect to the first. For example, if there is a line above and below normal captions, it is probably undesirable to have a double line in the middle of a bilingual caption. So, for bilingual captions the following may be done within the float before the caption:

```
\precaption{\rule{\linewidth}{0.4pt}\par}
\postcaption{}
\midbicaption{\precaption{}%
    \postcaption{\rule{\linewidth}{0.4pt}}}
```

This sets a line before the first of the two captions, then the \midbicaption{...} nulls the pre-caption line and adds a post-caption line for the second caption. The class initially specifies \midbicaption{}.

#### 10.9 Subcaptions

The subfigure package enables the captioning of sub-figures within a larger figure, and similarly for tables. The subfigure package may be used with the class, or you can use the class commands described below; these commands can only be used inside a float environment for which a subfloat<sup>3</sup> has been specified via \newsubfloat.

 $<sup>^{3}</sup>$ See §10.3.

```
\subcaption[\langle list-entry \rangle] \{\langle subtitle \rangle\}
```

The \subcaption command is similar to the \caption command and can only be used inside a float environment. It typesets an identified  $\langle subtitle \rangle$ , where the identification is an alphabetic character enclosed in parentheses. If the optional  $\langle list\text{-entry} \rangle$  argument is present,  $\langle list\text{-entry} \rangle$  is added to the caption listings for the float. If it is not present, then  $\langle subtitle \rangle$  is added to the listing.

The  $\langle subtitle \rangle$  is typeset within a box which is the width of the surrounding environment, so \subcaption should only be used within a fixed width box of some kind, for example a minipage as shown below.

```
\begin{figure}
\centering
\begin{minipage}{0.3\textwidth}
  \verb?Some verbatim text?
  \subcaption{First text}
\end{minipage}
\hfill
\begin{minipage}{0.3\textwidth}
  \verb?More verbatim text?
  \subcaption{Second text}
\end{minipage}
\caption{Verbatim texts}
\end{figure}
```

As the example code shows, the \subcaption command provides a means of putting verbatim elements into subfigures.

```
\label{list-entry} $$ \left(\frac{|st-entry|}{\langle subtitle|} \right) = \left(\frac{|st-entry|}{
```

The command \subtop puts a subcaption identifier on top of  $\langle text \rangle$ . If both optional arguments are present,  $\langle list-entry \rangle$  will be added to the appropriate listing, and  $\langle subtitle \rangle$  is placed above the  $\langle text \rangle$  with the identifier. If only one optional argument is present this is treated as being  $\langle subtitle \rangle$ ; the identifier and  $\langle subtitle \rangle$  are placed above the  $\langle text \rangle$  and  $\langle subtitle \rangle$  is added to the listing. Regardless of the optional arguments the identifier is always added to the listing and placed above the  $\langle text \rangle$ .

The \subbottom command is identical to \subtop except that the identifier, and potentially the  $\langle subtitle \rangle$ , is placed below the  $\langle text \rangle$ . Note that verbatim text cannot be used in the  $\langle text \rangle$  argument to \subbottom or \subtop.

The main caption can be at either the top or the bottom of the float. The positioning of the main and subcaptions are independent. For example

If a figure that includes subfigures is itself continued then it may be desirable to continue the captioning of the subfigures. For example, if Figure 3 has three subfigures, say A, B and C, and Figure 3 is continued then the subfigures in the continuation should be D, E, etc.

```
\label{eq:contsubcaption} $$ \operatorname{\langle list-entry \rangle} {\langle subtitle \rangle} $$ \operatorname{\langle list-entry \rangle} [\langle subtitle \rangle] {\langle text \rangle} $$ \operatorname{\langle list-entry \rangle} [\langle subtitle \rangle] {\langle text \rangle} $$ \operatorname{\langle bulled} $$
```

The \contsubcaption, \contsuptop and \contsubbottom commands are the continued versions of the respective subcaptioning commands. These continue the subcaption numbering scheme across (continued) floats. In any event, the main caption can be at the top or bottom of the float. The \subconcluded command is used to indicate that the continued (sub) float has been concluded and the numbering scheme is reinitialized. The command should be placed immediately before the end of the last continued environment. For example:

```
\begin{figure}
\subbottom{...} captioned as (a) below
\subbottom{...} captioned as (b) below
\caption{...}
\end{figure}
\begin{figure}
\contsubtop{...} captioned as (c) above
\contsubtop{...} captioned as (d) above
\contcaption{Concluded}
\subconcluded
\end{figure}
\begin{table}
\caption{...}
\subtop{...}
                captioned as (a) above
\subbottom{...} captioned as (b) below
\end{table}
 \subcaptionref{\langle labstr \rangle}
```

A \label command may be included in the  $\langle subtitle \rangle$  argument of the subcaptioning commands. Using the normal \ref macro to refer to the label will typeset the number of the float (obtained from a \labeled main \caption) and the subcaption identifier. If the \subcaptionref macro is used instead of \ref then only the subcaption identifier is printed.

In cases where the hyperref package is used, the \label command when used inside the  $\langle subtitle \rangle$  argument can take an optional  $\langle bookmark \rangle$  argument, enclosed in parenthese not square brackets, which will create a bookmark field of the form 'Subfigure 4.7(d)'.

As an example to show the difference between \subcaptionref and \ref, Figure 10.19 and the paragraph immediately following this one were produced by the code shown below.

```
Figure 10.19 has two subfigures, namely 10.19(a) and (b). Figure \ref{fig:twosubfig} has two subfigures, namely \ref{sf:1} and \subcaptionref{sf:2}. \begin{figure}
```

# SUBFIGURE ONE

SUBFIGURE TWO

(a) Subfigure 1

(b) Subfigure 2

Figure 10.19: Figure with two subfigures

```
\centering
\subbottom[Subfigure 1]{\fbox{SUBFIGURE ONE}\label{sf:1}}
\hfill
\subbottom[Subfigure 2]{\fbox{SUBFIGURE TWO}\label{sf:2}}
\caption{Figure with two subfigures} \label{fig:twosubfig}
\end{figure}
```

```
\tightsubcaptions \loosesubcaptions
```

As with many other aspects of typesetting the style of subcaptions may be specified. There is a small amount of vertical space surrounding a subcaption. More space is used after the \loosesubcaptions declaration compared to that produced after the default \tightsubcaptions declaration.

```
\subcaptionsize{\langle size \rangle} \\ subcaptionlabelfont{\langle fontspec \rangle} \\ subcaptionfont{\langle fontspec \rangle} \\
```

The size of the font used for subcaptions is specified by \subcaptionsize, and the fonts for the identifier and text are specified by \subcaptionlabelfont for the identifier and by \subcaptionfont for the title text. The defaults are:

```
\subcaptionsize{\footnotesize}
\subcaptionlabelfont{\normalfont}
\subcaptionfont{\normalfont}
```

```
\label{eq:local_subcaptionstyle} $$ \operatorname{\centering \raggedright \centerlastline} $$
```

The identifier and title of a subcaption is typeset as a block (i.e., non-indented) paragraph by specifying

```
\subcaptionstyle{}
```

Other styles are available by calling \subcaptionstyle with a styling  $\langle cmd \rangle$ . Values that you might use are: \centering for a centered paragraph, \raggedleft or \raggedright for ragged left or right paragraphs, or \centerlastline which calls for a block paragraph with the last line centered.

```
\hangsubcaption
\shortsubcaption
\normalsubcaption
```

The \hangsubcaption declaration causes subcaptions to be typeset with the identifier above the title. Following the \shortsubcaption declaration subcaptions

that are less than a full line in length are typeset left justified instead of centered. The \normalsubcaption declaration, which is the default, undoes any previous \hangsubcaption or \shortsubcaption declaration, so that subcaptions are normally centered.

```
10.10 Side captions
```

Typically captions are put either above or below the element they are describing. Sometimes it is desireable to put a caption at the side of the element instead.

The sidecaption environment is used for a sidecaption rather than a macro. The body of the float is put inside the environment. For example:

```
\begin{figure}
  \begin{sidecaption}{An illustration}[fig:ill]
  \centering
  \includegraphics{...}
  \end{sidecaption}
\end{figure}
```

whereby the caption, 'Figure N: An illustration', will be placed in the margin alongside the graphic, and for reference purposes will be given given the \label fig:ill.

```
\label{eq:linear_side_capse} $$ \setsidecaps { \langle sep \rangle } {\langle width \rangle }$
```

The caption is set in a box \sidecapwidth wide (the default is \marginparwidth) offset \sidecapsep (default \marginparsep) into the margin. The command \setsidcaps sets the \sidecapsep and \sidecapwidth to the given values. Changing the marginpar parameters, for example with \setmarginnotes, will not change the side caption settings. Note also that \checkandfixthelayout neither checks nor fixes the side caption parameters.

If the float is a single column float in a twocolumn document then the caption is always placed in the adjacent margin, otherwise the \sidecapmargin command controls the margin where the sidecaption will be placed. The possible values for  $\langle margin \rangle$  are one of: left, right, inner, or outer. If left or right is specified the caption will go into the left or right margin. If inner or outer is specified then in a two sided document the caption will be on different sides of the typeblock according to whether it is a recto or verso page; in a one sided document the caption margin is fixed. The left margin is the default.

When the caption is to be set in the left margin, \ifscapmargleft is set true, and for a right margin it is set false.

```
\sline \sline
```

<sup>&</sup>lt;sup>4</sup>Well, nearly always. See the \overridescapmargin command later.

By default a sidecaption is vertically centered with respect to the float it is captioning. This can be altered by using the \setsidecappos declaration. The allowed values for  $\langle pos \rangle$  are:

- t the top of the caption is aligned with the top of the float
- c (the default) the center of the caption is aligned with the center of the float
- the bottom of the caption is aligned with the bottom of the float

The other kinds of simple captions can also be put at the side of a float. The positioning and styling commands for these are exactly those for sidecaption. Bilingual captions, which are not considered to be simple, can only be placed above or below a float; no facilities are provided for setting them at the side..

Sidecaptions may be continued with the sidecontcaption environment.

Named legends may be set at the side with the sidenamedlegend environment.

```
\label{legend} $$ \left( \left\langle title \right\rangle \right)$ the body of the float $$ \left( sidelegend \right) $$
```

Legends may be set at the side with the sidelegend environment.

**Caveat**: Note that the side... envs expect the body of the float to be *taller* than the typeset caption/legend. In case you write a long caption/legend for a short float, you may want to visit this answer: http://tex.stackexchange.com/a/228412/3929.

```
10.10.1 Tweaks
```

```
\sidecapstyle
```

Just before the caption is set, the \sidecapstyle command is called. This may be used to set the styling for the particular caption. By default it sets captions that are in the left margin raggedleft, and those that are in the right margin are set raggedright. The default definition is:

```
\newcommand*{\sidecapstyle}{%
%% \captionnamefont{\bfseries}
  \ifscapmargleft
   \captionstyle{\raggedleft}%
  \else
   \captionstyle{\raggedright}%
  \fi}
```

You can change the command to suit your purposes; for example, uncommenting the \captionnamefont line would result in the caption's float name being set in a bold font.

Table 10.6: Permitted arguments for some sidecaption related commands

\overridescapmargin
left
right

Sometimes the caption may not be placed exactly where you want it — it may be in the wrong margin or at the wrong height.

The command \overridescapmargin will force the following caption into the  $\langle margin \rangle$  you specify which can only be left or right. In a two-sided document where \sidecapmargin is inner or outer and the caption goes in the wrong margin, it is likely that the declaration \strictpagecheck will solve the problem. The wrong margin might be chosen in a two-column document where the float is in the second column; use

\overridescapmargin{right}

to fix this.

The caption may not be at quite the height you want with respect to the float. The caption will be raised by the length \sidecapraise in addition to the calculated movement (or lowered if \sidecapraise is negative).

```
\sidecapfloatwidth{\langle length \rangle}
```

The float is set in a minipage with width sidecapfloatwidth, whose default definition is

\newcommand\*{\sidecapfloatwidth}{\linewidth}

That is, the normal width is the same as the current \linewidth. For a narrow table, say, you may want to reduce this, for example to half by

 $\verb|\command*{\sidecapfloatwidth}{0.5}| inewidth| \\$ 

Note that \sidecapfloatwidth is a macro, not a length, so it must be altered by using a \renewcommand\*, not by \setlength.

If you do reduce the \sidecapfloatwidth you may notice that the sidecaption is actually placed a distance \sidecapsep with respect to the float's minipage, not with respect to the text block.

Table 10.6 was created by the following code.

```
\newlength{\mylength}
\setlength{\mylength}{\linewidth}
\addtolength{\mylength}{-\sidecapsep}
\addtolength{\mylength}{-\sidecapwidth}
\begin{table}
```

```
\sidecapmargin{left}%
          \verb|\renewcommand*{\sidecapfloatwidth}{\mbox{\mbox{$\mbox{$\mbox{$}}$}} | $$ \cite{those of the command the comman
          \raggedleft
          \begin{sidecaption}{%
                    Permitted arguments for some sidecaption related commands}[scap:one]
          \centering
          \begin{tabular}{cc} \toprule
          \cs{sidecapmargin} & \cs{overridescapmargin} \\ \midrule
          \texttt{left}
                                                                                                           & \texttt{left}
          \texttt{right}
                                                                                                                & \texttt{right}
                                                                                                                                                                                                                                                    11
                                                                                                          & \\
         \texttt{inner}
         \texttt{outer}
                                                                                                           & \\ \bottomrule
         \end{tabular}
\end{sidecaption}
\end{table}
```

The calculations on the \mylength length are so that the sidecaption and float will just fit inside the typeblock.

Note that the \raggedleft command before the sidecaption environment makes the float's minipage be placed raggedleft (i.e., moved across to the right hand edge of the type-block) while the \centering centers the tabular within the minipage. You can get a variety of horizontal placements by judicious use of \raggedright, \centering and \raggedleft commands. If you do move the float sideways to leave space for the caption make sure that the caption will go to the side you want. In the example code I 'moved' the float to the right so I made sure that the caption would go on the left by explicitly setting

```
\sidecapmargin{left}
```

As far as TeX is concerned a sidecaption takes no horizontal space. If you use a sidecaption in a wrapped float from, say, the wrapfig package, make sure that the sidecaption gets placed where it won't be overlaid by the main text.

#### 10.11 How LaTeX makes captions

This section provides an overview of how LaTeX creates captions and gives some examples of how to change the captioning style directly. The section need not be looked at more than once unless you like reading LaTeX code or you want to make changes to LaTeX's style of captioning not supported by the class.

The LaTeX kernel provides tools to help in the definition of captions, but it is the particular class that decides on their format.

\@captype is defined by the code that creates a new float environment and is set to the environment's name (see the code for \@xfloat in ltfloat.dtx). For a figure environment, there is an equivalent to

\def\@captype{figure}

```
\cline{type} [\langle short \rangle] \{\langle long \rangle\}
```

The kernel also provides the \@caption macro as:

where  $\langle type \rangle$  is the name of the environment in which the caption will be used. Putting these three commands together results in the user's view of the caption command as  $\column{caption}[\langle short \rangle] \{\langle long \rangle\}$ .

It is the responsibilty of the class (or package) which defines floats to provide definitions for \ext@type, \fnum@type and \@makecaption which appear in the definition of \@caption (in the lines marked <- above).

```
\ext@type
```

This macro holds the name of the extension for a 'List of...' file. For example for the figure float environment there is the definition equivalent to

\newcommand{\ext@figure}{lof}

```
\fnum@type
```

This macro is responsible for typesetting the caption number. For example, for the figure environment there is the definition equivalent to

\newcommand{\fnum@figure}{\figurename~\thefigure}

```
\verb|\Qmakecaption{|\langle number\rangle|}{\langle text\rangle}|
```

The \@makecaption macro, where  $\langle number \rangle$  is a string such as 'Table 5.3' and  $\langle text \rangle$  is the caption text, performs the typesetting of the caption, and is defined in the standard classes (in classes.dtx) as the equivalent of:

```
\newcommand{\@makecaption}[2]{
  \vskip\abovecaptionskip     <- 1
  \sbox\@tempboxa{#1: #2}     <- 2
  \ifdim \wd\@tempboxa >\hsize
```

# A THOUSAND WORDS...

Figure 10.20: A picture is worth a thousand words

Vertical space is added before and after a caption (lines marked 1 and 4 in the code for \@makecaption above) and the amount of space is given by the lengths \abovecaptionskip and \belowcaptionskip. The standard classes set these to 10pt and 0pt respectively. If you want to change the space before or after a caption, use \setlength to change the values. In figures, the caption is usually placed below the illustration. The actual space between the bottom of the illustration and the baseline of the first line of the caption is the \abovecaptionskip plus the \parskip plus the \baselineskip. If the illustration is in a center environment then additional space will be added by the \end{center}; it is usually better to use the \centering command rather than the center environment.

The actual typesetting of a caption is effectively performed by the code in lines marked 2 and 3 in the code for \@makecaption; note that these are where the colon that is typeset after the number is specified. If you want to make complex changes to the default captioning style you may have to create your own version of \@caption using \renewcommand. On the other hand, many such changes can be achieved by changing the definition of the the appropriate \fnum@type command(s). For example, to make the figure name and number bold:

```
\renewcommand{\fnum@figure}{\textbf{\figurename~\thefigure}}
```

REMEMBER: If you are doing anything involving commands that include the @ character, and it's not in a class or package file, you have to do it within a  $\mbox{makeatletter}$  and  $\mbox{makeatother}$  pairing (see §E.4). So, if you modify the  $\mbox{fnum@figure}$  command anywhere in your document it has to be done as:

```
\makeatletter
\renewcommand{\fnum@figure}{.....}
\makeatother
As an example, Figure 10.20 was created by the following code:
\makeatletter
\renewcommand{\fnum@figure}{\textsc{\figurename~\thefigure}}
\makeatother
\begin{figure}
\centering
```

# ANOTHER THOUSAND WORDS...

Figure 10.21 — A different kind of figure caption

```
A THOUSAND WORDS\ldots \caption{A picture is worth a thousand words}\label{fig:sc} \end{figure}
```

As another example, suppose that you needed to typeset the \figurename and its number in a bold font, replace the colon that normally appears after the number by a long dash, and typeset the actual title text in a sans-serif font, as is illustrated by the caption for Figure 10.21. The following code does this.

Perhaps a little description of how this works is in order. Doing a little bit of TeX's macro processing by hand, the typesetting lines in \@makecaption (lines 2 and 3) get instantiated like:

```
\fnum@figure{\figurename~\thefigure}: text
```

Redefining \fnum@figure to take one argument and then not using the value of the argument essentially gobbles up the colon. Using

```
\textbf{\figurename~\thefigure}
```

in the definition causes \figurename and the number to be typeset in a bold font. After this comes the long dash. Finally, putting \sffamily at the end of the redefinition causes any following text (i.e., the actual title) to be typeset using the sans-serif font.

If you do modify \@makecaption, then spaces in the definition may be important; also you must use the comment (%) character in the same places as I have done above. Hopefully, though, the class provides the tools that you need to make most, if not all, of any likely caption styles.

# 10.12 Footnotes in captions

If you want to have a caption with a footnote, think long and hard as to whether this is really essential. It is not normally considered to be good typographic practice, and to rub the point in LaTeX does not make it necessarily easy to do. However, if you (or your publisher) insists, read on.

If it is present, the optional argument to \caption is put into the 'List of...' as appropriate. If the argument is not present, then the text of the required argument is put into the 'List of...'. In the first case, the optional argument is moving, and in the second case the required argument is moving. The \footnote command is fragile and must be \protected (i.e., \protect\footnote{}) if it is used in a moving argument. If you don't want the footnote to appear in the 'List of...', use a footnoteless optional argument and a footnoted required argument.

You will probably be surprised if you just do, for example:

```
\begin{figure}
...
\caption[For LoF]{For figure\footnote{The footnote}}
\end{figure}
```

because (a) the footnote number may be greater than you thought, and (b) the footnote text has vanished. This latter is because LaTeX won't typeset footnotes from a float. To get an actual footnote within the float you have to use a minipage, like:

```
\begin{figure}
\begin{minipage}{\linewidth}
...
\caption[For LoF]{For figure\footnote{The footnote}}
\end{minipage}
\end{figure}
```

If you are using the standard classes you may now find that you get two footnotes for the price of one. Fortunately this will not occur with this class, nor will an unexpected increase of the footnote number.

When using a minipage as above, the footnote text is typeset at the bottom of the minipage (i.e., within the float). If you want the footnote text typeset at the bottom of the page, then you have to use the \footnotemark and \footnotetext commands like:

```
\begin{figure}
...
\caption[For LoF]{For figure\footnotemark}
\end{figure}
\footnotetext{The footnote}
```

This will typeset the argument of the \footnotetext command at the bottom of the page where you called the command. Of course, the figure might have floated to a later page, and then it's a matter of some manual fiddling to get everything on the same page, and possibly to get the footnote marks to match correctly with the footnote text.

At this point, you are on your own.

10.13 The class versus the caption package (and its friends)

For some, the configurations for captions provided by the class, are either a bit too complicated or not complicated enough.

The caption package by Alex Sommerfeldt may provide a simpler and much more extensive configuration interface for captions. The package can be used with the class, but there are a few caveats:

- (a) All of the special configuration macros provided by the class will no longer have any effect (caption overwrites the core, and thus our interfaces will have no effect),
- (b) \abovecaptionskip will be reset to  $10\,\mathrm{pt}$ , and \belowcaptionskip to zero. (The class would set both to 0.5\onelineskip, so if you need to change these, move the change until after caption has been loaded)

# Eleven

# Rows and columns

The class provides extensions to the standard array and tabular environments. These are based partly on a merging of the array [MC18], dcolumn [Car14], delarray [Car14], tabularx [Car16], and booktabs [Fea16] packages. Much of the material in this chapter strongly reflects the documentation of these packages.

**Note.** As of September 2018: The array, delarray, tabularx and dcolumn packages are no longer embedded into the class, but rather being autoloaded from the  $ot\! ET_F X$  installation.  $^1$  As the embededed versions were just carbon copies, $^2$  we get the same result but just loading the packages, with less maintenance. Plus these packages are part of the LTFX core packages and thus is available in all LATEX installations.

We have kept the documentation we had written for the manual (and updated it slightly), but refer to [MC18], [Car14], [Car16] and [Car14] for the 100% up to date documentation.

Additionally, new kinds of tabular environments are also provided.

## 11.1 General

```
\[\langle pos \rangle\] \{\langle preamble \rangle\} \text{ rows } \[ \]
\begin{tabular}[\langle pos \rangle] {\langle preamble \rangle} rows \end{tabular}
\begin{tabular*}{\langle width\rangle}[\langle pos\rangle]{\langle preamble\rangle} rows \end{tabular*}
\begin{tabularx}{\langle width\rangle}[\langle pos\rangle]{\langle preamble\rangle} rows \end{tabularx}
```

The array and tabular environments are traditional and the others are extensions. The array is for typesetting math and has to be within a math environment of some kind. The tabular series are for typesetting ordinary text.

The optional  $\langle pos \rangle$  argument can be one of t, c, or b (the default is c), and controls the vertical position of the array or tabular; either the top or the center, or the bottom is aligned with the baseline. Each row consists of elements separated by &, and finished with \\. There may be as many rows as desired. The number and style of the columns is specified by the  $\langle preamble \rangle$ . The width of each column is wide enough to contain its longest entry and the overall width of the array or tabular is sufficient to contain all the columns. However, the tabular\* and tabularx environments provide more control over the width through their  $\langle width \rangle$  argument.

<sup>&</sup>lt;sup>1</sup>As the embedded versions were more or less carbon copies, it makes much more sense to let the LATEX-team maintain them, than us having to replace the embedded copy each time they are updated or bug fixed.

<sup>2</sup>With edited error messages.

Table 11.1: The array and tabular preamble options.

1	Left adjusted column.
С	Centered adjusted column.
r	Right adjusted column.
$p{\langle width \rangle}$	Equivalent to $\operatorname{parbox}[t]{\langle \operatorname{width} \rangle}$ .
$m\{\langle \mathrm{width} \rangle\}$	Defines a column of width \( \text{width} \). Every entry will be centered in proportion to the rest of the line. It is somewhat like \\parbox{\( \text{width} \)}.
$b{\langle width \rangle}$	Coincides with $\parbox[b]{\langle width \rangle}$ .
$> {\langle \operatorname{decl} \rangle}$	Can be used before an 1, $r$ , $c$ , $p$ , $m$ or a $b$ option. It inserts $\langle \text{decl} \rangle$ directly in front of the entry of the column.
$<\{\langle \mathrm{decl} \rangle\}$	Can be used after an 1, r, c, p{}, m{} or a b{} option. It inserts $\langle \text{decl} \rangle$ right after the entry of the column.
I	Inserts a vertical line. The distance between two columns will be enlarged by the width of the line.
${\tt Q}\{\langle { m decl}  angle\}$	Suppresses inter-column space and inserts (decl) instead.
$\mathop{!}\left\{\left\langle \operatorname{decl}\right\rangle \right\}$	Can be used anywhere and corresponds with the $\mid$ option. The difference is that $\langle \text{decl} \rangle$ is inserted instead of a vertical line, so this option doesn't suppress the normally inserted space between columns in contrast to $\mathfrak{Q}\{\ldots\}$ .
$*{\langle \text{num} \rangle} {\langle \text{opts} \rangle}$	Equivalent to \(\lambda\) copies of \(\lambda\)
$\texttt{w}\{\langle \text{align} \rangle\}\{\langle \text{width} \rangle\}$	New in array v2.4h, 2018. Here $\langle \text{align} \rangle$ is one of 1, c or r. The construction corresponds to every cell in the column being formated as $\text{makebox}[\langle \text{width} \rangle][\langle \text{align} \rangle] \{\langle \text{cell} \rangle\}$ . It will silently overprint if the contents are wider than $\langle \text{width} \rangle$ .
$\mathtt{W}\{\langle \mathrm{align} \rangle\} \{\langle \mathrm{width} \rangle\}$	New in in <i>array</i> v2.4h, 2018. Similar to w, but issues an overfull warning if the contents is too wide.
$D\{\langle \operatorname{ssep}\rangle\}\{\langle \operatorname{osep}\rangle\}\{\langle \operatorname{places}\rangle\}$	Column entries aligned on a 'decimal point'

# 11.2 The preamble

You use the  $\langle preamble \rangle$  argument to the array and tabular environments to specify the number of columns and how you want column entries to appear. The preamble consists of a sequence of options, which are listed in Table 11.1.

Examples of the options include:

• A flush left column with bold font can be specified with >{\bfseries}1.

\end{center}

• In columns which have been generated with p, m or b, the default value of \parindent is 0pt.

The \parindent for a particular column can be changed with, for example

>{\setlength{\parindent}{1cm}}p

```
\begin{center}
\begin{tabular}{>\setlength{\parindent}{5mm}}p{2cm} p{2cm}} \toprule
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 &
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 \\ \bottomrule
\end{tabular}
\end{center}
```

123456	12345678
78901234	90123456
567890	7890

- The specification >{\$}c<{\$} generates a column in math mode in a tabular environment. When used in an array environment the column is in LR mode (because the additional \$'s cancel the existing \$'s).
- Using c!{\hspace{1cm}}c you get space between two columns which is enlarged by one centimeter, while c@\hspace{1cm}}c gives you exactly one centimeter space between two columns.
- Elsewhere reasons are given why you should not use vertical lines (e.g., the | option) in tables. Any examples that use vertical lines are for illustrative purposes only where it is advantageous to denote column boundaries, for example to show different spacing effects.

# 11.2.1 D column specifiers

## Recommended alternative

As an alternative to the D column (through using the dcolumn package), you can use the siunitx package which have the added bonus of many more configuration and formatting features. See [Wri18] for details.

In financial tables dealing with pounds and pence or dollars and cents, column entries should be aligned on the separator between the numbers. The D column specifier is provided for columns which are to be aligned on a 'decimal point'. The specifier takes three arguments.

```
D\{\langle ssep \rangle\}\{\langle osep \rangle\}\{\langle places \rangle\}
```

- $\langle ssep \rangle$  is the single character which is used as the separator in the source .tex file. Thus it will usually be '.' or ','.
- $\langle osep \rangle$  is the separator in the output, this may be the same as the first argument, but may be any math-mode expression, such as  $\cdot$ . A D column always uses math mode for the digits as well as the separator.
- ⟨places⟩ should be the maximum number of decimal places in the column (but see below for more on this). If this is negative, any number of decimal places can be used in the column, and all entries will be centred on (the leading edge of) the separator. Note that this can cause a column to be too wide; for instance, compare the first two columns in the example below.

Here are some example specifications which, for convenience, employ the \newcolumntype macro described later.

```
\label{local_prop_d} $$\operatorname{D}_{.}^{D}. $$\cdot}_{\#1}$$
```

This defines d to be a column specifier taking a single argument specifying the number of decimal places, and the .tex file should use '.' as the separator, with  $\cdot$  (·) being used in the output.

```
\newcolumntype{.}{D{.}{.}{.}{-1}}
```

The result of this is that '.' specifies a column of entries to be centered on the '.'.

```
\newcolumntype{,}{D{,}{,}{2}}
```

And the result of this is that ', ' specifies a column of entries with at most two decimal places after a ','.

The following table is typeset from this code:

```
\begin{center}
\begin{tabular}{|d{-1}|d{2}|.|,|}
1.2 & 1.2 &1.2
                     &1,2
1.23 & 1.23 &12.5 &300,2 \\
1121.2& 1121.2&861.20 &674,29 \\
184 & 184 &10
                     &69
      & .4 &
                     &,4
                            //
      &
             &.4
                     &
\end{tabular}
\end{center}
```

1.2	1.2	1.2	1,2
1.23	1.23	12.5	300,2
1121.2	1121.2	861.20	$674,\!29$
184	184	10	69
.4	.4		,4
		.4	

Note that the first column, which had a negative  $\langle places \rangle$  argument is wider than the second column, so that the decimal point appears in the middle of the column.

The third  $\langle places \rangle$  argument may specify *both* the number of digits to the left and to the right of the decimal place. The third column in the next table below is set with D{.}{.}{5.1} and in the second table, D{.}{1.1}, to specify 'five places to the left and one to the right' and 'one place to the left and one to the right' respectively. (You may use ',' or other characters, not necessarily '.' in this argument.) The column of figures is then positioned such that a number with the specified numbers of digits is centred in the column.

Be careful if you have table headings inserted, say, with

```
\multicolumn{1}{c}{\ldots\}.\ldots\}
```

to over-ride the D column type, as the overall result may not look quite as good as you might expect. In the next pair of tabulars the first column is set with

```
D{.}{.}{-1}
```

to produce a column centered on the '.', and the second column is set with

```
D{.}{.}{1}
```

to produce a right aligned column.

```
Source for example 11.1
\begin{center}\small
\begin{tabular}[t]{|D..{-1}|D..{1}|D..{5.1}|}
\multicolumn{1}{|c|}{head} &
\mathcal{L}_{c}\
\multicolumn{1}{c|}{head}
                            \\[3pt]
1
       & 2
            & 3
                            11
       & 1.2
                & 1.2
1.2
                            //
11212.2 & 11212.2 & 11212.2 \\
       & .4
                 & .4
\end{tabular}
\hfill
\begin{tabular}[t]{|D..{-1}|D..{1}|D..{1.1}|}
\multicolumn{1}{|c|}{wide heading} &
\multicolumn{1}{c|}{wide heading} &
\multicolumn{1}{c|}{wide heading} \\[3pt]
1
       & 2
                & 3
                                 11
1.2
       & 1.2
                 & 1.2
                                 11
```

Typeset example 11.1: Tabular with narrow and wide headings

head	head	head	wide heading	wide heading	wide heading	
1	2	3	1	2	3	
1.2	1.2	1.2	1.2	1.2	1.2	
11212.2	11212.2	11212.2	.4	.4	.4	
.4	.4	.4				

```
.4 & .4 & .4 \\end{tabular} \\end{center}
```

In both of these tables the first column is set with  $D\{.\}\{.\}\{-1\}$  to produce a column centered on the '.', and the second column is set with  $D\{.\}\{.\}\{1\}$  to produce a right aligned column.

The centered (first) column produces columns that are wider than necessary to fit in the numbers under a heading as it has to ensure that the decimal point is centered. The right aligned (second) column does not have this drawback, but under a wide heading a column of small right aligned figures is somewhat disconcerting.

The notation for the  $\langle places \rangle$  argument also enables columns that are centred on the mid-point of the separator, rather than its leading edge; for example

```
D{+}{\,\pm\,}{3,3}
```

will give a symmetric layout of up to three digits on either side of a  $\pm$  sign.

11.2.2 Defining new column specifiers

You can easily type

```
>{\some declarations\}{c}<{\some more declarations\}}
```

when you have a one-off column in a table, but it gets tedious if you often use columns of this form. The \newcolumntype lets you define a new column option like, say

\newcolumntype{x}{\some declarations\}}{c}<{\some more declarations\}}

and you can then use the  $\boldsymbol{x}$  column specifier in the preamble wherever you want a column of this kind.

The  $\langle char \rangle$  argument is the character that identifies the option and  $\langle spec \rangle$  is its specification in terms of the regular preamble options. The \newcolumntype command is similar to \newcommand —  $\langle spec \rangle$  itself can take arguments with the optional  $\langle nargs \rangle$  argument declaring their number.

For example, it is commonly required to have both math-mode and text columns in the same alignment. Defining:

```
\newcolumntype{C}{>{$}c<{$}}
\newcolumntype{L}{>{$}1<{$}}</pre>
```

Then C can be used to get centred text in an array, or centred math-mode in a tabular. Similarly L and R are for left- and right-aligned columns.

The  $\langle spec \rangle$  in a \newcolumntype command may refer to any of the primitive column specifiers (see table 11.1 on page 206), or to any new letters defined in other \newcolumntype commands.

```
\showcols
```

A list of all the currently active \newcolumntype definitions is sent to the terminal and log file if the \showcols command is given.

# 11.2.3 Surprises

- A preamble of the form {wx\*{0}{abc}yz} is treated as {wxyz}
- An incorrect positional argument, such as [Q], is treated as [t].
- A preamble such as {cc\*{2}} with an error in a \*-form will generate an error message that is not particularly helpful.
- Error messages generated when parsing the column specification refer to the preamble argument *after* it has been re-written by the \newcolumntype system, not to the preamble entered by the user.
- Repeated < or > constructions are allowed.  $>{\langle decs1\rangle}>{\langle decs2\rangle}$  is treated the same as  $>{\langle decs2\rangle}{\langle decs1\rangle}$ .
  - The treatment of multiple < or > declarations may seem strange. Using the obvious ordering of >{ $\langle decs1 \rangle$ }{ $\langle decs2 \rangle$ } has the disadvantage of not allowing the settings of a \newcolumntype defined using these declarations to be overriden.
- The \extracolsep command may be used in @-expressions as in standard LaTeX, and also in !-expressions.

The use of \extracolsep is subject to the following two restrictions. There must be at most one \extracolsep command per 0, or ! expression and the command must be directly entered into the 0 expression, not as part of a macro definition. Thus

does not work. However you can use something like

\newcolumntype{e}{@{\extracolsep{\fill}}

instead.

- As noted by the LaTeX book [Lam94], a \multicolumn, with the exception of the first column, consists of the entry and the *following* inter-column material. This means that in a tabular with the preamble |1|1|1|1| input such as \multicolumn{2}{|c|} in anything other than the first column is incorrect.
  - In the standard array/tabular implementation this error is not noticeable as a | takes no horizontal space. But in the class the vertical lines take up their natural width and you will see two lines if two are specified another reason to avoid using |.

# 11.3 The array environment

Mathematical arrays are usually produced using the array environment.

```
\[\begin{array}[\langle pos \rangle]{\langle preamble \rangle} rows \end{array}\] \[\begin{array}[\langle pos \rangle]\langle left \rangle{\langle preamble \rangle}\langle right \rangle rows \end{array}\]
```

Math formula are usually centered in the columns, but a column of numbers often looks best flush right, or aligned on some distinctive feature. In the latter case the D column scheme is very handy.

```
\[ \begin{array}{lcr} a +b +c & d - e - f & 123 \\ g-h & j k & 45 \\ 1 & m & 6 \end{array} \] a+b+c
```

$$egin{array}{cccc} a+b+c & d-e-f & 123 \\ g-h & jk & 45 \\ l & m & 6 \\ \end{array}$$

Arrays are often enclosed in brackets or vertical lines or brackets or other symbols to denote math constructs like matrices. The delimeters are often large and have to be indicated using \left and \right commands.

$$\left[\begin{array}{cc} x_1 & x_2 \\ x_3 & x_4 \end{array}\right]$$

The class's array environment is an extension of the standard environment in that it provides a system of implicit \left \right pairs. If you want an array surrounded by parentheses, you can enter:

\[ \begin{array}({cc})a&b\\c&d\end{array} \]

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Or, a litle more complex

```
\[ \begin{array}({c})
    \begin{array}|{cc}|
        x_{1} & x_{2} \\
        x_{3} & x_{4}
    \end{array} \\
        y \\
        z
    \end{array} \]
```

$$\left(\begin{array}{c|c} x_1 & x_2 \\ x_3 & x_4 \\ y \\ z \end{array}\right)$$

And you can do things like this:

$$a = \left| \begin{array}{ccc} x - \lambda & 1 & 0 \\ 0 & x - \lambda & 1 \\ 0 & & x - \lambda \end{array} \right|^2$$

As another example, a construct equivalent to plain TeX's \cases could be defined by:

$$f(x) = \begin{cases} 0 & \text{if } x = 0\\ \sin(x)/x & \text{otherwise} \end{cases}$$

Here L denotes a column of left aligned L-R text, as described earlier. Note that as the delimiters must always be used in pairs, the '.' must be used to denote a 'null delimiter'.

This feature is especially useful if the [t] or [b] arguments are also used. In these cases the result is not equivalent to surrounding the environment by \left…\right, as can be seen from the following example:

```
\begin{array}[t]({c}) 1\\2\\3 \end{array}
\begin{array}[c]({c}) 1\\2\\3 \end{array}
\begin{array}[b]({c}) 1\\2\\3 \end{array}
\quad\mbox{not}\quad
\left(\begin{array}[t]{c} 1\\2\\3 \end{array}\right)
\left(\begin{array}[c]{c} 1\\2\\3 \end{array}\right)
\left(\begin{array}[b]{c} 1\\2\\3 \end{array}\right)
```

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad \text{not} \quad \begin{pmatrix} 1 \\ 1 \\ 2 \\ 3 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

		spann	spanner		
stub	head subhead	head subhead	head	head	
A	a	b	С	d	
В	e	f	g	h	
$\mathbf{C}$	i	j	k	1	
D	$\mathbf{m}$	$\mathbf{n}$	O	p	

Table 11.2: Demonstrating the parts of a table

## 11.4 Tables

A table is one way of presenting a large amount of information in a limited space. Even a simple table can presents facts that could require several wordy paragraphs — it is not only a picture that is worth a thousand words.

A table should have at least two columns, otherwise it is really a list, and many times has more. The leftmost column is often called the *stub* and it typically contains a vertical listing of the information categories in the other columns. The columns have *heads* (or *headings*) at the top indicating the nature of the entries in the column, although it is not always necessary to provide a head for the stub if the heading is obvious from the table's caption. Column heads may include subheadings, often to specify the unit of measurement for numeric data.

A less simple table may need two or more levels of headings, in which case *decked heads* are used. A decked head consists of a *spanner head* and the two or more column heads it applies to. A horizontal *spanner rule* is set between the spanner and column heads to show which columns belong to the spanner.

Double decking, and certainly triple decking or more, should be avoided as it can make it difficult following them down the table. It may be possible to use a *cut-in head* instead of double decking. A cut-in head is one that cuts across the columns of the table and applies to all the matter below it. To try and clarify, the parts of a table are shown diagrammatically in Table 11.2.

No mention has been made of vertical rules in a table, and this is deliberate. There should be no vertical rules in a table. Rules, if used at all, should be horizontal only, and these should be single, not double or triple. It's not that ink is expensive, or that practically no typesetting is done by hand any more, it is simply that the visual clutter should be eliminated.

For example, in Table 11.3 which was produced from the code below, Table 11.3(a) is from the LaTeX book and Table 11.3(b) is how Simon Fear [Fea16] suggests it should be cleaned up. Notice how both the revised code and the table are generally simpler than the originals.

```
\begin{table}
\centering
\caption{Two views of one table} \label{tab:twoviews}
```

Table 11.3: Two views of one table

#### (a) Before

gnats	gram	\$13.65
	each	.01
gnu	stuffed	92.50
emu		33.33
armadillo	frozen	8.99

(b) After

I	_	
Animal	Description	Price (\$)
Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.50
$\operatorname{Emu}$	stuffed	33.33
Armadillo	frozen	8.99

```
\subtop[Before]{\label{tab:before}%
 \begin{tabular}{||1|1r||} \hline
gnats
         & gram
                 & \$13.65 \\ \cline{2-3}
         & each
                   & .01 \\ \hline
         & stuffed & 92.50 \\ \cline{1-1} \cline{3-3}
gnu
                   & 33.33 \\ \hline
 emu
armadillo & frozen & 8.99 \\ \hline
\end{tabular}}
\hfill
\subtop[After]{\label{tab:after}%
\begin{tabular}{@{}llr@{}} \toprule
Animal & Description & Price (\$)\\ \midrule
Gnat
     & per gram & 13.65 \\
      & each & 0.01 \\
      & stuffed & 92.50 \\
Gnu
      Armadillo & frozen & 8.99 \\ \bottomrule
\end{tabular}
}
\end{table}
```

Columns of numbers often end with a line giving the total or result. A horizontal rule may be drawn to separate the result from the rest but a small amount of white space may do just as well, as in Table 11.4.

Some other points are:

- Put the units in the column heading (not in the body of the table).
- Always precede a decimal point by a digit; thus 0.1 *not* just .1.
- Do not use 'ditto' signs or any other such convention to repeat a previous value. In many circumstances a blank will serve just as well. If it won't, then repeat the value.

Not every table requires all the elements mentioned above. For instance, in Charles Dicken's *David Copperfield* (1849–1850) Mr Wilkins Micawber states:

Table 11.4: Micawber's law

Income	£20-0-0	£20-0-0
Expenditure	£19-0-6	£20-0-6
Result	happiness	misery

Table 11.5: A narrow table split half and half

Relative contents of odd isotopes for heavy elements

Element	Z	$\gamma$	Element	Z	$\gamma$
Sm	62	1.480	$\overline{\mathrm{w}}$	74	0.505
$\operatorname{Gd}$	64	0.691	Os	76	0.811
Dy	66	0.930	$\operatorname{Pt}$	78	1.160

'Annual income twenty pounds, annual expenditure nineteen six, result hapiness. Annual income twenty pounds, annual expenditure twenty pounds ought and six, result misery.'

This can also be represented in tabular form<sup>3</sup> as in Table 11.4.

Long narrow tables do not look well on the page. In such cases the table could be set half and half instead, as in Table 11.5.

# 11.5 Fear's rules

Simon Fear disapproves of the default LaTeX table rules and wrote the booktabs package [Fea16] to provide better horizontal rules. Like many typographers, he abhors vertical rules. The following is taken almost verbatim from the booktabs package.

In the simplest of cases a table begins with a top rule, has a single row of column headings, then a dividing rule, and after the columns of data it is finished off with a final rule. The top and bottom rules are normally set heavier (i.e., thicker or darker) than any intermediate rules.

```
\label{eq:continuity} $$ \operatorname{[\langle width\rangle] \wedge heavyrulewidth} $$ \aboverulesep \belowrulesep \end{tabular} $$ $$ \aboverulesep \end{tabular} $$ \ab
```

All the rule commands here go immediately after the closing  $\$  of the preceding row (except of course  $\$ toprule, which comes right after the start of the environment). Each rule has an optional length argument,  $\langle width \rangle$ , which you can use to locally change the default width of the rule.

\toprule draws a rule (with a default width of \heavyrulewidth), and \belowrulesep vertical space inserted below it.

 $\label{lightrulewidth} \ with \ \ aboverule sep space above it and \ \ below rule sep below it.$ 

<sup>&</sup>lt;sup>3</sup>But putting Josh Billings' (Henry Wheeler Shaw) corollary: 'Live within your income, even if you have to borrow to do it.' into tabular form would not work.

\bottomrule draws a rule with a default width of \heavyrulewidth. There is \aboverulesep space above it and \belowrulesep space below it.

If a rule immediately follows another the space between them is \doublerulesep, but as you are not going to use double rules you won't be concerned about this.

```
\label{lem:cmidrule} $$ \operatorname{cmidrule}(\langle width \rangle) (\langle m-n \rangle) $$ $$ \operatorname{cmidrulewidth}(\langle m-n \rangle) $$
```

Spanner rules do not extend the full width of the table, and the \cmidrule is provided for that purpose. This draws a rule, default thickness \cmidrulewidth, across columns  $\langle m \rangle$  to  $\langle n \rangle$  inclusive (where  $\langle m \rangle$  and  $\langle n \rangle$  are column numbers, with  $\langle m \rangle$  not greater than  $\langle n \rangle$ ).

Generally, this rule should not come to the full width of the end columns, and this is especially the case if you have to begin a \cmidrule straight after the end of another one. You can use the optional trimming argument commands, which are (r), (1) and (r1) or (lr), indicated whether the right and/or left ends of the rule should be trimmed. Note the exceptional use of parentheses instead of brackets for this optional argument.

\cmidrule draws a rule from column m to n with a default thickness of \cmidrulewidth. Adjacent \cmidrules, for example

```
\dots \ \cmidrule{2-3}\cmidrule{5-7}
```

have the same vertical alignment. It is best not to leave any space between these commands. The space above and below is normally \aboverulesep and \belowrulesep.

If you are forced into having double spanner rules then you will reluctantly have to insert the command \morecmidrules between the commands for the upper and lower rules. For example:

will draw double rules across columns 2 and 3. You must finish off the rules for one row before starting the lower set of rules. There must not be any space surrounding the \morecmidrules macro. The upper and lower rules are separated by \cmidrulesep.

Occasionally extra space between certain rows of a table may be helpful; for example, before the last row if this is a total. This is simply a matter of inserting  $\addlinespace$  after the  $\addlinespace$  as being a white rule of width  $\addlinespace$  as being a white rule of width  $\addlinespace$  which gives rather less than a whole line space. If another rule follows the amount of whitespace is increased by  $\addlinespace$ .

```
\specialrule{\langle width \rangle} {\langle abovespace \rangle} {\langle belowspace \rangle}
```

You can, but should not, generate peculiar spacing between rules by using \specialrule. The three required arguments are the width,  $\langle width \rangle$ , of the rule and the spaces above  $(\langle abovespace \rangle)$  and below  $(\langle belowspace \rangle)$ . \specialrule ignores a preceding rule but if there is a following one then  $\langle belowspace \rangle$  will be increased by \doublerulesep.

The default dimensions are

```
\heavyrulewidth = 0.08em
\lightrulewidth = 0.05em
\cmidrulewidth = 0.03em
\belowrulesep = 0.65ex
\aboverulesep = 0.4ex
```

```
\defaultaddspace = 0.5em \cite{cmidrulekern} = 0.25em
```

The last of these, \cmidrulekern, is the amount by which a \cmidrule is trimmed at the ends indicated in the () option. In the construction

```
\cmidrule(r){1-2}\cmidrule(1){3-4}
```

there is a total of 0.5em separating the two rules. Currently the only way to get special effects is to reset \cmidrulekern as appropriate; the amount of trimming is not available as an argument in the current implementation of \cmidrule.

An example of the commands in use was given by the code to produce Table 11.3(b) on page 215:

## 11.5.1 Fills

The rules described previously go between rows. Sometimes it may be desirable to to put a rule or something similar within a row.

```
\downbracefill \hrulefill \upbracefill
```

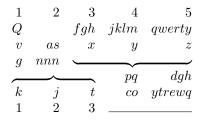
Examples of \downbracefill, \hrulefill, and \upbracefill are illustrated in Table 11.6, typeset from the code below. Surprisingly these are ordinary text commands, not math mode commands.

```
\begin{table}
\centering
\caption{Example table with fills} \label{tab:fills}
\begin{tabular}{rrrrr}
1 & 2 & 3 & 4
                    & 5 \\
       & fgh & jklm & qwerty \\
                           z \\
v & as & x & y &
g & nnn & \multicolumn{3}{c}{\upbracefill} \\
\multicolumn{3}{c}{\downbracefill} & pq & dgh \\
k & j
             & co
                    & ytrewq \\
       & t
1 & 2
             & \multicolumn{2}{c}{\hrulefill}
       & 3
\end{tabular}
\end{table}
```

Here are the same fills, but this time in an array environment. are shown afterwards. Notice the \$s in the array surrounding the fills. Normally \$...\$ is used to typeset a small amount of math mode material in the middle of text. In this case, as the array is already in math mode the \$...\$ are used to typeset a small amount of text material within math mode.

Table 11.6: Example table with fills

```
\begin{displaymath}
 \begin{array}{rrrrr}
 1 & 2 & 3
                                                                                                  & 4
                                                                                                                                                                & 5 \\
                                                             & fgh & jklm & qwerty \\
v & as & x
                                                                                                &
                                                                                                                                                у &
                                                                                                                                                                                                                     z \\
g & nnn & \multicolumn{3}{c}{$\upbracefill$} \\
 \multicolumn{3}{c}{$\downbracefill$} & pq & dgh \\
                                                             & t
                                                                                                          & co
                                                                                                                                                                & ytrewq \\
 1 & 2
                                                                                                          & \mbox{\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{}\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{
                                                            & 3
 \end{array}
\end{displaymath}
```



You can define your own 'fill'. For example:

\newcommand\*{\upbracketfill}{%
 \vrule height 4pt depth 0pt\hrulefill%
 \vrule height 4pt depth 0pt}

is a fill that has the appearance of a (horizontal) bracket. It can be used like this:

\begin{displaymath}
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
a & \multicolumn{2}{c}{\upbracketfill} & d \\
A & B & C & D
\end{array}
\end{displaymath}

Multicolumn entry! THRI		THREE	FOUR
one	The width of	three	Column four
	this column is		will act in the
	fixed $(5.5pc)$ .		same way as
			column two,
			with the same
			width.

Figure 11.1: Example of a regular tabular

## 11.6 Tabular environments

A table created using the tabular environment comes out as wide as it has to be to accommodate the entries. On the other hand, both the tabular\* and tabular\* environments let you specify the overall width of the table via the additional  $\langle width \rangle$  atrgument.

The tabular\* environment makes any necessary adjustment by altering the intercolumn spaces while the tabularx environment alters the column widths. Those columns that can be adjusted are noted by using the letter X as the column specifier in the  $\langle format \rangle$ . Once the correct column widths have been calculated the X columns are converted to p columns.

# 11.6.1 Examples

The following code is used for a regular tabular.

```
\begin{figure}
\centering
\caption{Example of a regular \texttt{tabular}}
\begin{tabular}{|c|p{5.5pc}|c|p{5.5pc}|} \hline
\multicolumn{2}{|c|}{Multicolumn entry!} & THREE & FOUR \\ hline
one &
\raggedright\arraybackslash The width of this column is fixed
(5.5pc). & three &
\raggedright\arraybackslash Column four will act in the same
way as column two, with the same width.\\
\hline
\end{tabular}
\end{figure}
```

The following examples use virtually the same contents, the major differences are the specifications of the environment.

Note that the horizontal rules extend beyond the last column. There are no X columns and the total width required to set the tabular\* is less than the 250pt specified for the width.

Compare the previous narrow tabular\* with the next one which is set with \begin{tabular\*}{300pt}%

Figure 11.2: Example tabularx and tabular\* with widths of 250pt

\begin{tabularx}{250pt}{|c|X|c|X|}

	, , , , , , , , , , , , , , , , , , , ,		
Multicolumn entry!		THREE	FOUR
one The width of		three	Column four will
	this column		act in the same
	depends on the		way as column
	width of the		two, with the
	table. <sup>4</sup>		same width.

\begin{tabular\*}{250pt}{|c|p{5.5pc}|c|p{5.5pc}|}

Multicolumn entry!		THREE	FOUR
one	The width of	three	Column four
	this column is		will act in the
	fixed $(5.5pc)$ .		same way as
			column two,
			with the same
			width.

Figure 11.3: Example tabularx and tabular\* with widths of 300pt

\begin{tabularx}{300pt}{|c|X|c|X|}

Multicolumn entry!		THREE	FOUR
one	The width of this	three	Column four will act
	column depends on		in the same way as
	the width of the table.		column two, with the
			same width.

\begin{tabular\*}{300pt}%
{|@{\extracolsep{\fill}}c|p{5.5pc}|c|p{5.5pc}|}

	(let (excracorpore) (illi) (clb(o.obc) (clb(o.obc)))			
	Multicolumn entry!	THREE	FOUR	
one	The width of	three	Column four	
	this column's		will act in the	
	text is fixed		same way as	
	(5.5pc).		column two,	
			with the same	
			width.	

 $\{ | @{\text{\colsep}\{fill}\}c|p{5.5pc}|c|p{5.5pc}| \} \\$ 

The main differences between the tabularx and tabular\* environments are:

- tabularx modifies the widths of the *columns*, whereas tabular\* modifies the widths of the intercolumn *spaces*.
- tabular and tabular\* environments may be nested with no restriction, however if one tabularx environment occurs inside another, then the inner one *must* be enclosed by { }.

- The body of the tabularx environment is in fact the argument to a command, and so certain constructions which are not allowed in command arguments (like \verb) may not be used.<sup>5</sup>
- tabular\* uses a primitive capability of TeX to modify the inter column space of an alignment. tabularx has to set the table several times as it searches for the best column widths, and is therefore much slower. Also the fact that the body is expanded several times may break certain TeX constructs.

```
\tracingtabularx
```

Following the \tracingtabularx declaration all later tabularx environments will print information about column widths as they repeatedly re-set the tables to find the correct widths.

By default the X specification is turned into  $p{\langle some\ value \rangle}$ . Such narrow columns often require a special format, which can be achieved by using the > syntax. For example, >{\small}X. Another format which is useful in narrow columns is raggedright, however LaTeX's \raggedright macro redefines \\ in a way which conflicts with its use in tabular or array environments.

```
\arraybackslash
```

For this reason the command \arraybackslash is provided; this may be used after a \raggedright, \raggedleft or \centering declaration. Thus a tabularx format may include

>{\raggedright\arraybackslash}X

These format specifications may of course be saved using the command, \newcolumntype. After specifying, say,

\newcolumntype{Y}{>{\small\raggedright\arraybackslash}X}

then Y could be used in the tabularx format argument.

```
\tabularxcolumn
```

The X columns are set using the p column, which corresponds to \parbox[t]. You may want them set using, say, the m column, which corresponds to \parbox[c]. It is not possible to change the column type using the > syntax, so another system is provided. \tabularxcolumn should be defined to be a macro with one argument, which expands to the tabular format specification that you want to correspond to X. The argument will be replaced by the calculated width of a column.

The default definition is

\newcommand{\tabularxcolumn}[1]{p{#1}}

This may be changed, for instance

\renewcommand{\tabularxcolumn}[1]{>{\small}m{#1}}

so that X columns will be typeset as m columns using the \small font.

Normally all X columns in a single table are set to the same width, however it is possible to make tabularx set them to different widths. A format argument of

<sup>&</sup>lt;sup>5</sup>Actually, \verb and \verb\* may be used, but they may treat spaces incorrectly, and the argument can not contain an unmatched { or }, or a % character.

#### {>{\hsize=.5\hsize}X>{\hsize=1.5\hsize}X}

specifies two columns, where the second will be three times as wide as the first. If you think you need to do things like this try and redesign your table. However, if you must you should follow these two rules.

- Make sure that the sum of the widths of all the X columns is unchanged. (In the above example, the new widths still add up to twice the default width, the same as two standard X columns.)
- Do not use \multicolumn entries which cross any X column.

tabularx will not set X columns to a negative width. If the widths of the 'normal' columns of the table already total more than the requested total width you will get the warning 'X columns too narrow (table too wide)'. The X columns will be set to a width of 1em and so the table itself will be wider than the requested total width given in the argument to the environment.

The standard \verb macro does not work inside a tabularx, just as it does not work in the argument to any macro.

```
\TX@verb
```

The 'poor man's \verb' (and \verb\*) defined here is based on page 382 of the *TeXbook*. As explained there, doing verbatim this way means that spaces are not treated correctly, and so \verb\* may well be useless. The mechanism is quite general, and any macro which wants to allow a form of \verb to be used within its argument may

\let\verb=\TX@verb

It must ensure that the real definition is restored afterwards.

This version of \verb and \verb\* are subject to the following restictions:

- 1. Spaces in the argument are not read verbatim, but may be skipped according to TeX's usual rules.
- 2. Spaces will be added to the output after control words, even if they were not present in the input.
- 3. Unless the argument is a single space, any trailing space, whether in the original argument, or added as in (2), will be omitted.
- 4. The argument must not end with \, so \verb|\| is not allowed, however, because of (3), \verb|\| produces \.
- 5. The argument must be balanced with respect to { and }. So \verb|{| is not allowed.
- 6. A comment character like % will not appear verbatim. It will act as usual, commenting out the rest of the input line!
- 7. The combinations ? ' and ! ' will appear as ? ` and ! ` if the Computer Typewriter font is being used.

# 11.7 Spaces and rules

# 11.7.1 Spacing

Sometimes tabular rows appear vertically challenged.

\arraystretch

The macro \arraystretch controls the spacing between rows. The normal space is multiplied by the value of \arraystretch, whose default definition is

```
\newcommand{\arraystretch}{1.0}
```

If this is changed to 1.25, for example, the row spacing is increased by 25%.

```
\extrarowheight
```

If the length \extrarowheight is positive, its value is added to the normal height of every row of the array or table, while the depth will remain the same. This is important for tables with horizontal lines because those lines normally touch the capital letters. For example

```
\begin{table}
\centering
\caption{The array and tabular format options.}%
\label{tab:tabpream}
  \setlength{\extrarowheight}{1pt}
\begin{tabular}{cp{9cm}} \toprule
...
```

was used for Table 11.1.

```
\arraycolsep \tabcolsep
```

The length \arraycolsep is half the width of the horizontal space between columns in an array environment and similarly the length \tabcolsep is half the space between columns in an tabular or tabular\* environment.

```
\arrayrulewidth \doublerulesep
```

The length \arrayrulewidth is the width of the line created by a | in the format, or by an \hline, \cline or \vline command. The length \doublerulesep is the space between lines created by two successive | options in the format or by successive \hline commands.

## 11.7.2 Special variations on horizontal lines

The family of tabular environments allows vertical positioning with respect to the baseline of the text in which the environment appears. By default the environment appears centered, but this can be changed to align with the first or last line in the environment by supplying a t or b value to the optional position argument. However, this does not work when the first or last element in the environment is a \hline command — in that case the environment is aligned at the horizontal rule.

Here is an example:

```
Tables with no
                   versus
                                  Tables
       hline
                                  \begin{tabular}[t]{1}
                                   with no\\ hline\\ commands
       commands
tables
                   used.
                                  \end{tabular} versus tables
       with some
                                  \begin{tabular}[t]{|1|}
       hline
                                   \hline
       commands
                                   with some \\ hline \\ commands \\
                                   \hline
                                  \end{tabular} used.
```

```
\firsthline \lasthline \extratabsurround
```

Using \firsthline and \lasthline will cure the problem, and the tables will align properly as long as their first or last line does not contain extremely large objects.

```
Tables with no
                  versus
                                 Tables
                                 \begin{tabular}[t]{1}
      line
                                   with no\\ line\\ commands
       commands
                                 \end{tabular} versus tables
      with some
tables
                  used.
                                 \begin{tabular}[t]{|1|}
      line
                                  \firsthline
      commands
                                   with some \\ line \\ commands \\
                                  \lasthline
                                 \end{tabular} used.
```

The implementation of these two commands contains an extra dimension, which is called \extratabsurround, to add some additional space at the top and the bottom of such an environment. This is useful if such tables are nested.

# 11.7.3 Handling of rules

There are two possible approaches to the handling of horizontal and vertical rules in tables:

- 1. rules can be placed into the available space without enlarging the table, or
- 2. rules can be placed between columns or rows thereby enlarging the table.

The class implements the second possibility while the default implementation in the LaTeX kernel implements the first concept.

With standard LaTeX adding rules to a table will not affect the width or height of the table (unless double rules are used), e.g., changing a format from 111 to 1|1|1 does not affect the document other than adding rules to the table. In contrast, with the class a table that just fits the \textwidth might now produce an overfull box. (But you shouldn't have vertical rules in the first place.)

# 11.8 Free tabulars

All the tabular environments described so far put the table into a box, which LaTeX treats like a large complex character, and characters are not broken across pages. If you have a long table that runs off the bottom of the page you can turn to, say, the longtable [Car98b] or xtab [Wil00e] packages which will automatically break tables across page boundaries. These have various bells and whistles, such as automatically putting a caption at the top of each page, repeating the column heads, and so forth.

#### 11.8.1 Continuous tabulars

The ctabular environment is similar to tabular, but with a couple of differences, the main one being that the table will merrily continue across page breaks. The  $\langle format \rangle$  argument is the same as for the previous array and tabular environments, but the optional  $\langle pos \rangle$  argument controls the horizontal position of the table, not the vertical. The possible argument value is one of the following characters:

```
| left justified,
    c centered, or
    r right justified;
the default is c.
    \begin{ctabular}{lcr} \toprule
    LEFT & CENTER & RIGHT \\    \midrule
    l & c & r \\
    l & c & r \\
```

LEFT	CENTER	RIGHT
1	С	r
1	С	r
1	С	r
1	С	r

An example use is for setting two texts in parallel, for instance a poem and it's translation, without having to be concerned about page breaks.

Je suis Françoys, dont il me pois, Né de Paris emprès Pointoise, Et de la corde d'une toise Sçaura mon col que mon cul poise.

I am François, which is unfortunate, born in Paris near Pointoise, and with a six-foot stretch of rope, my neck will know my arse's weight. François Villon, 1431–1463?

The ctabular environment will probably not be used within a table environment (which defeats the possibility of the table crossing page boundaries). To caption a ctabular you can define a fixed caption. For example:

 $\verb|\newfixedcaption{\freetabcaption}{table}|$ 

And then \freetabcaption can be used like the normal \caption within a table float.

# 11.8.2 Automatic tabulars

A tabular format may be used just to list things, for example the names of the members of a particular organisation, or the names of LaTeX environments.

Especially when drafting a document, or when the number of entries is likely to change, it is convenient to be able to tabulate a list of items without having to explicitly mark the end of each row.

The \autorows macro lists the  $\langle entries \rangle$  in rows; that is, the entries are typeset left to right and top to bottom. The  $\langle num \rangle$  argument is the number of columns. The  $\langle entries \rangle$  argument is a comma-separated list of the names to be tabulated; there must be no comma after the last of the names before the closing brace. Table 11.7 was set by \autorows using:

Table 11.7: Example automatic row ordered table

one	two	$_{ m three}$	four	five
six	seven	$_{ m eight}$	nine	$_{ m ten}$
eleven	twelve	thirteen	fourteen	

The  $\langle pos \rangle$  argument controls the horizontal position of the tabular and the  $\langle style \rangle$  argument specifies the location of the entries in the columns; each column is treated identically. The value of a  $\langle pos \rangle$  or  $\langle style \rangle$  argument is one of the following characters:

l left justified,c centered, orr right justified.

Each column is normally the same width, which is large enough to accommodate the widest entry in the list. A positive  $\langle width \rangle$  (e.g., {0.8\textwidth}), defines the overall width of the table, and the column width is calculated by dividing  $\langle width \rangle$  by the number of columns. Any negative value for the  $\langle width \rangle$  width lets each column be wide enough for the widest entry in that column; the column width is no longer a constant.

The examples in Figure 11.4 illustrate the effect of the  $\langle width \rangle$  argument (the default value is 0pt). The principal elements of the code for the Figure are:

The \autocols macro lists its  $\langle entries \rangle$  in columns, proceeding top to bottom and left to right. The arguments, are the same as for \autorows, except that a negative  $\langle width \rangle$  is treated as if it were zero. The column width is always constant throughout the table and

```
\langle \text{width} \rangle = -1 \text{pt}
                                             three
                                                           four
                                                                    five
                        one
                                   two
                         six
                                  seven
                                             eight
                                                          nine
                                                                    ten
                       eleven twelve thirteen fourteen
                              \langle \text{width} \rangle = \text{Opt (the default)}
                              two
                                             three
                                                             four
                                                                            five
              one
               six
                             seven
                                             eight
                                                             nine
                                                                             ten
                                                          fourteen
             eleven
                            twelve
                                           thirteen
                               \langle \text{width} \rangle = 0.9 \text{\textwidth}
                                             three
                                                                    four
                                                                                          five
 one
                       two
  six
                      seven
                                             eight
                                                                    nine
                                                                                           ten
                      twelve
                                           thirteen
                                                                 fourteen
eleven
```

Figure 11.4: Changing the width of a row ordered table

```
\langle \text{width} \rangle = \text{Opt (the default)}
                 one, two
                                   five
                                                eight
                                                              eleven
                                                                           thirteen
                   three
                                   six
                                                 nine
                                                              twelve
                                                                           fourteen
                    four
                                  seven
                                                  ten
                                         \langle \text{width} \rangle = 0.9 \text{textwidth}
                         five
                                             eight
                                                                 eleven
                                                                                      thirteen
one, two
  three
                         six
                                             nine
                                                                 twelve
                                                                                     fourteen
   four
                                              ten
                       seven
```

Figure 11.5: Changing the width of a column ordered table

is normally sufficient for the widest entry. A positive or zero  $\langle width \rangle$  has the same effect as for \autorows.

If you need to include a comma within one of the entries in the list for either \autorows or \autocols you have to use a macro. For instance:

\label{fig:acw}
\end{figure}

# Twelve

# Page notes

The standard classes provide the \footnote command for notes at the bottom of the page. The class provides several styles of footnotes and you can also have several series of footnotes for when the material gets complicated. The normal \marginpar command puts notes into the margin, which may float around a little if there are other \marginpars on the page. The class additionally supplies commands for fixed marginal notes and sidebars.

#### 12.1 Footnotes

A footnote can be considered to be a special kind of float that is put at the bottom of a page.

```
\label{local_footnote} $$ \footnote[\langle num\rangle] {\langle text\rangle}$
```

In the main text, the \footnote command puts a marker at the point where it is called, and puts the  $\langle text \rangle$ , preceded by the same mark, at the bottom of the page. If the optional  $\langle num \rangle$  is used then its value is used for the mark, otherwise the footnote counter is stepped and provides the mark's value. The \footnote command should be used in paragraph mode where it puts the note at the bottom of the page, or in a minipage where it puts the note at the end of the minipage. Results are likely to be peculiar if it is used anywhere else (like in a tabular).

```
\label{local_local_local_local_local} $$ \footnotemark[\langle num\rangle] {\langle text\rangle}$
```

You can use \footnotemark to put a marker in the main text; the value is determined just like that for \footnote. Footnote text can be put at the bottom of the page via \footnotetext; if the optional  $\langle num \rangle$  is given it is used as the mark's value, otherwise the value of the footnote counter is used. It may be helpful, but completely untrue, to think of \footnote being defined like:

\newcommand{\footnote}[1]{\footnotemark\footnotetext{#1}}

In any event, you can use a combination of  $\footnotemark$  and  $\footnotemark$  to do footnoting where LaTeX would normally get upset.

```
\footref{\langle label \rangle}
```

On occasions it may be desireable to make more than one reference to the text of a footnote. This can be done by putting a \label in the footnote and then using \footref to refer to the label; this prints the footnote mark. For example:

```
...\footnote{...values for the kerning.\label{fn:kerning}} ...
```

```
... The footnote\footref{fn:kerning} on pref{fn:kerning} ... \\
```

In this manual, the last line above prints:

```
... The footnote<sup>16</sup> on page 101 ...
\multfootsep
```

In the standard classes if two or more footnotes are applied sequentially  $^{1,2}$  then the markers in the text are just run together. The class, like the footmisc [Fai00] and ledmac packages, inserts a separator between the marks. In the class the macro \multfootsep is used as the separator. Its default definition is:

\newcommand\*{\multfootsep}{\textsuperscript{\normalfont,}}

```
\feetabovefloat \feetbelowfloat
```

In the standard classes, footnotes on a page that has a float at the bottom are typeset before the float. I think that this looks peculiar. Following the \feetbelowfloat declaration footnotes will be typeset at the bottom of the page below any bottom floats; they will also be typeset at the bottom of \raggedbottom pages as opposed to being put just after the bottom line of text. The standard positioning is used following the \feetabovefloat declaration, which is the default.

### 12.1.1 A variety of footnotes

```
\label{eq:continuous} $$ \operatorname{verbfootnote}[\langle num \rangle] \{\langle text \rangle\}$$
```

The macro \verbfootnote is like the normal \footnote except that its  $\langle text \rangle$  agument can contain verbatim material. For example, the next two paragraphs are typeset by this code:

```
Below, footnote~\ref{fn1} is a \verb?\footnote? while footnote~\ref{fn2} is a \verb?\verbfootnote?.

The \verb?\verbfootnote? command should appear\footnote{There may be some problems if color is used.\label{fn1}} to give identical results as the normal \verb?\footnote?, but it can include some verbatim text\verbfootnote{The \verb?\footnote? macro, like all other macros except for \verb?\verbfootnote?, can not contain verbatim text in its argument.\label{fn2}} in the \meta{text} argument.
```

Below, footnote 3 is a \footnote while footnote 4 is a \verbfootnote.

The \verbfootnote command should appear<sup>3</sup> to give identical results as the normal \footnote, but it can include some verbatim text<sup>4</sup> in the  $\langle text \rangle$  argument.

 $<sup>^1{\</sup>rm One}$  footnote

<sup>&</sup>lt;sup>2</sup>Immediately followed by another

 $<sup>^3</sup>$ There may be some problems if color is used.

 $<sup>^4</sup>$ The \footnote macro, like all other macros except for \verbfootnote, can not contain verbatim text in its argument.

```
\plainfootnotes
\twocolumnfootnotes
\threecolumnfootnotes
\paragraphfootnotes
```

Normally, each footnote starts a new paragraph. The class provides three other styles, making four in all. Following the \twocolumnfootnotes declaration footnotes will be typeset in two columns, and similarly they are typeset in three columns after the \threecolumnfootnotes declaration. Footnotes are run together as a single paragraph after the \paragraphfootnotes declaration. The default style is used after the \plainfootnotes declaration.

The style can be changed at any time but there may be odd effects if the change is made in the middle of a page when there are footnotes before and after the declaration. You may find it interesting to try changing styles in an article type document that uses \maketitle and \thanks, and some footnotes on the page with the title:

```
\title{...\thanks{...}}
\author{...\thanks{...}...}
...
\begin{document}
\paragraphfootnotes
\maketitle
\plainfootnotes
...
\footfudgefiddle
```

Paragraphed footnotes may overflow the bottom of a page. TeX has to estimate the amount of space that the paragraph will require once all the footnotes are assembled into it. It then chops off the main text to leave the requisite space at the bottom of the page, following which it assembles and typesets the paragraph. If it underestimated the size then the footnotes will run down the page too far. If this happens then you can change \footfudgefiddle to make TeX be more generous in its estimation. The default is 64 and a value about 10% higher should fix most overruns.

```
\renewcommand*{\footfudgefiddle}{70}
```

You must use an integer in the redefinition as the command is going to be used in a place where TeX expects an integer.

```
\newfootnoteseries{\(\series\)}
\plainfootstyle{\(\series\)}
\twocolumnfootstyle{\(\series\)}
\threecolumnfootstyle{\(\series\)}
\paragraphfootstyle{\(\series\)}
```

If you need further series you can create you own. A new footnote series is created by the  $\mbox{newfootseries}$  macro, where  $\langle series \rangle$  is an alphabetic identifier for the series. This is most conveniently a single (upper case) letter, for example P.

Calling, say,  $\ensuremath{\texttt{Newfootnoteseries}}\Q$  creates a set of macros equivalent to those for the normal  $\ensuremath{\texttt{footnote}}\$  but with the  $\ensuremath{\texttt{series}}\$  appended. These include  $\ensuremath{\texttt{footnote}}\$ Q,

By default, a series is set to typeset using the normal style of a paragraph per note. The series' style can be changed by using one of the \...footstyle commands.

For example, to have a  $\dot{P}$  (for paragraph) series using roman numerals as markers which, in the main text are superscript with a closing parenthesis and at the foot are on the baseline followed by an endash, and the text is set in italics at the normal footnote size:

The \newfootnoteseries macro does not create series versions of the footnote-related length commands, such as \footmarkwidth and \footmarksep, nor does it create versions of \footnoterule.

At the foot of the page footnotes are grouped according to their series; all ordinary footnotes are typeset, then all the first series footnotes (if any), then the second series, and so on. The ordering corresponds to the order of \newfootnoteseries commands.

If you can't specify a particular footnote style using the class facilities the footmisc package [Fai00] provides a range of styles. A variety of styles also comes with the ledmac package [Wil03b] which additionally provides several classes of footnotes that can be mixed on a page.

```
12.1.2 Styling
```

The parameters controlling the vertical spacing of footnotes are illustrated in Figure 12.1.

There is a discussion in §4.2 starting on page 67 about how to style the \thanks command; footnotes can be similarly styled.

The \footnote macro (and its relations) essentially does three things:

- Typesets a marker at the point where \footnote is called;
- Typesets a marker at the bottom of the page on which \footnote is called;
- Following the marker at the bottom of the page, typesets the text of the footnote.

```
\@makefnmark
\@thefnmark
```

The \footnote macro calls the kernel command \@makefnmark to typeset the footnote marker at the point where \footnote is called (the value of the marker is kept in the macro \@thefnmark which is defined by the \footnote or \footnotemark macros). The default definition typesets the mark as a superscript and is effectively

```
\newcommand*{\@makefnmark}{\hbox{\textsuperscript{\@thefnmark}}}
```



Figure 12.1: Footnote layout parameters

You can change this if, for example, you wanted the marks to be in parentheses at the baseline.

```
\renewcommand*{\@makefnmark}{{\footnotesize (\@thefnmark)}}
```

or, somewhat better to take account of the size of the surrounding text

\renewcommand\*{\@makefnmark}{\slashfracstyle{(\@thefnmark)}}

```
\label{local_continuity} $$ \footmarkstyle{\langle arg \rangle}$
```

The class macro for typesetting the marker at the foot of the page is \footfootmark. The appearance of the mark is controlled by \footmarkstyle. The default specification is

```
\footmarkstyle{\textsuperscript{#1}}
```

where the #1 indicates the position of \@thefnmark in the style. The default results in the mark being set as a superscript. For example, to have the marker set on the baseline and followed by a right parenthesis, do

\footmarkstyle{#1) }

```
\footmarkwidth \footmarksep \footparindent
```

The mark is typeset in a box of width \footmarkwidth If this is negative, the mark is outdented into the margin, if zero the mark is flush left, and when positive the mark is indented. The mark is followed by the text of the footnote. Second and later lines of the text are offset by the length \footmarksep from the end of the box. The first line of a paragraph within a footnote is indented by \footparindent. The default values for these lengths are:

Table 12.1: Some footnote text styles

\footmarkwidth	\footmarksep	Comment
1.8em	-1.8em	Flushleft, regular indented paragraph (the default)
$1.8\mathrm{em}$	$0\mathrm{em}$	Indented, block paragraph hung on the mark
$0\mathrm{em}$	$0\mathrm{em}$	Flushleft, block paragraph
-1.8em	$1.8\mathrm{em}$	Block paragraph, flushleft, mark in the
-1sp	0em	margin Block paragraph, flushleft, mark in the
		margin but flush against the text

\setlength{\footmarkwidth}{1.8em}
\setlength{\footmarksep}{-\footmarkwidth}
\setlength{\footparindent}{1em}

```
\foottextfont
```

The text in the footnote is typeset using the \footnotextfont font. The default is \footnotesize.

Altogether, the class specifies

\footmarkstyle{\textsuperscript{#1}}
\setlength{\footmarkwidth}{1.8em}
\setlength{\footmarksep}{-1.8em}
\setlength{\footparindent}{1em}

\newcommand{\foottextfont}{\footnotesize}

to replicate the standard footnote layout.

You might like to try the combinations of \footmarkwidth and \footmarksep listed in Table 12.1 to see which you might prefer. Not listed in the Table, to get the marker flushleft and then the text set as a block paragraph you can try:

```
\setlength{\footmarkwidth}{1.8em}
\setlength{\footmarksep}{0em}
\footmarkstyle{#1\hfill}
```

As an example of a rather different scheme, in at least one discipline the footnoted text in the main body has a marker at each end. It is possible to define a macro to do this:

The macro is based on a posting to CTT by Donald Arseneau in November 2003, and is used like this:

Some

\wrapfootnote{disciplines}{For example, Celtic studies.} require double marks in the text.

Some <sup>5</sup> disciplines <sup>5</sup> require double marks in the text.

```
\label{counter} $$ \gnormalfont{\counter}$ \gnormalfont{\counter}$
```

Any footnotes after this point will be set according to:

```
\setlength{\footmarkwidth}{-1.0em}
\setlength{\footmarksep}{-\footmarkwidth}
\footmarkstyle{#1}
```

The \fnsymbol macro typesets the representation of the counter  $\langle counter \rangle$  like a footnote symbol. Internally it uses the kernel \@fnsymbol macro which converts a positive integer  $\langle num \rangle$  to a symbol. If you are not fond of the standard ordering of the footnote symbols, this is the macro to change. Its original definition is:

```
\def\@fnsymbol#1{\ensuremath{%}
  \ifcase#1\or *\or \dagger\or \ddagger\or
  \mathsection\or \mathparagraph\or \|\or **\or
  \dagger\dagger \or \ddagger\ddagger \else\@ctrerr\fi}}
```

This, as shown by  $\Omega = 11, ... \Omega$  produces the series:

\*, †, ‡, 
$$\S$$
,  $\P$ ,  $\|$ , \*\*, ††, and ‡‡.

Robert Bringhurst quotes the following as the traditional ordering (at least up to  $\P$ ):

\*, †, ‡, 
$$\S$$
,  $\|$ ,  $\P$ , \*\*, ††, and ‡‡.

You can obtain this sequence by redefining \Ofnsymbol as:

```
\renewcommand*{\@fnsymbol}[1]{\ensuremath{%}
\ifcase#1\or *\or \dagger\or \ddagger\or
\mathsection\or \|\or \mathparagraph\or **\or \dagger\dagger
\or \ddagger\ddagger \else\@ctrerr\fi}}
```

not forgetting judicious use of \makeatletter and \makeatother if you do this in the preamble. Other authorities or publishers may prefer other sequences and symbols.

To get the footnote reference marks set with symbols use:

```
\renewcommand*{\thefootnote}{\fnsymbol{footnote}}
```

or to use roman numerals instead of the regular arabic numbers:

\renewcommand\*{\thefootnote}{\roman{footnote}}

```
\footnoterule
```

The rule separating footnotes from the main text is specified by \footnoterule:

<sup>&</sup>lt;sup>5--5</sup>For example, Celtic studies.

```
\newcommand*{\footnoterule}{%
  \kern-3pt%
  \hrule width 0.4\columnwidth
  \kern 2.6pt}
```

If you don't want a rule (but you might later), then the easiest method is:

```
\let\oldfootnoterule\footnoterule
\renewcommand*{\footnoterule}{}
```

and if you later want rules you can write:

\let\footnoterule\oldfootnoterule

In Figure 12.1 we see that the footnotes are separated from the text by \skip\footins. We provide a special interface to set this skip:

Added 2015/04/22

```
\strut \ \setfootins{(length for normal)}{(length for minipage)}
```

The default is similar to

\setfootins{\bigskipamount}{\bigskipamount}

Internally \setfootins also sets the skips being used by \twocolumnfootnotes and friends.

#### 12.2 Marginal notes

Some marginalia can also be considered to be kinds of floats. The class provides the standard margin notes via \marginpar. Remember that the width of the margin note, the separation from the text, and the separation from one \marginpar to another is controlled by \setmarginnotes, see Section 2.5 on page 21.

Just as a reminder, the  $\mbox{marginpar}$  macro puts  $\langle text \rangle$  into the margin alongside the type-block — the particular margin depends on the document style and the particular page.

The interface for specifying which margin \marginpar (and friends) write to, have long been quite cluttered, so we have in 2010 adopted a more textual and natural interface. For \marginpar the macro is named \marginparmargin{\langle placement \rangle} \ with possible placements: left, right, outer, and inner. The interpretation of which is explained in Figure 12.2. The default corresponds to \marginparmargin{\langle outer}.

The original convoluted methods of specifying the margin for \marginpar is deprecated, although still supported; if you need to know what they are then you can read all about them in memoir.dtx.

Sometimes LaTeX gets confused near a page break and a note just after a break may get put into the wrong margin (the wrong margin for the current page but the right one if the note fell on the previous page). If this occurs then inserting the \strictpagecheck declaration before any \marginpar command is used will prevent this, at the cost of at least one additional LaTeX run.

\Xmargin{\( \prightarrow\) \text{placement}\) \text{for possible placements: left, right, outer, and inner}

**Two column document** If the note is placed in the first column, to the left, otherwise to the right, irrespective the document being one- or two-side and of the users choices

**One sided document** If user specified left, notes are placed to the left, otherwise to the right.

Two sided document depends on whether a recto or verso page:

**Recto (odd) page** note is placed on the right if the user specified right or outer, otherwise the note is placed on the left.

**Verso (even) page** note is placed on the left if the user specified left or outer, otherwise the note is placed on the right.

Figure 12.2: Interpretation of the arguments to the \mathbb{Xmargin} commands for specifying the side in which to place side note like material. X here equals marginpar, sidepar, sidebar, or sidefoot.

### 12.3 Side notes

The vertical position of margin notes specified via \marginpar is flexible so that adjacent notes are prevented from overlapping.

The \sidepar macro is similar to \marginpar except that it produces side notes that do not float — they may overlap.

The same spacing is used for both \marginpar and \sidepar, namely the lengths \marginparsep and \marginparwidth. See \setmarginnotes, in Section 2.5 on page 21.

The length \sideparvshift can be used to make vertical adjustments to the position of \sidepar notes. By default this is set to a value of 0pt which should align the top of the note with the text line.

The command \sideparfont is used to specify the font used for the \sidepar, default is \normalfont\normalsize.

While \sideparfort holds the font settings for the sidepar, the local adjustment is kept in \sideparform, the default is

```
\newcommand*{\sideparform}{%
  \ifmemtortm\raggedright\else\raggedleft\fi}
```

Which is a special construction the makes the text go flush against the text block on side specified via \sideparmargin. Since the margin par area is usually quite narrow it might be an idea to use a ragged setup which enables hyphenation. This can be achieved by

```
\usepackage{ragged2e}
\newcommand*{\sideparform}{%
\ifmemtortm\RaggedRight\else\RaggedLeft\fi}
```

The macro  $\sideparmargin{\langle placement \rangle \}}$  can be used to specify which margin the side note should go to.  $\langle placement \rangle$  should be one of *left*, *right*, *outer*, or *inner*. Interpretation of which is explained in Figure 12.2. For some now forgotten reason the default corresponds to  $\sideparmargin{\{left\}.}^6$ 

By default the  $\langle right \rangle$  argument is put in the left margin. When the twoside option is used the  $\langle right \rangle$  argument is put into the right margin on the verso (even numbered) pages; however, for these pages the optional  $\langle left \rangle$  argument is used instead if it is present. For two column text the relevent argument is put into the 'outer' margin with respect to the column.

The original convoluted methods of specifying the margin for \sidepar is deprecated, although still supported; if you need to know what they are then you can read all about them in memoir.dtx.

\parnopar

When LaTeX is deciding where to place the side notes it checks whether it is on an odd or even page and sometimes TeX doesn't realise that it has just moved onto the next page. Effectively TeX typesets paragraph by paragraph (including any side notes) and at the end of each paragraph sees if there should have been a page break in the middle of the paragraph. If there was it outputs the first part of the paragraph, inserts the page break, and retains the second part of the paragraph, without retypesetting it, for eventual output at the top of the new page. This means that side notes for any given paragraph are in the same margin, either left or right. A side note at the end of a paragraph may then end up in the wrong margin. The macro \parnopar forces a new paragraph but without appearing to (the first line in the following paragraph follows immediately after the last element in the prior paragraph with no line break). You can use \parnopar to make TeX to do its page break calculation when you want it to, by splitting what appears to be one paragraph into two paragraphs.

Bastiaan Veelo has kindly provided example code for another form of a side note, the code is shown in Sniplet C.3 on page 398.

Bastiaan also noted that it provided an example of using the \foremargin length. If you want to try it out, either put the code in your preamble, or put it into a package (i.e., .sty file) without the \makeat... commands.

#### 12.4 Sidebars

Sidebars are typeset in the margin and usually contain material that is ancilliary to the main text. They may be long and extend for more than one page.<sup>7</sup>

 $\sidebar{\langle text \rangle}$ 

The \sidebar command is like \marginpar in that it sets the  $\langle text \rangle$  in the margin. However, unlike \marginpar the  $\langle text \rangle$  will start near the top of the page, and may continue onto later pages if it is too long to go on a single page. If multiple \sidebar commands are used on a page, the several  $\langle text \rangle$ s are set one after the other.

 $\sidebarmargin{\langle margin \rangle}$ 

<sup>6</sup> As not to change existing documents, we have decided to leave it like that.

<sup>7</sup> Donald Arseneau's help has been invaluable in getting the sidebar code to work.

The macro  $\sidebarmargin{\langle placement \rangle\}}$  can be used to specify which margin the side note should go to.  $\langle placement \rangle$  should be one of *left*, *right*, *outer*, or *inner*. Interpretation of which is explained in Figure 12.2. The default corresponds to  $\sidebarmargin{\{outer\}}$ .

```
\sidebarform
```

The sidebar  $\langle text \rangle$  is typeset using the \sidebarfont, whose initial definition is

```
\newcommand{\sidebarfont}{\normalsize\normalfont}
```

Sidebars are normally narrow so the text is set raggedright to reduce hyphenation problems and stop items in environments like itemize from overflowing. More accurately, the text is set according to \sidebarform which is defined as:

```
\newcommand*{\sidebarform}{%
  \ifmemtortm\raggedright\else\raggedleft\fi}
```

Which is a special construction the makes the text go flush against the text block on side specified via \sideparmargin. Since the margin par area is usually quite narrow it might be an idea to use a ragged setup which enables hyphenation. This can be achieved by

```
\usepackage{ragged2e}
\newcommand*{\sidebarform}{%
\ifmemtortm\RaggedRight\else\RaggedLeft\fi}
```

You may run into problems if the \sidebar command comes near a pagebreak, or if the sidebar text gets typeset alongside main text that has non-uniform line spacing (like around a \section). Further, the contents of sidebars may not be typeset if they are too near to the end of the document.

```
\sidebarwidth \sidebarhsep \sidebarvsep
```

The  $\langle text \rangle$  of a \sidebar is typeset in a column of width \sidebarwidth and there is a horizontal gap of \sidebarhsep between the main text and the sidebar. The length \sidebarvsep is the vertical gap between sidebars that fall on the same page; it also has a role in controlling the start of sidebars with respect to the top of the page.

The length \sidebartopsep controls the vertical position of the top of a sidebar. The default is 0pt which aligns it with the top of the typeblock. The command \setsidebarheight sets the height of sidebars to  $\langle height \rangle$ , without making any allowance for \sidebartopsep. The  $\langle length \rangle$  argument should be equivalent to an integral number of lines. For example:

```
\setsidebarheight{15\onelineskip}
```

The default is the \textheight.

Perhaps you would like sidebars to start two lines below the top of the typeblock but still end at the bottom of the typeblock? If so, and you are using the calc package [TJ05], then the following will do the job:

```
\setlength{\sidebartopskip}{2\onelineskip}
\setsidebarheight{\textheight-\sidebartopskip}
```

The alignment of the text in a sidebar with the main text may not be particularly good and you may wish to do some experimentation (possibly through a combination of \sidebarvsep and \setsidebarheight) to improve matters.

Although you can set the parameters for your sidebars individually it is more efficient to use the \setsidebars command; it *must* be used if you change the font and/or the height.

```
\setsidebars{\langle hsep \rangle}{\langle width \rangle}{\langle vsep \rangle}{\langle topsep \rangle}{\langle font \rangle}{\langle height \rangle}
```

The \setsidebars command can be used to set the sidebar parameters. \sidebarhsep is set to  $\langle hsep \rangle$ , \sidebarwidth is set to  $\langle width \rangle$ , \sidebarvsep is set to  $\langle vsep \rangle$ , \sidebartopsep is set to  $\langle topsep \rangle$ , \sidebarfont is set to  $\langle font \rangle$ , and finally \setsidebarheight is used to set the height to  $\langle height \rangle$ . The default is:

Any, or all, of the arguments can be a \*, in which case the parameter corresponding to that argument is unchanged. Repeating the above example of changing the topskip and the height, assuming that the other defaults are satisfactory except that the width should be 3cm and an italic font should be used:

Changing the marginpar parameters, for example with \setmarginnotes, will not affect the sidebar parameters.

Note that \checkandfixthelayout neither checks nor fixes any of the sidebar parameters. This means, for instance, that if you change the \textheight from its default value and you want sidebars to have the same height then after changing the \textheight you have to call \checkandfixthelayout and then call \setsidebars with the (new) \textheight. For instance:

```
...
\settypeblocksize{40\baselineskip}{5in}{*}
...
\checkandfixthelayout
\setsidebars{...}{...}{...}{\textheight}
```

Unfortunately if a sidebar is on a double column page that either includes a double column float or starts a new chapter then the top of the sidebar comes below the float or the chapter title. I have been unable to eliminate this 'feature'.

## 12.5 Side footnotes

Besides three already mentioned macros for writing in the margin (\marginpar, \sidepar, and \sidebar) memoir also provide a functionality to add side footnotes. Actually two ways: one is to internally make \footnote use \marginpar to write in the margin, the other is to collect all side footnotes bottom up in the margin.

```
\footnotesatfoot
\footnotesinmargin
```

\footnotesatfoot (the default) causes \footnote to place its text at the bottom of the page. By issuing footnotesinmargin \footnote (and friends like \footnotetext)will internally use \marginpar to write the footnote to the page.

### 12.5.1 Bottom aligned side footnotes

Bottom aligned footnotes works just like regular footnotes, just with a separate macro  $\sidefootenote{\langle text \rangle}$ , and here the side footnotes are placed at the bottom of the specified margin (more or like as if one had taken the footnotes from the bottom of the page and moved it to the margin instead). All the major functionality is the same as for the normal  $\sides$  footnote command.

```
\sidefootnote[\langle num\rangle] \{\langle text\rangle\} \\ \sidefootnotemark[\langle num\rangle] \\ \sidefootnotetext[\langle num\rangle] \{\langle text\rangle\} \\
```

By default the regular footnotes and the side footnotes use different counters. If one would like them to use the same counter, issue the following in the preamble:

\letcountercounter{sidefootnote}{footnote}

### 12.5.2 Setting the layout for \sidefootnote

There are several possibilities to change the appearance of the \sidefootnote: Specifying the margin in which the side footnote should go, is done by

```
\sidefootmargin{\langle keyword \rangle}
```

where  $\langle keyword \rangle$  can be *left*, *right*, *outer*, and *inner*, and their meaning is explained in Figure 12.2. The default is *outer*.

```
\sidefoothsep
\sidefootwidth
\sidefootvsep
```

\sidefoothsep is a length controlling the separation from the text to the side footnote column, default \marginparsep. \sidefootwidth is length controlling the width of the side footnote column, default \marginparwidth, and \sidefootvsep is the vertical distance between two side footnotes, default \onelineskip.

```
\label{eq:linear_state} $$ \sidefootheight{\langle height\rangle}$ $$ \sidefootfont
```

\sidefootadjust is a length which specifies the placement of the side footnote column in relation to the bottom of the text block, the default is 0pt. \setsidefootheight sets the maximal height of the side footnote column, default textwidth. Lastly \sidefootfont holds the general font setting for the side footnote, 9 default \normalfont\footnotesize.

The macro

```
\verb|\setsidefeet{|\langle hsep\rangle|} {\langle width\rangle} {\langle vsep\rangle} {\langle adj\rangle} {\langle font\rangle} {\langle height\rangle}
```

sets the specifications all six settings above in one go.. An  $^{\prime*\prime}$  means 'use the current value'. So memoir internally use the following default

<sup>3 \</sup>sidefootnote does not make sense inside minipages...

<sup>9</sup> There is a similar macro to control the font of the text alone.

It is recommended to use this macro along with the other macros in the preamble to specify document layout.

## 12.5.3 Styling \sidefootnote

```
\sidefootmarkstyle{\langle code \rangle}
```

controls how the side footnote counter is typeset in the side footnote. The default is

```
\sidefootmarkstyle{\textsuperscript{#1}}
```

The mark is typeset in a box of width \sidefootmarkwidth If this is negative, the mark is outdented into the margin, if zero the mark is flush left, and when positive the mark is indented. The mark is followed by the text of the footnote. Second and later lines of the text are offset by the length \sidefootmarksep from the end of the box. The first line of a paragraph within a footnote is indented by \sidefootparindent. The default values for these lengths are:

```
\setlength{\sidefootmarkwidth}{0em}
\setlength{\sidefootmarksep}{0em}
\setlength{\sidefootparindent}{1em}
```

Caveat: It is natural to specify a length as \sidefootparindent as a LATEX length, but it has a down side. If, as we do here, set the value to 1em, then since the size of the em unit changes with the current font size, one will actually end up with an indent corresponding to the font size being used when the

```
\setlength{\sidefootparindent}{1em}
```

was issued, not when it has used (where the font size most often will be \footnotesize).

At this point we consider this to be a *feature* not an error. One way to get pass this problem it the following

```
\begingroup% keep font change local
\sidefoottextfont
\global\setlength\sidefootparindent{1em}
\endgroup
```

Then it will store the value of em corresponding to the font being used.

```
\sidefoottextfont
```

holds the font being used by the side footnote, default \normalfont\footnotesize.

```
\sidefootform
```

is used to specify the raggedness of the text. Default

```
\newcommand*{\sidefootform}{\rightskip=\z@ \@plus 2em}
```

which is much like \raggedright but allows some hyphenation. One might consider using

```
\usepackage{ragged2e}
\newcommand*{\sidefootform}{\RaggedRight}
```

Which does something similar.

### 12.5.4 Side footnote example

In the margin you will find the result of the following code:

```
Testing\sidefootnote{This is test} bottom aligned footnotes.\sidefootnote{This is another side footnote, spanning several lines.
```

And several paragraphs}\sidefootnote{And number three}

Testing<sup>1</sup> bottom aligned footnotes.<sup>2,3</sup>

12.6 Endnotes

*Reimplemented, December* 2010<sup>10</sup>

Endnotes are often used instead of footnotes so as not to interrupt the flow of the main text. Although endnotes are normally put at the end of the document, they may instead be put at the end of each chapter.

The endnotes package already uses the command \endnote for an endnote, so the class uses \pagenote for an endnote so as not to clash if you prefer to use the package.

The general principle is that notes are written out to a file which is then input at the place where the notes are to be printed. The note file has an ent extension, like the table of contents file has a toc extension.

You have to put \makepagenote in your preamble if you want endnotes. This will open the ent note file which is called \jobname.ent.

In the body of the text use use \pagenote to create an endnote, just as you would use \footnote to create a footnote. In the books that I have checked there are two common methods of identifying an endnote:

- 1. Like a footnote, put a number in the text at the location of the note and use the same number to identify the note when it finally gets printed.
- 2. Put no mark in the text, but when it is finally printed use a few words from the text to identify the origin of the note. The page number is often used as well with this method.

The  $\langle text \rangle$  argument of \pagenote is the contents of the note and if the optional  $\langle id \rangle$  argument is not used the result is similar to having used \footnote — a number in the main text and the corresponding number in the endnotes listing (as in 1 above). For the second reference style (2 above) use the optional  $\langle id \rangle$  argument for the 'few words', and no mark will be put into the main text but  $\langle id \rangle$  will be used as the identification in the listing.

For one set of endnotes covering the whole document put \printpagenotes where you want them printed, typically before any bibliography or index. The \printpagenotes macro inputs the ent endnote file for printing and then closes it to any further notes.

The former implementation had some difficulties handling certain types of input. A few of the macros used to format the output are no longer supported/used in the new implementation.

 $<sup>^{1}</sup>$ This is test

<sup>&</sup>lt;sup>2</sup>This is another side footnote, spanning several lines.

And several paragraphs

<sup>&</sup>lt;sup>3</sup>And number three

Figure 12.3: Example endnote listing

For notes at the end of each chapter put \printpagenotes\*, which inputs the ent file for printing then empties it ready for more notes, at the end of each chapter.

The simple use is like this:

```
\documentclass[...]{memoir}
 \makepagenote
 \begin{document}
 \chapter{One}
 ...\pagenote{An end note.} ...
 ...\pagenote{Fascinating information.}
 \chapter{Last}% chapter 9
 ...\pagenote{Another note.}% 30th note
 . . .
 \printpagenotes
 \end{document}
This will result in an endnote listing looking like Figure 12.3.
   For notes at the end of each chapter:
 \documentclass[...]{memoir}
 \makepagenote
 \begin{document}
 \chapter{One}
 ...\pagenote{An end note.} ...
```

```
\rintpagenotes*
\chapter{Last}
...\pagenote{Another note.} ...
\rintpagenotes*
%%% no more chapters
...
\end{document}
\continuousnotenums
\notepageref
```

The pagenote counter is used for the notes. By default the endnotes are numbered per chapter. If you want the numbering to be continuous throughout the document use the \continuousnotenums declaration. Normally the information on which page a note was created is discarded but will be made available to notes in the endnote listing following the \notepageref declaration. Both \continuousnotenums and \notepageref can only be used in the preamble.

```
\notesname \notedivision
```

When \printpagenotes (or \printpagenotes\*) is called the first thing it does is call the macro \notedivision. By default this is defined as:

\newcommand\*{\notedivision}{\chapter{\notesname}}

with

```
\newcommand*{\notesname}{Notes}
```

In other words, it will print out a heading for the notes that will be read from the ent file. \print... then closes the ent file for writing and after this \inputs it to get and process the notes.

12.6.1 Changing the appearance

In the text

```
\n
```

The pagenote counter is used for pagenotes. The macro \notenumintext is called by \pagenote with the value of the pagenote counter as the  $\langle num \rangle$  argument to print the value of the pagenote counter in the main text. By default it is printed as a superscript, but this can be changed, or even eliminated.

\newcommand\*{\notenumintext}[1]{\textsuperscript{#1}}

In the page note list

To better understand how a page note entry is formatted in the page note list, we start with the following pseudo code (it is not exactly what you will see in the .ent file, but macros will end up being called in this manner)

```
\label{eq:continuous} $$\operatorname{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\noteinnotes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_{\notes}_
```

At the start and end we have the two macros \prenoteinnotes and \postnoteinnotes, they take care of preparing for and ending an entry in the list. The list is typeset in a manner where each item is (at least) a paragraph, so the default definition is

```
\newcommand{\prenoteinnotes}{\par\noindent}
\newcommand{\postnoteinnotes}{\par}
```

A user could change this to make it look a bit more like a list construction. For example the following would give a hanging indentation

\renewcommand{\prenoteinnotes}{\par\noindent\hangindent 2em}

The \noteidinnotes calls \idtextinnotes to print the note  $\langle id \rangle$  if it was given as the optional argument to pagenote, otherwise it calls \notenuminnotes to print the note number.

```
\label{localization} $$ \operatorname{dinnotes}_{\langle id\rangle} $$ \operatorname{ditextinnotes}_{\langle num\rangle} $$ \notenuminnotes_{\langle num\rangle} $$
```

These are defined respectively as:

```
\newcommand*{\idtextinnotes}[1]{[#1]\space}
\newcommand*{\notenuminnotes}[1]{\normalfont #1.\space}
```

Next we execute  $\pageinnotes{\langle note \ label \ key \rangle}$  which does nothing by default. But if  $\notepageref$  is issued in the preamble two things happen, (1) each page note issues a label such that we can refer back to its page, and (2)  $\pageinnotes$  calls  $\printpageinnotes$  (or if hyperref is loaded  $\printpageinnotes$ hyperref)

```
\pageinnotes{\(\lambda\) uto generated note label key\)}
\printpageinnotes{\(\lambda\) uto generated note label key\)}
\printpageinnoteshyperref{\(\lambda\) uto generated note label key\)}
\pagerefname
```

Default definitions

```
\newcommand*{\printpageinnotes}[1]{%
    (\pagerefname\ \pageref{#1})\space}
\newcommand\printpageinnoteshyperref[1]{%
    (\hyperref[#1]{\pagerefname\ \pageref*{#1}})\space}
```

That is if hyperref is loaded the entire text  $\langle page 3 \rangle$  will be the text of a hyperlink.

```
\prenotetext \postnotetext
```

The actual text part of the page note is enclosed by \prenotetext and postnotetext. By default they do nothing, but could easily be redefined such that (only) the entry text would be in italic:

```
\renewcommand\prenotetext{\begingroup\itshape}
\renewcommand\postnotetext{\endgroup}
```

```
\addtonotes{\langle text \rangle}
```

The macro \addtonotes inserts  $\langle text \rangle$  into the ent file.

**Note.** As the argument to \pagenote and \addtonotes is moving you may have to \protect any fragile commands. If you get strange error messages, try using \protect and see if they go away.

Internally in \pagenote \addtonotes is used to provide chapter devisions into the note list. It will detect both numbered and unnumbered chapters. The actual text is provided using

```
\pagenotesubhead{$\langle chapapp\rangle$} {\langle num\rangle$} {\langle title\rangle$} \\ pagenotesubheadstarred{$\langle chapapp\rangle$} {\langle num\rangle$} {\langle title\rangle$} \\ pnchap \pnschap
```

The macro \pagenotesubhead typesets the subheadings in an endnote list. The  $\langle chapapp \rangle$  argument is normally \chaptername but if the notes are from an appendix then \appendixname is used instead.  $\langle num \rangle$  is the number of the chapter, or blank if there is no number. Lastly,  $\langle title \rangle$  is \pnchap for regular chapters which defaults to the ToC entry, or \pnschap for starred chapters which defaults to the normal title. The default definition of \pagenotesubhead is very simply:

```
\newcommand*{\pagenotesubhead}[3]{%
  \section*{#1 #2 #3}}
\newcommand\pagenotesubheadstarred{\pagenotesubhead} % i.e. the same
```

By default this means that the header for starred chapters will be something like »Chapter Title«, which may look odd. In that case redefine \pagenotesubheadstarred to something similar to

```
\renewcommand\pagenotesubheadstarred[3]{\section*{#3}}
```

Just remember that unless you have specified \continuousnotenums in the preamble the note counter (pagenote) will only be reset at the start of any numbered chapters (because it is tied to changes in the chapter counter).

The scheme is set up under the assumption that notes will only be printed at the end of the document. If you intend to put them at the end of each chapter, then you will probably want to change the definitions of the \notedivision and \pagenotesubhead macros. For example:

```
\renewcommand*{\notedivision}{\section*{\notesname}}
\renewcommand*{\pagenotesubhead}[3]{}
```

and remember to use \printpagenotes\* at each place you want the current set of notes to be printed.

Say you have written a document with footnotes, but later on decide on using end notes (page notes) instead. In that case you can use \footnotepagenote to make \footnote, \footnotemark and \footnotetext works as if it was implemented using end notes. On the other hand \pagetofootnote makes all page notes into footnotes (note that this might not work, because there are places where page notes can be issued but foot notes cannot).

\foottopagenote \pagetofootnote

In either conversion the optional argument will be ignored as for \pagenote it can be arbitrary text whereas for \footnote it must be a number.

# Thirteen

# Decorative text

Too servile a submission to the books and opinions of the ancients has spoiled many an ingenious man, and plagued the world with an abundance of pedants and coxcombs.

James Puckle (1677?–1724)

By now we have covered most aspects of typesetting. As far as the class is concerned this chapter describes the slightly more fun task of typesetting epigraphs.

Some authors like to add an interesting quotation at either the start or end of a chapter. The class provides commands to assist in the typesetting of a single epigraph. Other authors like to add many such quotations and the class provides environments to cater for these as well. Epigraphs can be typeset at either the left, the center or the right of the typeblock. A few example epigraphs are exhibited here, and others can be found in an article by Christina Thiele [Thi99] where she reviewed the epigraph package [Wil00a] which is included in the class.

## 13.1 Epigraphs

The original inspiration for \epigraph was Doug Schenck's for the epigraphs in our book [SW94]. That was hard wired for the purpose at hand. The version here provides much more flexibility.

```
\ensuremath{\texttt{epigraph}}{\langle text \rangle} {\langle source \rangle}
```

The command \epigraph typesets an epigraph using  $\langle text \rangle$  as the main text of the epigraph and  $\langle source \rangle$  being the original author (or book, article, etc.) of the quoted text. By default the epigraph is placed at the right hand side of the typeblock, and the  $\langle source \rangle$  is typeset at the bottom right of the  $\langle text \rangle$ .

```
\begin{epigraphs} \\ \qitem{\langle text \rangle} {\langle source \rangle} \\ ... \\ \end{epigraphs} \end{epigraphs}
```

The epigraphs environment typesets a list of epigraphs, and by default places them at the right hand side of the typeblock. Each epigraph in an epigraphs environment is specified

by a \qitem (analagous to the \item command in ordinary list environments). By default, the  $\langle source \rangle$  is typeset at the bottom right of the  $\langle text \rangle$ .

13.2 General

Example is the school of mankind, and they will learn at no other.

Letters on a Regicide Peace
Edmund Burke

The commands described in this section apply to both the \epigraph command and the epigraphs environment. But first of all, note that an epigraph immediately after a heading will cause the first paragraph of the following text to be indented. If you want the initial paragraph to have no indentation, then start it with the \noindent command.

```
\verb|\epigraphwidth| \\ epigraphposition{|\langle flush \rangle|}
```

The epigraphs are typeset in a minipage of width \epigraphwidth. The default value for this can be changed using the \setlength command. Typically, epigraphs are typeset in a measure much less than the width of the typeblock. The horizontal position of an epigraph in relation to the main typeblock is controlled by the  $\langle flush \rangle$  argument to the \epigraphposition declaration. The default value is flushright, so that epigraphs are set at the right hand side of the typeblock. This can be changed to flushleft for positioning at the left hand side or to center for positioning at the center of the typeblock.

```
\P
```

In order to avoid bad line breaks, the epigraph  $\langle text \rangle$  is normally typeset raggedright. The  $\langle flush \rangle$  argument to the \epigraphtextposition declaration controls the  $\langle text \rangle$  typesetting style. By default this is flushleft (which produces raggedright text). The sensible values are center for centered text, flushright for raggedleft text, and flushleftright for normal justified text.

If by any chance you want the  $\langle text \rangle$  to be typeset in some other layout style, the easiest way to do this is by defining a new environment which sets the paragraphing parameters to your desired values. For example, as the  $\langle text \rangle$  is typeset in a minipage, there is no paragraph indentation. If you want the paragraphs to be indented and justified then define a new environment like:

\newenvironment{myparastyle}{\setlength{\parindent}{1em}}{}

and use it as:

\epigraphtextposition{myparastyle}

```
\verb|\epigraphsourceposition{| \langle flush \rangle|}
```

The  $\langle flush \rangle$  argument to the \epigraphsourceposition declaration controls the position of the  $\langle source \rangle$ . The default value is flushright. It can be changed to flushleft, center or flushleftright.

For example, to have epigraphs centered with the  $\langle source \rangle$  at the left, add the following to your document.

```
\epigraphposition{center}
\epigraphsourceposition{flushleft}
```

```
\ensuremath{\verb{\ensuremath{\mathsf{e}}}}
```

Epigraphs are often typeset in a smaller font than the main text. The  $\langle fontsize \rangle$  argument to the \epigraphfontsize declaration sets the font size to be used. If you don't like the default value (\small), you can easily change it to, say \footnotesize by:

\epigraphfontsize{\footnotesize}

```
\epigraphrule
```

By default, a rule is drawn between the  $\langle text \rangle$  and  $\langle source \rangle$ , with the rule thickness being given by the value of \epigraphrule. The value can be changed by using \setlength. A value of Opt will eliminate the rule. Personally, I dislike the rule in the list environments.

```
\beforeepigraphskip
\afterepigraphskip
```

The two ...skip commands specify the amount of vertical space inserted before and after typeset epigraphs. Again, these can be changed by \setlength. It is desireable that the sum of their values should be an integer multiple of the \baselineskip.

Note that you can use normal LaTeX commands in the  $\langle text \rangle$  and  $\langle source \rangle$  arguments. You may wish to use different fonts for the  $\langle text \rangle$  (say roman) and the  $\langle source \rangle$  (say italic).

The epigraph at the start of this section was specified as:

13.3 Epigraphs before chapter headings

If all else fails, immortality can always be assured by spectacular error.

John Kenneth Galbraith

The \epigraph command and the epigraphs environment typeset an epigraph at the point in the text where they are placed. The first thing that a \chapter command does is to start off a new page, so another mechanism is provided for placing an epigraph just before a chapter heading.

```
\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath}\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath{\mbox{\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremath}\ensuremat
```

The \epigraphhead macro stores  $\langle text \rangle$  for printing at  $\langle distance \rangle$  below the header on a page.  $\langle text \rangle$  can be ordinary text or, more likely, can be either an \epigraph command or an epigraphs environment. By default, the epigraph will be typeset at the righthand margin. If the command is immediately preceded by a \chapter or \chapter\* command, the epigraph is typeset on the chapter title page.

The default value for the optional  $\langle distance \rangle$  argument is set so that an  $\epsilon$  sisting of a single line of quotation and a single line denoting the source is aligned with the

bottom of the 'Chapter X' line produced by the \chapter command using the *default* chapterstyle. In other cases you will have to experiment with the  $\langle distance \rangle$  value. The value for  $\langle distance \rangle$  can be either a integer or a real number. The units are in terms of the current value for \unitlength. A typical value for  $\langle distance \rangle$  for a single line quotation and source for a \chapter\* might be about 70 (points). A positive value of  $\langle distance \rangle$  places the epigraph below the page heading and a negative value will raise it above the page heading.

Here's some example code:

```
\chapter*{Celestial navigation}
\epigraphhead[70]{\epigraph{Star crossed lovers.}{\textit{The Bard}}}
```

The  $\langle text \rangle$  argument is put into a minipage of width \epigraphwidth. If you use something other than \epigraph or epigraphs for the  $\langle text \rangle$  argument, you may have to do some positioning of the text yourself so that it is properly located in the minipage. For example

```
\chapter{Short}
\renewcommand{\epigraphflush}{center}
\epigraphhead{\centerline{Short quote}}
```

The \epigraphhead command changes the page style for the page on which it is specified, so there should be no text between the \chapter and the \epigraphhead commands. The page style is identical to the *plain* page style except for the inclusion of the epigraph. If you want a more fancy style for epigraphed chapters you will have to do some work yourself.

```
\ensuremath{\verb| epigraphforheader[\langle distance\rangle] \{\langle text\rangle\}} \\ \ensuremath{\verb| epigraphpicture|}
```

The \epigraphforheader macro takes the same arguments as \epigraphhead but puts  $\langle text \rangle$  into a zero-sized picture at the coordinate position (0,-<distance>); the macro \epigraphpicture holds the resulting picture. This can then be used as part of a chapter pagestyle, as in

```
\makepagestyle{mychapterpagestyle}
...
\makeoddhead{mychapterpagestyle}{}{\epigraphpicture}
```

Of course the  $\langle text \rangle$  argument for \epigraphforheader need not be an \epigraph, it can be arbitrary text.

```
\label{lemgth} $$ \dropchapter{\langle length\rangle}$ undodrop
```

If a long epigraph is placed before a chapter title it is possible that the bottom of the epigraph may interfere with the chapter title. The command  $\dropchapter$  will lower any subsequent chapter titles by (length); a negative (length) will raise the titles. The command  $\dropchapter$  to their default positions. For example:

```
\dropchapter{2in}
\chapter{Title}
\epigraphhead{long epigraph}
\undodrop
```

```
\cline{cleartoevenpage}[\langle text \rangle]
```

On occasions it may be desirable to put something (e.g., an epigraph, a map, a picture) on the page facing the start of a chapter, where the something belongs to the chapter that is about to start rather than the chapter that has just ended. In order to do this in a document that is going to be printed doublesided, the chapter must start on an odd numbered page and the pre-chapter material put on the immediately preceding even numbered page. The \cleartoevenpage command is like \cleardoublepage except that the page following the command will be an even numbered page, and the command takes an optional argument which is applied to the skipped page (if any).

Here is an example:

```
... end previous chapter.
\cleartoevenpage
\begin{center}
\begin{picture}... \end{picture}
\end{center}
\chapter{Next chapter}
```

If the style is such that chapter headings are put at the top of the pages, then it would be advisable to include \thispagestyle{empty} (or perhaps plain) immediately after \cleartoevenpage to avoid a heading related to the previous chapter from appearing on the page.

If the something is like a figure with a numbered caption and the numbering depends on the chapter numbering, then the numbers have to be hand set (unless you define a special chapter command for the purpose). For example:

```
... end previous chapter.
\cleartoevenpage[\thispagestyle{empty}] % a skipped page to be empty
\thispagestyle{plain}
\addtocounter{chapter}{1} % increment the chapter number
\setcounter{figure}{0} % initialise figure counter
\begin{figure}
...
\caption{Pre chapter figure}
\end{figure}

\addtocounter{chapter}{-1} % decrement the chapter number
\chapter{Next chapter}
 % increments chapter & resets figure numbers
\addtocounter{figure}{1} % to account for pre-chapter figure
```

#### 13.3.1 Epigraphs on book or part pages

If you wish to put an epigraphs on \book or \part pages you have to do a little more work than in other cases. This is because these division commands do some page flipping before and after typesetting the title.

One method is to put the epigraph into the page header as for epigraphs before \chapter titles. By suitable adjustments the epigraph can be placed anywhere on the page, independently of whatever else is on the page. A similar scheme may be used for epigraphs on other kinds of pages. The essential trick is to make sure that the *epigraph* pagestyle is

used for the page. For an epigraphed bibliography or index, the macros \prebibhook or \preindexhook can be appropriately modified to do this.

The other method is to subvert the \beforepartskip command for epigraphs before the title, or the \afterpartskip command for epigraphs after the title (or the equivalents for \book pages).

For example:

```
\let\oldbeforepartskip\beforepartskip % save definition
\renewcommand*{\beforepartskip}{%
  \epigraph{...}% an epigraph
  \vfil}
\part{An epigraphed part}
...
\renewcommand*{\beforepartskip}{%
  \epigraph{...}% another epigraph
  \vfil}
\part{A different epigraphed part}
...
\let\beforepartskip\oldbeforepartskip % restore definition
\part{An unepigraphed part}
...
\...
```

# Fourteen

# Poetry

The typesetting of a poem should ideally be dependent on the particular poem. Individual problems do not usually admit of a general solution, so this manual and code should be used more as a guide towards some solutions rather than providing a ready made solution for any particular piece of verse.

The doggerel used as illustrative material has been taken from [Wil??].

Note that for the examples in this section I have made no attempt to do other than use the minimal (La)TeX capabilities; in particular I have made no attempt to do any special page breaking so some stanzas may cross onto the next page — most undesireable for publication.

The standard LaTeX classes provide the verse environment which is defined as a particular kind of list. Within the environment you use \\ to end a line, and a blank line will end a stanza. For example, here is the text of a single stanza poem:

```
\newcommand{\garden}{
I used to love my garden \\
But now my love is dead \\
For I found a bachelor's button \\
In black-eyed Susan's bed.
}
```

When this is typeset as a normal LaTeX paragraph (with no paragraph indentation), i.e.,

\noident\garden

it looks like:

I used to love my garden But now my love is dead For I found a bachelor's button In black-eyed Susan's bed.

Typesetting it within the verse environment produces:

I used to love my garden But now my love is dead For I found a bachelor's button In black-eyed Susan's bed.

The stanza could also be typeset within the alltt environment, defined in the standard alltt package [Bra97], using a normal font and no \\ line endings.

```
\begin{alltt}\normalfont
I used to love my garden
But now my love is dead
For I found a bachelor's button
In black-eyed Susan's bed.
\end{alltt}
which produces:
```

I used to love my garden But now my love is dead For I found a bachelor's button In black-eyed Susan's bed.

The alltt environment is like the verbatim environment except that you can use LaTeX macros inside it. In the verse environment long lines will be wrapped and indented but in the alltt environment there is no indentation.

Some stanzas have certain lines indented, often alternate ones. To typeset stanzas like this you have to add your own spacing. For instance:

```
\begin{verse}
There was an old party of Lyme \\
Who married three wives at one time. \\
\hspace{2em} When asked: 'Why the third?' \\
\hspace{2em} He replied: 'One's absurd, \\
And bigamy, sir, is a crime.'
\end{verse}
is typeset as:
    There was an old party of Lyme
    Who married three wives at one time.
        When asked: 'Why the third?'
        He replied: 'One's absurd,
        And bigamy, sir, is a crime.'
```

Using the alltt environment you can put in the spacing via ordinary spaces. That is, this:

```
\begin{alltt}\normalfont
There was an old party of Lyme
Who married three wives at one time.
     When asked: 'Why the third?'
     He replied: 'One's absurd,
And bigamy, sir, is a crime.'
\end{alltt}
is typeset as
```

There was an old party of Lyme

```
Who married three wives at one time.
    When asked: 'Why the third?'
    He replied: 'One's absurd,
And bigamy, sir, is a crime.'
   More exotically you could use the TeX \parshape command<sup>1</sup>:
 \parshape = 5 Opt \linewidth Opt \linewidth
                2em \linewidth 2em \linewidth Opt \linewidth
 \verb|\noindent There was an old party of Lyme \verb|\| \\
 Who married three wives at one time. \
 When asked: 'Why the third?' \\
 He replied: 'One's absurd, \\
 And bigamy, sir, is a crime.' \par
which will be typeset as:
There was an old party of Lyme
Who married three wives at one time.
    When asked: 'Why the third?'
    He replied: 'One's absurd,
And bigamy, sir, is a crime.'
```

This is about as much assistance as standard (La)TeX provides, except to note that in the verse environment the \\* version of \\ will prevent a following page break. You can also make judicious use of the \needspace macro to keep things together.

Some books of poetry, and especially anthologies, have two or more indexes, one, say for the poem titles and another for the first lines, and maybe even a third for the poets' names. If you are not using memoir then the index [Jon95] and multind [Lon91] packages provide support for multiple indexes in one document.

#### 14.1 Classy verse

The code provided by the memoir class is meant to help with some aspects of typesetting poetry but does not, and cannot, provide a comprehensive solution to all the requirements that will arise.

The main aspects of typesetting poetry that differ from typesetting plain text are:

- Poems are usually visually centered on the page.
- Some lines are indented, and often there is a pattern to the indentation.
- When a line is too wide for the page it is broken and the remaining portion indented with respect to the original start of the line.

These are the ones that the class attempts to deal with.

```
\label{length} $$ \left( \langle length \rangle \right] ... \end{verse} $$ \left( \langle length \rangle \right) ... \end{verse} $$ \left( \langle lengt
```

<sup>1</sup> See the TeXbook for how to use this.

The verse environment provided by the class is an extension of the usual LaTeX environment. The environment takes one optional parameter, which is a length; for example \begin{verse}[4em]. You may have noticed that the earlier verse examples are all near the left margin, whereas verses usually look better if they are typeset about the center of the page. The length parameter, if given, should be about the length of an average line, and then the entire contents will be typeset with the mid point of the length centered horizontally on the page.

The length \versewidth is provided as a convenience. It may be used, for example, to calculate the length of a line of text for use as the optional argument to the verse environment:

```
\settowidth{\versewidth}{This is the average line,}
\begin{verse}[\versewidth]
```

```
\vleftmargin
```

In the basic LaTeX verse environment the body of the verse is indented from the left of the typeblock by an amount \leftmargini, as is the text in many other environments based on the basic LaTeX list environment. For the class's version of verse the default indent is set by the length \vleftmargin (which is initially set to leftmargini). For poems with particularly long lines it could perhaps be advantageous to eliminate any general indentation by:

\setlength{\vleftmargin}{0em}

If necessary the poem could even be moved into the left margin by giving \vleftmargin a negative length value, such as -1.5em.

```
\stanzaskip
```

The vertical space between stanzas is the length \stanzaskip. It can be changed by the usual methods.

```
\vin
\vgap
\vindent
```

The command \vin is shorthand for \hspace\*{\vgap} for use at the start of an indented line of verse. The length \vgap (initially 1.5em) can be changed by \setlength or \addtolength. When a verse line is too long to fit within the typeblock it is wrapped to the next line with an initial indent given by the value of the length vindent. Its initial value is twice the default value of \vgap.

```
\\[\left\[\left\] \\* [\left\] \\\! [\left\] \\! [\left\] \\ [\left\
```

Each line in the verse environment, except possibly for the last line in a stanza, must be ended by  $\$  which comes in several variants. In each variant the optional  $\langle length \rangle$  is the vertical space to be left before the next line. The  $\$  form prohibits a page break after the line. The  $\$ ! form is to be used only for the last line in a stanza when the lines are being numbered; this is because the line numbers are incremented by the  $\$  macro. It would normally be followed by a blank line.

Using \verselinebreak will cause later text in the line to be typeset indented on the following line. If the optional  $\langle length \rangle$  is not given the indentation is twice \vgap, otherwise it is  $\langle length \rangle$ . The broken line will count as a single line as far as the altverse and patverse environments are concerned. The macro \\> is shorthand for \verselinebreak, and unlike other members of the \\ family the optional  $\langle length \rangle$  is the indentation of the following partial line, not a vertical skip. Also, the \\> macro does not increment any line number.

```
\vee inphantom{\langle text \rangle}
```

Verse lines are sometimes indented according to the space taken by the text on the previous line. The macro  $\$  vinphantom can be used at the start of a line to give an indentation as though the line started with  $\langle text \rangle$ . For example here are a few lines from the portion of *Fridthjof's Saga* where Fridthjof and Ingeborg part:

```
Source for example 14.1
\settowidth{\versewidth}{Nay, nay, I leave thee not,
                         thou goest too}
\begin{verse}[\versewidth]
\ldots \\*
His judgement rendered, he dissolved the Thing. \\*
\flagverse{Ingeborg} And your decision? \\*
\flagverse{Fridthjof} \vinphantom{And your decision?}
                      Have I ought to choose? \\*
Is not mine honour bound by his decree? \\*
And that I will redeem through Angantyr \\*
His paltry gold doth hide in Nastrand's flood. \\*
Today will I depart. \/*
\flagverse{Ingeborg} \vinphantom{Today will I depart.}
                     And Ingeborg leave? \/*
\flagverse{Fridthjof} Nay, nay, I leave thee not,
                      thou goest too. \\*
\flagverse{Ingeborg} Impossible! \\*
\flagverse{Fridthjof} \vinphantom{Impossible!}
                      O! hear me, ere thou answerest.
\end{verse}
```

Use of \vinphantom is not restricted to the start of verse lines — it may be used anywhere in text to leave some some blank space. For example, compare the two lines below, which are produced by this code:

```
\noindent Come away with me and be my love --- Impossible. \\
Come away with me \vinphantom{and be my love} --- Impossible.

Come away with me and be my love — Impossible.

Come away with me — Impossible.

\vleftofline{\langle text \rangle}
```

	Typeset example 14.1: Phantom text in verse	
	His judgement rendered, he dissolved the Thing.	
Ingeborg	And your decision?	
Fridthjof	Have I ought to choose?	
<b>y</b>	Is not mine honour bound by his decree?	
	And that I will redeem through Angantyr	
	His paltry gold doth hide in Nastrand's flood.	
	Today will I depart.	
Ingeborg	And Ingeborg leave?	
Fridthjof	Nay, nay, I leave thee not, thou goest too.	
Ingeborg	Impossible!	
Fridthjof	O! hear me, ere thou answerest.	

A verse line may start with something, for example open quote marks, where it is desireable that it is ignored as far as the alignment of the rest of the line is concerned  $^2$  — a sort of 'hanging left punctuation'. When it is put at the start of a line in the verse environment the  $\langle text \rangle$  is typeset but ignored as far as horizontal indentation is concerned. Compare the two examples.

```
Source for example 14.2

\noindent ''No, this is what was spoken by the prophet Joel:
\begin{verse}
''\,'\,''In the last days,'' God says, \\
''I will pour out my spirit on all people. \\
Your sons and daughters will prophesy, \\
\ldots \\
And everyone who calls \ldots ''\,'
\end{verse}
```

```
Source for example 14.3

\noindent ''No, this is what was spoken by the prophet Joel:
\begin{verse}
\vleftofline{''\,'\,''}In the last days,'' God says, \\
\vleftofline{''}I will pour out my spirit on all people. \\
Your sons and daughters will prophesy, \\
\ldots \\
```

 $<sup>2\</sup>quad$  Requested by Matthew Ford who also provided the example text.

# Typeset example 14.2: Verse with regular quote marks

```
"'No, this is what was spoken by the prophet Joel:

"'"In the last days," God says,

"I will pour out my spirit on all people.

Your sons and daughters will prophesy,
...

And everyone who calls ..."
```

## Typeset example 14.3: Verse with hanging left quote marks

```
"No, this is what was spoken by the prophet Joel:

""In the last days," God says,

"I will pour out my spirit on all people.

Your sons and daughters will prophesy,

...

And everyone who calls ..."
```

```
And everyone who calls \ldots ''\,' \end{verse}
```

# 14.1.1 Indented lines

Within the verse environment stanzas are normally separated by a blank line in the input.

```
\begin{altverse} ... \end{altverse}
```

Individual stanzas within verse may, however, be enclosed in the altverse environment. This has the effect of indenting the 2nd, 4th, etc., lines of the stanza by the length  $\$ vgap.

```
\begin{patverse} ... \end{patverse} \\ begin{patverse*} ... \end{patverse*} \\ \indentpattern{$\langle digits \rangle$} \\ \end{patverse} \\
```

As an alternative to the altverse environment, individual stanzas within the verse environment may be enclosed in the patverse environment. Within this environment the indentation of each line is specified by an indentation pattern, which consists of an array of digits,  $d_1$  to  $d_n$ , and the nth line is indented by  $d_n$  times \vgap. However, the first line is not indented, irrespective of the value of  $d_1$ .

The indentation pattern for a patverse or patverse\* environment is specified via the \indentpattern command, where  $\langle digits \rangle$  is a string of digits (e.g., 3213245281). With the patverse environment, if the pattern is shorter than the number of lines in the stanza, the

trailing lines will not be indented. However, in the patverse\* environment the pattern keeps repeating until the end of the stanza.

### 14.1.2 Numbering

```
\label{eq:local_local_local_local} $$ \left\{ \left\langle flag \right\rangle \right\} $$ \volength{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\mbox{\m\s\m\m\\\\m\s\m\\novint\\\m\s\\\\\novinty\sen\sin\\novint\\mox{\\mbox{\m
```

Putting \flagverse at the start of a line will typeset  $\langle flag \rangle$ , for example the stanza number, ending at a distance \vleftskip before the line. The default for \vleftskip is 3em.

The lines in a poem may be numbered.

```
\label{linenumberfrequency} $$ \left( \frac{nth}{s} \right) + \left( \frac{f(rst)}{s(rst)} \right) $$
```

The declaration \linenumberfrequency{ $\langle nth \rangle$ } will cause every  $\langle nth \rangle$  line of succeeding verses to be numbered. For example, \linenumberfrequency{5} will number every fifth line. The default is \linenumberfrequency{0} which prevents any numbering. The \setverselinenums macro can be used to specify that the number of the first line of the following verse shall be  $\langle first \rangle$  and the first printed number shall be  $\langle startat \rangle$ . For example, perhaps you are quoting part of a numbered poem. The original numbers every tenth line but if your extract starts with line 7, then

```
\linenumberfrequency{10}
\setverselinenums{7}{10}
```

is what you will need.

The poemline counter is used in numbering the lines, so the number representation is \thepoemline, which defaults to arabic numerals, and they are typeset using the font specified via \linenumberfont; the default is

\linenumberfont{\small\rmfamily}

for small numbers in the roman font.

```
\verselinenumbersright
\verselinenumbersleft
\vrightskip
```

Following the declaration \verselinenumbersright, which is the default, any verse line numbers will be set in the righthand margin. The \verselinenumbersleft declaration will set any subsequent line numbers to the left of the lines. The numbers are set at a distance \vrightskip (default 1em) into the margin.

### 14.2 Titles

The \PoemTitle command is provided for typesetting titles of poems.

The \PoemTitle command takes the same arguments as the \chapter command; it typesets the title for a poem and adds it to the ToC. Following the declaration \NumberPoemTitle the title is numbered but there is no numbering after the \PlainPoemTitle declaration.

```
\poemtoc{\langle sec \rangle}
```

The kind of entry made in the ToC by \PoemTitle is defined by \poemtoc. The initial definition is:

\newcommand{\poemtoc}{section}

for a section-like ToC entry. This can be changed to, say, chapter or subsection or ....

```
\label{lemark} $$ \operatorname{poemtitlemark}_{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\colored{\
```

The macro \poemtitlemark is called with the argument \( \for head \)\) so that it may be used to set marks for use in a page header via the normal mark process. The \poemtitlepstyle macro, which by default does nothing, is provided as a hook so that, for example, it can be redefined to specify a particular pagestyle that should be used. For example:

```
\renewcommand*{\poemtitlemark}[1]{\markboth{#1}{#1}}
\renewcommand*{\poemtitlepstyle}{%
  \pagestyle{headings}%
  \thispagestyle{empty}}

\PoemTitle*[\(\forhead\)] \{\(\title\)}
  \poemtitlestarmark{\(\forhead\)}
  \poemtitlestarpstyle
\]
```

The \PoemTitle\* command produces an unnumbered title that is not added to the ToC. Apart from that it operates in the same manner as the unstarred version. The \poemtitlestarmark and \poemtitlestarpstyle can be redefined to set marks and pagestyles.

14.2.1 Main Poem Title layout parameters

The essence of the code used to typeset a numbered  $\langle title \rangle$  from a \PoemTitle is:

```
\PoemTitleheadstart
\printPoemTitlenum
\afterPoemTitlenum
\printPoemTitletitle{title}
\afterPoemTitle
```

If the title is unnumbered then \printPoemTitlenonum is used instead of the \printPoemTitlenum and \afterPoemTitlenum pair of macros.

The various elements of this can be modified to change the layout. By default the number is centered above the title, which is also typeset centered, and all in a \large font.

The elements are detailed in the next section.

### 14.2.2 Detailed Poem Title layout parameters

```
\beforePoemTitleskip
\PoemTitlenumfont
\midPoemTitleskip
\PoemTitlefont
\afterPoemTitleskip
```

As defined,  $\PoemTitleheadstart$  inserts vertical space before a poem title. The default definition is:

```
\newcommand*{\PoemTitleheadstart}{\vspace{\beforePoemTitleskip}}
\newlength{\beforePoemTitleskip}
\setlength{\beforePoemTitleskip}{1\onelineskip}
```

\printPoemTitlenum typesets the number for a poem title. The default definition, below, prints the number centered and in a large font.

```
\newcommand*{\printPoemTitlenum}{\PoemTitlenumfont \thepoem}
\newcommand*{\PoemTitlenumfont}{\normalfont\large\centering}
```

The definition of  $\printPoemTitlenonum$ , which is used when there is no number, is simply

```
\newcommand*{\printPoemTitlenonum}{}
```

\afterPoemTitlenum is called between setting the number and the title. It ends a paragraph (thus making sure any previous \centering is used) and then may add some vertical space. The default definition is:

```
\newcommand*{\afterPoemTitlenum}{\par\nobreak\vskip \midPoemTitleskip}
\newlength{\midPoemTitleskip}{0pt}
\setlength{\midPoemTitleskip}{0pt}
```

The default definition of \printPoemTitletitle is below. It typesets the title centered and in a large font.

```
\newcommand*{\printPoemTitletitle}[1]{\PoemTitlefont #1}
\newcommand*{\PoemTitlefont}{\normalfont\large\centering}
```

The macro \afterPoemTitle finishes off the title typesetting. The default definition is:

```
\newcommand*{\afterPoemTitle}{\par\nobreak\vskip \afterPoemTitleskip}
\newlength{\afterPoemTitleskip}{1\onelineskip}
```

## 14.3 Examples

Here are some sample verses using the class facilities.

First a Limerick, but titled and centered:

```
\renewcommand{\poemtoc}{subsection}
\PlainPoemTitle
\PoemTitle{A Limerick}
\settowidth{\versewidth}{There was a young man of Quebec}
\begin{verse}[\versewidth]
There was a young man of Quebec \\
Who was frozen in snow to his neck. \\
\vin When asked: 'Are you friz?' \\
\vin He replied: 'Yes, I is, \\
But we don't call this cold in Quebec.'
\end{verse}
```

which gets typeset as below. The \poemtoc is redefined to subsection so that the \poemtitle titles are entered into the ToC as subsections. The titles will not be numbered because of the \PlainPoemTitle declaration.

#### A Limerick

There was a young man of Quebec Who was frozen in snow to his neck. When asked: 'Are you friz?' He replied: 'Yes, I is, But we don't call this cold in Quebec.'

Next is the Garden verse within the altverse environment. Unlike earlier renditions this one is titled and centered.

```
\settowidth{\versewidth}{But now my love is dead}
\PoemTitle{Love's lost}
\begin{verse} [\versewidth]
\begin{altverse}
\garden
\end{altverse}
\end{verse}
```

Note how the alternate lines are automatically indented in the typeset result below.

### Love's lost

I used to love my garden But now my love is dead For I found a bachelor's button In black-eyed Susan's bed.

It is left up to you how you might want to add information about the author of a poem. Here is one example of a macro for this:

```
\newcommand{\attrib}[1]{%
  \nopagebreak{\raggedleft\footnotesize #1\par}}
```

This can be used as in the next bit of doggerel.

```
\PoemTitle{Fleas}
\settowidth{\versewidth}{What a funny thing is a flea}
\begin{verse}[\versewidth]
What a funny thing is a flea. \\
You can't tell a he from a she. \\
But he can. And she can. \\
Whoopee!
\end{verse}
\attrib{Anonymous}
```

#### Fleas

What a funny thing is a flea. You can't tell a he from a she. But he can. And she can. Whoopee!

Anonymous

The next example demonstrates the automatic line wrapping for overlong lines.

# In the beginning

Then God created Newton,
And objects at rest tended to remain at rest,
And objects in motion tended to remain in motion,
And energy was conserved and momentum was conserved and
matter was conserved
And God saw that it was conservative.

Possibly from Analog, circa 1950

The following verse demonstrates the use of a forced linebreak; I have used the \\> command instead of the more descriptive, but discursive, \verselinebreak. It also has a slightly different title style.

```
\renewcommand{\PoemTitlefont}{%
              \normalfont\large\itshape\centering}
\poemtitle{Mathematics}
\settowidth{\versewidth}{Than Tycho Brahe, or Erra Pater:}
\begin{verse}[\versewidth]
In mathematics he was greater \\
Than Tycho Brahe, or Erra Pater: \\
For he, by geometric scale, \\
Could take the size of pots of ale; \\
\settowidth{\versewidth}{Resolve by}%
Resolve, by sines \\>[\versewidth] and tangents straight, \\
If bread or butter wanted weight; \\
And wisely tell what hour o' the day \\
The clock does strike, by Algebra.
\end{verse}
\attrib{Samuel Butler (1612--1680)}
```

#### Mathematics

In mathematics he was greater Than Tycho Brahe, or Erra Pater: For he, by geometric scale, Could take the size of pots of ale; Resolve, by sines

and tangents straight, If bread or butter wanted weight; And wisely tell what hour o' the day The clock does strike, by Algebra.

Samuel Butler (1612–1680)

Another limerick, but this time taking advantage of the patverse environment. If you are typesetting a series of limericks a single \indentpattern will do for all of them.

```
\settowidth{\versewidth}{There was a young lady of Ryde}
\indentpattern{00110}
\needspace{7\onelineskip}
\PoemTitle{The Young Lady of Ryde}
\begin{verse}[\versewidth]
\begin{patverse}
There was a young lady of Ryde \\
Who ate some apples and died. \\
The apples fermented \\
Inside the lamented \\
And made cider inside her inside.
```

```
\end{patverse}
\end{verse}
```

Note that I used the \needspace command to ensure that the limerick will not get broken across a page.

The Young Lady of Ryde

There was a young lady of Ryde Who ate some apples and died. The apples fermented Inside the lamented And made cider inside her inside.

The next example is a song you may have heard of. This uses \flagverse for labelling the stanzas, and because the lines are numbered they can be referred to.

```
\settowidth{\versewidth}{In a cavern, in a canyon,}
\PoemTitle{Clementine}
\begin{verse}[\versewidth]
\linenumberfrequency{2}
\begin{altverse}
\flagverse{1.} In a cavern, in a canyon, \\
Excavating for a mine, \\
Lived a miner, forty-niner, \label{vs:49} \\
And his daughter, Clementine. \\!
\end{altverse}
\begin{altverse}
\flagverse{\textsc{chorus}} Oh my darling, Oh my darling, \\
Oh my darling Clementine. \\
Thou art lost and gone forever, \
Oh my darling Clementine.
\end{altverse}
\linenumberfrequency{0}
\end{verse}
The 'forty-niner' in line~\ref{vs:49} of the song
refers to the gold rush of 1849.
```

#### Clementine

In a cavern, in a canyon,
 Excavating for a mine,
 Lived a miner, forty-niner,
 And his daughter, Clementine.

2

4

CHORUS Oh my darling, Oh my darling,

Oh my darling Clementine. Thou art lost and gone forever, Oh my darling Clementine. 6

The 'forty-niner' in line 3 of the song refers to the gold rush of 1849.

The last example is a much more ambitious use of \indentpattern. In this case it is defined as:

# Mouse's Tale

```
Fury said to
   a mouse, That
          he met
                 in the
                 house,
                 'Let us
             both go
          to law:
      I will
      prosecute
   уои. —
   Come, I'll
      take no
          denial;
             We must
                    have a
                           trial:
                              For
                           really
                        this
                       morning
                       nothing
                       to do.'
                 Said the
                mouse to
             the cur,
Such a
                trial,
dear sir,
jury or
judge,
would be
wasting
our breath.'
'I'll be
judge,
I'll be
jury.'
Said
cunni
          With no
               y.
id
cunning
old Fury;
T'll try
the whole
                        and
                       condemn
                           you
to
```

Lewis Carrol, Alice's Adventures in Wonderland, 1865

# Fifteen

# Boxes, verbatims and files

The title of this chapter indicates that it deals with three disconnected topics, but there is method in the seeming peculiarity. By the end of the chapter you will be able to write LaTeX code that lets you put things in your document source at one place and have them typeset at a different place, or places. For example, if you are writing a text book that includes questions and answers then you could write a question and answer together yet have the answer typeset at the end of the book.

Writing in one place and printing in another is based on outputting stuff to a file and then inputting it for processing at another place or time. This is just how LaTeX produces the ToC. It is often important when writing to a file that LaTeX does no processing of any macros, which implies that we need to be able to write verbatim. One use of verbatim in LaTeX is to typeset computer code or the like, and to clearly distinguish the code from the main text it is often typeset within a box. Hence the chapter title.

The class extends the kinds of boxes normally provided, extends the default verbatims, and provides a simple means of writing and reading files.

One problem with verbatims is that they can not be used as part of an argument to a command. For example to typeset something in a framed minipage the obvious way is to use the minipage as the argument to the \fbox macro:

```
\fbox{\begin{minipage}{6cm}
    Contents of framed minipage
    \end{minipage}}
```

This works perfectly well until the contents includes some verbatim material, whereupon you will get nasty error messages. However this particular conundrum is solvable, even if the solution is not particularly obvious. Here it is.

We can put things into a box, declared via \newsavebox, and typeset the contents of the box later via \usebox. The most common way of putting things into a save box is by the \sbox or \savebox macros, but as the material for saving is one of the arguments to these macros this approach fails. But, lrbox is an environment form of \sbox, so it can handle verbatim material. The code below, after getting a new save box, defines a new framedminipage environment which is used just like the standard minipage. The framedminipage starts an lrbox environment and then starts a minipage environment, after which comes the contents. At the end it closes the two environments and calls \fbox with its argument being the contents of the saved box which have already been typeset.

```
\newsavebox{\minibox}
\newenvironment{framedminipage}[1]{%
\begin{lrbox}{\minibox}\begin{minipage}{#1}}%
```

```
{\end{minipage}\end{lrbox}\fbox{\usebox{\minibox}}}
```

**Question 1.** Can you think of any improvements to the definition of the framedminipage environment?

**Question 2.** An answer to question 1 is at the end of this chapter. Suggest how it was put there.

#### 15.1 Boxes

LaTeX provides some commands to put a box round some text. The class extends the available kinds of boxes.

```
\begin{framed} text \end{framed}
\begin{shaded} text \end{shaded}
\begin{snugshade} text \end{snugshade}
```

The framed, shaded, and snugshade environments, which were created by Donald Arseneau as part of his framed package [Ars07], put their contents into boxes that break across pages. The framed environment delineates the box by drawing a rectangular frame. If there is a pagebreak in the middle of the box, frames are drawn on both pages.

The shaded environment typesets the box with a shaded or colored background. This requires the use of the color package [Car05], which is one of the required LaTeX packages, or the xcolor package [Ker07]. The shading color is shadecolor, which you have to define before using the environment. For example, to have a light gray background:

```
\definecolor{shadecolor}{gray}{0.9}
```

For complete information on this see the documentation for the color or xcolor packages, or one of the LaTeX books like the *Graphics Companion* [ $GM^+07$ ]. In the snugshaded environment the box clings more closely to its contents than it does in the shaded environment.

#### Recommended alternative

Since the class was originally written, much have happened in the gfx generating capabilities in LaTeX, especially the popularity of TikZ has provided many more extensive box and graphics generating packages.

As of 2018 one of the most impressive packages for all sorts of boxes is the tcolorbox package by Thomas F. Sturm.

Be aware that the boxes we present in this manual are somewhat delicate; they do not work in all circumstances. For example they will not work with the multicol package [Mit18], and any floats or footnotes in the boxes will disappear.

```
\FrameRule \FrameSep \FrameHeightAdjust
```

The framed environment puts the text into an '\fbox' with the settings:

```
\setlength{\FrameRule}{\fboxrule}
\setlength{\FrameSep}{3\fboxsep}
```

The macro \FrameHeightAdjust specifies the height of the top of the frame above the baseline at the top of a page; its initial definition is:

\providecommand\*{\FrameHeightAdjust}{0.6em}

Internally, the environments are specified using the MakeFramed environment. The \( \setting \) should contain any adjustments to the text width (applied to \hsize and using the \width of the frame itself) and a 'restore' command, which is normally the provided \FrameRestore macro. The frame itself is drawn via the \FrameCommand, which can be changed to obtain other boxing styles. The default definition equates to an \fbox and is:

```
\newcommand*{\FrameCommand}{%
  \setlength{\fboxrule}{\FrameRule}\setlength{\fboxsep}{\FrameSep}%
  \fbox}
```

For example, the framed, shaded and snugshade environments are defined as

```
\newenvironment{framed}{% % uses default \FrameCommand
  \MakeFramed{\advance\hsize -\width \FrameRestore}}%
  {\endMakeFramed}
\newenvironment{shaded}{% % redefines \FrameCommand as \colorbox
  \def\FrameCommand{\fboxsep=\FrameSep \colorbox{shadecolor}}%
  \MakeFramed{\FrameRestore}}%
  {\endMakeFramed}
\newenvironment{snugshade}{% A tight version of shaded
  \def\FrameCommand{\colorbox{shadecolor}}%
  \MakeFramed{\FrameRestore\@setminipage}}%
  {\par\unskip\endMakeFramed}
```

If you wanted a narrow, centered, framed environment you could do something like this:

where 22pc will be the width of the new framed environment.

```
\begin{leftbar} text \end{leftbar}
```

The leftbar environment draws a thick vertical line at the left of the text. It is defined as

```
\newenvironment{leftbar}{%
  \def\FrameCommand{\vrule width 3pt \hspace{10pt}}%
  \MakeFramed{\advance\hsize -\width \FrameRestore}}%
  {\endMakeFramed}
```

By changing the  $\langle setting \rangle$  for \MakeFramed and the definition of \FrameCommand you can obtain a variety of framing styles. For instance, to have rounded corners to the frame instead of the normal sharp ones, you can use the fancybox package [Zan98] and the following code:

```
\usepackage{fancybox}
\newenvironment{roundedframe}{%
  \def\FrameCommand{%
   \cornersize*{20pt}%
   \setlength{\fboxsep}{5pt}%
   \ovalbox}%
   \MakeFramed{\advance\hsize-\width \FrameRestore}}%
   {\endMakeFramed}
```

A framed environment is normally used to distinguish its contents from the surrounding text. A title for the environment may be useful, and if there was a pagebreak in the middle, a title on the continuation could be desireable. Doing this takes a bit more work than I have shown so far. This first part was inspired by a posting to CTT by Donald Arseneau. 1.

```
\newcommand{\FrameTitle}[2]{%
 \fboxrule=\FrameRule \fboxsep=\FrameSep
 \fbox{\vbox{\nobreak \vskip -0.7\FrameSep
    \rlap{\strut#1}\nobreak\nointerlineskip% left justified
    \vskip 0.7\FrameSep
    \hbox{#2}}}
\newenvironment{framewithtitle}[2][\FrameFirst@Lab\ (cont.)]{%
  \def\FrameFirst@Lab{\textbf{#2}}%
 \def\FrameCont@Lab{\textbf{#1}}%
 \def\FrameCommand##1{%
    \FrameTitle{\FrameFirst@Lab}{##1}}%
 \def\FirstFrameCommand##1{%
    \FrameTitle{\FrameFirst@Lab}{##1}}%
 \def\MidFrameCommand##1{%
    \FrameTitle{\FrameCont@Lab}{##1}}%
 \def\LastFrameCommand##1{%
    \FrameTitle{\FrameCont@Lab}{##1}}%
 \MakeFramed{\advance\hsize-\width \FrameRestore}}%
 {\endMakeFramed}
```

The framewithtitle environment, which is the end goal of this exercise, acts like the framed environment except that it puts a left-justified title just after the top of the frame box and before the regular contents.

The  $\langle title \rangle$  is set in a bold font. If the optional  $\langle cont\text{-}title \rangle$  argument is given then  $\langle cont\text{-}title \rangle$  is used as the title on any succeeding pages, otherwise the phrase ' $\langle title \rangle$  (cont.)' is used for the continuation title.

If you would like the titles centered, replace the line marked 'left justified' in the code for \FrameTitle with the line:

<sup>1</sup> On 2003/10/24 in the thread framed.sty w/heading?. The particulars are no longer applicable as the framing code in question then has since been revised.

\rlap{\centerline{\strut#1}}\nobreak\nointerlineskip% centered

The code for the frametitle environment is not obvious. The difficulty in creating the environment was that the underlying framing code goes through the 'stuff' to be framed by first trying to fit it all onto one page (\FrameCommand). If it does not fit, then it takes as much as will fit and typesets that using \FirstFrameCommand, then tries to typeset the remainder on the next page. If it all fits then it uses \LastFrameCommand. If it doesn't fit, it typesets as much as it can using \MidFrameCommand, and then tries to set the remainder on the following page. The process repeats until all has been set.

If you would prefer to have the title at the top outside the frame the above code needs adjusting.

```
\newcommand{\TitleFrame}[2]{%
  \fboxrule=\FrameRule \fboxsep=\FrameSep
 \vbox{\nobreak \vskip -0.7\FrameSep
    \rlap{\strut#1}\nobreak\nointerlineskip% left justified
    \vskip 0.7\FrameSep
    \noindent\fbox{#2}}}
\newenvironment{titledframe}[2][\FrameFirst@Lab\ (cont.)]{%
  \def\FrameFirst@Lab{\textbf{#2}}%
  \def\FrameCont@Lab{\textbf{#1}}%
  \def\FrameCommand##1{%
    \TitleFrame{\FrameFirst@Lab}{##1}}
  \def\FirstFrameCommand##1{%
    \TitleFrame{\FrameFirst@Lab}{##1}}
  \def\MidFrameCommand##1{%
    \TitleFrame{\FrameCont@Lab}{##1}}
  \def\LastFrameCommand##1{%
    \TitleFrame{\FrameCont@Lab}{##1}}
  \MakeFramed{\hsize\textwidth
              \advance\hsize -2\FrameRule
              \advance\hsize -2\FrameSep
              \FrameRestore}}%
  {\endMakeFramed}
  \begin{titledframe} [\langle cont\text{-}title \rangle] {\langle title \rangle} \ text \end{titledframe}
```

The titledframe environment is identical to framewithtitle except that the title is placed just before the frame. Again, if you would like a centered title, replace the line marked 'left justified' in \TitleFrame by

```
\rlap{\centerline{\strut#1}}\nobreak\nointerlineskip% centered
```

You can adjust the code for the framewithtitle and titledframe environments to suit your own purposes, especially as they are not part of the class so you would have to type them in yourself anyway if you wanted to use them, using whatever names you felt suitable.

The class provides two further environments in addition to those from the framed package.

```
\begin{qframe} text \end{qframe} \begin{qshade} text \end{qshade}
```

When used within, say, a quotation environment, the framed and shaded environments do not closely box the indented text. The qframe and qshade environments do provide close boxing.<sup>2</sup> The difference can be seen in the following quotation.

This is the start of a quotation environment. It forms the basis showing the difference between the framed and qframe environments.

This is the second paragraph in the quotation environment and in turn it is within the qframe environment.

This is the third paragraph in the quotation environment and in turn it is within the framed environment.

This is the fourth and final paragraph within the quotation environment and is not within either a qfame or framed environment.

If you want to put a frame inside an adjustwidth environment then you may well find that qframe or qshade meet your expections better than framed of shaded. Of course, it does depend on what your expectations are.

### 15.2 Long comments

The % comment character can be used to comment out (part of) a line of TeX code, but this gets tedious if you need to comment out long chunks of code.

```
\begin{comment} text to be skipped over \end{comment}
```

As an extreme form of font changing, although it doesn't actually work that way, anything in a comment environment will not appear in the document; effectively, LaTeX throws it all away. This can be useful to temporarily discard chunks of stuff instead of having to mark each line with the % comment character.

```
\label{eq:local_name} $$\operatorname{commentsoff}_{name} $$ \operatorname{commentson}_{name} $$
```

The class lets you define your own comment environment via the \newcomment command which defines a comment environment called  $\langle name \rangle$ . In fact the class itself uses \newcomment{comment} to define the comment environment. A comment environment  $\langle name \rangle$  may be switched off so that its contents are not ignored by using the \commentsoff declaration. It may be switched on later by the \commentson declaration. In either case  $\langle name \rangle$  must have been previously declared as a comment environment via \newcomment.

Suppose, for example, that you are preparing a draft document for review by some others and you want to include some notes for the reviewers. Also, you want to include some private comments in the source for yourself. You could use the comment environment for your private comments and create another environment for the notes to the reviewers. These notes should not appear in the final document. Your source might then look like:

<sup>2</sup> Donald Arseneau has said that he may put something similar in a later version the framed package.

```
\newcomment{review}
\ifdraftdoc\else
   \commentsoff{review}
\fi
...
\begin{comment}
Remember to finagle the wingle!
\end{comment}
...
\begin{review}
\textit{REVIEWERS: Please pay particular attention to this section.}
\end{review}
```

Comment environments cannot be nested, nor can they overlap. The environments in the code below will not work in the manner that might be expected:

```
\newcomment{acomment} \newcomment{mycomment}
\begin{comment} %% comments cannot be nested
...
  \end{acomment}
...
  \begin{mycomment}
...
  \end{comment}
...
  \end{comment}
...
  \end{mycomment} %% comments cannot overlap
```

More encompassing comment environments are available if you use Victor Eijkhout's comment package [Eij99].

# 15.3 Verbatims

Standard LaTeX defines the \verb and \verb\* commands for typesetting short pieces of text verbatim, short because they cannot include a linebreak. For longer verbatim texts the verbatim or verbatim\* environments can be used. The star forms indicate spaces in the verbatim text by outputing a  $\square$  mark for each space. The class extends the standard verbatims in various ways.

If you have to write a lot of \verb text, as I have had to do for this book, it gets tedious to keep on typing this sort of thing: \verb!verbatim text!. Remember that the character immediately after the \verb, or \verb\*, ends the verbatim processing.

The \MakeShortVerb macro takes a character preceded by a backslash as its argument, say \!, and makes that character equivalent to \verb!. Using the character a second time will stop the verbatim processing. Doing, for example \MakeShortVerb{\!}, lets you then use !verbatim text! instead of the longer winded \verb!verbatim text!.

You have to pick as the short verb character one that you are unlikely to use; a good choice is often the | bar character as this rarely used in normal text. This choice, though may be unfortunate if you want to have any tabulars with vertical lines, as the bar character is used to specify those. The \DeleteShortVerb macro is provided for this contingency; give it the same argument as an earlier \MakeShortVerb and it will restore the short verb character to its normal state.

The \MakeShortVerb and \DeleteShortVerb macros come from the shortvrb package which is part of the LaTeX base system, but I have found them so convenient that I added them to the class.

```
\setverbatimfont{\( font-declaration \) \}
```

The default font for verbatims is the normal sized monospaced font. The declaration \setverbatimfont can be used to specify a different font. The class default is

\setverbatimfont{\normalfont\ttfamily}

To use a smaller version simply say

\setverbatimfont{\normalfont\ttfamily\small}

A monospaced font is normally chosen as verbatim text is often used to present program code or typewritten text. If you want a more exotic font, try this

\setverbatimfont{\fontencoding{T1}\fontfamily{cmss}\selectfont}

and your verbatim text will then look like

We are no longer using the boring old typewriter font for verbatim text. We used the T1 encoding to make sure that characters that are often ligatures like '', or '', or ---, or <, or >, print as expected. After this we will switch back to the default verbatim font via \setverbatimfont{\normalfont\ttfamily}

In the normal way of things with an OT1 fontencoding, typesetting the ligatures mentioned above in the sans font produces: ligatures like ", or ", or ---, or  $_{i}$ , or  $_{\iota}$ , which is not what happens in the \verbatim environment.

```
\begin{verbatim} anything \end{verbatim}
\begin{verbatim*} anything \end{verbatim*}
```

In the verbatim environment<sup>3</sup> you can write anything you want (except \end{verbatim}), and it will be typeset exactly as written. The verbatim\* environment is similar except, like with \verb\*, spaces will be indicated with a  $\sqcup$  mark.

```
\tabson[\langle number \rangle] \tabsoff
```

The standard verbatim environment ignores any TAB characters; with the class's environment after calling the \tabson declaration the environment will handle TAB characters. By default 4 spaces are used to represent a TAB; the optional  $\langle number \rangle$  argument to the declaration will set the number of spaces for a TAB to be  $\langle number \rangle$ . Some folk like to use 8

<sup>3</sup> This version of the verbatim environment is heavily based on the verbatim package [SRR99] but does provide some extensions.

spaces for a TAB, in which case they would need to declare \tabson[8]. Unremarkably, the declaration \tabsoff switches off TABs. The class default is \tabsoff.

As noted, whatever is written in a verbatim environment is output just as written, even if lines are too long to fit on the page. The declaration \wrappingon lets the environment break lines so that they do not overflow. The declaration \wrappingoff restores the normal behaviour.

The following is an example of how a wrapped verbatim line looks. In the source the contents of the verbatim was written as a single line.

This is an example of line wrapping in the verbatim environment. It is a % single line in the source and the \wrappingon declaration has been % used.

The wrapped portion of verbatim lines are indented from the left margin by the length \verbatimindent. The value can be changed by the usual length changing commands. The end of each line that has been wrapped is marked with the  $\langle char \rangle$  character of the \verbatimbreakchar macro. The class default is \verbatimbreakchar{\char'\%}, so that lines are marked with %. To put a '/' mark at the end of wrapped lines you can do

```
\setverbatimbreak{\char'\/}
```

or similarly if you would like another character. Another possibility is

```
\setverbatimchar{\char'\/\char'\*}
```

which will make '/\*' the end marker.

### 15.3.1 Boxed verbatims

Verbatim environments are often used to present program code or, as in this book, LaTeX code. For such applications it can be useful to put the code in a box, or to number the code lines, or perhaps both.

```
\begin{fboxverbatim} anything \end{fboxverbatim}
```

The fboxverbatim environment typesets its contents verbatim and puts a tightly fitting frame around the result; in a sense it is similar to the \fbox command.

```
\begin{boxedverbatim} anything \end{boxedverbatim} \begin{boxedverbatim*}
```

The boxedverbatim and boxedverbatim\* environments are like the verbatim and verbatim\* environments except that a box, allowing page breaks, may be put around the verbatim text and the lines of text may be numbered. The particular format of the output can be controlled as described below.

```
\bvbox \bvtopandtail \bvsides \nobvbox \bvboxsep
```

Four styles of boxes are provided and you can extend these. Following the \bvbox declaration, a box is drawn round the verbatim text, breaking at page boundaries if necessary; this is the default style. Conversely, no boxes are drawn after the \nobvbox declaration. With the \bvtopandtail declaration horizontal lines are drawn at the start and end of the verbatim text, and with the \bvsides declarations, vertical lines are drawn at the left and right of the text. The separation between the lines and the text is given by the length \bvboxsep.

The following hooks are provided to set your own boxing style.

```
\bvtoprulehook \bvtopmidhook \bvendrulehook \bvrightsidehook
```

The macros \bvtoprulehook and \bvendrulehook are called at the start and end of the boxedverbatim environment, and before and after page breaks. The macros \bvleftsidehook and \bvrightsidehook are called at the start and end of each verbatim line. The macro \bvtopmidhook is called after \bvtoprulehook at the start of the environment. It can be used to add some space if \bvtoprulehook is empty.

The command \bvperpagetrue indicates that a box should be visibly broken at a page-break, while there should be no visible break for \bvperpagefalse. If the box continues on to another page then it may be advantageous to place some sort of heading before the verbatim continues. Following the declaration \bvperpagetrue the  $\langle text \rangle$  argument to \bvtopofpage will be typeset after any pagebreak. For example you could set:

```
\bvtopofpage{continued}
to print 'continued' in the normal text font.
By default, the class sets
\bvendofpage{\hrule\kern-.4pt}
```

which causes the \hrule to be drawn at the end of a page as the visible break (the rule is 0.4pt thick and the kern backs up that amount after the rule, so it effectively takes no vertical space). This is not always suitable. For instance, if there will be a 'continued' message at the top of the following page it may seem odd to draw a line at the bottom of the previous page. In this case, setting

```
\bvendofpage{}
```

will eliminate the rule.

As examples of the use of these hooks, here is how some of the boxed verbatim styles are defined.

The default style is \bvbox, which puts separate full boxes on each page.

```
\newcommand{\bvbox}{%
  \bvperpagetrue
  \renewcommand{\bvtoprulehook}{\hrule \nobreak \vskip-.1pt}%
  \renewcommand{\bvleftsidehook}{\vrule}%
  \renewcommand{\bvrightsidehook}{\vrule}%
  \renewcommand{\bvendrulehook}{\hrule}%
  \renewcommand{\bvendrulehook}{\hrule}%
}
```

The \nobvbox turns off all boxing, and is defined as

```
\newcommand{\nobvbox}{%
  \bvperpagefalse
  \renewcommand{\bvtoprulehook}{}%
  \renewcommand{\bvleftsidehook}{}%
  \renewcommand{\bvrightsidehook}{}%
  \renewcommand{\bvendrulehook}{}%
  \renewcommand{\bvtopmidhook}{\rule{0pt}{2\fboxsep} \hss}%
}
```

The definitions of the other styles, \bvtopandtail and \bvsides, are intermediate between \bvbox and \nobvbox in the obvious manner.

The command \linenumberfrequency controls the numbering of lines in a boxedverbatim — every  $\langle nth \rangle$  line will be numbered. If  $\langle nth \rangle$  is 0 or less, then no lines are numbered, if  $\langle nth \rangle$  is 1 then each line is numbered, and if  $\langle nth \rangle$  is n, where n is 2 or more, then only every nth line is numbered. Line numbering is continuous from one instance of the boxedverbatim environment to the next. Outside the environment the line numbers can be reset at any time by the command \resetbvlinenumber.

The \setbvlinenums macro can be used to specify that the number of the first line of the following boxedverbatim shall be  $\langle first \rangle$  and the first printed number shall be  $\langle startat \rangle$ .

The  $\label{linenumberfont}$  declaration sets  $\langle font\ declaration \rangle$  as the font for the line numbers. The default specification for this is:

```
\linenumberfont{\footnotesize\rmfamily}
```

Line numbers are always set at the left of the lines because there is no telling how long a line might be and it might clash with a line number set at the right.

```
\bvnumbersinside
\bvnumbersoutside
```

Line numbers are typeset inside the box after the declaration \bvnumberinside and are typeset outside the box after the declaration \bvnumbersoutside. The default is to print the numbers inside the box.

Verbatim tabbing, but not wrapping, applies to the boxedverbatim environment.

#### Recommended alternative

Again the tcolorbox package offers boxes vs verbatim text.

### 15.3.2 New verbatims

The class implementation of verbatims lets you define your own kind of verbatim environment. Unfortunately this is not quite as simple as saying

```
\newverbatim{myverbatim}{...}{...}
```

as you can for defining normal environments. Instead, the general scheme is

```
\newenvironment{myverbatim}%
{<non-verbatim stuff> \verbatim <more non-verbatim stuff>}%
{\endverbatim}
```

In particular, you cannot use either the \begin or \end macros inside the definition of the new verbatim environment. For example, the following code will not work

```
\newenvironment{badverbatim}%
   {NBG\begin{verbatim}} {\end{verbatim}}

and this won't work either
   \newenvironment{badverbatim}%
   {\begin{env}\verbatim}{\endverbatim\end{env}}
```

And, as with the standard verbatim environment, you cannot use the new one in the definition of a new command.

For an example of something that does work, this next little piece of typesetting was done in a new verbatim environment I have called verbexami, which starts and ends with a horizontal rule, and it shows the definition of verbexami.

The verbexami environment

```
\newenvironment{verbexami}%
  {\par\noindent\hrule The verbexami environment \verbatim}%
  {\endverbatim\hrule}
```

And this is a variation on the theme, with the environment again being enclosed by horizontal rules.

```
Verbexamii
```

```
Is THIS FUN?
\newenvironment{verbexamii}%
  {\vspace{0.5\baselineskip}\hrule \vspace{0.2\baselineskip}}
  Verbexamii \verbatim \textsc{Is this fun?}}%
  {\endverbatim\hrule\vspace{0.3\baselineskip}}
```

As no doubt you agree, these are not memorable examples of the typesetter's art but do indicate that you can define your own verbatim environments and may need to take a bit of care to get something that passes muster.

I will give some more useful examples, but mainly based on environments for writing verbatim files as I think that these provide a broader scope.

#### 15.3.3 Example: the lcode environment

In this manual all the example LaTeX code has been typeset in the lcode environment; this is a verbatim environment defined especially for the purpose. Below I describe the code for defining my lcode environment, but first here is a simple definition of a verbatim environment, which I will call smallverbatim, that uses the \small font instead of the normalsize font.

```
\newenvironment{smallverbatim}%
  {\setverbatimfont{\normalfont\ttfamily\small}%
   \verbatim}%
  {\endverbatim}
```

The verbatim environment is implemented as a kind of trivlist, and lists usually have extra vertical space before and after them. For my environment I did not want any extra spacing so I defined the macro \@zeroseps to zero the relevant list spacings. I also wanted the code lines to be inset a little, so I defined a new length called \gparindent to use as the indentation.

Unless you are intimately familiar with the inner workings of the verbatim processing you deserve an explanation of the lcode definition.

Extremely roughly, the code for \verbatim looks like this:

```
\def\verbatim{%
  \verbatim@font
  % for each line, until \end{verbatim}
  \verbatim@startline
   % collect the characters in \verbatim@line
  \verbatim@processline{\the\verbatim@line\par}
   % repeat for the next line
}
```

The code first calls \verbatim@font to set the font to be used. Then, for each line it does the following:

- Calls the macro \verbatim@startline to start off the output version of the line.
- Collects all the characters comprising the line as a single token called \verbatim@line.
- If the characters are the string '\end{verbatim}' it finishes the verbatim environment.
- Otherwise it calls the macro \verbatim@processline whose argument is the characters in the line, treated as a paragraph. It then starts all over again with the next line.

I configured the \verbatim@startline macro to indent the line of text using a horizontal skip of \gparindent. The rest of the initialisation code, before calling \verbatim to do the real processing, just sets up the vertical spacing.

### 15.4 Files

LaTeX reads and writes various files as it processes a document. Obviously it reads the document source file, or files, and it writes the log file recording what it has done. It also reads and writes the aux file, and may read and write other files like a toc file.

On occasions it can be useful to get LaTeX to read and/or write other files of your own choosing. Unfortunately standard LaTeX does not provide any easy method for doing this. The memoir class tries to rectify this.

```
\jobname
```

When you run LaTeX on your source file, say fred.tex, LaTeX stores the name of this file (fred) in the macro \jobname. LaTeX uses this to name the various files that it writes out — the dvi or pdf file, the log file, the aux file, etc.

TeX can read from 16 input streams and can write to 16 output streams. Normally an input stream is allocated for each kind of file that will be read and an output stream for each kind of file that will be written. On the input side, then, at least two streams are allocated, one for the source tex file and one for the aux file. On the output side again at least two streams are allocated, one for the log file and one for the aux file. When toc and other similar files are also part of the LaTeX process you can see that many of the 16 input and output streams may be allocated before you can get to use one yourself.

The macros \newoutputstream and \newinputstream respectively create a new output and input stream called  $\langle stream \rangle$ , where  $\langle stream \rangle$  should be a string of alphabetic characters, like myout or myin. The  $\langle stream \rangle$  names must be unique, you cannot use the same name for two streams even if one is a input stream and the other is an output stream. If all the 16 streams of the given type have already been allocated TeX will issue a message telling you about this, of the form:

```
No room for a new write % for an output stream
No room for a new read % for an input stream
```

The two \new...stream commands also provide two empty macros called \atstreamopen<stream> and \atstreamclose<stream>. If these macros already exist then they are left undisturbed. For example if you do:

```
\newcommand{\atstreamopenmyout}{...}
\newoutputstream{myout}
\newinputstream{myin}
```

Then you will find that three new commands have been created like:

```
\newcommand{\atstreamclosemyout}{}
\newcommand{\atstreamopenmyin}{}
\newcommand{\atstreamclosemyin}{}
```

You can use \renewcommand to change the definitions of these if you wish.

```
\label{lifstreamOpen} $$ \IfStreamOpen{\langle stream\rangle} {\langle true-code\rangle} {\langle false-code\rangle} $$
```

The macro \IfStreamOpen checks whether or not the  $\langle stream \rangle$  stream is open. If it is then the  $\langle true\text{-}code \rangle$  argument is processed, while when it is not open the  $\langle false\text{-}code \rangle$  argument is processed.

#### 15.4.1 Writing to a file

One stream may be used for writing to several different files, although not simultaneously.

```
\label{loss} $$\operatorname{\operatorname{const}}_{\langle stream\rangle}$$ $$\operatorname{\operatorname{const}}_{\langle stream\rangle}$$
```

The command \openoutputfile opens the file called  $\langle filename \rangle$ , either creating it if it does not exist, or emptying it if it already exists. It then attaches the file to the output stream called  $\langle stream \rangle$  so that it can be written to, and then finally calls the macro named \atstreamopen<stream>.

The command  $\closeoutputstream$  firstly calls the macro named  $\atstream\close<$  stream> then closes the output stream  $\alpha$ , and finally detaches and closes the associated file.

```
\addtostream{\langle stream \rangle}{\langle text \rangle}
```

The \addtostream command writes  $\langle text \rangle$  to the output stream  $\langle stream \rangle$ , and hence to whatever file is currently attached to the stream. The  $\langle stream \rangle$  must be open. Any commands within the  $\langle text \rangle$  argument will be processed before being written. To prevent command expansion, precede the command in question with \protect.

Writing verbatim text to a file is treated specially as it is likely to be the most common usage.

```
\begin{verbatimoutput}{\langle file \rangle} anything \end{verbatimoutput} \begin{writeverbatim}{\langle stream \rangle} anything \end{writeverbatim}
```

The text within a verbatimoutput environment is written verbatim to the  $\langle file \rangle$  file. Alternatively, the contents of the writeverbatim environment are written verbatim to the  $\langle stream \rangle$  stream.

Specifically, verbatimoutput opens the specified file, writes to it, and then closes the file. This means that if verbatimoutput is used more than once to write to a given file, then only the contents of the last of these outputs is captured in the file. On the other hand, you can use writeverbatim several times to write to the file attached to the stream and, providing the stream has not been closed in the meantime, all will be captured.

#### 15.4.2 Reading from a file

One stream may be used for reading from several files, although not simultaneously.

```
\label{losseinputfile} $$\operatorname{diename}_{{\langle stream \rangle}}$$ $$ \operatorname{diename}_{{\langle stream \rangle}}$$
```

The command \openinputfile opens the file called  $\langle filename \rangle$  and attaches it to the input stream called  $\langle stream \rangle$  so that it can be read from. Finally it calls the macro named \atstreamopen<stream>. It is an error if  $\langle filename \rangle$  can not be found.

The command  $\closeinputstream$  calls the macro named  $\atstreamclose<stream>$ , closes the output stream  $\slashed{stream}$ , and then detaches and closes the associated file.

```
\rack {stream}
```

The command \readstream reads the entire contents of the file currently associated with the input stream  $\langle stream \rangle$ . This provides the same functionality as \input{ $\langle filename \rangle$ }.

```
\response \
```

The \readaline reads what TeX considers to be one line from the file that is currently associated with the input stream \( \stream \).

Multiple lines can be read by calling \readaline multiple times. A warning is issued if there are no more lines to be read (i.e., the end of the file has been reached).

Just as for writing, reading files verbatim is treated specially.

The commands \verbatiminput and \boxedverbatiminput, and their starred versions, act like the verbatim and boxedverbatim environments, except that they get their text from the  $\langle file \rangle$  file. It is an error if  $\langle file \rangle$  cannot be found. Similarly, \readverbatim and \readboxedverbatim get their text from the file currently attached to the  $\langle stream \rangle$  input stream. It is an error if  $\langle stream \rangle$  is not open for input.

### 15.4.3 Example: endnotes

In an earier version of the manual, this section contained an example as to how one could make endnotes. The example is now irrelevant, since memoir contain something similar to end notes called page notes, see section 12.6 on page 243.

Those interested in the code from the old example, can find it in the manual source (it has just been commented out).

#### 15.4.4 Example: end floats

There are some documents where all figures are required to be grouped in one place, for instance at the end of the document or perhaps at the end of each chapter. Grouping at the end of a document with chapters is harder, so we'll tackle that one.

The basic idea is to write out verbatim each figure environment and then read them all back in at the end. We will use a stream, let's call it tryout, and call our file for figures tryout.fig.

```
\newoutputstream{tryout}
\openoutputfile{tryout.fig}{tryout}

If all were simple, in the document we could then just do
\begin{writeverbatim}{tryout}
\begin{figure} ... \end{figure}
\end{writeverbatim}
...
\closeoutputstream{tryout}
\input{tryout.fig}
```

So, what's the problem?

By default figure captions are numbered per chapter, and are preceded by the chapter number; more precisely, the definition of a figure number is

```
\thechapter.\arabic{figure}
```

If we simply lump all the figures at the end, then they will all be numbered as if they were in the final chapter. For the sake of argument assume that the last chapter is number 10. The nth figure will then be numbered 10.n. One thing that we can do rather simply is to change the definition of the figure by using another counter, let's call it pseudo, instead of the chapter.

```
\newcounter{pseudo}
\renewcommand{\thepseudo}{\arabic{pseudo}}
\renewcommand{\thefigure}{\thepseudo.\arabic{figure}}
```

Now, all we should have to do is arrange that the proper value of pseudo is available before each figure is typeset at the end. The code around the figure environments might then look like this

```
\addtostream{tryout}{\protect\setcounter{pseudo}{\thechapter}}
\begin{writeverbatim}{tryout}
\begin{figure}...
and a part of the file might then look like
...
\setcounter{pseudo}{4}
\begin{figure}...
```

The \protect before the \setcounter command will stop it from expanding before it is written to the file, while the \thechapter command will be expanded to give the actual number of the current chapter. This looks better as now at least the figure will be numbered 4.n instead of 10.n.

There is one last snag — figure numbers are reset at the start of each chapter — but if we just dump the figures at the end of the document then although the chapter part of the number will alter appropriately because of the pseudo process, the second part of the number will just increase continuously. It looks as though we should write out a change to the chapter counter at the start of each chapter. If we do that, then we should be able to get rid of the pseudo counter, which sounds good. But, and this is almost the last but, what if there are chapters after we have read in the figure file? To cater for this the chapter number of the last chapter before the file must be saved, and then restored after the figures have been processed.

Finally, wouldn't it be much better for the user if everything was wrapped up in an environment that handled all the messy stuff?

Here is the final code that I am going to produce which, by the way, is displayed in the boxedverbatim environment with line numbers and the following settings, just in case there is a page break in the middle of the box.

```
\bvendofpage{}
   \resetbvlinenumber
   \linenumberfrequency{1}
   \bvnumbersoutside
   \linenumberfont{\footnotesize\rmfamily}
   \begin{boxedverbatim}
    \newoutputstream{tryout}
1
    \openoutputfile{\jobname.fig}{tryout}
2
   \newcounter{pseudo}
3
4
   \renewcommand{\thefigure}{\thepseudo.\arabic{figure}}
5
    \newenvironment{writefigure}{%
      \ifnum\value{chapter}=\value{pseudo}\else
6
        \setcounter{pseudo}{\value{chapter}}
        \addtostream{tryout}{\protect\stepcounter{chapter}}
8
9
        \addtostream{tryout}{\protect\addtocounter{chapter}{-1}}
10
        \addtostream{tryout}{%
          \protect\setcounter{pseudo}{\thechapter}}
11
      \fi
12
      \addtostream{tryout}{\protect\begin{figure}}
13
      \writeverbatim{tryout}}%
14
15
     {\endwriteverbatim\finishwritefigure}
    \newcommand{\finishwritefigure}{%
16
      \addtostream{tryout}{\protect\end{figure}}}
17
    \newcommand{\printfigures}{%
18
      \closeoutputstream{tryout}%
19
20
      \input{\jobname.fig}%
   }
21
```

The above code should be either put in the preamble or in a separate package file.

The first four lines of the code perform the initial setup described earlier. Lines 1 and 2 set up for outputting to a file \jobname.fig, which is where the figures will be collected. Lines 3 and 4 create the new counter we need and change the construction of the figure number. The rest of the code defines a new environment writefigure which is to be used instead of the figure environment. It writes its content out to the tryout stream.

In line 6 a check is made to see if the current values of the chapter and pseudo counters are the same; nothing is done if they are. If they are different, it means that this is the first figure in the chapter and we have to put appropriate information into the figure file. Line 7 sets the pseudo counter to the value of the chapter counter (if there is another writefigure in the chapter it will then skip over the code in lines 7 to 11). The next lines put (where N is the number of the current chapter):

```
\stepcounter{chapter}
\addtocounter{chapter}{-1}
\setcounter{pseudo}{N}
```

into the figure file. Stepping the chapter number (by one) resets the following figure number, and then subtracting one from the stepped number returns the chapter number to its original value. Finally the counter pseudo is set to the number of the current chapter.

```
Line 13 puts \begin{figure}
```

into the figure file, and line 14 starts the writeverbatim environment.

For the end of the writefigure environment (line 15), the writeverbatim environment is ended and after that the \finishwritefigure macro is called. This is defined in lines 16 and 17, and simply writes

```
\end{figure}
```

out to the figure file. The \endwriteverbatim, and any other kind of \end...verbatim, command is very sensitive to anything that follows it, and in this case did not like to be immediately followed by an \addtostream{...}, but did not mind it being wrapped up in the \finishwritefigure macro.

The \printfigures macro defined in the last three lines of the code simply closes the output stream and then inputs the figures file.

As an example of how this works, if we have the following source code:

```
\chapter{The fifth chapter}
...
\begin{writefigure}
%% illustration and caption
\end{writefigure}
...
\begin{writefigure}
%% another illustration and caption
\end{writefigure}
```

then the figure file will contain the following (shown verbatim in the fboxverbatim environment).

```
\stepcounter{chapter}
\addtocounter{chapter}{-1}
\setcounter{pseudo}{5}
\begin{figure}
%% illustration and caption
\end{figure}
\begin{figure}
%% another illustration and caption
\end{figure}
```

#### 15.4.5 Example: questions and answers

Text books often have questions at the end of a chapter. Sometimes answers are also provided at the end of the book, or in a separate teachers guide. During the draft stages of such a book it is useful to keep the questions and answers together in the source and paper drafts, only removing or repositioning the answers towards the end of the writing process.

This example provides an outline for meeting these desires. For pedagogical purposes I use a \label and \ref technique although there are better methods. The example also

shows that not everything works as expected — it is a reasonably accurate rendition of the process that I actually went through in designing it.

First we need a counter for the questions and we'll use an environment for questions as these may be of any complexity. The environment takes one argument — a unique key to be used in a \label.

```
\newcounter{question} \setcounter{question}{0}
\renewcommand{\thequestion}{\arabic{question}}
\newenvironment{question}[1]%
   {\refstepcounter{question}
    \par\noindent\textbf{Question \thequestion:}\label{#1}}%
   {\par}
```

I have used \refstepcounter to increment the counter so that the \label will refer to it, and not some external counter.

We will use a file, called \jobname.ans to collect the answers and this will be written to by a stream. There is also a convenience macro, \printanswers, for the user to call to print the answers.

```
\newoutputstream{ansout}
```

A matching environment for answers is required. The argument to the environment is the key of the question.

In draft mode it is simple, just typeset the answer and no need to bother with any file printing (remember that \ifdraftdoc is true for a draft mode document).

In final mode the answer environment must write its contents verbatim to the ans file for printing by \printanswers. Dealing with these in reverse order, this is the definition of \printanswer when not in draft mode.

```
\newcommand{\printanswers}{%
  \closeoutputstream{ansout}
  \input{\jobname.ans}}
```

Now for the tricky bit, the answer environment. First define an environment that makes sure our output stream is open, and which then writes the answer title to the stream.

```
\newenvironment{@nswer}[1]{\@bsphack
  \IfStreamOpen{ansout}{}{%
    \openoutputfile{\jobname.ans}{ansout}%
  }%
  \addtostream{ansout}{\par\noindent\textbf{Answer \ref{#1}:}}%
  }{\@esphack}
```

The macros \@bsphack and \@esphack are LaTeX kernel macros that will gobble extraneous spaces around the environment. In other words, this environment will take no space in the

typeset result. The \IfStreamOpen macro is used to test whether or not the stream is open, and if it isn't then it opens it. The answer title is then written out to the stream. Now we can define the answer environment so that its contents get written out verbatim.

When I was testing this code I had a surprise as I got nasty error messages from LaTeX the first time around, but it worked fine when I processed the source a second time! The problem lies in the code line

```
\addtostream{ansout}{\par\noindent\textbf{Answer \ref{#1}:}}%
```

The first time around, LaTeX processed the \ref command and of course it was undefined. In this case \ref gets replaced by the code to print the error message, which involves macros that have @ in their names, which LaTeX only understands under special circumstances. The second time around \ref gets replaced by the question number and all is well. I then remembered that some commands need protecting when they are written out, so I tried (I've wrapped the line to fit)

```
\addtostream{ansout}{\par\noindent
  \protect\makeatletter\textbf{Answer
  \protect\ref{#1}:}\protect\makeatother}%
```

which did work but seemed very clumsy.

I then took another line of attack, and looked at the definition of \ref to see if I could come up with something that didn't expand into @ names. The result of this was

```
\addtostream{ansout}{\par\noindent\textbf{Answer} \quietref{#1}:}}%
```

In the kernel file ltxref.dtx I found the definition of \ref and it used a macro \@setref (shown below) to do its work. My \quietref locally changes the definition of \@setref and then calls \ref, which will then use the modified \@setref.

```
\def\@setref#1#2#3{%
                            %% kernel definition
 \int x#1\relax
    \protect\G@refundefinedtrue
    \nfss@text{\reset@font\bfseries ??}%
    \@latex@warning{Reference '#3' on page \thepage \space
                    undefined}%
 \else
    \expandafter#2#1\null
 \fi}
\DeclareRobustCommand{\quietref}[1]{\begingroup
 \def\@setref##1##2##3{%
   \ifx##1\relax ??\else
      \expandafter##2##1\null
   \fi
 \ref{#1}\endgroup}
```

Having gone all round the houses, the simplest solution was actually one that I had skipped over

The advantage of using the \label and \ref mechanism is that a question and its answer need not be adjacent in the source; I think that you have seen some of the disadvantages. Another disadvantage is that it is difficult to use, although not impossible, if you want the answers in a separate document.

The real answer to all the problems is force an answer to come immediately after the question in the source and to use the question counter directly, as in the endnotes example. In the traditional manner, this is left as an exercise for the reader.

#### 15.5 Answers

**Question 1.** As a convenience, the argument to the environment could be made optional, defaulting, say, to the current line width. If the default width is used the frame will be wider than the line width, so we really ought to make the width argument specify the width of the frame instead of the minipage. This means calculating a reduced width for the minipage based on the values of \fboxsep and \fboxrule.

```
\newsavebox{\minibox}
\newlength{\minilength}
\newenvironment{framedminipage}[1][\linewidth]{%
\setlength{\minilength}{#1}
\addtolength{\minilength}{-2\fboxsep}
\addtolength{\minilength}{-2\fboxrule}
\begin{lrbox}{\minibox}\begin{\minipage}{\minilength}}%
{\end{\minipage}\end{\linex}\fbox{\usebox{\minibox}}}
```

**Question 2.** There are at least three reasonable answers. In increasing or decreasing order of probability (your choice) they are:

- I took Sherlock Holmes' advice and followed the methods outlined in the chapter;
- I used a package, such as the answer package which is designed for the purpose;
- I just wrote the answers here.

# Sixteen

# Cross referencing

A significant aspect of LaTeX is that it provides a variety of cross referencing methods, many of which are automatic. An example of an automatic cross reference is the way in which a \chapter command automatically adds its title and page number to the ToC, or where a \caption adds itself to a 'List of...'.

Some cross references have to be specifically specified, such as a reference in the text to a particular chapter number, and for these LaTeX provides a general mechanism that does not require you to remember the particular number and more usefully does not require you to remember to change the reference if the chapter number is later changed.

#### 16.1 Labels and references

The general LaTeX cross reference method uses a pair of macros.

You can put a \label command where you want to label some numbered element in case you might want to refer to the number from elsewhere in the document. The  $\langle labstr \rangle$  argument is a sequence of letters, digits, and punctuation symbols; upper and lower case letters are different. The \ref macro prints the number associated with  $\langle labstr \rangle$ . The \pageref macro prints the number of the page where the \label specifying the  $\langle labstr \rangle$  was placed.

The \label and \ref mechanism is simple to use and works well but on occasions you might be surprised at what \ref prints.

LaTeX maintains a current ref value which is set by the \refstepcounter command. This command is used by the sectioning commands, by \caption, by numbered environments like equation, by \item in an enumerate environment, and so on. The \label command writes an entry in the aux file consisting of the  $\langle labstr \rangle$ , the current ref value and the curent page number. A \ref command picks up the ref value for  $\langle labstr \rangle$  and prints it. Similarly \pageref prints the page number for  $\langle labstr \rangle$ .

The critical point is that the  $\label$  command picks up the value set by the *most recent* visible  $\label$   $\label$ 

- A \label after a \section picks up the \section number, not the \chapter number.
- A \label after a \caption picks up the caption number.
- A \label before a \caption picks up the surrounding sectional number.

<sup>1</sup> Remember, a change within a group, such as an environment, is not visible outside the group.

If you are defining your own macro that sets a counter, the counter value will be invisible to any \label unless it is set using \refstepcounter.

```
\label{linear_label} $$ \left( \langle labstr \rangle \right) \to \left( \langle labstr \rangle \right) \to \left( \langle labstr \rangle \right) $$ pagerefname $$ \left( \langle labstr \rangle \right) \to \left( \langle labstr \rangle \right) $$
```

The class provides these more particular named references to a figure, table or page. For example the default definitions of \fref and \pref are

```
\newcommand{\fref}[1]{\figurerefname~\ref{#1}}
\newcommand{\pref}[1]{\pagerefname~\pageref{#1}}
and can be used as
\ldots footnote parameters are shown in~\fref{fig:fn}
on~\pref{fig:fn}.
```

which in this document prints as:

Similarly, specific commands are supplied for referencing sectional divisions; \Aref for Appendix , \Bref for Book , \Pref for Part , \Cref for Chapter , and \Sref for divisions below Chapter . For example:

```
This is \Sref{sec:lab&ref} in \Cref{chap:xref}.

This is \$16.1 in Chapter 16.
```

#### 16.2 Reference by name

In technical works it is common to reference a chapter, say, by its number. In non-technical works such cross references are likely to be rare, and when they are given it is more likely that the chapter title would be used instead of the number, as in:

```
The chapter \textit{\titleref{chap:bringhurst}} describes \ldots
```

The chapter *An example book design* describes ···

There are two packages, nameref [Rahtz01] and titleref [Ars01a], that let you refer to things by name instead of number.

Name references were added to the class as a consequence of adding a second optional argument to the sectioning commands. I found that this broke the nameref package, and hence the hyperref package as well, so they had to be fixed. The change also broke Donald Arseneau's titleref package, and it turned out that nameref also clobbered titleref. The class also provides titles, like \poemtitle, that are not recognised by either of the packages. From my viewpoint the most efficient thing to do was to enable the class itself to provide name referencing.

```
	ext{ \titleref} \{ \langle labstr \rangle \}
```

Typeset example 16.1: Named references should be to titled elements

Labels may be applied to:

- 1. Chapters, sections, etc.
- 2. Captions
- 3. Legends
- 4. Poem titles
- 5. Items in numbered lists, etc. ...

Item 1 in section Reference by name mentions sections while item Named references should be to titled elements, on page 295 in the same section, mentions things like items in enumerated lists that should not be referenced by \titleref.

The macro \titleref is a class addition to the usual set of cross referencing commands. Instead of printing a number it typesets the title associated with the labelled number. This is really only useful if there *is* a title, such as from a \caption or \section command. For example, look at this code and its result.

```
Labels may be applied to:
\begin{enumerate}
\item Chapters, sections, etc. \label{sec:nxref:1}
...
\item Items in numbered lists, etc. \ldots \label{sec:nxref:5}
\end{enumerate}
Item \ref{sec:nxref:1} in section \textit{\titleref{sec:nxref}}
mentions sections while item \titleref{sec:nxref:5}, on page
\pageref{sec:nxref:5} in the same section, mentions things like
items in enumerated lists that should not be referenced
by \verb?\titleref?.
```

As the above example shows, you have to be a little careful in using \titleref. Generally speaking, \titleref{ $\langle key \rangle$ } produces the last named thing before the \label that defines the  $\langle key \rangle$ .

```
\headnameref \tocnameref
```

There can be three possibilities for the name of a sectional division; the full name, the name in the ToC, and the name in the page header. As far as \titleref is concerned it does not use the fullname, and so the choice simplifies to the ToC or page header. Following the declaration \headnameref it uses the page header name. Following the opposite declaration \tocnameref, which is the default, it uses the ToC name.

NOTE: Specifically with the memoir class, do not put a \label command inside an

### Typeset example 16.2: Current title

This sentence in the section titled 'Current title' is an example of the use of the command \currenttitle.

argument to a \chapter or \section or \caption, etc., command. Most likely it will either be ignored or referencing it will produce incorrect values. This restriction does not apply to the standard classes, but in any case I think it is good practice not to embed any \label commands.

#### \currenttitle

If you just want to refer to the current title you can do so with \currenttitle. This acts as though there had been a label associated with the title and then \titleref had been used to refer to that label. For example:

### Source for example 16.2

This sentence in the section titled '\currenttitle' is an example of the use of the command \verb?\currenttitle?.

# \theTitleReference $\{\langle num \rangle\}\{\langle text \rangle\}$

Both \titleref and \currenttitle use the \theTitleReference to typeset the title. This is called with two arguments — the number,  $\langle num \rangle$ , and the text,  $\langle text \rangle$ , of the title. The default definition is:

\newcommand{\theTitleReference}[2]{#2}

so that only the  $\langle \textit{text} \rangle$  argument is printed. You could, for example, change the definition to

\renewcommand{\theTitleReference}[2]{#1\space \textit{#2}}

to print the number followed by the title in italics. If you do this, only use \titleref for numbered titles, as a printed number for an unnumbered title (a) makes no sense, and (b) will in any case be incorrect.

The commands  $\titleref$ ,  $\titleref$  and  $\titleref$  are direct equivalents of those in the titleref package [Ars01a].

# \namerefon \namerefoff

The capability for referencing by name has one potentially unfortunate side effect — it causes some arguments, such as that for \legend, to be moving arguments and hence any fragile command in the argument will need \protecting. However, not every document will require the use of \titleref and so the declaration \namerefoff is provided to switch it off (the argument to \legend would then not be moving). The declaration \namerefon, which is the default, enables name referencing.

# Seventeen

# Back matter

The back matter consists of reference and supportive elements for the main matter; things like bibliographies, indexes, glossaries, endnotes, and other material. The class provides additional elements and features of such matter that are not in the standard classes.

# 17.1 Bibliography

Just as a reminder the bibliography is typeset within the thebibliography environment.

```
\bibname
\begin{thebibliography}{\(\langle exlabel\)\}
\bibitem ...
\end{thebibliography}
```

The environment takes one required argument,  $\langle exlabel \rangle$ , which is a piece of text as wide as the widest label in the bibliography. The value of \bibname (default 'Bibliography') is used as the title.

```
\bibintoc \nobibintoc
```

The declaration \bibintoc will cause the thebibliography environment to add the title to the ToC, while the declaration \nobibintoc ensures that the title is not added to the ToC. The default is \bibintoc.

```
\cite[\langle detail \rangle] \{ \langle labstr-list \rangle \}
```

Within the text you call out a bibliographic reference using the \cite command, where  $\langle labstr-list \rangle$  is a comma-separated list of identifiers for the cited works; there must be no spaces in this list. The optional  $\langle detail \rangle$  argument is for any additional information regarding the citation such as a chapter or page number; this is printed after the main reference.

Various aspects of a bibliography can be changed and at this point it may be helpful to look at some of the internals of the thebibliography environment, which is defined like this

```
\newenvironment{thebibliography}[1]{%
  \bibsection
  \begin{bibitemlist}{#1}}%
  {\end{bibitemlist}\postbibhook}
```

The bibliographic entries are typeset as a list, the bibitemlist.

```
\bibsection
```

The macro \bibsection defines the heading for the thebibliography environment; that is, everything before the actual list of items. It is effectively defined as

```
\newcommand{\bibsection}{%
  \chapter*{\bibname}
  \bibmark
  \ifnobibintoc\else
    \phantomsection
    \addcontentsline{toc}{chapter}{\bibname}
  \fi
  \prebibhook}
```

If you want to change the heading, redefine \bibsection. For example, to have the bibliography as a numbered section instead of an unnumbered chapter, redefine it like

```
\renewcommand{\bibsection}{%
  \section{\bibname}
  \prebibhook}
```

If you use the natbib [Dal99a] and/or the chapterbib [Ars01b] packages with the sectionbib option, then they change \bibsection appropriately to typeset the heading as a numbered section.

```
\bibmark
```

\bibmark may be used in pagestyles for page headers in a bibliography. Its default definition is:

```
\newcommand*{\bibmark}{}
```

but could be redefined like, say,

\renewcommand\*{\bibmark}{\markboth{\bibname}{}}

```
\prebibhook \postbibhook
```

The commands \prebibhook and postbibhook are called after typesetting the title of the bibliography and after typesetting the list of entries, respectively. By default, they are defined to do nothing. You may wish to use one or other of these to provide some general information about the bibliography. For example:

```
\label{lem:command} $$ \operatorname{CTAN} is the Comprehensive \tx\ Archive Network and URLS for the several CTAN mirrors can be found at \url{http://www.tug.org}.}
```

```
\biblistextra
```

Just at the end of setting up the bibitemlist the \biblistextra command is called. By default this does nothing but you may change it to do something useful. For instance, it can be used to change the list parameters so that the entries are typeset flushleft.

```
\renewcommand*{\biblistextra}{%
  \setlength{\leftmargin}{0pt}%
  \setlength{\itemindent}{\labelwidth}%
  \addtolength{\itemindent}{\labelsep}}
```

```
\setbiblabel\{\langle style \rangle\}
```

The style of the labels marking the bibliographic entries can be set via \setbiblabel. The default definition is

```
\setbiblabel{[#1]\hfill}
```

where #1 is the citation mark position, which generates flushleft marks enclosed in square brackets. To have marks just followed by a dot

\setbiblabel{#1.\hfill}

```
\label{label} $$ \left( \langle label \rangle \right] \left( \langle labstr \rangle \right) $$ \newblock
```

Within the bibitemlist environment the entries are introduced by \bibitem instead of the more normal \item macro. The required  $\langle labstr \rangle$  argument is the identifier for the citation and corresponds to a  $\langle labstr \rangle$  for \cite. The items in the list are normally labelled numerically but this can be overriden by using the optional  $\langle label \rangle$  argument. The \newblock command can be used at appropriate places in the entry for encouraging a linebreak (this is used by the openbib option).

```
\bibitemsep
```

In the listing the vertical space between entries is controlled by the length \bibitemsep, which by default is set to the normal \itemsep value. The vertical space is (\bibitemsep + \parsep). If you wish to eliminate the space between items do

\setlength{\bibitemsep}{-\parsep}

# 17.1.1 BibTex

Often the BibTeX program [Pat88a] is used to generate the bibliography list from database(s) of bibliographic data. For BibTeX a bibliographic data base is a bib file containing information necessary to produce entries in a bibliography. BibTeX extracts the raw data from the files for each citation in the text and formats it for typesetting according to a particular style.

The bibliography will be printed at the location of the \bibliography command. Its argument is a comma-separated list of BibTeX bib files which will be searched by BibTeX to generate the bibliography. Only the file name(s) should be supplied, the extension must not be given.

```
\nocite{\langle labstr \rangle} \nocite{*}
```

The command \nocite causes BibTeX to make an entry in the bibliography but no citation will appear in the text. The special case \nocite{\*} causes every entry in the database to be listed in the bibliography.

```
\begin{tabular}{ll} \beg
```

Many different BibTeX styles are available and the particular one to be used is specified by calling \bibliographystyle before the bibliography itself. The 'standard' bibliography \(\style\)\s follow the general schemes for mathematically oriented papers and are:

- plain The entry format is similar to one suggested by Mary-Claire van Leunen [Leu92], and entries are sorted alphabetically and labelled with numbers.
- unsrt The format is the same as plain but that entries are ordered by the citation order in the text.
- alpha The same as plain but entries are labelled like 'Wil89', formed from the author and publication date.

abbrv The same as plain except that some elements, like month names, are abbreviated.

There are many other styles available, some of which are used in collaboration with a package, one popular one being Patrick Daly's natbib [Dal99a] package for the kinds of authoryear citation styles used in the natural sciences. Another package is jurabib [Ber02] for citation styles used in legal documents where the references are often given in footnotes rather than listed at the end of the document.

I assume you know how to generate a bibliography using BibTeX, so this is just a quick reminder. You first run LaTeX on your document, having specified the bibliography style, cited your reference material and listed the relevant BibTeX database(s). You then run BibTeX, and after running LaTeX twice more the bibliography should be complete. After a change to your citations you have to run LaTeX once, BibTeX once, and then LaTeX twice more to get an updated set of references.

The format and potential contents of a BibTeX database file (a bib file) are specified in detail in Lamport [Lam94] and the first of the *Companions* [M $G^+04$ ]. Alternatively there is the documentation by Oren Patashnik [Pat88a] who wrote the BibTeX program.

A BibTeX style, specified in a bst file, is written using an anonymous stack based language created specifically for this purpose. If you can't find a BibTeX style that provides what you want you can either use the makebst [Dal99b] package which leads you through creating your own style via a question and answer session, or you can directly write your own. If you choose the latter approach then Patashnik's *Designing BibTeX files* [Pat88b] is essential reading. As he says, it is best to take an existing style and modify it rather than starting from scratch.

#### 17.2 Index

It is time to take a closer look at indexing. The class allows multiple indexes and an index may be typeset as a single or a double column.

The general process is to put indexing commands into your source text, and LaTeX will write this raw indexing data to an idx file. The raw index data is then processed, not by LaTeX but by yourself if you have plenty of spare time on your hands, or more usually by a separate program, to create a sorted list of indexed items in a second file (usually an ind file). This can then be input to LaTeX to print the sorted index data.

#### 17.2.1 Printing an index

In order to make LaTeX collect indexing information the declaration \makeindex must be put in the preamble. By default the raw index data is put into the jobname.idx file. If the optional \( \frac{file}{\} \) argument is given then index data can be output to file.idx. Several \( \makeindex \) declarations can be used provided they each call for a different file.

The \printindex command will print an index where by default the indexed items are assumed to be in a file called jobname.ind. If the optional  $\langle file \rangle$  argument is given then the indexed items are read from the file called file.ind.

```
\begin{theindex} entries \end{theindex}
\onecolindex \twocolindex
\indexname
```

The index entries are typeset within the theindex environment. By default it is typeset with two columns but following the \onecolindex declaration the environment uses a single column. The default two column behaviour is restored after the \twocolindex declaration. The index title is given by the current value of \indexname (default 'Index').

```
\indexintoc \noindexintoc
```

The declaration \indexintoc will cause the theindex environment to add the title to the ToC, while the declaration \noindexintoc ensures that the title is not added to the ToC. The default is \indexintoc.

```
\indexcolsep
\indexrule
```

The length \indexcolsep is the width of the gutter between the two index columns The length \indexrule, default value 0pt, is the thickness of a vertical rule separating the two columns.

```
\preindexhook
```

The macro \preindexhook is called after the title is typeset and before the index listing starts. By default it does nothing but can be changed. For example

\renewcommand{\preindexhook}{Bold page numbers are used
to indicate the main reference for an entry.}

```
\indexmark
```

\indexmark may be used in pagestyles for page headers in an index. Its default definition is:

\newcommand\*{\indexmark}{}

but could be redefined like, say,

\renewcommand\*{\indexmark}{\markboth{\indexname}{\indexname}}

```
\ignorenoidxfile
\reportnoidxfile
```

Following the declaration \ignorenoidxfile, which is the default, LaTeX will silently pass over attempts to use an idx file which has not been declared via \makeindex. After the declaration \reportnoidxfile LaTeX will whine about any attempts to write to an unopened file.

# 17.2.2 Preparing an index

It it is easy for a computer to provide a list of all the words you have used, and where they were used. This is called a concordance. Preparing an index, though, is not merely a gathering of words but is an intellectual process that involves recognising and naming concepts,

constructing a logical hierarchy of these and providing links between related concepts. No computer can do that for you though it can help with some tasks, such as sorting things into alphabetical order, eliminating duplicates, and so forth.

Several iterations may be required before you have an acceptable index. Generally you pick out the important words or phrases used on the first pass. Part of the skill of indexing is finding appropriate words to describe things that may not be obvious from the text. If there are several ways of describing something they may all be included using a 'see' reference to the most obvious of the terms, alternatively you could use 'see also' references between the items. Entries should be broken down into subcategories so that any particular item will not have a long string of page numbers and your reader is more likely to quickly find the relevant place. After having got the first index you will most probably have to go back and correct all the sins of ommission and commission, and start the cycle again.

I found that indexing this manual was the most difficult part of preparing it. It was easy to index the names of all the macros, environments, and so on as I had commands that would simultaneously print and index these. It was the concepts that was difficult. I inserted \index commands as I went along at what seemed to be appropriate places but turned out not to be. I would use slightly different words for the same thing, and what was worse the same word for different things. It took a long time to improve it to its present rather pitiful state.

```
\lceil \lceil (file) \rceil \{ \langle stuff \rangle \}
```

The \index macro specifies that  $\langle stuff \rangle$  is to appear in an index. By default the raw index data — the  $\langle stuff \rangle$  and the page number — will be output to the jobname.idx file, but if the optional  $\langle file \rangle$  argument is given then output will be to the file.idx file.

This book has two indexes. The main index uses the default indexing commands, while the second index does not. They are set up like this:

```
% in preamble
\makeindex
\makeindex[lines]
% in body
...\index{main} ...\index[lines]{First line} ...
. . .
% at the end
\clearpage
% main index
\pagestyle{Index}
\renewcommand{\preindexhook}{%
The first page number is usually, but not always,
the primary reference to the indexed topic.\vskip\onelineskip}
\printindex
% second index
\clearpage
\pagestyle{ruled}
\renewcommand{\preindexhook}{}
\renewcommand{\indexname}{Index of first lines}
\onecolindex
```

\printindex[lines]

```
\specialindex{\langle file\rangle}{\langle counter\rangle}{\langle stuff\rangle}
```

The \index command uses the page number as the reference for the indexed item. In contrast the \specialindex command uses the  $\langle counter \rangle$  as the reference for the indexed  $\langle stuff \rangle$ . It writes  $\langle stuff \rangle$  to the file.idx file, and also writes the page number (in parentheses) and the value of the  $\langle counter \rangle$ . This means that indexing can be with respect to something other than page numbers. However, if the hyperref package is used the special index links will be to pages even though they will appear to be with respect to the  $\langle counter \rangle$ ; for example, if figure numbers are used as the index reference the hyperref link will be to the page where the figure caption appears and not to the figure itself.

```
\showindexmarks \hideindexmarks \indexmarkstyle
```

The declaration \showindexmarks causes the argument to practically any \index and \specialindex to be printed in the margin of the page where the indexing command was issued. The argument is printed using the \indexmarkstyle which is initially specified as

\indexmarkstyle{\normalfont\footnotesize\ttfamily}

For reasons I don't fully understand, spaces in the argument are displayed as though it was typeset using the starred version of \verb. The \hideindexmarks, which is the default, turns off \showindexmarks.

The standard classes just provide the plain \index command with no optional  $\langle file \rangle$  argument. In those classes the contents of the jobname.idx file is limited to the index entries actually seen in the document. In particular, if you are using \include for some parts of the document and \includeonly to exclude one or more files, then any \index entries in an excluded file will not appear in the jobname.idx file. The new implementation of indexing eliminates that potential problem.

```
\item\subitem\subsubitem
```

The theindex environment supports three levels of entries. A \item command flags a main entry; a subentry of a main entry is indicated by \subitem and a subentry of a subentry is flagged by \subsubitem. For example a portion of an index might look like:

```
\item bridge, 2,3,7 bridge, 2,3,7 \subitem railway, 24 railway, 24 \subsubitem Tay, 37 Tay, 37
```

if the Tay Bridge<sup>1</sup> was mentioned on page 37.

Beautiful Railway Bridge of the Silv'ry Tay!

Alas! I am very sorry to say

That ninety lives have been taken away

On the last Sabbath day of 1879,

Which will be remember'd for a very long time.

A railway (railroad) bridge in Scotland that collapsed in 1879 killing 90 people. The disaster lives for ever in the poem The Tay Bridge Disaster by William McGonagall (1830--?), the first verse of which goes:

```
\index{Alf}
                                  \indexentry{Alf}{v}
p. 1:
      \index{Alf}
                                  \indexentry{Alf}{1}
p. 2:
      \index{Alf}
                                  \indexentry{Alf}{2}
p. 3:
      \index{Alf}
                                  \indexentry{Alf}{3}
      \index{Alfabet|see{Bet}}
p. 5:
                                  \indexentry{Alfabet|see{Bet}}{5}
p. 7:
      \index{Alf@\textit{Alf}}
                                  \indexentry{Alf@\textit{Alf}}{7}
       \index{Bet|textbf}
                                  \indexentry{Bet|textbf}{7}
p. 8:
      \index{Alf!Bet!Con}
                                  \indexentry{Alf!Bet!Con}{8}
p. 9:
      \index{Alf!Dan}
                                  \indexentry{Alf!Dan}{9}
```

Figure 17.1: Raw indexing: (left) index commands in the source text; (right) idx file entries

#### 17.2.3 MakeIndex

It is possible, but time consuming and error prone, to create your index by hand from the output of the \index commands you have scattered throughout the text. Most use the MakeIndex program to do this for them; there is also the xindy program [Keh98] but this is much less known.

```
\xindyindex
```

It turns out that xindy cannot handle a memoir hyperindex (which can be obtained with the aid of the hyperref package), although Makelndex can do so.<sup>2</sup> If you are going to use xindy to process the raw index data put \xindyindex in the preamble, which will prevent hyperindexing.

MakeIndex reads an idx file containg the raw index data (which may include some commands to MakeIndex itself), sorts the data, and then outputs an ind file containing the sorted data, perhaps with some LaTeX commands to control the printing. MakeIndex was created as a general purpose index processing program and its operation can be controlled by a 'makeindex configuration file' (by default this is an ist file). Such a file consists of two parts. The first part specifies MakeIndex commands that can be included in the  $\langle stuff \rangle$  argument to \index. The second part controls how the sorted index data is to be output.

I will only describe the most common elements of what you can put in an ist file; consult the MakeIndex manual [CH88], or the *Companion* [MG $^+04$ ], for all the details.

You can embed commands, in the form of single characters, in the argument to \index that guide Makelndex in converting the raw idx file into an ind file for final typesetting. The complete set of these is given in Table 17.1. They all have defaults and you can modify these via a Makelndex configuration file.

In the simplest case you just use the name of the index item as the argument to the  $\index$  command. However, spaces are significant as far as MakeIndex is concerned. The following three uses of  $\index$  will result in three different entries in the final index  $\index\{\{\{u\}\}\}$  index $\{\{u\}\}$ 

The ! character

The level specifier starts a new minor level, or subitem, with a maximum of two sub-levels. The default level specifier is the special character!. For example:

\index{item!sub item!sub sub item}

<sup>2</sup> This deficiency in xindy was discovered by Frederic Connes, who also provided the \xindyindex command.

Table 17.1: MakeIndex configuration file input parameters

Keyword	Default	Description
keyword (s)	"\\indexentry"	The argument to this command is a Makeln-
		dex index entry
${\tt arg\_open}\ (c)$	<b>'{</b> '	Argument start delimeter
${ t arg\_close}\ ({ t c})$	'}'	Argument end delimeter
${\tt range\_open}\ (c)$	'('	Start of an explicit page range
$ exttt{range\_close}\left( ext{c} ight)$	')'	End of an explicit page range
level(c)	, į ,	Character denoting a new subitem level
$\mathtt{actual}\ (\mathrm{c})$	·@·	Character denoting that the following text is
		to appear in the actual index file
$\mathtt{encap}\ (\mathrm{c})$	,   ,	Character denoting that the rest of the argu-
		ment is to be used as an encapsulating com-
		mand for the page number
$ extsf{quote} \ ( ext{c})$	) II )	Character that escapes the following charac-
		ter
$\operatorname{escape}\ (\operatorname{c})$	,//,	Symbol with no special meaning unless fol-
		lowed by the quote character, when both
		characters will be printed. The quote and
		escape characters must be different.
${\tt page\_compositor}\;(s)$	"_"	Composite number separator

(s) of type string, (c) of type character

\begin{theindex}	
\item Alf, v, 1-3	Alf, v, 1-3
\subitem Bet	Bet
\subsubitem Con, 8	Con, 8
\subitem Dan, 9	Dan, 9
\item \textit{Alf}, 7	Alf, 7
\item Alfabet, \see{Bet}{5}	Alfabet, see Bet
\indexspace	_
\item Bet, \textbf{7}	$\mathrm{Bet},7$
\end{theindex}	

Figure 17.2: Processed index: (left) alphabeticized ind file; (right) typeset index

The @ character

An indexable item may be represented in two portions, separated by the actual specifier, which by default is the @ character. The portion before the @ is used when MakeIndex sorts the raw index data, and the portion after the @ is used as the entry text. For example:

```
\index{MakeIndex@\textit{MakeIndex}}
```

will result in the final index entry of \textit{MakeIndex} in the alphabetic position accorded to MakeIndex. The same treatment can be applied for subitems:

\index{program!MakeIndex@\textit{MakeIndex}!commands}

The / character

Anything after the encap specifier, which by default is the | character, is treated as applying to the page number. In general

```
\index{...|str}
```

will produce a page number, say n, in the form

```
\str{n
```

For example, if you wanted the page number of one particular entry to be in a bold font, say to indicate that this is where the entry is defined, you would do

```
\index{entry|textbf}
```

As a special case, if you want an index item to have a page range put the two characters | ( at the end of the argument on the first page, and the character pair |) at the end of the argument on the last page. For example:

```
... \index{range|(} pages about range \index{range|)} ...
```

The two arguments must match exactly except for the final ( and ). You can also do

```
\index{...|(str}
```

which will produce a page range of the form

```
\t n-m
```

In this case, if the range is only a single page, the result is simply

\str{n}

```
\see\{\langle text \rangle\} \{\langle page \rangle\}\ seename \seealso\{\langle text \rangle\} \{\langle page \rangle\}\ alsoname
```

The macros \see and \seealso are specifically for use in an \index command after the |. The \see command replaces the page number by the phrase 'see  $\langle text \rangle$ ', while the \seealso command adds 'see also  $\langle text \rangle$ ' to the entry. For example, in the source for this manual I have

```
\index{chapter!style|see{chapterstyle}}
\index{figure|seealso{float}}
```

A \see or \seealso should be used once only for a particular entry. The 'see' texts for \see and \seealso are stored in \seename and \alsoname, whose default definitions are:

```
\newcommand*{\seename}{see}
\newcommand*{alsoname}{see also}
```

```
The " and \ characters
```

If, for some reason, you want to index something that includes one of the !, @, | or " characters there is the difficulty of persuading MakeIndex to ignore the special meaning. This is solved by the quote specifier, which is normally the " character. The character immediately after " is treated as non-special. For example, if you needed to index the @ and ! characters:

```
\index{"@ (commercial at)}
\index{"! (exclamation)}
```

The leading " is stripped off before entries are alphabetized.

The escape specifier is used to strip the special meaning from the quote specifier. This is usually the \ character. So, to index the double quote character itself:

```
\index{\" (double quote)}
```

Example of using the special characters

Here is a short example of indexing the special characters. Given an input like this in the document

```
\index{exclamation mark ("!)}
\index{vicious|see{circle}}
\index{atsign@\texttt{"@} sign|\textbf}
\index{quote!double ("")}
\index{circle|see{vicious}}
```

then an index could eventually be produced that looks like:

```
@ sign, 30
circle, see vicious
exclamation mark (!), 21
quote
double ("), 47
vicious, see circle
```

# 17.2.4 Controlling MakeIndex output

Table 17.2 lists the parameters that control MakeIndex's output, except for the keywords that control the setting of page numbers. The special characters and strings are not fixed within the MakeIndex program. The program will read an ist file in which you can redefine all of MakeIndex's defaults.

I have used a file called memman.ist for configuring MakeIndex for this manual. Here it is:

```
% MakeIndex style file memman.ist
% @ is a valid character in some entries, use ? instead
actual '?'
```

Table 17.2: MakeIndex configuration file output parameters

Keyword	Default	Description
preamble (s) postamble (s)	"\\begin{theindex}\n" "\n\n\\end{theindex}\n"	Text for the start of the output file Text at the end of the output file
group_skip (s) heading_prefix (s) heading_suffix (s) headings_flag (n)	"\n\n\\indexspace\n" "" 0	Vertical space before a new letter group Prefix for heading for a new letter group Suffix for heading for a new letter group A value = 0 inserts nothing between letter groups. A value > 0 includes an uppercase instance of the new symbol, while a value < 0 includes a lowercase instance, all within heading_prefix and heading_suffix
<pre>item_0 (s) item_1 (s) item_2 (s) item_01 (s) item_12 (s) item_x1 (s)</pre>	"\n\item " "\n \subitem " "\n \subsubitem " "\n \subsubitem " "\n \subsubitem "	Command inserted in front of a level 0 entry As above for a level 1 entry As above for a level 2 entry Command inserted in front of a level 1 entry starting at level 0 Command inserted in front of a level 2 entry starting at level 1 Command inserted in front of a level 1 entry when the parent level has no page numbers
item_x2 (s)	"\n \subitem "	As above for a level 2 entry
<pre>delim_0 (s)  delim_1 (s) delim_2 (s) delim_n (s) delim_r (s)</pre>	", " ", " ", " ", " ", "	Delimiter between level 0 entry and first page number As above for level 1 entry As above for level 2 entry Delimiter between page numbers Designator for a page range
encap_prefix (s) encap_infix (s) encap_suffix (s)	"\\" "{" "}"	Prefix in front of a page encapsulator Infix for a page encapsulator Suffix for a page encapsulator
page_precedence (s)	"rnaRA"	Page number precedence for sorting. r and R are lower- and uppercase roman; a and A are lower- and uppercase alphabetic; n is numeric
line_max (n) indent_space (s) indent_length (n)	"72" "\t\t" "16"	Maximum length of an output line Indentation commands for wrapped lines Indentation length for wrapped lines

<sup>(</sup>s) of type string, (n) of type number, "\n" and "\t" are newline and tab.

```
% output main entry <entry> as: \item \idxmark{<entry>},
item_0 "\n\\item \\idxmark{"
delim_0 "}, "
% not forgetting the subitem case
item_x1 "} \n \\subitem "
% Wrap and uppercase head letters
headings_flag 1
heading_prefix "\\doidxbookmark{"
heading suffix "}"
```

Many items that I need to index include @ as part of their names, which is one of the special characters. The actual line says that the character? performs the same function as the default @ (and by implication, @ is not a special character as far as MakeIndex is concerned).

The item\_0 line says that a main entry in the generated index starts

```
\item \idxmark{
and the delim_0 and item_x1 lines say that the main entry ends
}, % or
}
\subitem
```

Thus, main entries will appear in an ind file like

```
\item \idxmark{a main entry}, <list of page numbers>
\item \idxmark{a main entry with no page numbers}
\subitem subitem, <list of page numbers>
```

Read the MakeIndex manual [CH88] for full details on how to get MakeIndex to do what you want.

The \doidxbookmark that is used to wrap around the letter group headers, can be used to not only style the group header, but can also be used to add the headers in the bookmarks list. This can be done using

```
\newcommand{\doidxbookmark}[1]{{\def\@tempa{Symbols}\def\@tempb{#1}%
  \centering\bfseries \ifx\@tempa\@tempb %
  Analphabetics
  \phantomsection%
  \pdfbookmark[0]{Analphabetics}{Analphabetics-idx}%
% \label{AnalphabeticsAnalphabeticsAnalphabetics-idx}%
\else
  #1%
  \phantomsection%
  \pdfbookmark[0]{#1}{#1-idx}%
% \label{#1#1#1-idx}%
\fi%
  \vskip\onelineskip\par}}
```

The labels are generally not needed but can be used to add a visual representation of the index bookmarks into the index itself.

# 17.2.5 Indexing and the **natbib** package

The natbib package [Dal99a] will make an index of citations if \citeindextrue is put in the preamble after the natbib package is called for.

```
\citeindexfile
```

The name of the file for the citation index is stored in the macro \citeindexfile. This is initially defined as:

```
\newcommand{\citeindexfile}{\jobname}
```

That is, the citation entries will be written to the default idx file. This may be not what you want so you can change this, for example to:

```
\renewcommand{\citeindexfile}{names}
```

```
If you do change \citeindexfile then you have to put
  \makeindex[\citeindexfile]

before
  \usepackage[...]{natbib}
  So, there are effectively two choices, either along the lines of
  \renewcommand{\citeindexfile}{authors} % write to authors.idx
  \makeindex[\citeindexfile]
  \usepackage{natbib}
  \citeindextrue
  ...
  \renewcommand{\indexname}{Index of citations}
  \printindex[\citeindexfile]

or along the lines of
  \usepackage{natbib}
  \citeindextrue
  \makeindex
  ...
```

#### 17.3 Glossaries

\printindex

Unlike indexes, LaTeX provides less than minimal support for glossaries. It provides a \makeglossary command for initiating a glossary and a \glossary command which puts its argument, plus the page number, into a glo file, and that's it. memoir, combined with the MakeIndex program [CH88], enables you to generate and print a glossary in your document. The commands for creating a glossary are similar to those for indexes.

```
\label{eq:makeglossary} [\langle file 
angle]
```

You have to put \makeglossary in your preamble if you want a glossary. This opens a file called by default \jobname.glo. If you use the optional  $\langle file \rangle$  argument the file file.glo will be opened. A glossary glo file is analogous to an index idx file.

```
\printglossary[\langle file \rangle]
```

To print a glossary call \printglossary which will print the glossary from file \jobname.gls, or from file.gls if the optional argument is used. A glossary gls file is analagous to an index ind file.

```
\label{eq:glossary} $$ \geqslant (\langle key \rangle) {\langle term \rangle} {\langle desc \rangle}$
```

Use the \glossary command to add a  $\langle term \rangle$  and its description,  $\langle desc \rangle$ , to a glossary file. By default this will be \jobname.glo but if the optional  $\langle file \rangle$  argument is given then the information will be written to file.glo. The  $(\langle key \rangle)$  argument is optional. If present then  $\langle key \rangle$  will be added to the file to act as a sort key for the  $\langle term \rangle$ , otherwise  $\langle term \rangle$  will be used as the sort key.

By using the optional  $\langle file \rangle$  arguments you can have several glossaries, subject to TeX's limitations on the number of files that can be open at any one time.

A simple glossary entry might be:

\glossary{glossary}{A list of terms and their descriptions.}

The glossary facilities are designed so that the MakeIndex program can be used to convert the raw glossary data in a glo file into the printable glossary in a gls file.

```
\begin{theglossary} entry list \end{theglossary}
```

Glossary entries are typeset in a theglossary environment. It is assumed that a gls file will contain a complete theglossary environment, from \begin{theglossary} all the way through to \end{theglossary}.

```
\label{lossitem} $$\left( \operatorname{desc} \right) = \left( \operatorname{desc} \right) = \left( \operatorname{num} \right)
```

A \glossitem is a glossary entry within a theglossary environment for a  $\langle term \rangle$  with  $\langle description \rangle$ . The  $\langle num \rangle$  argument is the page or section where the corresponding \glossary was issued. The  $\langle ref \rangle$  argument, if not empty, might be the section or page number corresponding to the  $\langle num \rangle$  page or section number. The default definition is

\newcommand{\glossitem}[4]{#1 #2 #3 #4}

which is not very exciting. You may well prefer to use your own definition.

17.3.1 Controlling the glossary

Setting up makeindex

If you just run Makelndex on a glo file you will get lots of errors; Makelndex has to be configured to read a glo file and generate a useful gls file as by default it expects to read an index idx file and produce an index ind file. A configuration file like an index ist file will be needed. There is no recommended extension for such a file but I have come to favour gst. The command line for Makelndex to create a sorted glossary from the raw data in a glo file, say fred.glo, using a configuration file called, say basic.gst, is

```
makeindex -s basic.gst -o fred.gls fred.glo
```

For other jobs just change the file names appropriately.

So, what is in a gst file? The potential contents were described earlier, but at a minimum you need this:

```
%%% basic.gst basic makindex glossary style file
%%% Output style parameters
preamble "\begin{theglossary}"
postamble "\n\end{theglossary}\n"
item_0 "\n\glossitem"
delim_0 "{\memglonum{"
encap_suffix "}}}"
heading_flag 1
heading_prefix "\doglobookmark{"
heading_suffix "}"
%%% Input style parameters
keyword "\glossaryentry"
```

The keyword line says that each entry in an input (glo) file will be of the form:

```
\glossaryentry{entry text}{number}
```

and by a miracle of coding, this is what memoir will put in a glo file for each \glossary command.

The preamble and postamble lines tell the program to start and end its output file with \begin{theglossary} and \end{theglossary}, respectively. The item\_0 tells the program to start each output entry with \glossitem. The delim\_0 says that {\memglonum{ should be put between the end of the entry text and the (page) number. Finally encap\_suffix requests }}} to be put after any 'encapsulated' (page) number.

A complete listing of the possible entries in a configuration file, also called a style file, for MakeIndex is in Table 17.1 and 17.2 with the exception of the output file page number setting keywords.

The \doglobookmark macro can be used to add bookmarks for the letter groups. In the case of this manual we do not write anything, just add the letters to the glossary entry in the bookmark list. In memsty \doglobookmark is defined as

```
\newcommand\doglobookmark[1]{%
  \def\@tempa{Symbols}\def\@tempb{#1}%
  \ifx\@tempa\@tempb %
  \phantomsection\pdfbookmark[0]{Analphabetics}{Analphabetics-glo}%
  \else%
  \phantomsection\pdfbookmark[0]{#1}{#1-glo}%
  \fi%
}
```

MakeIndex uses the word 'Symbols' to specify the group that does not start with a letter.

Raw input data

```
\label{lower_lower} $$ \end{area} $$ \end{
```

The \glossary macro writes its arguments to the aux file in the form of arguments to the \@@wrglom@m internal macro. In turn this calls a series of other macros that eventually write the data to the  $\langle file \rangle$  glo file in the format (where @ is the actual flag):

which MakeIndex then effectively converts into

These macros can be redefined to format the various parts of a glossary entry. Their default definitions are simply

```
\newcommand{\memgloterm}[1]{#1}
\newcommand{\memglodesc}[1]{#1}
\newcommand{\memgloref}[1]{#1}
\newcommand{\memglonum}[1]{#1}
```

For example, if you wanted the term in bold, the description in italics, and no numbers:

```
\renewcommand{\memgloterm}[1]{\textbf{#1}}
\renewcommand{\memglodesc}[1]{\textit{#1}}
\renewcommand{\memglonum}[1]{}
```

There are several macros that effect a glossary entry but which must not be directly modified (the \memglonumf shown above as part of the \glossaryentry is one of these). Each of the following \changegloss... macros takes an optional  $\langle file \rangle$  argument. The changes to the underlying macro apply only to the glossary of that particular  $\langle file \rangle$  (or the \jobname file if the argument is not present.

\changeglossactual sets  $\langle char \rangle$  as the actual character for the  $\langle file \rangle$  glossary. It is initially @. This must match with the actual specified for the gst file that will be applied.

\changeglossref specifies that  $\langle the counter \rangle$  should be used to generate the  $\langle ref \rangle$  for the  $\langle file \rangle$  glossary. It is initially nothing.

\changeglossnum specifies that  $\langle the counter \rangle$  should be used to generate the  $\langle num \rangle$  for the  $\langle file \rangle$  glossary. It is initially \thepage.

\changeglossnumformat specifies that  $\langle format \rangle$  should be used to format the  $\langle num \rangle$  for the  $\langle file \rangle$  glossary. The format of  $\langle format \rangle$  is |form, where | is the encap character specified in the gst file, and form is a formatting command, taking one argument (the number), without any backslash. For example

```
\changeglossnumformat{|textbf}
```

to get bold numbers. It is initially set as |memjustarg, where this is defined as:

```
\newcommand{\memjustarg}[1]{#1}
```

There must be a format defined for the  $\langle num \rangle$  otherwise the arguments to \glossitem will not be set correctly.

The  $\mbox{\mbox{makeglossary command uses the $\mbox{\mbox{change...}} commands to define the initial versions, so only use the <math>\mbox{\mbox{\mbox{change...}} macros after $\mbox{\mbox{\mbox{makeglossary}}.}$  In this document an early version of the glossary was set up by

```
\makeglossary
\changeglossactual{?}
\makeatletter
\changeglossnum{\@currentlabel}
\makeatother
\changeglossnum{\thepage}
```

The first call of \changeglossnum makes the number the current numbered chapter, or numbered section, or numbered …. I didn't like that when I tried it, so the second call resets the number to the page number.

The listing

The final glossary data in the gls file is typeset in the theglossary environment, which is much like the theindex and thebibliography environments.

The environment starts off with a chapter-style unnumbered title. There are several macros for specifying what happens after that.

```
\glossaryname
\glossarymark
\glossaryintoc \noglossaryintoc
```

The title for the glossary is \glossaryname whose initial definition is

\newcommand\*{\glossaryname}{Glossary}

\glossarymark, which by default does nothing, can be redefined to set marks for headers. The glossary title will be added to the ToC if the \glossaryintoc declaration is in force, but will not be added to the ToC following the \noglossaryintoc.

```
\preglossaryhook
```

The macro \preglossaryhook is called after the glossary title has been typeset. By default it does nothing, but you could redefine it to, for example, add some explanatory material before the entries start.

```
\onecolglossary\twocolglossary
\glossarycolsep\glossaryrule
```

The glossary can be typeset in two columns (\twocolglossary) but by default (\onecolglossary) it is set in one column. When two columns are used, the length \glossarycolsep is the distance between the columns and the length \glossaryrule is the width (default 0) of a vertical rule between the columns.

```
\begintheglossaryhook
\atendtheglossaryhook
```

The last thing that \begin{theglossary} does is call \begintheglossaryhook. Similarly, the first thing that is done at the end of the environment is to call \atendtheglossaryhook. By default these macros do nothing but you can redefine them.

For example, if you wanted the glossary in the form of a description list, the following will do that.

```
\renewcommand*{\begintheglossaryhook}{\begin{description}}
\renewcommand*{\atendtheglossaryhook}{\end{description}}
\renewcommand{\glossitem}[4]{\item[#1:] #2 #3 #4}
```

The glossary for this document

The following is the code I have used to produce the glossary in this document. This is the code in the sty file that I used.

```
\makeglossary
\changeglossactual{?}
\changeglossnum{\thepage}
\changeglossnumformat{|hyperpage}%% for hyperlinks
\renewcommand*{\glossaryname}{Command summary}
\renewcommand*{\glossarymark}{\markboth{\glossaryname}{\glossaryname}}
\renewcommand{\glossitem}{4]{%
\sbox\@tempboxa{#1 \space #2 #3 \makebox[2em]{#4}}%
\par\hangindent 2em
\ifdim\wd\@tempboxa<0.8\linewidth
#1 \space #2 #3 \dotfill \makebox[2em][r]{#4}\relax
\else
#1 \dotfill \makebox[2em][r]{#4}\\
#2 #3
\fi}</pre>
```

The redefinition of  $\glossitem$  works as follows (it is similar to code used in the setting of a  $\colonsymbol{\colonsym$ 

- 1. Put the whole entry into a temporary box.
- 2. Set up a hanging paragraph with 2em indentation after the first line.
- 3. Check if the length of the entry is less than 80% of the linewidth.
- 4. For a short entry set the name, description, and any reference then fill the remainder of the line with dots with the number at the right margin.
- 5. For a longer entry, set the title and number on a line, separated by a line of dots, then set the description (and reference) on the following lines.

The gst file I have used for this document has a few more items than the basic one.

```
%%% memman.gst makindex glossary style file for memman and friends
%%% Output style parameters
preamble "\\begin{theglossary}\\n"
postamble "\n\\end{theglossary}\\n"
group_skip "\n\\glossaryspace\\n"
item_0 "\n\\glossitem"
delim_0 "{\\memglonum{"}
encap_suffix "}}\"
indent_space "\t"
indent_length 2
%%% Input style parameters
keyword "\\glossaryentry"
```

```
actual '?'
page_compositor "."
```

The group\_skip line asks that \glossaryspace be put between the last entry for one letter and the first for the next letter. The indent\_space and indent\_length give a smaller indent for continuation lines in the output than the default.

The actual entry says that the input file will use? instead of the default @ as the flag for separating a key from the start of the real entry. The page\_compositor indicates that any compound numbers will be like 1.2.3 instead of the default 1-2-3.

In the document the raw data is collected by the \glossary commands in the body of the text. For instance, although I have not actually used the first two:

```
\glossary(cs)%
  {\cs{cs}\marg{name}}%
  {Typesets \texttt{name} as a macro name with preceding backslash,
    e.g., \cs{name}.}%
\glossary(gmarg)%
  {\cs{gmarg}\marg{arg}}%
  {Typesets \texttt{arg} as a required argument, e.g., \marg{arg}.}
\glossary(glossaryname)%
  {\cs{glossaryname}}%
  {Name for a glossary}%
\glossary(memgloterm)%
  {\cs{memgloterm}\marg{term}}%
  {Wrapper round a glossary term.}%
```

Any change to the glossary entries will be reflected in the glo produced from that La-TeX run. Makelndex has to be run the glo file using the appropriate gst configuration file, and then LaTeX run again to get the corrected, sorted and formatted result printed by \printglossary.

In particular, for this document, which also includes an index so that can be processed when the glossary is processed.

# Eighteen

# Miscellaneous

In which we talk of many things, but not shoes or ships or sealing wax, nor cabbages and kings.

This chapter started with the \chapterprecis command to add the above text, which is also added to the ToC.

The class provides a miscellany of minor facilities, which are described here.

#### 18.1 Draft documents

The draft option marks any overfull lines with black rectangles, otherwise the appearance is the same as for a final document.

```
\ifdraftdoc
```

The \ifdraftdoc macro is provided so that you can add extra things that you might want to happen when processing a draft; for example you might want to have each page header (or footer) include the word 'DRAFT'. The code to do this can be put into a construct like the following:

```
\ifdraftdoc
   % special things for a draft document
\else
   % perhaps special things for a non-draft document
\fi
```

# 18.2 Change marks

When preparing a manuscript it normally goes through several iterations. The macros described in this section can be used to identify changes made to a document throughout its lifecycle.

The commands below implement a simplified version of the change marking in the iso class [Wil00b].

```
\changemarks \nochangemarks
```

The change marking macros only work properly when the draft class option is used. Additionally, the marks are only printed when the \changemarks declaration is in effect. Using \nochangemarks switches off any marking.

```
\label{change-id} $$ \deleted{\change-id}$ $$ \changed{\change-id}$
```

Each of these macros puts a symbol and  $\langle change-id \rangle$  into the margin near where the command is given. The \added macro indicates that something has been added to the manuscript and uses  $\oplus$  as its symbol. The \deleted macro is for indicating that something has been deleted and uses the  $\neq$  symbol. The macro \changed uses the  $\Leftrightarrow$  symbol to indicate that something has been changed.

These marking commands should be attached to some word or punctuation mark in the text otherwise extraneous spaces may creep into the typeset document.

#### 18.3 Trim marks

When the class showtrims option is used, trim marks can be placed on each page, usually to indicate where the stock should be trimmed to obtain the planned page size.

```
\showtrimsoff \showtrimson
```

If the showtrims class option has been used then the \showtrimsoff declaration switches off the trim marks; the \showtrimson declaration, which is the default, switches on the trim marks. These declarations do nothing if the showtrims option has not been used.

Caveat. Most modern LTEX editors make use of the synctex features in, say, pdfLTEX to enable reverse search from the PDF previewer back to the source code in the editor. That is, click in a certain manner in the PDF previewer and you will be taken to the corresponding (or there about) line in the source code.

Apparently the synctex feature does not like the trimmarks the class provide, or the showlocs page style. The code line one is referred back to may be off by tens of lines.

Currently, there is no known workarounds for this deficiency.

Trim marks can be placed at each corner of the (trimmed) page, and also at the middle of each side of the page.

```
\trimXmarks
\trimLmarks
\trimFrame
\trimNone
\trimmarkscolor
```

Some predefined trimming styles are provided. After the declaration \trimXmarks marks in the shape of a cross are placed at the four corners of the page. The declaration \trimLmarks calls for corner marks in the shape of an 'L', in various orientations depending on the particular corner. After \trimFrame a frame will be drawn around each page, coinciding with the page boundaries. The declaration \trimNone disables all kinds of trim marking. All three plus \quarkmarks (described below) is visibly shown on Figure 18.1.

The macro \trimmarkscolor is despite its name a normal (empty) macro. By redefining it one can change the color of the trimmarks, handy for example if the document has a dark background color. To make them blue use

\newcommand\*{\trimmarkscolor}{\color{blue}}

```
\trimmarks
\tmarktl \tmarktr \tmarkbr \tmarkbl
\tmarktm \tmarkbm \tmarkml
```

The \trimmarks macro is responsible for displaying up to 8 marks. The marks are defined as zero-sized pictures which are placed strategically around the borders of the page.

The command \trimmarks places the pictures \tmarktl, \tmarktr, \tmarkbl, and \tmarkbr at the top left, top right, bottom right and bottom left corners of the page. The pictures \tmarktm, \tmarkmr, \tmarkbm, and \tmarkml are placed at the top middle, middle right, bottom middle and middle left of the edges of the page. All these \tmark.. macros should expand to zero-sized pictures.

```
\trimmark
```

The declaration \trimXmarks uses \trimmark for the corner crosses. This is defined as

```
\newcommand{\trimmark}{%
  \begin{picture}(0,0)
  \setlength{\unitlength}{1cm}\thicklines
  \put(-2,0){\line(1,0){4}}
  \put(0,-2){\line(0,1){4}}
\end{picture}}
```

which produces a zero-sized picture of a 4cm cross. Then \trimXmarks is defined as:

```
\newcommand*{\trimXmarks}{%
  \let\tmarktl\trimmark
  \let\tmarktr\trimmark
  \let\tmarkbr\trimmark
  \let\tmarkbl\trimmark}
```

As an example, to draw short lines marking the half-height of the page, try this:

```
\renewcommand*{\tmarkml}{%
  \begin{picture}(0,0)%
    \unitlength 1mm
  \thinlines
  \put(-2,0){\line(-1,0){10}}
  \end{picture}}
\renewcommand*{\tmarkmr}{%
  \begin{picture}(0,0)%
    \unitlength 1mm
  \thinlines
  \put(2,0){\line(1,0){10}}
  \end{picture}}
```

Thin horizontal lines of length 10mm will be drawn at the middle left and middle right of the page, starting 2mm outside the page boundary. This is what we do (now) by default for all four middle parts.

```
\label{eq:linear_loss} $$\operatorname{\model{mark}}$
```

Following the declaration \quarkmarks and trim marks will be in the style of Quark Xpress registration marks.\(^1\) The marks will be typeset using \registrationColour. The default definition is simply

<sup>1</sup> The code for this was donated by William Adams.

```
\newcommand*{\RegistrationColour}[1]{#1}
```

but you can change that to, say, print the marks in particular color. See Figure 18.1.

examples only inserted in pdf-mode, now we can write and work in DVI-mode,  $\mbox{\sc /daleif}$ 

Figure 18.1: The four trimmark types

# 18.4 Sheet numbering

One purpose of trim marks is to show a printer where the stock should be trimmed. In this application it can be useful to also note the sheet number on each page, where the sheet number is 1 for the first page and increases by 1 for each page thereafter. The sheet number is independent of the page number.

```
sheetsequence \thesheetsequence
```

The macro \thesheetsequence typesets the current sheet sequence number and is analogous to the \thepage macro.

```
lastsheet
lastpage
```

The counter lastsheet holds the number of sheets processed during the *previous* run of LaTeX. Similarly, the counter lastpage holds the number of the last page processed during the previous run. Note that the last page number is not necessarily the same as the last sheet number. For example:

*In this document this is sheet 344 of 549 sheets, and page 320 of 525.* 

The previous sentence was the result of processing the following code

```
\textit{In this document this is
    sheet \thesheetsequence\ of \thelastsheet\ sheets,
    and page \thepage\ of \thelastpage.}
```

You may wish to use the sheet and/or page numbers as part of some trim marks. The following will note the sheet numbers above the page.

```
\newcommand*{\trimseqpage}{%
  \begin{picture}(0,0)
  \unitlength 1mm
  \put(0,2){\makebox(0,0)[b]{Sheet: \thesheetsequence\ of \thelastsheet}}
  \end{picture}}
\let\tmarktm\trimseqpage
```

# 18.5 Gatherings or signatures

Sometimes publishers request that they be supplied with a total number of pages that meet their planned *gatherings*.<sup>2</sup> For instance a gathering may consist of 8 leaves, and as there are

<sup>2</sup> There was a thread on ctt, pagenumber mod 4? about this in 2008.

two pages to a leaf this is equivalent to 16 pages. To meet this particular requirement there must be a total of 8N leaves, or equivalently 16N pages, where N will be the number of gatherings.

```
\lceil \langle num \rangle \rceil
```

The command \leavespergathering ensures that there will be exactly the right number of pages output to make a complete set of gatherings of  $\langle num \rangle$  leaves  $(2\langle num \rangle$  pages) each — if necessary blank pages will be output at the end to make up the correct tally. If  $\langle num \rangle$  is less than two (the default) then no additional pages will be output.

#### 18.6 Time

```
\printtime \printtime*
\hmpunct \amname \pmname
```

The \printtime command<sup>3</sup> prints the time of day when the document is processed using the 24 hour clock while \printtime\* uses a 12 hour clock. For example, the effect of the next piece of code is shown below.

This document was processed on: \today\ at \printtime\ (\printtime\*).

This document was processed on: August 11, 2019 at 6:47 (6:47 am).

The punctuation between the hours and minutes is \hmpunct which defaults to a colon (:). The macros \amname and \pmname hold the abbreviations for *ante meridiem* and *post meridiem*, respecitively; the defaults are 'am' and 'pm'.

According to the *Chicago Manual of Style* [Chi93] there should be no punctuation between the hours and minutes in the 24 hour system. For the 12 hour system it recommends that small caps be used for the divisions of the day (e.g., A.M. and P.M.) and also that the American practice is to use a colon as the separator between hours and minutes whereas the English practice is to use a period (known to the English as a 'full stop'). I don't know what the traditions are in other orthographies.

The \quarkmarks declaration uses \printtime, so be careful if you change it.

Nicola Talbot's datetime package [Tal06] provides a much more comprehensive collection of styles for printing the time; also for dates.

# 18.7 Page breaks before lists

A sentence or two may be used to introduce a list (e.g., itemize) and it can be annoying if there is a page break between the introductory words and the first item.

```
\noprelistbreak
```

Putting \noprelistbreak immediately before the \begin{itemize} should prevent a page break. Ideally, there should be no blank lines between the end of the introduction and the start of the list.

<sup>3</sup> I based the code on a similar macro in TeX for the Impatient [AHK90].

# 18.8 Changing counters

This is effectively a bundling of the chngcntr package [Wil01e].

In LaTeX a new counter called, say ctr, is created by the \newcounter command as \newcounter{ctr}. If the optional  $\langle within \rangle$  argument is given, the counter ctr is reset to zero each time the counter called within is changed; the within counter must exist before it is used as the optional argument. The command \thectr typesets the value of the counter ctr. This is automatically defined for you by the \newcounter command to typeset arabic numerals

```
\label{counterwithin} $$ \operatorname{counterwithin}_{\langle ctr\rangle}_{\langle within\rangle} $$ counterwithin*_{\langle ctr\rangle}_{\langle within\rangle}$
```

The \counterwithin macro makes a  $\langle ctr \rangle$  that has been initially defined by \newcounter{ctr} act as though it had been defined by \newcounter{ctr} [within]. It also redefines the \thectr command to be \thewithin.\arabic{ctr}. The starred version of the command does nothing to the original definition of \thectr.

```
\label{eq:counterwithout} $$ \operatorname{counterwithout}_{\langle ctr\rangle}_{\langle within\rangle} $$ $$ \operatorname{counterwithout}_{\langle ctr\rangle}_{\langle within\rangle}_{} $$
```

The \counterwithout macro makes the ctr counter that has been initially defined by \newcounter{ctr}[within] act as though it had been defined by \newcounter{ctr}. It also redefines the \thectr command to be \arabic{ctr}. The starred version of the command does nothing to the original definition of \thectr.

Any number of \counterwithin{ctr}{...} and \counterwithout{ctr}{...} commands can be issued for a given counter ctr if you wish to toggle between the two styles. The current value of ctr is unaffected by these commands. If you want to change the value use \setcounter, and to change the typesetting style use \renewcommand on \thectr.

Caveat. As of 2018 \counterwithout and \counterwithin will be provided by the LTEX-kernel, thus we only provide it if it does not already exist.

```
\label{letcountercounter} $$ \operatorname{counter}(\langle counterA\rangle) + (\langle counterB\rangle) $$ $$ \operatorname{counter}(\langle counterA\rangle) $$
```

At times it is handy to 'let' one counter act as if it was a different counter. Say you have two constructions, each with their own counter A and B, now you want them to cooperate, counting in unison. This can be done using the \letcountercounter.

\letcountercounter{ $\langle counterA \rangle$ }{ $\langle counterB \rangle$ } \lets (make the same)  $\langle counterA \rangle$  to  $\langle counterB \rangle$ . The original of  $\langle counterA \rangle$  is kept, such that you can unlet it later.

 $\under(counterA)$  restores  $\langle counterA \rangle$  to its un\let condition.

This feature can be quite handy. Say for instance you want figures and tables to counter within the same counter (say table), then we need each change to the figure counter to actually act on the table counter. \letcountercounter{figure}{table} solves the problem.

# 18.9 New new and provide commands

```
\label{locality} $$ \left( cmd \right) { \left( string \right) } $$ \end{tabular}
```

The class provides means of creating new math log-like functions. For example you might want to do

```
\newloglike{\Ei}{Ei}
```

if you are using the exponential integral function in your work. The starred version of the command creates a function that takes limits (like the \max function).

The LaTeX kernel defines the \providecommand macro that acts like \newcommand if the designated macro has not been defined previously, otherwise it does nothing. The class adds to that limited repertoire.

The macros \provideenvironment, \providelength and \providecounter take the same arguments as their \new... counterparts. If the environment, length or counter has not been defined then it is defined accordingly, otherwise the macros do nothing except produce a warning message for information purposes.

The \provideloglike commands are for math log-like functions, but they do not produce any warning messages.

# 18.10 Changing macros

Commands are provided for extending simple macro definitions.

```
\label{lambdaddtodef} $$ \addtodef{\macro}}{\addtoiargdef{\macro}}{\addtoiargdef}{\append}}{\addtoiargdef}$$
```

The macro \addtodef inserts  $\langle prepend \rangle$  at the start of the current definition of  $\langle macro \rangle$  and puts  $\langle append \rangle$  at the end, where  $\langle macro \rangle$  is the name of a macro (including the backslash) which takes no arguments. The \addtoiargdef macro is similar except that  $\langle macro \rangle$  is the name of a macro that takes one argument.

For example the following are two equivalent definitions of \mymacro:

```
\newcommand{\mymacro}[1]{#1 is a violinist in spite of being tone deaf}
and
```

```
\newcommand{\mymacro}[1]{#1 is a violinist}
\addtoiargdef{\mymacro}{}{ in spite of being tone deaf}
```

The  $\addtoiargdef$  (and  $\addtodef$ ) commands can be applied several times to the same  $\langle macro \rangle$ . Revising the previous example we could have

```
\addtoiargdef{\mymacro}{}{ and a bagpiper}
which is equivalent to
\newcommand{\mymacro}[1]{%
  Although somewhat elderly, #1 is a violinist
  in spite of being tone deaf and a bagpiper}
```

The  $\langle prepend \rangle$  and  $\langle append \rangle$  arguments may include LaTeX code, as shown in this extract from the class code:

```
\addtoiargdef{\date}{}{%
  \begingroup
  \renewcommand{\thanks}[1]{}
  \renewcommand{\thanksmark}[1]{}
  \renewcommand{\thanksgap}[1]{}
  \protected@xdef\thedate{#1}
  \endgroup}
```

Note that in the case of  $\addtoiargdef$  an argument can also refer to the original argument of the  $\langle macro \rangle$ .

```
\label{lem:addtodef*{macro}} $$ \addtoiargdef*{{macro}}{{prepend}}{{append}} $$
```

These starred versions are for use when the original  $\langle macro \rangle$  was defined via \newcommand\*. Using the starred versions is like using \renewcommand\* and the unstarred versions are like having used \renewcommand. It is the version (starred or unstarred) of a sequence of \addto... commands that counts when determining whether the equivalent \renew... is treated as starred or unstarred.

The \addto... macros cannot be used to delete any code from  $\langle macro \rangle$  nor to add anything except at the start and end. Also, in general, they cannot be used to change the definition of a macro that takes an optional argument, or a starred macro.

```
\label{eq:local_patch_command} $$ \operatorname{\command}_{\command}(\command,\command,\command,\command}(\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\command,\comma
```

The \patchcommand is from the late Michael Downes' patchcmd package [Dow00]. It inserts the  $\langle start\text{-}code \rangle$  at the start of the current definition of the macro  $\langle macro \rangle$ , and inserts  $\langle end\text{-}code \rangle$  at the end of its current definition. The  $\langle macro \rangle$  can have zero to nine parameters. If  $\langle macro \rangle$  uses \futurelet (e.g., it is a starred command or takes an optional argument) only  $\langle start\text{-}code \rangle$  is useful —  $\langle end\text{-}code \rangle$  must be empty otherwise things get messed up. If  $\langle macro \rangle$  has any delimited arguments then \patchcommand cannot be used.

#### 18.11 String arguments

In the code for the class I have sometimes used macro arguments that consist of a 'string', like the \* arguments in the page layout macros (e.g., \settypeblocksize), or the flushleft, center and flushright strings for the \makeheadposition macro.

```
\label{eq:lambda} $\operatorname{\sc str 1} = \frac{\langle str 1 \rangle}{\langle str 2 \rangle}$$ if same name
```

The macro \nametest takes two strings as the arguments  $\langle str1 \rangle$  and  $\langle str2 \rangle$ . It sets \ifsamename true if  $\langle str1 \rangle$  is the same as  $\langle str2 \rangle$ , otherwise it sets \ifsamename false. For

the purposes of \nametest, a string is a sequence of characters which may include spaces and may include the \ backslash character; strings are equal if and only if their character sequences are identical.

Typically, I have used it within macros for checking on argument values. For example:

```
\newcommand{\amacro}[1]{%
  \nametest{#1}{green}
  \ifsamename
%    code for green
  \fi
  \nametest{#1}{red}
  \ifsamename
%    code for red
  \fi
  ...
}
```

## 18.12 Odd/even page checking

It is difficult to check robustly if the current page is odd or even but the class does provide a robust method based on writing out a label and then checking the page reference for the label. This requires at least two LaTeX runs to stabilise. This has been extracted from the original chngpage package (which is no longer available). (The class code and chngpage code is similar but not identical. There is a later package, changepage [Wil08a] which contains code that is identical to the class.)

```
\checkoddpage
\ifoddpage
\strictpagecheck \easypagecheck
```

The macro \checkoddpage sets \ifoddpage to true if the current page number is odd, otherwise it sets it to false (the page number is even). The robust checking method involves writing and reading labels, which is what is done after the command \strictpagecheck is issued; it may take more than one run before everything settles down. The simple method is just to check the current page number which, because of TeX's asynchronous page breaking algorithm, may not correspond to the actual page number where the \checkoddpage command was issued. The simple, but faster, page checking method is used after the \easypagecheck command is issued.

```
\cplabel
```

When strict page checking is used the labels consist of a number preceded by the value of \cplabel, whose default definition is ^\_ (e.g., a label may consist of the characters ^\_21). If this might clash with any of your labels, change \cplabel with \renewcommand, but the definition of \cplabel must be constant for any given document.

# 18.13 Moving to another page

Standard LaTeX provides the \newpage, \clearpage and \cleardoublepage commands for discontinuing the current page and starting a new one. The following is a bundling of the nextpage package [Wil00c].

```
\needspace{\langle length \rangle}
```

This macro decides if there is  $\langle length \rangle$  space at the bottom of the current page. If there is, it does nothing, otherwise it starts a new page. This is useful if  $\langle length \rangle$  amount of material is to be kept together on one page. The \needspace macro depends on penalties for deciding what to do which means that the reserved space is an approximation. However, except for the odd occasion, the macro gives adequate results.

```
\label{eq:length} $$ \end{space} $
```

Like \needspace, the \Needspace macro checks if there is  $\langle length \rangle$  space at the bottom of the current page and if there is not it starts a new page. The command should only be used between paragraphs; indeed, the first thing it does is to call \par. The \Needspace command checks for the actual space left on the page and is more exacting than \needspace.

If either \needspace or \Needspace produce a short page it will be ragged bottom even if \flushbottom is in effect. With the starred \Needspace\* version, short pages will be flush bottom if \flushbottom is in effect and will be ragged bottom when \raggedbottom is in effect.

Generally speaking, use \needspace in preference to \Needspace unless it gives a bad break or the pages must be flush bottom.

```
\begin{tabular}{ll} $$ \mbox{movetoevenpage}[\langle text \rangle] \\ \mbox{cleartoevenpage}[\langle text \rangle] \\ \end{tabular}
```

The \movetoevenpage stops the current page and starts typesetting on the next even numbered page. The \clear... version flushes out all floats before going to the next even page. The optional  $\langle text \rangle$  is put on the skipped page (if there is one).

```
\label{local_movetooddpage} $$ \clear tooddpage [\langle text \rangle] $$
```

These macros are similar to the \...evenpage ones except that they jump to the next odd numbered page.

A likely example for the optional  $\langle text \rangle$  argument is

\cleartooddpage[\vspace\*{\fill}THIS PAGE LEFT BLANK\vspace\*{\fill}]

which will put 'THIS PAGE LEFT BLANK' in the centre of any potential skipped (empty) even numbered page.

```
\cleartorecto \cleartoverso
```

These are slightly simpler forms  $^4$  of \cleartooddpage and \cleartoevenpage. For example, if you wanted the ToC to start on a verso page, like in *The TeXbook* [Knu84], then do this:

<sup>4</sup> Perhaps more robust.

```
\cleartoverso
\tableofcontents
```

# 18.14 Number formatting

Several methods are provided for formatting numbers. Two classes of number representations are catered for. A 'numeric number' is typeset using arabic digits and a 'named number' is typeset using words.

The argument to the number formatting macros is a 'number', essentially something that resolves to a series of arabic digits. Typical arguments might be:

- Some digits, e.g., \ordinal{123} -> 123rd
- A macro expanding to digits, e.g., \def\temp{3}\ordinal{\temp} -> 3rd
- The value of a counter, e.g., \ordinal{\value{page}} -> 327th
- The arabic representation of a counter, e.g., \ordinal{\thepage} -> 327th
   However, if the representation of a counter is not completely in arabic digits, such as \thesection which here prints as 18.14, it will produce odd errors or peculiar results if it is used as the argument. For instance: \ordinal{\thesection} -> .1418th

#### 18.14.1 Numeric numbers

The macro  $\final$  prints its  $\langle number \rangle$  argument formatted using  $\final$  prints its  $\langle number \rangle$  argument formatted using  $\final$  frumbersep between each triple of digits. The default definition of  $\final$  frumbersep is:

```
\newcommand{\fnumbersep}{,}

Here are some examples:
\fcardinal{12} -> 12
\fcardinal{1234} -> 1,234
\fcardinal{1234567} -> 1,234,567
\renewcommand*{\fnumbersep}{\:}\fcardinal{12345678} -> 12 345 678
\renewcommand*{\fnumbersep}{,\:}
```

The \cardinal macro is like \fcardinal except that there is no separation between any of the digits.

The \fordinal macro typesets its  $\langle number \rangle$  argument as a formatted ordinal, using \fnumbersep as the separator. The macro \ordinal is similar except that there is no separation between any of the digits.

```
Some examples are:
\fordinal{3} -> 3rd
\fordinal{12341} -> 12,341st
```

Typeset example 18.1: TeX's minimum number in words (English style)

The minimum number in TeX is minus two billion, one hundred and forty-seven million, four hundred and eighty-three thousand, six hundred and forty-seven (i.e., -2, 147, 483, 647)

This is the  $\chapter$  chapter. -> This is the  $18^{th}$  chapter.

The characters denoting the ordinal (ordination?) are typeset as the argument of \ordscript, whose default definition is:

```
\newcommand{\ordscript}[1]{#1}
```

As the above examples show, this can be changed to give a different appearance.

```
\nthstring \iststring \iindstring \iiirdstring
```

The ordinal characters are the values of the macros \nthstring (default: th) for most ordinals, \iststring (default: st) for ordinals ending in 1 like 21<sup>st</sup>, \iindstring (default: nd) for ordinals like 22<sup>nd</sup>, and \iiirdstring (default: rd) for ordinals like 23<sup>rd</sup>.

#### 18.14.2 Named numbers

The macro \numtoname typesets its  $\langle number \rangle$  argument using lowercase words. The other two macros are similar, except that \numtoName uses uppercase for the initial letter of the first word and \numToName uses uppercase for the initial letters of all the words.

As examples:

```
\numtoname{12345} -> twelve thousand, three hundred and forty-five \numtoName{12345} -> Twelve thousand, three hundred and forty-five \NumToName{12345} -> Twelve Thousand, Three Hundred and Forty-Five
```

```
Source for example 18.1
```

```
The minimum number in TeX is \sum_{i=0}^{\infty} 147483647 (i.e., \sum_{i=0}^{\infty} 147483647)
```

```
 \label{eq:condition} $$ \operatorname{\mathcal{n}umber} $$ \operatorname{\mathcal{n}umber} $$ \operatorname{\mathcal{n}umber} $$ \operatorname{\mathcal{n}umber} $$ $$ \operatorname{\mathcal{n}umber} $$ $$ \operatorname{\mathcal{n}umber} $$ $$ $$ $$
```

Typeset example 18.2: TeX's maximum number in words (American style)

The maximum number in TeX is two billion one hundred forty-seven million four hundred eighty-three thousand six hundred forty-seven (i.e., 2147483647).

These three macros are similar to \numtoname, etc., except that they typeset the argument as a wordy ordinal.

For example:

This is the  $\operatorname{\def}$  chapter. -> This is the eighteenth chapter.

```
\namenumberand \namenumbercomma \tensunitsep
```

By default some punctuation and conjunctions are used in the representation of named numbers. According to both American and English practice, a hyphen should be inserted between a 'tens' name (e.g., fifty) and a following unit name (e.g., two). This is represented by the value of \tensunitsep. English practice, but not American, is to include commas (the value of \namenumbercomma) and conjunctions (the value of \namenumberand) in strategic positions in the typesetting. These macros are initially defined as:

```
\newcommand*{\namenumberand}{ and }
\newcommand*{\namenumbercomma}{, }
\newcommand*{\tensunitsep}{-}
```

The next example shows how to achieve American typesetting.

```
Source for example 18.2

\renewcommand*{\namenumberand}{ }

\renewcommand*{\namenumbercomma}{ }

The maximum number in TeX is \numtoname{2147483647}

(i.e., \cardinal{2147483647}).
```

```
\minusname \lcminusname \ucminusname
```

When a named number is negative, \minusname is put before the spelled out number. The definitions of the above three comands are:

```
\newcommand*{\lcminusname}{minus }
\newcommand*{\ucminusname}{Minus }
\let\minusname\lcminusname
```

which means that 'minus' is normally all lowercase. To get 'minus' typeset with an initial uppercase letter simply:

\let\minusname\ucminusname

# Typeset example 18.3: Varieties of fractions in text

In summary, fractions can be typeset like 3/4 or 3/4 or 3/4 or 3/4 because several choices are provided.

Only one version of \namenumberand is supplied as I consider that it is unlikely that 'and' would ever be typeset as 'And'. If the initial uppercase is required, renew the macro when appropriate.

There is a group of macros that hold the names for the numbers. To typeset named numbers in a language other than English these will have to be changed as appropriate. You can find them in the class and patch code. The actual picking and ordering of the names is done by an internal macro called \n@me@number. If the ordering is not appropriate for a particular language, that is the macro to peruse and modify. Be prepared, though, for the default definitions to be changed in a later release in case there is a more efficient way of implementing their functions.

If you want to express numbers that fall outside TeX's range, Edward Reingold has produced a set of macros that can write out in words any number within the range  $-10^{66}to10^{66}$ , that is, up to a thousand vigintillion [Rei07].

#### 18.14.3 Fractions

When typesetting a simple fraction in text there is usually a choice of two styles, like 3/4 or  $\frac{3}{4}$ , which do not necessarily look as though they fit in with their surroundings. These fractions were typeset via:

```
... like 3/4 or $\frac{3}{4}$ ...
```

```
\slashfrac{\langle top \rangle}{\langle bottom \rangle}
\slashfracstyle{\langle num \rangle}
```

The class provides the \slashfrac command which typesets its arguments like  $^3/_4$ . Unlike the \frac command which can only be used in math mode, the \slashfrac command can be used in text and math modes.

The \slashfrac macro calls the \slashfracstyle command to reduce the size of the numbers in the fraction. You can also use \slashfracstyle by itself.

# Source for example 18.3

In summary, fractions can be typeset like 3/4 or  $\frac{3}{4}$  or  $\frac{3}{4}$  or  $\frac{3}{4}$  or  $\frac{3}{4}$  because several choices are provided.

```
\label{eq:linear_loss} $$ \text{textsubscript}_{\langle sub \rangle} $$
```

```
Typeset example 18.4: Super- and subscripts in text
```

In normal text you can typeset superscripts like H<sup>+</sup> and subscripts like H<sub>2</sub>O without going into math mode.

While on the subject of moving numbers up and down, the kernel provides the \textsuperscript macro for typesetting its argument as though it is a superscript. The class also provides the \textsubscript macro for typesetting its argument like a subscript.

# Source for example 18.4

In normal text you can typeset superscripts like H\textsuperscript{+} and subscripts like H\textsubscript{2}O without going into math mode.

## 18.15 An array data structure

The class includes some macros for supporting the patverse environment which may be more generally useful.

```
\label{lowarray} $$ \operatorname{newarray}(\langle arrayname \rangle)_{\langle low \rangle}_{\langle high \rangle}$
```

\newarray defines the  $\langle arrayname \rangle$  array, where  $\langle arrayname \rangle$  is a name like MyArray. The lowest and highest array indices are set to  $\langle low \rangle$  and  $\langle high \rangle$  respectively, where both are integer numbers.

```
\label{eq:lement} $$ \operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}(\operatorname{cont}
```

The \setarrayelement macro sets the  $\langle index \rangle$  location in the  $\langle arrayname \rangle$  array to be  $\langle text \rangle$ . Conversely, \getarrayelement sets the parameterless macro  $\langle result \rangle$  to the contents of the  $\langle index \rangle$  location in the  $\langle arrayname \rangle$  array. For example:

```
\setarrayelement{MyArray}{23}{$2^{23}$}
\getarrayelement{MyArray}{23}{\result}
```

will result in \result being defined as  $\left( \frac{\$2^{23}}{.} \right)$ .

\checkarrayindex checks if  $\langle arrayname \rangle$  is an array and if  $\langle index \rangle$  is a valid index for the array. If both conditions hold then \ifbounderror is set false, but if either  $\langle arrayname \rangle$  is not an array or, if it is,  $\langle index \rangle$  is out of range then \ifbounderror is set true.

```
\stringtoarray{\langle arrayname \rangle} {\langle string \rangle} 
\arraytostring{\langle arrayname \rangle} {\langle result \rangle}
```

The macro \stringtoarray puts each character from  $\langle string \rangle$  sequentially into the  $\langle arrayname \rangle$  array, starting at index 1. The macro \arraytostring assumes that  $\langle arrayname \rangle$ 

is an array of characters, and defines the macro  $\langle \textit{result} \rangle$  to be that sequence of characters. For example:

The command \checkifinteger ckecks if  $\langle num \rangle$  is an integer (not less than zero). If it is then \iffinteger is set true, otherwise it is set false.

*Note.* Please note that \checkifinteger may only work on simple input.

## 18.16 Checking the processor

## 18.16.1 Checking for pdfLaTeX

Both LaTeX and pdfLaTeX can be run on the same document. LaTeX produces a .dvi file as output, while pdfLaTeX can produce either a .dvi or a .pdf file. On modern systems pdfLaTeX produces a pdf file by default.

If you want a dvi file output use LaTeX and if you want a pdf file use pdfLaTeX.

```
\ifpdf ... \fi
```

The class provides \ifpdf (by autoloading the ifpdf package) which is true when the document is being processed by pdfLaTeX and false otherwise. You can use it like this:

```
\ifpdf
  code for pdfLaTeX only
\else
  code for LaTeX only
\fi
```

If there is no LaTeX specific code then don't write the \else part.

# 18.16.2 Checking for etex

Modern LaTeX distributions use etex, which is an extended version of TeX, as the underlying engine. etex provides some more primitives than TeX, which may be useful, but not everybody has etex available (Though, as of 2018, this is *very* rare).

```
\ifetex
```

\ifetex can be used to determine if etex is being used as the underlying engine; it is analogous to \ifpdf which tests for pdfLaTeX (provided by autoloading the ifetex package). For example:

```
\ifetex
   %%% code only processible by etex
\else
```

```
\typeout{etex is not available}
\fi
```

## 18.16.3 Checking for XeTeX

You have been able to use \ifpdf to check if pdfLaTeX is being used to process the document.

```
\ifxetex
```

In a similar manner you can use \ifxetex to check if the document is being processed by XeTeX (provided by autoloading the ifxetex package).

```
\RequireXeTeX
```

The ifxetex package also provides \RequireXeTeX, which generates an error if XeTeX is not being used to process the document. This can be useful if you make your own class building upon memoir.

18.16.4 Checking for LuaTeX

Similarly you can use

```
\ifluatex
```

to check if the doc is being process by LuaTeX.

# 18.17 Leading

LaTeX automatically uses different leading for different font sizes.

```
\baselineskip \onelineskip
```

At any point in a document the standard LaTeX \baselineskip length contains the current value of the leading<sup>5</sup>. The class provides the length \onelineskip which contains the initial leading for the normal font. This value is useful if you are wanting to specify length values in terms of numbers of lines of text.

## 18.18 Minor space adjustment

The kernel provides the \, macro for inserting a thin space in both text and math mode. There are other space adjustment commands, such as \! for negative thin space, and \: and \; for medium and thick spaces, which can only be used in math mode.

```
\thinspace \medspace \: \!
```

On occasions I have found it useful to be able to tweak spaces in text by some fixed amount, just as in math mode. The kernel macro \thinspace specifies a thin space, which is 3/18 em. The class \medspace specifies a medium space of 4/18 em. As mentioned, the kernel macro \: inserts a medium space in math mode. The class version can be used in both math and text mode to insert a medium space. Similarly, the class version of \! can be used to insert a negative thin space in both text and math mode.

The math thick space is  $5/18\,\mathrm{em}$ . To specify this amount of space in text mode you can combine spacing commands as:

<sup>5</sup> This statement ignores any attempts to stretch the baseline.

\:\:\!

which will result in an overall space of 5/18 em (from (4+4-3)/18).

# 18.19 Adding a period

Much earlier, when showing the code for the sectional division styles for this document, I used the macro \addperiod.

This puts a period (a full stop) at the end of  $\langle text \rangle$ . I used it to add a period at the end of the \paragraph and \subparagaph titles. When sectional titles, like \paragraph are run-in, it is customary to end them with a period (or occasionally a colon).

## 18.20 Words and phrases

The class provides several macros that expand into English words or phrases. To typeset in another language these need to be changed, or an author or publisher may want some changes made to the English versions. Table 18.1 lists the macros, their default values, and where they used.

Most, if not all, of the tabulated definitions are simple — for example

```
\newcommand*{\partname}{Part}
\newcommand*{\partrefname}{Part~}
```

and so can be also changed simply.

The definitions of the macros for the names of numbers are more complex — for example for the number 11 (eleven)

```
\newcommand*{\nNamexi}{\iflowertonumname e\else E\fi leven}
```

That is, each definition includes both a lowercase and an uppercase initial letter, so a bit more care has to be taken when changing these. For specifics read the documentation of the class code.

# 18.21 Symbols

LaTeX lets you typeset an enormous variety of symbols. The class adds nothing to the standard LaTeX capabilities in this respect. If you want to see what symbols are available then get a copy of Scott Pakin's *The Comprehensive LaTeX Symbol List* [Pak01]. You may have to do a little experimentation to get what you want, though.

For example, the \texttrademark command produces the trademark symbol  $^{\text{\tiny IM}}$ , but the \textregistered command produces  $^{\text{\tiny IM}}$ . When I wanted to use the registered trademark symbol it needed to be typeset like a superscript instead of on the baseline. The \textsuperscipt macro typesets its argument like a superscript, so using

```
\textsuperscript{\textregistered}
```

gave the required result<sup>®</sup>.

Table 18.1: Defined words and phrases

Table 10.1. Defined words and phrases				
Macro	Default	Usage		
\abstractname	Abstract	title for abstract environment		
\alsoname	see also	used by \seealso		
\amname	am	used in printing time of day		
\appendixname	Appendix	name for an appendix heading		
\appendixpagename	Appendices	name for an \appendixpage		
\appendixtocname	Appendices	ToC entry announcing appendices		
\bibname	문헌목록	title for \thebibliography		
\bookname	Book	name for \book heading		
\bookrefname	Book	used by \Bref		
\chaptername	Chapter	name for \chapter heading		
\chapterrefname	Chapter	used by \Cref		
\contentsname	목차	title for \tableofcontents		
\figurename	Figure	name for figure \caption		
\figurerefname	Figure	used by \fref		
\glossaryname	Glossary	title for \theglossary		
\indexname	색인	title for \theindex		
\lcminusname	minus	used in named number formatting		
\listfigurename	圖 목차	title for \listoffigugres		
\listtablename	表 목차	title for \listoftables		
\minusname	minus	used in named number formatting		
\namenumberand	and	used in named number formatting		
\namenumbercomma	,	used in named number formatting		
\notesname	Notes	title of \notedivision		
\pagename	page	for your use		
\pagerefname	page	used by \pref		
\partname	Part	name for \part heading		
\partrefname	Part	used by \Pref		
\pmnane	pm	used in printing time of day		
\sectionrefname	§	used by \Sref		
\seename	see	used by \see		
\tablename	Table	name for table \caption		
\tablerefname	Table	used by \tref		
\ucminusname	Minus	used in named number formatting		

# 18.22 Two simple macros

There are two trivial macros that can be generally useful.

```
\label{eq:local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_
```

The \memjustarg macro just uses its argument and is defined as:

```
\newcommand*{\memjustarg}[1]{#1}
```

The \memgobble macro gobbles down and swallows its argument. Its definition is:

```
\newcommand{\memgobble}[1]{}
```

Do *not* redefine either \memjustarg or \memgobble; if you do various pieces of code will behave in unexpected ways that you will not like.

### 18.23 Vertical centering

Earlier there was a description of a method for centering text vertically. The vplace environment provides a simpler and more general way.

```
\lceil \sqrt{\text{place}} \rceil  text \rceil
```

The contents of the vplace environment are vertically centered. The optional  $\langle num \rangle$  argument can be used to specify the ratio of the upper space to the lower space. You can put other text on the page above or below the centered text. The environment may be useful for title pages.

#### 18.24 For package writers

The facilities described in this section are for anyone to use but I suspect that they may be most useful to package developers.

## 18.24.1 Emulating packages

```
\label{lem:latedPackage} $$ \operatorname{Density}_{\langle ade\rangle} [\langle date\rangle] $$ \operatorname{Density}_{\langle ade\rangle}_{\langle ade\rangle} [\langle date\rangle] $$
```

These commands are for package writers; they are based on a conversation with Donald Arseneau on CTT. They fool LaTeX into thinking that the  $\langle package \rangle$  has already been loaded so it won't try loading it again. These are probably only useful if your package includes the actual code for  $\langle package \rangle$ .

memoir does include code from several packages and uses a similar internal command to ensure that the packages are not loaded following some later \usepackage command. The names of the emulated packages are written to the log file. At the time of writing the emulated packages are: abstract, appendix, array, booktabs, ccaption, chngcntr, crop, dcolumn, delarray, enumerate, epigraph, ifmtarg, ifpdf, index, makeidx, moreverb, needspace, newfile, nextpage, pagenote, patchcmd, parskip, setspace, shortvrb, showidx, tabularx, titleref, tocbibind, tocloft, verbatim, and verse. As well as the emulated packages memoir provides functions equivalent to those in the following packages, although the class does not prevent you from using them: fancyhdr, framed, geometry, sidecap, subfigure, and titlesec.

```
\verb|\DisemulatePackage{\langle package \rangle}|
```

This command undoes any prior  $\mbox{EmulatedPackage}$  or  $\mbox{EmulatedPackageWithOptions}$  for the  $\mbox{package}$  package. For example, if you wish to use the index package instead of memoir's emulation then put

```
\DisemulatePackage{index}
\usepackage{index}
```

in your preamble.

18.24.2 Inserting code before and after a file, package or class

The kernel provides two commands, \AtBeginDocument and \AtEndDocument which can only be used in the preamble, for inserting code at the start and end of the document environment.

The kernel also provides the macros  $\AtEndOfPackage\{\langle code\rangle\}\$  and  $\AtEndOfClass\{\langle code\rangle\}\$  for inserting code at the end of the current package or class. More precisely, these macros call the  $\langle code\rangle$  after the package or class file has been input via  $\InputIfFileExists$ .

The class provides a more comprensive set of macros for code insertions, which should be used before the relevant file is called for.

The \AtBeginFile macro inserts  $\langle code \rangle$  just before the  $\langle file \rangle$  file is \input (or \included, etc.). Similarly \AtEndFile inserts the  $\langle code \rangle$  immediately after the  $\langle file \rangle$ . The  $\langle file \rangle$  argument must be the same as used in the corresponding \input command. If  $\langle file \rangle$  includes an extension, for example fred.def, then that is taken as the complete name, otherwise if there is no extension, for instance fred, then the .tex extension is automatically appended making the full name fred.tex.

The  $\At...File$  commands must be issued *before* the corresponding  $\langle file \rangle$  is input otherwise nothing will happen.

The \AtBeginPackage command will insert  $\langle code \rangle$  just before the  $\langle pack \rangle$  package is used. Similarly \AtEndPackage will insert the  $\langle code \rangle$  immediately after the  $\langle pack \rangle$ . The  $\langle pack \rangle$  argument must be the same as used in the corresponding \usepackage command, that is, without any extension. The \At...Package commands must be issued *before* the corresponding  $\langle pack \rangle$  is used otherwise nothing will happen.

The \RequireAtEndPackage command will, like \AtEndPackage, insert  $\langle code \rangle$  at the end of the  $\langle pack \rangle$  package if it has not yet been used. If the package has already been used then the  $\langle code \rangle$  is called immediately.

```
\label{lass} $$ \AtBeginClass{\langle class\rangle} {\langle code\rangle} $$ \AtEndClass{\langle class\rangle} {\langle code\rangle} $$ \RequireAtEndClass{\langle class\rangle} {\langle code\rangle} $$
```

The \AtBeginClass command will insert  $\langle code \rangle$  just before the  $\langle class \rangle$  class is used. Similarly \AtEndClass will insert the  $\langle code \rangle$  immediately after the  $\langle class \rangle$ . The  $\langle class \rangle$  argument must be the same as used in the corresponding \LoadClass command, that is, without any

extension. The At...Class commands must be issued *before* the corresponding  $\langle class \rangle$  is used otherwise nothing will happen.

The \RequireAtEndClass command will, like \AtEndClass, insert  $\langle code \rangle$  at the end of the  $\langle class \rangle$  class if it has not yet been used. If the class has already been used then the  $\langle code \rangle$  is called immediately.

There is an unfortunate interaction between the kernel's \AtEndOfPackage and the class's \AtEndPackage, and similarly for the \AtEndOfClass and \AtEndClass. I discovered this when I tried to automate using the memhfixc package if hyperref was being used by putting the following into the memoir code

```
\AtEndPackage{hyperref}{\usepackage{memhfixc}}
```

which caused all sorts of problems.

The kernel scheme looks like this:

```
\newcommand{\usepackage}[1]{%
    ...
  \InputIfFileExists{#1}
<AtEndOfPackage code>}
```

The basic mechanism for implementing the class macros is by modifying the kernel's  $\label{linear}$  InputIfFileExists macro, which internally uses a form of  $\label{linear}$  to read in the file, so that the inserted  $\langle code \rangle$  comes immediately before and after the  $\label{linear}$  somewhat like:

```
\renewcommand{\InputIfFileExists}[1]{%
...
<before code> \input{#1} <after code>}
```

If  $\AtEndPackage$  is applied to a package that has an internal  $\AtEndOfPackage$  then the result can be sketched as:

```
\newcommand{\usepackage}[1]{%
    ...
    <before code>
    \input{#1}
    <after code>
    <AtEndOfPackage code>
}
```

In other words the body of the package is read in, the \AtEndPackage code is called, and then after that the \AtEndOfPackage code is called.

The hyperref package internally uses \AtEndOfPackage to read some files and memhfixc had to be input after these. A way to automate memhfixc after hyperref is:

```
\AtEndPackage{hyperref}{%
  \AtBeginDocument{\usepackage{memhfixc}}}
```

but this seems more trouble than it's worth especially since Heiko Oberdiek has kindly updated hyperref so that versions after 2006/11/15 will automatically load the memhfixc package.

```
18.25 Heading hooks
```

On 2nd September 2005 I posted two messages to the comp.text.tex newsgroup saying that I was creating a new version of memoir and that I would consider inserting hooks into

the class code that package writers might find useful. I got no requests for any hooks or anything else from package writers. I therefore assume that no package author sees any problems if a memoir class document author uses the package.

However, I have provided macros that may be useful for those who want to do things with the contents of section headings, captions, and the like. The macros are called within the relevant heading or caption code, and by default are defined to do nothing.

Hooks for the \book and \book\* commands.

```
\label{lembookinfo} $$ \mathbf{thebook} {\langle thebook \rangle} {\langle title \rangle} $$ \mathbf{title} $$
```

Hooks for the \part and \part\* commands.

In many cases a \mem...info macro includes an argument related to the heading's number ( $\langle thepart \rangle$  for \mempartinfo). In certain circumstances, such as a \chapter in the \frontmatter, there might not be a number even though the normal unstarred version of the command is used. In these cases the number argument ( $\langle thechapter \rangle$  in the case of \memchapinfo) is left empty.

Hooks for the \chapter and \chapter\* commands. Note that regular chapters and those as appendices are treated differently.

Hooks for \section, \subsection, etc., and their starred versions.  $\langle name \rangle$  is the type of section (e.g., section, or subsection or subsubsection or ...

Hooks for appendix-like page headings.

Hooks for \poemtitle, \PoemTitle, and their starred versions.

Hooks for the several kinds of \caption and \legend commands.

```
\label{eq:local_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_continuous_cont
```

As an example of how one of these macros might be used, just before the start of this section I put

and now I'm putting

```
The arguments are: (1) '\Margi', (2) '\Margii', (3) '\Margiii', (4) '\Margiv', (5) '\Margv'.
```

The arguments are: (1) 'section', (2) '18.25', (3) 'Heading hooks', (4) 'Heading hooks', (5) 'Heading hooks'.

**Warning:** Be very careful with the fifth argument of this macro when using hyperref, this argument will then contain a hyper link anchor, whih may cause problems when used out of context.

# 18.26 Documenting LaTeX commands

The class provides a few macros to help you if you want to describe LaTeX commands.

```
\label{local_cond} $$ \cs{\langle name \rangle} \cmdprint{\langle cmd \rangle} \cmd{\langle cmd \rangle}$
```

The macro \bs simply prints the '' backslash.

The macro \cs prints its argument, putting a backslash in front of it. For example \cs{name} prints \name.

The argument to \cmdprint should be the name of a macro, including the backslash. It is then printed as is. For instance \cmdprint{\amacro} prints \amacro.

The argument to \cmd should be the name of a macro, including the backslash. It is then printed, using \cmdprint, and also added to the index file with the assumption that? will be used as the 'actual' character (the default is @ which is not of much use if you are trying to index macro names that have @ as part of their names).

```
\label{eq:loss_arg} $$ \operatorname{darg}  \ \operatorname{darg}  \ \operatorname{darg}  \ \ \operatorname{darg}  \ $$
```

The macro \meta{ $\langle arg \rangle$ } prints  $\langle arg \rangle$  for an argument to a macro.

The macro  $\mbox{marg}\{\langle arg \rangle\}$  prints  $\{\langle arg \rangle\}$  for a required argument.

The macro  $\backslash \text{oarg}\{\langle arg \rangle\}$  prints  $[\langle arg \rangle]$  for an optional argument.

The macro \parg{ $\langle arg \rangle$ } prints ( $\langle arg \rangle$ ) for a parenthesized argument.

# Nineteen

# For package users

Many packages work just as well with memoir as with any of the standard classes. In some instances, though, there may be a problem. In other instances the class and package have been designed to work together but you have to be careful about any code that you might write.

# 19.1 Class/package name clash

A typical indication of a problem is an error message saying that a command has already been defined (see page 454).

When the class and a package both use the same name for a macro that they define something has to give. For the sake of an example, assume that memoir has defined a macro called \foo and a package pack that is used in the document also defines \foo. There are several options that you can choose which might resolve the difficulty.

1. Just keep the class definition:

2. Just keep the package definition:

3. Keep the class definition and rename the package definition:

4. Keep the package definition and rename the class definition:

A potential problem with these options can occur after the package is loaded and there are class or package commands that you use that, knowingly or not, call \foo expecting to get the class or the package definition, one of which is now not available (except under a different name).

The memoir class has been available since 2001. It seems likely that if older packages clash with memoir then, as eight years have gone by, the authors are unlikely to do anything about it. If a newer package clashes then contact the author of the package.

If all else fails, ask for help on the CTT newsgroup.

# 19.2 Support for bididirectional typesetting

The bidi system [Kha10] provides means of bidirectional typesetting. The class has built in support for bidi but this means that if you are defining your own macros there are some things you need to be aware of.

When dealing with bidirectional texts the left-to-right (LTR) direction is the familiar one and LaTeX is set up for this. When typesetting right-to-left (RTL) bidi interchanges left and right. The support in memoir consists of replacing many, but not all, of the right- and left-specific constructs. The replacement macros are:

```
\memRTLleftskip \memRTLrightskip \memRTLvrightskip \memRTLvrightskip \memRTLraggedright \memRTLraggedleft
```

In certain places, but not everywhere:
\memRTLleftskip is used instead of \leftskip
\memRTLrightskip is used instead of \rightskip
\memRTLvleftskip is used instead of \vleftskip
\memRTLvrightskip is used instead of \vrightskip
\memRTLraggedleft is used instead of \raggedleft
\memRTLraggedright is used instead of \raggedright

So, if you are defining any macros that use the \...skip or \ragged... macros you may have to use the \memRTL... version instead. The memoir definitions of these macros are simply:

```
\newcommand*{\memRTLleftskip}{\leftskip}
\newcommand*{\memRTLrightskip}{\rightskip}
...
\newcommand*{\memRTLraggedleft}{\raggedleft}
```

The bidi system redefines them to suit its purposes. If your work will only be set LTR there is no need to use the \memRTL... macros in any of your code. If you might ever be producing a bidirectional document then you may have to use the \memRTL... versions of the standard commands in your code. To determine where you might have to use them you will have to consult the bidi documentation as not every use of the standard commands needs to be replaced.

#### 20.1 introduction

In this chapter I will work through a reasonably complete design exercise. Rather than trying to invent something myself I am taking the design of Bringhurst's *The Elements of Typographic Style* [Bri99] as the basis of the exercise. This is sufficiently different from the normal LaTeX appearance to demonstrate most of the class capabilities, and also it is a design by a leading proponent of good typography.

As much as possible, this chapter is typeset according to the results of the exercise to provide both a coding and a graphic example.

# 20.2 design requirements

The *Elements of Typographic Style* is typeset using Minion as the text font and Syntax (a sans font) for the captions.

The trimmed page size is 23 by 13.3cm. The foredge is 3.1cm and the top margin is 1.9cm.

As already noted, the font for the main text is Minion, with 12pt leading on a 21pc measure with 42 lines per page. For the purposes of this exercise I will assume that Minion can be replaced by the font used for this manual. The captions to figures and tables are unnamed and unnumbered and typeset in Syntax. The captions give the appearance of being in a smaller font size than the main text, which is often the case. I'll assume that the \small\sfseries font will reasonably do for the captions.

The footer is the same width as the typeblock and the folio is placed in the footer at the fore-edge. There are two blank lines between the bottom of the typeblock and the folio.

There is no header in the usually accepted sense of the term but the chapter title is put on recto pages and section titles are on verso pages. The running titles are placed in the fore-edge margin level with the seventh line of the text in the typeblock. The recto headers are typeset raggedright and the verso ones raggedleft.

Bringhurst also uses many marginal notes, their maximum width being about 51pt, and typeset raggedright in a smaller version of the textfont.

Chapter titles are in small caps, lowercase, in a larger font than for the main text, and a rule is placed between the title and the typeblock. The total vertical space used by a chapter title is three text lines. Chapters are not numbered in the text but are in the ToC.

Section titles are again in small caps, lowercase, in the same size as the text font. The titles are numbered, with both the chapter and section number.

A subsection title, which is the lowest subdivision in the book, is in the italic form of the textfont and is typeset as a numbered non-indented paragraph. These are usually multiline as Bringhurst sometimes uses them like an enumerated list, so on occasion there is a subsection title with no following text.

Only chapter titles are put into the ToC, and these are set raggedright with the page numbers immediately after the titles. There is no LoF or LoT.

Note that unlike the normal LaTeX use of fonts, essentially only three sizes of fonts are used — the textfont size, one a bit larger for the chapter titles, and one a bit smaller for marginal notes and captions. Also, bold fonts are not used except on special occasions, such

as when he is comparing font families and uses large bold versions to make the differences easier to see.

20.3 specifying the page and typeblock

The first and second things to do are to specify the sizes of the page after trimming and the typeblock. The trimmed size is easy as we have the dimensions.

Specifying the page and typeblock

```
\strimmedsize{23cm}{13.3cm}{*}
```

We want 42 lines of text, so that's what we set as the height of the typeblock; however, we have to remember to ask for lines as the optional  $\langle algorithm \rangle$  argument when we finally call  $\c$ 

```
\settypeblocksize{42\onelineskip}{21pc}{*}
```

To make life easier, we'll do no trimming of the top of the stock

```
\setlength{\trimtop}{0pt}
```

but will trim the fore-edge. The next set of calculations first sets the value of the \trimedge to be the \stockwidth; subtracting the trimmed \paperwidth then results in \trimedge being the amount to trim off the foredge.

```
\setlength{\trimedge}{\stockwidth} \addtolength{\trimedge}{-\paperwidth}
```

The sizes of the trimmed page and the typeblock have now been specified. The typeblock is now positioned on the page. The sideways positioning is easy as we know the fore-edge margin to be 3.1cm.

```
\setlrmargins{*}{3.1cm}{*}
```

The top margin is specified as 1.9cm, which is very close to four and a half lines of text. Just in case someone might want to use a different font size, I'll specify the top margin so that it is dependent on the font size. The \footskip can be specified now as well (it doesn't particularly matter what we do about the header-related lengths as there isn't anything above the typeblock).

```
\setulmargins{4.5\onelineskip}{*}{*}
\setheadfoot{\onelineskip}{3\onelineskip}
\setheaderspaces{\onelineskip}{*}{*}
```

Lastly define the dimensions for any marginal notes.

```
\setmarginnotes{17pt}{51pt}{\onelineskip}
```

If this was for real, the page layout would have to be checked and implemented.

```
\checkandfixthelayout[lines]
```

It is possible to implement this layout just for this chapter but I'm not going to tell you either how to do it, or demonstrate it. Except under exceptional circumstances it is not good to make such drastic changes to the page layout in the middle of a document. However, the picture on page 347 illustrates how this layout would look on US letterpaper stock. Looking at the illustration suggests that the layout would look rather odd unless the stock was trimmed down to the page size — another reason for not switching the layout here.

Dashed lines represent the actual page size after trimming the stock.



An example book design

```
Lengths are to the nearest pt.
\stockheight = 795pt
                             \stockwidth = 614pt
\pageheight = 654pt
\textheight = 502pt
                              \pagewidth = 378pt
\textwidth = 252pt
\trimtop = Opt
                              \trimedge = 236pt
\uppermargin = 54pt
                              \spinemargin = 38pt
\headheight = 12pt
                              \headsep = 30pt
\footskip = 36pt
                              \marginparsep = 17pt
\marginparpush = 12pt
                              \columnsep = 10pt
\columnseprule = 0.0pt
```

An illustration of Bringhurst's page layout style when printed on US letter paper stock. Also shown are the values used for the page layout parameters for this design.

20.4 specifying the sectional titling styles

#### 20.4.1 The chapter style

Recapping, chapter titles are in small caps, lowercase, in a larger font than for the main text, and a rule is placed between the title and the typeblock. The total vertical space used by a chapter title is three text lines. Chapters are not numbered in the text but are in the ToC. Titles in the ToC are in mixed case.

Specifying the sectional titling styles

The definition of the chapterstyle is remarkably simple, as shown below.

```
%% Bringhurst chapter style
\makechapterstyle{bringhurst}{%
  \renewcommand{\chapterheadstart}{}
  \renewcommand{\printchaptername}{}
  \renewcommand{\chapternamenum}{}
  \renewcommand{\printchapternum}{}
  \renewcommand{\afterchapternum}{}
  \renewcommand{\printchaptertitle}[1]{%
   \raggedright\Large\scshape\MakeLowercase{##1}}
  \renewcommand{\afterchaptertitle}{%
   \vskip\onelineskip \hrule\vskip\onelineskip}
}
```

Most of the specification consists of nulling the majority of the normal LaTeX specification, and modifying just two elements.

The chapter title (via \printchaptertitle) is typeset raggedright using the \Large smallcaps fonts. The \MakeLowercase macro is used to ensure that the entire title is lowercase before typesetting it. Titles are input in mixed case.

After the title is typeset the \afterchaptertitle macro specifies that one line is skipped, a horizontal rule is drawn and then another line is skipped.

# 20.4.2 Lower level divisions

Section titles are in small caps, lowercase, in the same size as the text font. The titles are numbered, with both the chapter and section number.

The specification is:

```
\setsecheadstyle{\raggedright\scshape\MakeLowercase}
\setbeforesecskip{-\onelineskip}
\setaftersecskip{\onelineskip}
```

The macro \setsecheadstyle lowercases the title and typesets it small caps.

The default skips before and after titles are rubber lengths but this does not bode well if we are trying to line something up with a particular line of text — the presence of section titles may make slight vertical adjustments to the text lines because of the flexible spacing. So, we have to try and have fixed spacings. A single blank line is used before (\setbeforesecskip) and after (\setaftersecskip) the title text.

A subsection title, which is the lowest subdivision in the book, is in the italic form of the textfont and is typeset as a numbered non-indented paragraph. The code for this is below.

```
\setsubsecheadstyle{\sethangfrom{\noindent ##1}\raggedright\itshape}
\setbeforesubsecskip{-\onelineskip}
\setaftersubsecskip{\onelineskip}
```

As in the redefinition of the \section style, there are fixed spaces before and after the title text. The title is typeset (\setsubsecheadstyle) raggedright in a normal sized italic font. The macro \sethangfrom is used to to redefine the internal \@hangfrom macro so that the title and number are typeset as a block paragraph instead of the default hanging paragraph style. Note the use of the double ## mark for denoting the position of the argument to \@hangfrom.

An example book design

```
20.5 specifying the pagestyle
```

The pagestyle is perhaps the most interesting aspect of the exercise. Instead of the chapter and section titles being put at the top of the pages they are put in the margin starting about seven lines below the top of the typeblock. The folios are put at the bottom of the page aligned with the outside of the typeblock.

As the folios are easy, we'll deal with those first.

```
%% Bringhurst page style
\makepagestyle{bringhurst}
\makeevenfoot{bringhurst}{\thepage}{}{}
\makeoddfoot{bringhurst}{}{\thepage}
```

Putting text at a fixed point on a page is typically done by first putting the text into a zero width picture (which as far as LaTeX is concerned takes up zero space) and then placing the picture at the required point on the page. This can be done by hanging it from the header.

We might as well treat the titles so that they will align with any marginal notes, which are \marginparsep (17pt) into the margin and \marginparwidth (51pt) wide. Earlier in the manual I defined two lengths called \pwlayi and \pwlayii which are no longer used. I will use these as scratch lengths in performing some of the necessary calculations.

For the recto page headers the picture will be the  $\langle right \rangle$  part of the header and for the verso pages the picture will be the  $\langle left \rangle$  part of the header, all other parts being empty.

For the picture on the  $\langle right \rangle$  the text must be 17pt to the right of the origin, and some distance below the origin. From some experiments, this distance turns out to be the \headsep plus the \topskip plus 7.3 lines, which is calculated as follows:

```
\setlength{\pwlayi}{\headsep}
\addtolength{\pwlayi}{\topskip}
\addtolength{\pwlayi}{7.3\onelineskip}
```

There is a nifty internal LaTeX macro called \strip@pt which you probably haven't heard about, and I have only recently come across. What it does is strip the 'pt' from a following length, reducing it to a plain real number. Remembering that the default \unitlength is 1pt we can do the following, while making sure that the current \unitlength is 1pt:

```
\makeatletter
\newcommand{\bringpicr}[1]{%
\setlength{\unitlength}{1pt}}
```

```
\begin{picture}(0,0)
  \put(\strip@pt\marginparsep, -\strip@pt\pwlayi){%
  \begin{minipage}[t]{\marginparwidth}
     \raggedright\itshape #1
  \end{minipage}}
  \end{picture}
}
\makeatother
```

Specifying the pagestyle

The new macro \bringpicr{ $\langle text \rangle$ } puts  $\langle text \rangle$  into a minipage of width \marginparwidth, typeset raggedright in an italic font, and puts the top left of the minipage at the position (\marginparsep, -\pwlayi) in a zero width picture.

We need a different picture for the  $\langle \textit{left} \rangle$  as the text needs to be typeset raggedleft with the right of the text 17pt from the left of the typeblock. I will use the length \pwlayii to calculate the sum of \marginparsep and \marginparwidth. Hence:

```
\makeatletter
\setlength{\pwlayii}{\marginparsep}
\addtolength{\pwlayii}{\marginparwidth}
\newcommand{\bringpicl}[1]{%
\setlength{\unitlength}{1pt}
\begin{picture}(0,0)
\put(-\strip@pt\pwlayii, -\strip@pt\pwlayi){%
\begin{minipage}[t]{\marginparwidth}
\raggedleft\itshape #1
\end{minipage}}
\end{picture}
}
\makeatother
```

The new macro \bringpicl{ $\langle text \rangle$ } puts  $\langle text \rangle$  into a minipage of width \marginparwidth, typeset raggedleft in an italic font, and puts the top left of the minipage at the position (-(\marginparsep + \marginparwidth), -\pwlayi) in a zero width picture.

Now we can proceed with the remainder of the pagestyle specification. The next bit puts the chapter and section titles into the  $\verb+\...mark$  macros.

```
\makeatletter
\makepsmarks{bringhurst}{%
  \def\chaptermark##1{\markboth{##1}{##1}}
  \def\sectionmark##1{\markright{##1}}
}
\makeatother
```

Finally, specify the evenhead using \bringpicl with the section title as its argument, and the oddhead using \bringpicr with the chapter title as its argument.

```
\makeevenhead{bringhurst}{\bringpic1{\rightmark}}{}{\
\makeoddhead{bringhurst}{}{}\bringpicr{\leftmark}}
```

# 20.6 captions and the toc

The captions to figures and tables are set in a small sans font and are neither named nor numbered, and there is no LoF or LoT. Setting the caption titles in the desired font is simple:

```
\captiontitlefont{\small\sffamily}
```

There are two options regarding table and figure captioning: either use the \legend command (which produces an anonymous unnumbered title) instead of the \caption command, or use the \caption command with a modified definition. Just in case the design might change at a later date to required numbered captions, it's probably best to use a modified version of \caption. In this case this is simple, just give the \caption command the same definition as the \legend command.

An example book design

```
\let\caption\legend
```

An aside: I initially used the default caption style (block paragraph) for the diagram on page 347, but this looked unbalanced so now it has the last line centered. As a float environment, like any other environment, forms a group, you can make local changes within the float. I actually did it like this:

```
\begin{figure}
\captiontitlefont{\small\sffamily}
\captionstyle{\centerlastline}
...
\legend{...} \label{...}
\end{figure}
```

For fine typesetting you may wish to change the style of particular captions. The default style for a single line caption works well, but for a caption with two or three lines either the centering or centerlastline style might look better. A very long caption is again probably best done in a block paragraph style.

Only chapter titles are included in the ToC. To specify this we use the \settocdepth command.

```
\settocdepth{chapter}
```

The ToC is typeset raggedright with no leaders and the page numbers coming immediately after the chapter title. This is specified via:

```
\renewcommand{\cftchapterfont}{\normalfont}
\renewcommand{\cftchapterpagefont}{\normalfont}
\renewcommand{\cftchapterpresnum}{\bfseries}
\renewcommand{\cftchapterleader}{}
\renewcommand{\cftchapterafterpnum}{\cftparfillskip}
```

# 20.7 preamble or package?

When making changes to the document style, or just defining a new macro or two, there is the question of where to put the changes — in the preamble of the particular document or into a separate package?

If the same changes/macros are likely to be used in more than one document then I suggest that they be put into a package. If just for the single document then the choice remains open.

I have presented the code in this chapter as though it would be put into the preamble, hence the use of \makeatletter and \makeatother to surround macros that include the @ character (see §E.4). The code could just as easily be put into a package called, say, bringhurst. That is, by putting all the code, except for the \makeatletter and \makeatother commands, into a file called bringhurst.sty. It is a good idea also to end the code in the file with \endinput; LaTeX stops reading the file at that point and will ignore any possible garbage after \endinput.

Preamble or package?

You then use the bringhurst package just like any other by putting

\usepackage{bringhurst}

in your document's preamble.

# Twenty-one

# An example thesis design

Many universities in the United States have strict regulations about the typography of theses. The title and administrative pages are inherently specific to a particular university, but often the design for the body of the thesis clashes with normally accepted typographic practice. This chapter presents fairly typical guidelines for some US universities and code intended to meet them. Let's call the university in question the *Archibald Smythe University*, or ASU for short.

The requirements that are listed below are not from any single university but I have, over the years, seen each one of them. In reality there are likely to be many more nit-picking rules than I have shown.

It amuses me that I have never seen a printed set of requirements that followed the rules laid down therein.

Universities outside the US tend to be more relaxed with the result that theses from these establishments are very often more attractive (certainly less bulky) and more readable. The ASU requirements lead to an exceptionally dull and unattractive appearance.

#### 21.1 Example US thesis typographic requirements

# 21.1.1 General

Paper size The thesis shall be printed on 8.5 by 11 inch plain white paper.

Single-sided The thesis shall be printed single-sided; that is, one side of each sheet of paper will be blank.

Margins Every page of the document shall meet the requirements of a 1.5 inch margin on the left and a 1 inch margin at the top, right, and bottom of the page. Nothing shall appear in any margin.

Fonts The thesis may be set in 10, 11 or 12pt Arial, Century, Garamond, Lucida Bright (10pt only), Tahoma, Times, or Verdana. The same font and size shall be used throughout the thesis.

There shall be no bold type.

Italic type (or underlining) is limited to the names of species, genera, book titles, musical compositions, or foreign words.

Line Spacing All text shall be double-spaced, except material in tables and the optional biographical sketch (which must be single-spaced). You shall single-space individual footnotes and endnotes with a double space between each entry.

# 21.1.2 Preliminary matter

The preliminary matter consists of the following pages in this order:

- 1. Title page
- 2. Approval page
- 3. Abstract
- 4. Dedication (optional)
- 5. Acknowledgements (optional)
- 6. Table of contents
- 7. List of tables (if there are any tables)
- 8. List of figures (if there are any figures)
- 9. Other lists (e.g., nomenclature, definitions, glossary of terms, etc.)
- 10. Preface (optional but must be less than ten pages)
- 11. Under special circumstances further sections may be allowed

The heading for each preliminary page (except the Dedication which shall not have a heading) is centered between the margins, in all capital letters, double-spaced and begin on the first line below the top margin.

The title and approval page are counted as pages one and two, but no page numbers shall appear on them. All subsequent preliminary pages are paginated with lowercase Roman numerals. Starting with 'iii' on the abstract page, place all page numbers at the bottom of the page, centered between the left and right margins and upon the 1 inch bottom margin. Continue numbering consecutively on the subsequent pages up to the first page of the main text.

# Title page

- 1. All text shall be centered between the side margins.
- 2. Set the title in all capital letters, double-spaced, starting at the top of the page (but below the top margin).
- 3. On a new line (double-spaced) type 'by' in lowercase letters.
- 4. On a new line (double-spaced) type your full legal name.
- 5. At the center of the page type the appropriate description for your degree with the exact wording and line breaks as shown, and single-spaced:

Α	Presented in Partial Fulfillment
	of the Requirements for the Degree

Replace the blanks with the appropriate wording: Thesis and Master of Arts or Dissertation and Doctor of Philosophy.

- 6. At the bottom of the page type 'ARCHIBALD SMYTHE UNIVERSITY' in all capitals.
- 7. Type the month and year of the date you will graduate, with the month in title case and no comma between the month and year.
- 8. The space between your name and the degree description should equal the space between the degree description and the name of the University.

# Approval page

- 1. All text shall be centered between the side margins.
- 2. Set the title in all capital letters, double-spaced, starting at the top of the page (but below the top margin).
- 3. On a new line (double-spaced) type 'by' in lowercase letters.
- 4. On a new line (double-spaced) type your full legal name in title-cased letters.
- Add two double-spaced lines (four single-spaced lines) and type 'has been aproved' in lowercase
- 6. Add a double-space.
- 7. Type the month and year of your oral defense, with the month in title case and no comma between the month and year.
- 8. At about the center of the page type 'Graduate Supervisory Committee:'
- 9. A blank (double-spaced) line
- 10. Type the members' names, without titles, one per line, single-spaced, as follows:
  - a) If you have one chair, type: the chair's name, comma, space 'Chair'
  - b) If two chairs, type: comma, space, 'Co-Chair' after the first two names
  - c) Follow with the other members' names.
- 11. At the bottom of the page, type 'ACCEPTED BY THE GRADUATE COLLEGE'
- 12. The space between the date and 'Graduate Supervisory Committe' lines should equal the space between the last member's name and the 'ACCEPTED...' line.

# Abstract page

Center the title 'ABSTRACT' at the top of the page. Number the page at the bottom, centered with the Roman numeral 'iii'. If there is a second page, number it similarly with 'iv'.

Dedication and Acknowledgements (optional)

- The dedication and acknowledgements together must not exceed three pages.
- The dedication page is not titled and the text should be centered both vertically and horizontally.
- The heading for the acknowledgements page is 'ACKNOWLEDGEMENTS', centered and at the top of the page.
- Continue the page numbering in lowercase Roman, at the bottom and centered.

# 21.1.3 Table of contents

- 1. Type 'TABLE OF CONTENTS' centered at the top of the page.
- 2. On the next line type the word 'Page' right justified
- 3. Begin listing any preliminary pages that follow the table of contents (e.g., lists) in ALL CAPS. The title is left justified, the page number is right justified and a dotted line fills the gap between.
- 4. Double space between entries.

- 5. Chapter headings and subheadings to three levels shall be listed, with a lower level being indented with respect to a higher level.
- 6. The wording of headings shall correspond exactly to those in the main body.
- 7. The page number is centered at the bottom of the page.
- 8. If the listing continues for more than one page, subsequent pages shall be headed with one line consisting of 'Chapter' left justified and 'Page' right justified.

# 21.1.4 Lists

For a given kind of list (often figures or tables) called, say, 'things':

- 1. Type 'LIST OF THINGS' centered at the top of the page.
- 2. On the next line type 'Thing' left justified and 'Page' right justified.
- 3. List, double-spaced, the caption or title of the thing left justified and the page number right justified, with a dotted line between them.
- 4. Use Roman lowercase to number the page(s) at the bottom, centered.
- 5. If the listing continues for more than one page, subsequent pages shall be headed with one line consisting of 'Thing' left justified and 'Page' right justified.

# 21.1.5 Main text

Nothing shall appear in the margins.

The top line on a page is the line immediately below the top margin. The top text line is the one following that (i.e. the second line below the margin).

# Page numbering

All pages are counted, but the first page of each chapter is not numbered (paginated); other pages are paginated. The first page of the main text is counted as number 1. Numbered pages have the number right justified on the top line.

### Headings

Chapter headings shall be centered. On the top line type 'Chapter' followed by the number. On the top text line type the heading in all uppercase. Type the text on the subsequent lines.

Subheadings, consisting of the number and title (not in all caps), shall be centered, with one blank line before and after.

#### Captions

Table captions, which are left justified, shall be put before the table itself. The first line consists of 'Table' followed by the number; the caption wording commences on the next line.

Captions for figures are similar, except that they shall be put below the figure and 'Table' replaced by 'Figure'.

Tables and figures shall be single-spaced.

# Notes

Notes may be placed at the bottom of the page (i.e., footnotes), or grouped in the backmatter (i.e., endnotes) before the reference list.

All notes shall be introduced by a superior number in the text, with the same number used for the text of the note. Notes should be single spaced, with double spacing between them.

#### 21.1.6 Backmatter

The backmatter consists of the following pages, in order (all of which are optional).

- 1. Notes (if you are using endnotes and grouping them at the end)
- 2. References (AKA 'Bibliography' or 'Works Cited')
- 3. Appendices
- 4. Biographical sketch (optional)

Pagination continues from the main text; but as with chapters, the first pages of any notes, references, or appendices are not numbered. A biographical sketch, if it is included, is the last page and is neither counted nor paginated.

Headings for the backmatter sections shall be in uppercase, centered, and on the top line.

#### Notes

The section for endnotes should begin on a new, unnumbered page. Subsequent pages should be numbered.

Use 'NOTES', centered and at the top, as the heading for the notes section.

#### References

Use the reference heading appropriate for your discipline, in uppercase, centered and at the top of the page. Individual refences should be single-spaced with the second and later lines of a multiline reference indented with respect to the first line. There should be double-spacing between references.

# Appendices

The heading for an appendix consists of the word 'APPENDIX' followed by the uppercase letter signifying its position in the sequence of appendices (e.g., A or B or C or ···). This shall be centered on the top line. The title of the appendix, in uppercase, is centered on the following line. This page is not numbered. Subsequent pages are numbered and the text commences on the top text line of the following page.

# Biography

The title for the optional biographical page is 'BIOGRAPHICAL SKETCH', in the usual position. The text shall not exceed the one page.

# 21.2 Code

Given the above set of requirements we can produce code that, hopefully, will generate documents that will not fall foul of the inspectorate. For simplicity I'll do the code in the form of a package called pwasu.sty. I will be using some LaTeX kernel commands that you won't normally come across. Some of the macros include @ as part of their name but this is safe as they are in a package, otherwise they would have to be within a \makeatletter \cdots \makeatletter grouping (see §E.4).

# 21.2.1 Initialisation

First, identify the package and its basic requirements.

```
%%%% file pwasu.sty
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{pwasu}[2009/04/18 v0.3 ASU thesis]
```

This is only going to work with memoir, so check if the class is being used, and if not then give an error message and stop reading the package code.

ASU is very strict about only using a single font in a single size. It is probable that at least one reference will be made to a location on the web. With LaTeX such references are set using the url package, which defaults to using a monospaced font for urls. For ASU we have to make sure that they will be made using the body font (and the same applies to any verbatim text, such as computer code listings, for which we can use the \setverbatimfont command).

```
\usepackage{url}
\urlstyle{same}
\setverbatimfont{\normalfont\rmfamily}% make verbatims use the body font
```

Ensuring that footnotes use the body font is a simple matter of redefining \foottextfont.

```
\renewcommand*{\foottextfont}{\normalfont\normalsize}
```

Noticing that the requirements can involve switching between double and single spacing, some of which will be done internally in the package, give the user a chance to change the default double spacing. The argument to \setasuspacing can be either \OnehalfSpacing or \DoubleSpacing. The result sets \AsuSpacing to be one of these commands.

```
% To enable spacing to be changed if neccesary by the user
\newcommand*{\setasuspacing}[1]{%
  \let\AsuSpacing#1
  \AsuSpacing}
\setasuspacing{\DoubleSpacing}
```

# 21.2.2 Page layout

As this is for an American University on letterpaper sized paper we'll assume letterpaper stock, so no cropping will be needed. Setting the side margins is easy:

```
% left, right margins and textwidth
\setlrmarginsandblock{1.5in}{1in}{*}
```

Setting the top and bottom margins requires more thought. LaTeX provides the header and footer areas for page numbers. However, the requirements state that page numbers must be either on the top or bottom line (of the textblock) with the text extending down to the bottom line or up to the top line (of the textblock). I'll organise it according to the layout for the main body. Here, the top of the header is 1 inch below the top of the paper and the bottom of the text is 1 inch above the bottom of the paper (with the footer below that).

```
%% for main body, bottom of text at 1in, footer below
 %% top of header at 1in, first text line double spaced
 %% below base of header
 \newlength{\linespace}
 \setlength{\linespace}{\baselineskip} % current equivalent of \onelineskip
 \setlength{\headheight}{\onelineskip}
 \setlength{\headsep}{\linespace}
 \addtolength{\headsep}{-\topskip}
 \setlength{\uppermargin}{1in}
 \addtolength{\uppermargin}{\headheight}
 \addtolength{\uppermargin}{\headsep}
And for the bottom margin:
 %% and for the bottom
 \setlength{\lowermargin}{1in}
 \setlength{\textheight}{\paperheight}
 \addtolength{\textheight}{-\uppermargin}
 \addtolength{\textheight}{-\lowermargin}
And finally for footnotes:
 %% footnote settings
 \setlength{\footskip}{\onelineskip}
 \setlength{\footnotesep}{\onelineskip}
```

The layout on the preliminary pages is different, so I'll need some handy lengths in order to change the layout as appropriate. The user can also make adjustments with these if necessary.

```
%% the fiddle lengths (..ta.. for title/approval page, others for prelims)
\newlength{\toptafiddle} \setlength{\toptafiddle}{2\linespace}
\newlength{\bottafiddle} \setlength{\bottafiddle}{0pt}
\newlength{\topfiddle} \setlength{\toptafiddle} \toptafiddle}
\newlength{\botfiddle} \setlength{\botfiddle}{\onelineskip}
```

That's it for the general layout, except for increasing the paragraph indentation as the line spacing is larger than normal.

```
\setlength{\parindent}{2em}
\checkandfixthelayout[nearest]
```

As the layout is set up, the bottom of the text is one inch above the bottom of the paper. This is fine for the main text and the Title and Approval pages but the text height must be decreased, temporarily, for the other pages in the prelims. Changes to the page layout may be accomplished by the following sneaky procedure. First change from one- to two-column (or vice-versa), which starts a new page, make the changes, then change from two- to one-column (or vice-versa) which starts the same new page again but with the layout changes implemented. The following code implements this for the \textheight in a one column document, which is what we are dealing with here.

```
\newcommand*{\addtotextheight}[1]{%
  \twocolumn
  \addtolength{\textheight}{#1}%
  \onecolumn}
```

# 21.2.3 Page styles

Next I'll tackle the page styles. The style for the main body is simple, with the page number top right, and the *empty* style is the one for chapter pages. The page number for the preliminary pages is centered at the bottom, and the *plain* page style provides that. For the main text define the *asu* page style.

```
%%%% pagestyles
%% the main text
\makepagestyle{asu}
\makeevenhead{asu}{\thepage}{}{}
\makeoddhead{asu}{}{\thepage}
```

Any 'continuation' pages for the ToC, etc., have a header that consists of a name at the left and the word 'Page' at the right. We need a header for each kind of listing.

```
%% for continuation pages of the ToC, LoF, LoT
\makepagestyle{toc}
  \makeevenfoot{toc}{}{\thepage}{}
  \makeoddfoot{toc}{}{\thepage}{}
  \makeevenhead{toc}{Chapter}{}{Page}
  \makeoddhead{toc}{Chapter}{}{Page}
\makepagestyle{lof}
  \makeevenfoot{lof}{}{\thepage}{}
  \makeoddfoot{lof}{}{\thepage}{}
  \makeevenhead{lof}{Figure}{}{Page}
  \makeoddhead{lof}{Figure}{}{Page}
\makepagestyle{lot}
  \makeevenfoot{lot}{}{\thepage}{}
  \makeoddfoot{lot}{}{\thepage}{}
  \makeevenhead{lot}{Table}{}{Page}
  \makeoddhead{lot}{Table}{}{Page}
```

# 21.2.4 The ToC and friends

While we're at it, do the code for the LoF and LoT, which is simpler than that needed for the ToC. We have to specify our new pagestyles which can be done by extending the \listof... macros. The \addtodef macro makes this rather easy.

```
%%%% The LoF
\renewcommand{\listfigurename}{LIST OF FIGURES}
\addtodef{\listoffigures}{\clearpage\pagestyle{lof}}{}
```

For the titles, these have to be moved up into the header area, so that they come just below the top margin, and set the initial pagestyle as *plain*. After the title we can insert the relevant column headers.

```
\renewcommand*{\lofheadstart}{\vspace*{-\topfiddle}}
\renewcommand*{\afterloftitle}{\thispagestyle{plain}%
\par\nobreak {\normalfont Figure \hfill Page}\par\nobreak}
```

And the same for the LoT.

```
%%% The LoT
\renewcommand{\listtablename}{LIST OF TABLES}
\addtodef{\listoftables}{\clearpage\pagestyle{lot}}{}
\renewcommand*{\lotheadstart}{\vspace*{-\topfiddle}}
\renewcommand*{\afterlottitle}{\thispagestyle{plain}%
\par\nobreak {\normalfont Table \hfill Page}\par\nobreak}
```

The ToC is similar but we also have to deal with the entries themselves.

```
%%%% Do the ToC
\renewcommand{\contentsname}{TABLE OF CONTENTS}
\addtodef{\tableofcontents}{\clearpage\pagestyle{toc}}{}
\renewcommand*{\tocheadstart}{\vspace*{-\topfiddle}}
\renewcommand*{\aftertoctitle}{\thispagestyle{plain}%
\par\nobreak \mbox{}\hfill{\normalfont Page}\par\nobreak}
```

And the changes to the entries, all of which are set in the normal font and with dotted leaders, with no extra space between any of the entries.

```
\renewcommand*{\cftchapterfont}{\normalfont}
\renewcommand*{\cftchapterpagefont}{\normalfont}
\renewcommand*{\cftchapterleader}{\%
\cftchapterfont\cftdotfill{\cftchapterdotsep}}
\renewcommand*{\cftchapterdotsep}{\cftdotsep}
```

Unlike the typical LaTeX ToC there must be no additional space before chapter entries; also, there should be no additional space inserted by the chapters in the LoF or LoT which just requires a simple redefinition of \insertchapterspace..

```
%%% no extra space before the entry, or in the LoF/LoT
\setlength{\cftbeforechapterskip}{0pt plus 0pt}
\renewcommand*{\insertchapterspace}{}
```

#### 21.2.5 Chapter styling

Moving on to styling the chapter titles, the first line must be moved up into the header area, and other spacings set to give blank lines. The fonts are just the regular body font. Call the chapterstyle *asu* and make sure that the *empty* pagestyle is used for it.

```
%% chapter style
\makechapterstyle{asu}{%
  \setlength{\beforechapskip}{-\topfiddle}
  \setlength{\midchapskip}{1.0\onelineskip}
  \setlength{\afterchapskip}{1.0\onelineskip}
  \renewcommand*{\chapnamefont}{\normalfont}
  \renewcommand*{\chapnumfont}{\chapnamefont}
  \renewcommand*{\printchapternum}{\centering\chapnumfont \thechapter}
  \renewcommand*{\chaptitlefont}{\normalfont\centering}
  \renewcommand*{\printchapternonum}{}}
}
\aliaspagestyle{\chapter}{\empty}
```

The chapterstyle for any appendices is slightly different as the title, all in uppercase, is on one page by itself and the text starts on the following page. Call this the *asuappendix* chapterstyle.

```
%% chapter style for appendices, text comes on following page
\makechapterstyle{asuappendix}{%
  \setlength{\beforechapskip}{-\topfiddle}
  \setlength{\midchapskip}{1.0\onelineskip}
  \setlength{\afterchapskip}{1.0\onelineskip}
  \renewcommand*{\chapnamefont}{\normalfont}
  \renewcommand*{\chapnumfont}{\chapnamefont}
  \renewcommand*{\printchaptername}{%
    \chapnamefont\MakeUppercase{\@chapapp}}
  \renewcommand*{\printchapternum}{\centering\chapnumfont \thechapter}
  \renewcommand*{\chaptitlefont}{\normalfont\centering}
  \renewcommand*{\chaptitlefont}{\normalfont\centering}
  \renewcommand*{\printchapternonum}{}
  \renewcommand*{\afterchaptertitle}{\clearpage}}
}
```

We have to extend the \appendix command to use the new chapter style, and also to ensure that double spacing will be used (certain elements that come before the appendices are single spaced).

```
%%% different chapter style for appendices, (and double spaced) addtodef{appendix}{}{\chapterstyle{asuappendix}\AsuSpacing}
```

# 21.2.6 Section, etc., styling

Set up the section headings so that they are centered, use the normal font, and have a blank line before and after.

```
%%% (subsub)section styles
\setsecheadstyle{\centering\normalfont}
\setbeforesecskip{-1\onelineskip plus -1ex minus -.2ex}
\setaftersecskip{1\onelineskip plus .2ex}
\setsubsecheadstyle{\centering\normalfont}
\setbeforesubsecskip{-1\onelineskip plus -1ex minus -.2ex}
\setaftersubsecskip{1\onelineskip plus .2ex}
```

### 21.2.7 Captions

The captions are set flushleft and raggedright with the name and number on one line and the title on the following line. Fortunately floats are automatically set single spaced, which is what the requirements specify.

```
%% Captions
\captiontitlefont{\normalfont}% title font
\precaption{\raggedright}% for Caption N
\captiondelim{\newline}% newline
\captionstyle{\raggedright}% for title
\setlength{\belowcaptionskip}{\onelineskip}
```

# 21.2.8 The bibliography

The requirements imply that the title is likely to be 'REFERENCES'. The bibliography is set single spaced but with a blank line between the entries.

```
%% for REFERENCE section
\renewcommand*{\bibname}{REFERENCES}
\setlength{\bibitemsep}{\onelineskip}
```

The second and later lines of any entry are to be indented. We use the \biblistextra hook for setting this up.

```
\renewcommand*{\biblistextra}{%
  \setlength{\itemsep}{\bibitemsep}
  \setlength{\labelwidth}{0pt}
  \setlength{\leftmargin}{3em}% hanging indent
  \setlength{\itemindent}{-\leftmargin}}
```

The title for the bibliography is set via the \bibsection macro. The heading is unnumbered but is added to the ToC. To get the spacing right the heading, set as a \chapter\*, which must be called double spaced, and then single spacing is called for after that.

```
\renewcommand*{\bibsection}{%
  \AsuSpacing
  \chapter*{\bibname}\addcontentsline{toc}{chapter}{\bibname}
  \SingleSpacing}
```

# 21.2.9 End notes

The heading for the Notes section is similar to the bibliography heading.

```
%% endnotes
\renewcommand*{\notesname}{\NOTES}
\renewcommand*{\notedivision}{%
  \AsuSpacing
  \chapter*{\notesname}
  \addcontentsline{\toc}{\chapter}{\notesname}
  \SingleSpacing}
```

The rest of the code for endnotes ensures that they are numbered continuously throughout the text, the number is set as a superscript, that there is a blank line between each entry, and that there are no subdivisions within the listing.

```
\continuousnotenums
\renewcommand*{\notenuminnotes}[1]{\textsuperscript{#1}\space}
\renewcommand{\noteinnotes}[1]{#1\\}
\renewcommand*{\pagenotesubhead}[3]{}\% no subheads
```

# 21.2.10 Preliminary headings

There can be any number of sections in the prelims. The titles for these are located in the LaTeX header area. Here's a general macro for setting these.

```
%%% general macro for Abstract, etc., headings
\newcommand*{\pretoctitle}[1]{{\clearpage\centering
\vspace*{-\topfiddle}#1\par}}
%%% Start the ACKNOWLEDGEMENTS
\newcommand{\asuacknowledgements}{\pretoctitle{ACKNOWLEDGEMENTS}}}
```

The Abstract is the first section after the title and approval pages. At this point we must reduce the textheight in order to raise the footer area.

```
%%% Start the ABSTRACT
\newcommand{\asuabstract}{%
\addtotextheight{-\botfiddle}%
\pretoctitle{ABSTRACT}}
```

While we are at this, the textheight must be reset to its default value just before the first chapter in the main matter. A simple addition to \mainmatter handles this.

```
\addtodef{\mainmatter}{\addtotextheight{\botfiddle}{}
```

The dedication, if any, does not have a heading and the text is centered horizontally and vertically.

There may be sections in the prelims that come after the ToC, and the titles of these are added to the ToC.

```
%% for any headings after the tocloft and before the main body
\newcommand{\prelimtitle}[1]{%
\pretoctitle{#1}\addcontentsline{toc}{chapter}{#1}}
```

# 21.2.11 Components of the title and approval pages

There are several items that are set on the title and approval pages. In order to separate the information from the particular layout, I've defined a macro for defining each item.

```
%%%% for the title page and approval page.
% your title
\newcommand{\settitle}[1]{\def\asutitle{#1}}
% you
\newcommand{\setauthor}[1]{\def\asuauthor{#1}}
\% document type (e.g., thesis)
\newcommand{\setdoctype}[1]{\def\asudoctype{#1}}
% possible degree
\newcommand{\masters}{\def\asudegree{Master of Arts}}
\newcommand{\doctors}{\def\asudegree{Doctor of Philosophy}}
% defence date
\newcommand{\setdefdate}[1]{\def\asudefdate{#1}}
% graduation date
\newcommand{\setgraddate}[1]{\def\asugraddate{#1}}
% committe chair
\newcommand{\setchair}[1]{\def\asuchair{#1, Chair}}
% committe co-chairs
\newcommand{\setchairs}[2]{%
  \def\asuchair{#1, Co-chair \\ #2, Co-chair}}
% other members (separated by \\s)
```

Just for fun, create some default settings for these. The successful user will have changed them all!

# 21.2.12 The title and approval pages

An example of a title page is shown in Figure 21.1 and an example of the corresponding approval page is in Figure 21.2.

Now we can set up the layouts for the title and approval pages. The information typeset on these pages is obtained from the previous set of commands. Note that the last line on each of these pages has to be set upon the bottom margin. The ASU inspectorate is likely to be very keen on this, perhaps using a ruler to measure the actual distance from the bottom of the page to ensure that it is the magic 1 inch. I have included an \enlargethispage by the

# CONTINUING CONUNDRUMS IN THE TYPOGRAPHIC REQUIREMENTS FOR THESES

by

Ima A. Student

A Polemic Presented in Partial Fulfillment of the Requirements for the Degree Master of Arts

# ARCHIBALD SMYTHE UNIVERSITY

May 2021

Figure 21.1: Example Archibald Smythe University title page

# CONTINUING CONUNDRUMS IN THE TYPOGRAPHIC REQUIREMENTS FOR THESES

by

Ima A. Student

has been approved

April 2018

# Graduate Supervisory Committee:

S. Holmes, Co-Chair J. Moriarty, Co-Chair G. E. Challenger A. Quartermain J. H. Watson

ACCEPTED BY THE GRADUATE COLLEGE

Figure 21.2: Example Archibald Smythe University approval page

amount \bottafiddle so the user can make any fine adjustment that might be requested. Similarly, the length \toptafiddle may be altered to fine tune the position of the title. Hopefully, neither of these tweaks will be needed, but if so, then use \addtolength instead of \setlength to make an adjustement.

```
%%% typesets the TITLE page
\newcommand{\thetitlepage}{{%
  \clearpage
  \thispagestyle{empty}
  \centering
  \vspace*{-\toptafiddle}
  \asutitle \\ by \\ \asuauthor
  \vfill
  {\SingleSpace
  A \asudoctype\ Presented in Partial Fulfillment \\
  of the Requirements for the Degree \\
  \asudegree\par}
  \vfill
  ARCHIBALD SMYTHE UNIVERSITY \\
  \asugraddate
  \par
  \enlargethispage{\bottafiddle}
  \clearpage}}
 And similary for the approval page.
%%% typesets the APPROVAL page
\newcommand{\approvalpage}{{%
  \thispagestyle{empty}
  \centering
  \vspace*{-\toptafiddle}
  \asutitle \\ by \\ \asuauthor \\[3\onelineskip]
  has been approved \\
  \asudefdate
  \vfill
  Graduate Supervisory Committee: \\[-0.5\onelineskip]
  {\SingleSpacing
  \asuchair \\
  \asumembers}
  \vfill
  ACCEPTED BY THE GRADUATE COLLEGE
  \par
  \enlargethispage{\bottafiddle}
```

```
\clearpage}}
```

#### 21.2.13 The last bits

The biographical sketch has a title (which is not added to the ToC), the text is single spaced and there is no page number. It is easiest to provide this as an environment.

```
%%% put your biographical text in this environment
%% \begin{biosketch} I'm a person who has accomplished .... \end{biosketch}
\newenvironment{biosketch}{%
  \pretoctitle{BIOGRAPHICAL SKETCH}\thispagestyle{empty}\SingleSpacing}%
{}
```

Make sure that the requisite initial page style and appropriate chapter style is used.

```
%% use the asu chapterstyle and plain pagestyle
\chapterstyle{asu}
\pagestyle{plain}

%%%%%%%%%%%%%%%%% end of *.sty file
\endinput
%%%%%%%%%%%%%%%%%
```

#### 21.3 Usage

This is a sketch of how an ASU thesis could be written.

With the wide textblock, 12pt is too small for reading ease, so best not to use 11pt or 10pt.

Times Roman comes with LaTeX, but you are effectively writing a book, not a newspaper column. If you have Garamond or Lucida Bright then seriously consider using one or other of them. Lucida Bright is probably more appropriate if your thesis includes mathematics while Garamond is perhaps more in keeping if your thesis topic falls into the humanities area. If the requirements did not limit your choices then there are other fonts that might better express your work. In any case I suggest that you do not use a sans font (e.g., Arial, Tahoma or Verdana from the ASU list).

```
\documentclass[oneside,12pt]{memoir}
\usepackage{mathptmx} % Times New Roman
\usepackage{pwasu} % the package
```

The general sequence in your document, after you have set the data for the TITLE and APPROVAL pages and any other specifics and packages in the preamble, is:

```
% if you can get away without the default \DoubleSpacing, then
%\setasuspacing{\OnehalfSpacing}
%% if you use endnotes, then
```

Garamond is a commercial font and, for example, is available along with many other fonts from FontSite (http://www.fontsite.com) with LaTeX support from http://contrapunctus.net/fs500tex. Lucida Bright, another commercial font, is available from TUG at http://tug.org/store/lucida and is supported by several LaTeX packages.

```
\makepagenote
\begin{document}
\maxtocdepth{subsection} % put 3 levels into the ToC
\frontmatter
\thetitlepage
\approvalpage
\asuabstract
   abstract text
%% if you have any acknowledgements, then
  \asuacknowledgements
    acknowledgements text
\mbox{\ensuremath{\mbox{\%}}} \asudedication{ text } \mbox{\ensuremath{\mbox{\%}}} if you want a dedication
\tableofcontents
% \listoffigures % if you have any figures
% \listoftables % if you have any tables
\%\% if you have more prelim sections, then
%%% \clearpage
%%% \pagestyle{plain}
\ensuremath{\mbox{\%\%}}\xspace \prelimtitle{title} text % for each section before main text
\mainmatter
\pagestyle{asu}
\chapter{...} % start of your main text
... report on lots of incredible work, now you are on your
own until...
%% if endnotes then
   \printpagenotes
%% if a bibliography then
   \verb|\begin{the bibliography}|...\\end{the bibliography}|
%% if appendices, then
  \appendix
  \chapter{...}
%% if Biographical sketch then
   \begin{biosketch} ... \end{biosketch}
\end{document}
```

If you actually try any of the above code and it does not work, then I may have made a typo or two, or maybe you have, or perhaps we both have. In any event, the code is more of a sketch of what might be needed than a prescription of how it must be done.

# 21.4 Comments

Having read through the requirements you will have realised that whatever committees set them had not advanced beyond the 19th century technology of the typewriter. When I wrote my thesis some forty years ago it was, of necessity, single sided so that carbon copies could be made by the typist (who would have objected strongly to having to type the hundred and fifty or so pages six times). I must admit, though, that the sixth copy was almost too faint and blurry to be read comfortably even though the typist had used thin paper and kept replacing the carbon paper. In this day of double sided printers and double sided copiers I see no reason except inertia to keep a single sided requirement. Many students, and faculty members, have beaten their heads against the diehards and very rarely have they managed to prevail.

In contrast to the ASU style I have a copy of a doctoral thesis [Sne04] for Vrije Universiteit, Amsterdam. This is a professionally printed 100 page, double sided, glossy paperbound book with an attractive coloured photograph of a sunset on the front and rear covers. The page size is 40.5pc by 57pc with spine and top margins of 5pc and foredge and bottom margins of 7pc.<sup>3</sup> The textblock, then, is 28.5 by 45pc set with 45 lines of a 10pt Lucida Bright seriffed font. Chapter and section heads are in a sans font, with the chapter heads larger than the section heads. Caption names are a bold sans with the caption text in an italic. Headers on the verso pages are the chapter title with the section title as the recto header. The page numbers are in the footers by the foredge margin. Altogether, a much more appealing production than Archibald Smythe University will permit.

<sup>2</sup> Remington sold their first commercial typewriter in 1873 which even then had the QWERTY keyboard layout. By 1910 typewriter designs were pretty well standardised.

<sup>3</sup> Professional printers use points and picas for their measurements.



## А

## Packages and macros

The memoir class does not provide for everything that you have seen in the manual. I have used some packages that you are very likely to have in your LaTeX distribution, and have supplemented these with some additional macros, some of which I will show you.

#### A.1 Packages

The packages that I have used that you are likely to have, and if you do not have them please consider getting them, are:

- etex lets you take advantage of eTeX's extended support for counters and such. Note that from 2015 and onwards, the allocation of extra registers have now been build into the LaTeX kernel. Thus in most cases the etex package is no longer necessary. There are how ever extra very special features left in etex that *some* users may need. In that case please remember to load etex by placing \RequirePackage{etex} before \documentclass!
- url [Ars99] is for typesetting URL's without worrying about special characters or line breaking.
- fixltx2e [MC00] eliminates some infelicities of the original LaTeX kernel. In particular it maintains the order of floats on a twocolumn page and ensures the correct marking on a twocolumn page.
  - Note that as of 2015, the functionality of this package has been merged into the LATEX kernel. Loading this package does nothing.
- alltt [Bra97] is a basic package which provides a verbatim-like environment but \, {, and } have their usual meanings (i.e., LaTeX commands are not disabled).
- graphicx [CR99] is a required package for performing various kinds of graphical functions.
- color [Car05] is a required package for using color, or xcolor [Ker07] is an enhanced version of color.
- latexsym gives access to some extra symbols.
- amsmath for when you are doing anything except the simplest kind of maths typeseting.
- fontenc for using fonts with anything other than the original OT1 encoding (i.e., for practically any font).
- pifont for typesetting Pifonts (i.e., Symbol and Zapf Dingbats)

Apart from the packages that are supplied as part of the memoir distribution, the packages that I used and you most likely do not have are:

• layouts [Wil03a]. I used it for all the layout diagrams. For example, Figure 6.2 and Figure 6.3 were drawn simply by:

```
\begin{figure}
\centering
\setlayoutscale{1}
\drawparameterstrue
\drawheading{}
\caption{Displayed sectional headings} \label{fig:displaysechead}
\end{figure}
\begin{figure}
\centering
\setlayoutscale{1}
\drawparameterstrue
\runinheadtrue
\drawheading{}
\caption{Run-in sectional headings} \label{fig:runsechead}
\end{figure}
```

The package also lets you try experimenting with different layout parameters and draw diagrams showing what the results would be in a document.

The version of layouts used for this manual is v2.4 dated 2001/04/30. Earlier versions will fail when attempting to draw some figures (e.g., to draw Figure 2.3).

• fonttable [Wil09a]. I used this for the font tables (e.g., Table 3.2). You must have at least version 1.3 dated April 2009 for processing the manual (earlier versions are likely to produce errors in the number formatting area with minor, but odd looking, effect on the printed result).

#### A.2 Macros

Originally the preamble of the manual contained many macro definitions, probably more than most documents would because:

- I am having to typeset many LaTeX commands, which require some sort of special processing;
- I have tried to minimize the number of external packages needed to LaTeX this manual satisfactorily, and so have copied various macros from elsewhere;
- I wanted to do some automatic indexing;
- I wanted to set off the syntax specifications and the code examples from the main text.

I have since put the majority of these into a package file called memsty.sty. To get the whole glory you will have to read the preamble, and the memsty package file but I show a few of the macros below as they may be of more general interest.

```
\Ppstyle{\(\rangle pagestyle \rangle \) \pstyle{\(\rangle pagestyle \rangle \)}
```

The command \Ppstyle prints its argument in the font used to indicate pagestyles and the command \pstyle prints its pagestyle argument and also makes a pagestyle entry in the index. Its definition is

```
\newcommand*{\pstyle}[1]{\Ppstyle{#1}%
  \index{#1 pages?\Ppstyle{#1} (pagestyle)}%
  \index{pagestyle!#1?\Ppstyle{#1}}}
```

The first part prints the argument in the text and the second adds two entries to the idx file. The fragment #1 pages is what the MakeIndex program will use for sorting entries, and the fragment following the ? character is what will be put into the index.

```
\Pcstyle{\langle chapterstyle \rangle} \cstyle{\langle chapterstyle \rangle}
```

The command \Pcstyle prints its argument in the font used to indicate chapterstyles and \cstyle prints its chapterstyle argument and also makes a chapterstyle entry in the index. Its definition is

```
\newcommand*{\cstyle}[1]{\Pcstyle{#1}%
  \index{#1 chaps?\Pcstyle{#1} (chapterstyle)}%
  \index{chapterstyle!#1?\Pcstyle{#1}}}
```

which is almost identical to \pstyle.

There is both a *companion* chapterstyle and a *companion* pagestyle. The strings used for sorting the index entries for these are companion chaps and companion pages respectively, so the chapterstyle will come before the pagestyle in the index. The reason for distinguishing between the string used for sorting and the actual entry is partly to distinguish between different kinds of entries for a single name and partly to avoid any formatting commands messing up the sorted order.

```
\begin{syntax} syntax \end{syntax}
```

The syntax environment is for specifying command and environment syntax. Its definition is

It is implemented in terms of the tabular environment, centered within the typeblock, which forms a box that will not be broken across a pagebreak. The box frame is just the normal horizontal and vertical lines that you can use with a tabular. The width is fixed at 90% of the text width. As it is a tabular environment, each line of syntax must be ended with \\. Note that normal LaTeX processing occurs within the syntax environment, so you can effectively put what you like inside it. The center environment is defined in terms of a trivlist and \centering. I wanted to be able to control the space before and after the '\centering' so I defined the tightcenter environment which enabled me to do this.

```
\begin{lcode} LaTeX code \end{lcode}
```

I use the lcode environment for showing examples of LaTeX code. It is a special kind of verbatim environment where the font size is \small but the normal \baselineskip is used, and each line is indented.

At the bottom the environment is defined in terms of a list, although that is not obvious from the code; for details see the class code [Wil09b]. I wanted the environment to be a tight list and started off by defining two helper items.

The macro \@zeroseps sets the before, after and middle skips in a list to 0pt (\z@ is shorthand for 0pt). The length \gparindent will be the line indentation in the environment.

```
% Now we can do the new lcode verbatim environment.
% This has no extra before/after spacing.
\newenvironment{lcode}{\@zeroseps
\renewcommand{\verbatim@startline}%
    {\verbatim@line{\hskip\gparindent}}
\small\setlength{\baselineskip}{\onelineskip}\verbatim}%
    {\endverbatim
    \vspace{-\baselineskip}\noindent}
```

The fragment {\hskip\gparindent} puts \gparindent space at the start of each line. The fragment \small\setlength{\baselineskip}{\onelineskip} sets the font size to be \small, which has a smaller \baselineskip than the normal font, but this is corrected for by changing the local \baselineskip to the normal skip, \onelineskip. At the end of

the environment there is a negative space of one line to compensate for a one line space that LaTeX inserts.

## **Showcases**

The memoir memoir class has several features that involve a *style* and it provide several of these styles. This chapter is used to showcase these styles.

### B.1 Chapter styles

For more about defining chapter styles, see section 6.5, page 84.

## Chapter 1

## Demonstration of the default chapter style

The above is a demonstration of the default chapters tyle. It is one of several styles that come as part of the  ${\sf memoir}$  class.

Figure B.1: The default chapterstyle

## 2 Demonstration of the section chapter style

The above is a demonstration of the section chapters tyle. It is one of several styles that come as part of the memoir class.

Figure B.2: The section chapterstyle

# 3 Demonstration of the hangnum chapter style

The above is a demonstration of the hangnum chapterstyle. It is one of several styles that come as part of the memoir class.

Figure B.3: The hangnum chapterstyle

## Demonstration of the companion chapter style

The above is a demonstration of the companion chapterstyle. It is one of several styles that come as part of the memoir class.

Figure B.4: The companion chapterstyle

## 5 Demonstration of the article chapter style

The above is a demonstration of the article chapters tyle. It is one of several styles that come as part of the memoir class.

Figure B.5: The article chapterstyle

# Demonstration of the bianchi chapter style

The above is a demonstration of the bianchi chapters tyle. It is one of several styles that come as part of the memoir class.

Figure B.6: The bianchi chapterstyle

demonstration of the bringhurst chapter style

The above is a demonstration of the bringhurst chapters tyle. It is one of several styles that come as part of the memoir class.

Figure B.7: The bringhurst chapterstyle

# Demonstration of the brotherton chapter style

The above is a demonstration of the brotherton chapters tyle. It is one of several styles that come as part of the memoir class.

Figure B.8: The brotherton chapterstyle

## Chapter 9

Demonstration of the chappell chapter style

The above is a demonstration of the chappell chapterstyle. It is one of several styles that come as part of the memoir class.

Figure B.9: The chappell chapterstyle

## 10 Demonstration of the crosshead chapter style

The above is a demonstration of the crosshead chapterstyle. It is one of several styles that come as part of the memoir class.

Figure B.10: The crosshead chapterstyle

## I Demonstration of the culver chapter style

The above is a demonstration of the culver chapterstyle. It is one of several styles that come as part of the memoir class.

Figure B.11: The culver chapterstyle

— 2 —

Demonstration of the dash chapter style

The above is a demonstration of the dash chapterstyle. It is one of several styles that come as part of the memoir class.

Figure B.12: The dash chapterstyle

3

# Demonstration of the demo2 chapter style

The above is a demonstration of the demo2 chapterstyle. It is one of several styles that come as part of the memoir class.

Figure B.13: The demo2 chapterstyle

### Chapter 4

## Demonstration of the dowding chapter style

The above is a demonstration of the dowding chapters tyle. It is one of several styles that come as part of the memoir class.

Figure B.14: The dowding chapterstyle

F

## Demonstration of the ell chapter style

The above is a demonstration of the ell chapterstyle. It is one of several styles that come as part of the memoir class.

Figure B.15: The ell chapters tyle

## Chapter 6

Demonstration of the ger chapter style

The above is a demonstration of the ger chapterstyle. It is one of several styles that come as part of the memoir class.

Figure B.16: The ger chapterstyle

## 7 Demonstration of the komalike chapter style

The above is a demonstration of the komalike chapters tyle. It is one of several styles that come as part of the memoir class.

Figure B.17: The komalike chapterstyle

Demonstration of the lyhne chapter style

The above is a demonstration of the lyhne chapterstyle. It is one of several styles that come as part of the memoir class.

Figure B.18: The lyhne chapterstyle. This style requires the graphicx package

# Demonstration of the madsen chapter style

The above is a demonstration of the madsen chapters tyle. It is one of several styles that come as part of the memoir class.

Figure B.19: The madsen chapterstyle. This style requires the graphicx package

Demonstration of the ntglike chapter style

The above is a demonstration of the ntglike chapters tyle. It is one of several styles that come as part of the memoir class.

Figure B.20: The ntglike chapterstyle

## 1 Demonstration of the southall chapter style

The above is a demonstration of the southall chapters tyle. It is one of several styles that come as part of the <code>memoir</code> class.

Figure B.21: The southall chapters tyle

## 2 Demonstration of the tandh chapter style

The above is a demonstration of the tandh chapterstyle. It is one of several styles that come as part of the memoir class.

Figure B.22: The tandh chapterstyle

### chapter 3

## DEMONSTRATION OF THE THATCHER CHAPTER STYLE

The above is a demonstration of the that cher chapterstyle. It is one of several styles that come as part of the <code>memoir</code> class.

Figure B.23: The thatcher chapterstyle

4

CHAPTER

# Demonstration of the veelo chapter style

The above is a demonstration of the veelo chapterstyle. It is one of several styles that come as part of the memoir class.

Figure B.24: The veelo chapterstyle. This style requires the graphicx package

# 5. Demonstration of the verville chapter style

The above is a demonstration of the verville chapterstyle. It is one of several styles that come as part of the memoir class.

Figure B.25: The verville chapterstyle

# 6 Demonstration of the wilsondob chapter style

The above is a demonstration of the wilsondob chapterstyle. It is one of several styles that come as part of the memoir class.

Figure B.26: The wilsondob chapterstyle

The code for some of these styles is given in below. For details of how the other chapter styles are defined, look at the documented class code. This should give you ideas if you want to define your own style.

Note that it is not necessary to define a new chapterstyle if you want to change the chapter headings — you can just change the individual macros without putting them into a style.

#### B.1.1 Chappell

A style that includes rules is one that I based on the chapter heads in [CB99] and which I have called *chappell* after the first author. The style, which is shown in Figure B.9, can easily form the basis for general heads in non-technical books.

```
\makechapterstyle{chappell}{%
  \setlength{\beforechapskip}{0pt}
  \renewcommand*{\chapnamefont}{\large\centering}
  \renewcommand*{\chapnumfont}{\large}
  \renewcommand*{\printchapternonum}{%
  \vphantom{\printchaptername}%
  \vphantom{\chapnumfont 1}%
  \afterchapternum
  \vskip -\onelineskip}
  \renewcommand*{\chaptitlefont}{\Large\itshape}
  \renewcommand*{\printchaptertitle}[1]{%
  \hrule\vskip\onelineskip \centering\chaptitlefont ##1}}
```

The style centers the chapter number, draws a rule across the page under it, and below that comes the title, again centered. All the fiddling in the \printchapternonum macro is to try and ensure that the rule above the title is at the same height whether or not the chapter is numbered (the ToC being an example of an unnumbered heading).

#### B.1.2 Demo, Demo2 and demo3

I created a *demo* chapterstyle quite a time ago and used it on occasions in earlier editions of this Manual. Here is the original code.

```
\makechapterstyle{demo}{%
  \renewcommand*{\printchaptername}{\centering}
  \renewcommand*{\printchapternum}{\chapnumfont \numtoName{\c@chapter}}
  \renewcommand*{\chaptitlefont}{\normalfont\Huge\sffamily}
  \renewcommand*{\printchaptertitle}[1]{%
    \hrule\vskip\onelineskip \raggedleft \chaptitlefont ##1}
  \renewcommand*{\afterchaptertitle}%
    {\vskip\onelineskip \hrule\vskip \afterchapskip}}
}% end demo
```

This has one serious failing and what I now believe is a poor design decision. The failing is that if you have any appendices that use the *demo* chapterstyle then they are numbered instead of being lettered. The poor design is that the position of the title with respect to the top of the page is not the same for numbered and unnumbered chapters. The *demo2* chapterstyle below fixes both of these at the expense of simplicity (at least for me).

```
\makechapterstyle{demo2}{%
  \renewcommand*{\printchaptername}{\centering}
  \renewcommand*{\printchapternum}{\chapnumfont
     \ifanappendix \thechapter \else \numtoName{\c@chapter}\fi}
  \renewcommand*{\chaptitlefont}{\normalfont\Huge\sffamily}
  \renewcommand*{\printchaptertitle}[1]{%
    \hrule\vskip\onelineskip \raggedleft \chaptitlefont ##1}
  \renewcommand*{\afterchaptertitle}{%
    \vskip\onelineskip \hrule\vskip \afterchapskip}
  \setlength{\beforechapskip}{3\baselineskip}
  \renewcommand*{\printchapternonum}{%
    \vphantom{\chapnumfont One}
    \afterchapternum%
    \vskip\topskip}
  \setlength{\beforechapskip}{2\onelineskip}
}% end{demo2}
```

You may find it instructive to compare the code for the demo and demo2 chapterstyles.

Thec *demo* chapterstyle is still available in the class for backward compatibility reasons, but I strongly advise against anyone using it.

By chance I inadvertantly typest a chapterstyle that was a mixture of the *pedersen* and *demo*2 styles. As a result there is now a *demo*3 chapterstyle as well. The only difference between the two styles is in the definition of \chapnumfont which in *demo*3 is:

\renewcommand\*{\chapnumfont}{\normalfont\HUGE\itshape}

#### B.1.3 Pedersen

I have modified Troels Pedersen's original code to make it a little more efficient and flexible.

```
\newcommand*{\colorchapnum}{}
```

```
\newcommand*{\colorchaptitle}{}
\makechapterstyle{pedersen}{%
 \setlength{\beforechapskip}{-20pt}
 \setlength{\afterchapskip}{10pt}
 \renewcommand*{\chapnamefont}{\normalfont\LARGE\itshape}
 \renewcommand*{\chapnumfont}{\normalfont\HUGE\itshape\colorchapnum}
 \renewcommand*{\chaptitlefont}{\normalfont\huge\itshape\colorchaptitle}
 \renewcommand*{\afterchapternum}{}
 \renewcommand*{\printchaptername}{}
 \setlength{\midchapskip}{20mm}
 \renewcommand*{\chapternamenum}{}
 \renewcommand*{\printchapternum}{%
    \sidebar{\raisebox{0pt}[0pt][0pt]{\makebox[0pt][1]{%
     \resizebox{!}{\midchapskip}{\chapnumfont\thechapter}}}}
 \renewcommand*{\printchaptertitle}[1]{\chaptitlefont ##1}
}
```

The chapter number is scaled up from its normal size and set in a sidebar.

```
\colorchapnum \colorchaptitle
```

The title is set with colorchaptitle and the number with colorchapnum, both of which default to doing nothing. Lars Madsen has suggested an attractive red color for these:

```
\usepackage{color}
\definecolor{ared}{rgb}{.647,.129,.149}
\renewcommand{\colorchapnum}{\color{ared}}
\renewcommand{\colorchaptitle}{\color{ared}}
\chapterstyle{pedersen}
```

The uncolored version is used for the chaptersyle for this chapter; because of setting the number in a sidebar it does not display well anywhere other than as a real chapter head.

#### B.1.4 Southall

On 2006/01/08 Thomas Dye posted his *southall* chapterstyle on comp.text.tex and kindly gave me permission to include it here. It is based on the headings in a Cambridge Press book<sup>1</sup> by Aidan Southall. It produces a simple numbered heading with the title set as a block paragraph, and with a horizontal rule underneath. His original code called for lining figures for the number but I have commented out that bit. I also changed the code to eliminate the need for the two new lengths that Thomas used.

```
\makechapterstyle{southall}{%
  \setlength{\afterchapskip}{5\baselineskip}
  \setlength{\beforechapskip}{36pt}
  \setlength{\midchapskip}{\textwidth}
  \addtolength{\midchapskip}{-\beforechapskip}
  \renewcommand*{\chapterheadstart}{\vspace*{2\baselineskip}}
  \renewcommand*{\chaptitlefont}{\huge\rmfamily\raggedright}
  \renewcommand*{\chaptuffont}{\chaptitlefont}}
```

<sup>1</sup> Which I haven't seen

```
\renewcommand*{\printchaptername}{}
\renewcommand*{\chapternamenum}{}
\renewcommand*{\printchapternum}{%}
\renewcommand*{\printchapternum}{%}
\begin{minipage}[t][\baselineskip][b]{\beforechapskip}
\{\vspace{0pt}\chapnumfont%%%\figureversion{lining}
\thechapter}
\end{minipage}}
\renewcommand*{\printchaptertitle}[1]{%}
\hfill\begin{minipage}[t]{\midchapskip}
\{\vspace{0pt}\chaptitlefont ##1\par}\end{minipage}}\\renewcommand*{\afterchaptertitle}{%}
\par\vspace{\baselineskip}%
\hrulefill \par\nobreak\noindent \vskip\afterchapskip}}
```

The resulting style is shown in Figure B.21.

#### B.1.5 Veelo

Bastiaan Veelo posted the code for a new chapter style to CTT on 2003/07/22 under the title [memoir] [contrib] New chapter style. His code, which I have slightly modified and changed the name to veelo, is below. I have also exercised editorial privilege on his comments.

I thought I'd share a new chapter style to be used with the memoir class. The style is tailored for documents that are to be trimmed to a smaller width. When the bound document is bent, black tabs will appear on the fore side at the places where new chapters start as a navigational aid. We are scaling the chapter number, which most DVI viewers will not display accurately.

Bastiaan.

In the style as I modified it, \beforechapskip is used as the height of the number and \midchapskip is used as the length of the black bar.

```
\newlength{\numberheight}
\newlength{\barlength}
\makechapterstyle{veelo}{%
 \setlength{\afterchapskip}{40pt}
 \renewcommand*{\chapterheadstart}{\vspace*{40pt}}
 \renewcommand*{\afterchapternum}{\par\nobreak\vskip 25pt}
 \renewcommand*{\chapnamefont}{\normalfont\LARGE\flushright}
 \renewcommand*{\chapnumfont}{\normalfont\HUGE}
 \renewcommand*{\chaptitlefont}{\normalfont\HUGE\bfseries\flushright}
 \renewcommand*{\printchaptername}{%
    \chapnamefont\MakeUppercase{\@chapapp}}
 \renewcommand*{\chapternamenum}{}
 \setlength{\beforechapskip}{18mm}
 \setlength{\midchapskip}{\paperwidth}
 \addtolength{\midchapskip}{-\textwidth}
 \addtolength{\midchapskip}{-\spinemargin}
 \renewcommand*{\printchapternum}{%
```

```
\makebox[0pt][1]{\hspace{.8em}%
  \resizebox{!}{\numberheight}{\chapnumfont \thechapter}%
  \hspace{.8em}%
  \rule{\midchapskip}{\beforechapskip}%
  }}%
\makeoddfoot{plain}{}{\thepage}}
```

If you use this style you will also need to use the graphicx package [CR99] because of the \resizebox macro. The *veelo* style works best for chapters that start on recto pages.

## **Sniplets**

This chapter is (over time) meant to hold various pieces of code for memoir that we have gathered over the years or others have contributed, and which we think might be useful for others. In some cases they will have been moved from the text to this place, in order to make the manual less cluttered.

If you have some memoir related code you would like to share, feel free to send it to daleif@math.au.dk.

#### Sniplet overview

```
397
398
398
399
399
Sniplet C.6 (Autoadjusted number widths in the ToC) . . . . . . . . . . . . . . . . . .
              399
401
403
```

Sniplet C.1 (Mirroring the output)

The memoir class is not quite compatible with the crop package. This is usually not a problem as we provide our own crop marks. But crop provide one feature that we do not: mirroring of the output. The following sniplet was posted on CTT by Heiko Oberdiek (2009/12/05, thread *Memoir and mirrored pdf output*)

```
\usepackage{atbegshi}
\usepackage{graphicx}
\AtBeginShipout{%
  \sbox\AtBeginShipoutBox{%
  \kern-1in\relax
  \reflectbox{%
  \rlap{\kern1in\copy\AtBeginShipoutBox}%
  \kern\stockwidth
  }%
  }%
}
```

### Sniplet C.2 (Remove pagenumber if only one page)

Memoir counts all the pages used. You can use this information in various ways. For example, say you are preparing a setup to write small assignments in, these may or may not be just one page. How do we remove the footer automatically if there is only one page? Easy, place the following in the preamble (compile at least twice):

 $\label{last sheet} $$ \Delta End Document \in {\last sheet} = 1 \times {empty} \in {empt$ 

#### Sniplet C.3 (A kind of draft note)

Bastiaan Veelo has kindly provided example code for another form of a side note, as follows.

```
%% A new command that allows you to note down ideas or annotations in
"" the margin of the draft. If you are printing on a stock that is wider
%% than the final page width, we will go to some length to utilise the
"" paper that would otherwise be trimmed away, assuming you will not be
"" trimming the draft. These notes will not be printed when we are not
%% in draft mode.
\makeatletter
   \ifdraftdoc
     \newlength{\draftnotewidth}
     \newlength{\draftnotesignwidth}
     \newcommand{\draftnote}[1]{\@bsphack%
       {\%} do not interfere with settings for other marginal notes
         \strictpagecheck%
         \checkoddpage%
         \setlength{\draftnotewidth}{\foremargin}%
         \addtolength{\draftnotewidth}{\trimedge}%
         \addtolength{\draftnotewidth}{-3\marginparsep}%
         \ifoddpage
           \setlength{\marginparwidth}{\draftnotewidth}%
           \marginpar{\flushleft\textbf{\textit{\HUGE !\ }}\small #1}%
         \else
           \settowidth{\draftnotesignwidth}{\textbf{\textit{\HUGE\ !}}}%
           \addtolength{\draftnotewidth}{-\draftnotesignwidth}%
           \marginpar{\raggedleft\makebox[0pt][r]{\\\\\\\\ around
               \parbox[t]{\draftnotewidth}{\%\%\%\%\% funny behaviour
                 \raggedleft\small\hspace{0pt}#1%
               }}\textbf{\textit{\HUGE\ !}}%
           }%
         \fi
       }\@esphack}
   \else
     \newcommand{\draftnote}[1]{\@bsphack\@esphack}
   \fi
```

#### \makeatother

Bastiaan also noted that it provided an example of using the \foremargin length. If you want to try it out, either put the code in your preamble, or put it into a package (i.e., .sty file) without the \makeat... commands.

```
Sniplet C.4 (Adding indentation to footnotes)
```

At times a document design calls for a footnote configuration equal to the default but everything indented more to the right. This can be achieved via

```
\newlength\myextrafootnoteindent
\setlength\myextrafootnoteindent{\parindent}
\renewcommand\makefootmarkhook{%
  \addtolength{\leftskip}{\myextrafootnoteindent}}
```

In this example we indent the footnotes to match the overall paragraph indentation. We need to save the current value of \parindent since it is reset in the footnotes.

```
Sniplet C.5 (Background image and trimmarks)
```

This sniplet comes from another problem described in CTT. If one use the eso-pic package to add a background image, this image ends up on top of the trim marks. To get it *under* the trim marks Rolf Niepraschk suggested the following trick

```
\RequirePackage{atbegshi}\AtBeginShipoutInit
\documentclass[...,showtrims]{memoir}
...
\usepackage{eso-pic}
...
```

Sniplet C.6 (Autoadjusted number widths in the ToC)

When the ToC is typeset the chapter, section etc. number is typeset within a box of a certain fixed with (one width for each sectional type). If this width is too small for the current document, the user have to manually adjust this width.

In this sniplet we present a method where we automatically record the widest.

It a later memoir version, we may add similar code to the core.

There are two ways to record the widest entries of the various types, either preprocess the entire ToC or measure it as a part of the ToC typesetting, store it in the .aux and reuse it on the next run. We will use the later approach. There is one caveat: The .aux file is read at \begin{document}, so we need to postpone our adjustments via \AtBeginDocument.

The following solution use some ToC related hooks within the class, plus the etoolbox and calc packages.

First we create some macros to store information within the .aux file, and retrieve it again.

```
\makeatletter
\newcommand\mem@auxrestore[2]{\csgdef{stored@value@#1}{#2}}
\newcommand\memstorevalue[2]{%
  \@bsphack%
  \immediate\write\@mainaux{\string\mem@auxrestore{#1}{#2}}%
  \@esphack}
\newcommand\RetrieveStoredLength[1]{%
  \ifcsdef{stored@value@#1}{\csuse{stored@value@#1}}{\opt}}%
\makeatletter
```

Here \RetrieveStoredLength can be used in most \setlength cases, at least when the calc package is loaded. The argument will be the name of the variable one asked to be stored. If no corresponding value has been found for a given name, 0 pt is returned.

Next we need to prepare the hooks. In this case we will show how to take care of \chapter, \section and \subsection. \chapter is relatively easy:<sup>1</sup>

We use an alternative syntax to make the \widestchapter global.

Handling \section and \subsection is slightly more tricky, as they both use numberline. Instead we rely on the local value of the magic macro \cftwhatismyname.

```
\newlength\widestsection
\renewcommand\numberlinehook[1]{%
    % use a loop handler to loop over a list of possible
    % types. \forcsvlist comes from etoolbox
    \forcsvlist{\ToCHookListHandler{#1}}{section, subsection, subsubsection,%
        paragraph, subparagraph, figure, table}}
% the actual handler.
\newcommand\ToCHookListHandler[2]{%
    \edef\tmpstr{#2}%
    \ifdefstrequal{\cftwhatismyname}{\tmpstr}{%
    \settowidth\tmplen{\hbox{\csuse{cft\cftwhatismyname font}#1}}%
    \ifcslength{widest#2}{% is this length defined?
    \ifdimgreater\tmplen{\csuse{widest#2}}{%
    \global\csuse{widest#2}=\tmplen}{}}}}}
```

Even though the list mention more macros, we only use those we have added corresponding lengths for.

Next we need to store the values at the end of the document

<sup>1</sup> In some cases you may want to use {\@chapapp@head\@cftbsnum #1\@cftasnum}

```
\AtBeginDocument{\AtEndDocument{
   \memstorevalue{widestchapter}{\the\widestchapter}
   \memstorevalue{widestsection}{\the\widestsection}
   \memstorevalue{widestsubsection}{\the\widestsubsection}
}}
```

Here is how to get the standard class setup for a three level TOC. We also add a little extra padding to the boxes. Remember that it may take a few compilations before the ToC settles down.

Sniplet C.7 (Using class tools to make a chapter ToC)

By using a few hooks, we will be able to create a simple chapter toc. First a few notes:

- (a) In this class, the TOC data can be reused, thus we can load the TOC data as many times as we would like.
- (b) Data in the TOC is stored as arguments the \contentsline macro, say (see also Figure 9.1 on page 141)

```
\contentsline{chapter}{\chapternumberline {1}Test}{3}
```

where the first argument determins which macro is used to process the data. Each of these macros look at the value of the tocdepth counter to know whether to typeset or not.

(c) Using some hooks we can insert local changes to tocdepth in order to only typeset the sections from the current chapter.

The idea is to be able to add hooks at key points in the ToC data, and then use these hooks to enable and disable typesetting.

We will need to add hooks just after a chapter line (like the one above), and we will need to be able to insert hooks just before items that mark the end of a chapter, that is the next \chapter, \part, \book, plus a macro like \appendixpage which also write to the ToC.

First we define hooks that add hooks into the TOC. We use a counter to make each start and end hook unique. We add *end markers* above the ToC entries for \chapter, \part and \book.

```
\newcounter{tocmarker}
\renewcommand\mempreaddchaptertotochook{\cftinserthook{toc}{end-\thetocmarker}}
\renewcommand\mempreaddparttotochook {\cftinserthook{toc}{end-\thetocmarker}}
\renewcommand\mempreaddbooktotochook {\cftinserthook{toc}{end-\thetocmarker}}
\% start marker
\renewcommand\mempostaddchaptertotochook{\%
\stepcounter{tocmarker}\cftinserthook{toc}{start-\thetocmarker}}
\let\normalchangetocdepth\changetocdepth \% for later
```

The hooks inserted into the TOC file, does nothing by default. You will notice that the line above will now look like:

```
\cftinsert {end-0}
\contentsline{chapter}{\chapternumberline {1}Test}{3}
\cftinsert {start-1}
...
\cftinsert {end-1}
\contentsline{chapter}{\chapternumberline {2}Test}{5}
```

Thus to get a chapter toc command we need to make sure that (1) all entries are disabled, (2) at start-1 we reenable TOC entries, and (3) at end-1 disable TOC entries again. Here is the rest of the code, explained via comments.

```
\makeatletter
\newcommand\chaptertoc{
 % make changes local, remember counters a global
 \begingroup
 % store current value, to be restored later
 \setcounter{@memmarkcntra}{\value{tocdepth}}
 % when ever \settocdepth is used, it adds the new value to the
 \% ToC data. This cause problems when we want to disable all
 \% entries. Luckily the data is added via a special macro, we we
 % redefine it, remember we stored the original value earlier.
 \let\changetocdepth\@gobble
 % disable all entries (using our copy from above)
 \normalchangetocdepth{-10}
 % enable toc data within our block, we go as far as subsubsection
 \cftinsertcode{start-\thetocmarker}{\normalchangetocdepth{3}}
 % when the block is done, disable the remaining
 \cftinsertcode{end-\thetocmarker}{\normalchangetocdepth{-10}}
 % remove the spacing above the toc title
 \let\tocheadstart\relax
 % remove the toc title itself
```

```
\let\printtoctitle\@gobble
 % remove space below title
 \let\aftertoctitle\relax
 % reformat TOC entries:
 \setlength{\cftsectionindent}{Opt}
 \setlength{\cftsubsectionindent}{\cftsectionnumwidth}
 \setlength{\cftsubsubsectionindent}{\cftsubsectionindent}
 \addtolength{\cftsubsubsectionindent}{\cftsubsectionnumwidth}
 \renewcommand\cftsectionfont{\small}
 \renewcommand\cftsectionpagefont{\small}
 \renewcommand\cftsubsectionfont{\small}
 \renewcommand\cftsubsectionpagefont{\small}
 \renewcommand\cftsubsubsectionfont{\small}
 \renewcommand\cftsubsubsectionpagefont{\small}
 % include the actual ToC data
 \tableofcontents*
 \endgroup
 % restore tocdepth
 \setcounter{tocdepth}{\value{@memmarkcntra}}
 % to indent or not after the chapter toc
 \m@mindentafterchapter
 % space between chapter toc and text
 \par\bigskip
 \% handles indentation after the macro
 \@afterheading}
\makeatother
```

Note that if the \chapterprecistoc or \chapterprecis has been used then that data is also added to the ToC data, and we will need to locally disable it in the chapter ToC. This can be done by adding

```
\let\precistoctext\@gobble
```

to the \chaptertoc definition above, just make sure it is added before calling before \tableofcontents\*.

```
Sniplet C.8 (An appendix ToC)

Here we assume a structure like

\tableofcontents*
\chapter
\chapter
\chapter
\appendix
\appendixpage
\appendixtableofcontents
```

\chapter

\endgroup

\makeatother

}

```
\chapter
\chapter
where the first ToC should just show until (and including) \appendixpage, and
\appendixtableofcontents should only list the appendices.
   We also assume that no \settocdepth's have been issued after \appendixpage.
   We only need a single hook after \appendixpage.
 \renewcommand\mempostaddapppagetotochook{\cftinserthook{toc}{BREAK}}
 \cftinsertcode{BREAK}{\changetocdepth{-10}}
 \let\normalchangetocdepth\changetocdepth
                                               % needed for later
Then the definition of the actual appendix ToC:
 \makeatletter
 \newcommand\appendixtableofcontents{
   \begingroup
   \let\changetocdepth\@gobble
   \verb|\normalchangetocdepth{\{-10\}}|
   \cftinsertcode{BREAK}{\normalchangetocdepth{3}}
   \renewcommand\contentsname{Appendices overview}
   \tableofcontents*
```

## **Pictures**

There are many freely available LaTeX introductions on CTAN and other places. One thing that these apparently are not covering is the traditional picture environment. It can be very handy in many applications, though for more complex drawings the reader might be better of with TiKz/pgf or PSTricks. For the benefit of the general reader we here provide a lesson in the standard picture environment.

Writers comment: There are many extensions to the stock picture environment provided by the LaTeX kernel. We have chosen not to deal with them in this chapter but instead concentrate on what you get as is from the kernel. But there are a few handy packages that the reader might want to explore: picture (by Heiko Oberdiek) which extends the \put syntax to include arbitrary lengths, like 50mm; pict2e which is mentioned in [GM+07] but just recently was released; eepic. All packages are available from CTAN.

This chapter describes how to draw diagrams and pictures using LaTeX. Pictures are drawn in the picture environment. You can draw straight lines, arrows and circles; you can also put text into your pictures.

Most often pictures are drawn within a figure environment, but they may also be drawn as part of the normal text.

#### D.1 Basic principles

The positions of picture elements are specified in terms of a two-dimensional cartesian coordinate system. A *coordinate* is a number, such as 7, –21 or 1.78. In the cartesian coordinate system, a pair of coordinates (i.e., a pair of numbers) specifies a position relative to the position designated as (0,0). This special position is called the *origin*. The first of the coordinate pair gives the value of the horizontal distance from the origin to the position. A positive coordinate is an offset to the right and a negative number is an offset to the left. The first value of a coordinate pair is called the *x coordinate*. The second value of a coordinate pair is called the *y coordinate* and gives the vertical offset from the origin (positive upwards and negative downwards).

#### \unitlength

To draw a picture we also need to specify the units of measurement. By default, LaTeX takes the printer's point (there are 72.27 points to an inch) as the measurement of length. The value of the unit of length measurement within a picture environment is actually given by the value of the \unitlength length declaration. This can be changed to any length that you like via the \setlength command. For example,



Figure D.1: Some points in the cartesian coordinate system

\setlength{\unitlength}{2mm}

will make the value of \unitlength to be two millimeters.

Figure D.1 shows the positions of some points and their coordinate values. Coordinate pairs are typed as a pair of numbers, separated by a comma, and enclosed in parentheses.

```
\label{lines} $$ \thicklines $$ \linethickness{\langle len \rangle}$
```

In general, LaTeX can draw lines of only two thicknesses, thin and thick. The required thickness is specified via either a \thicklines or a \thinlines declaration, with the latter being the default.

There is another declaration, \linethickness, which can be used to change the thickness of horizontal and vertical lines only. It sets the thickness of these lines to  $\langle len \rangle$ , but has no effect on any sloping lines.

A picture environment has a required size pair argument that specifies the width and height of the picture, in terms of the value of \unitlength.

```
\begin{picture}(\langle width, height \rangle) \langle contents \rangle \end{picture} \\ begin{picture}(\langle width, height \rangle) (\langle llx, lly \rangle) \langle contents \rangle \end{picture} \\
```

The environment creates a box of size  $\langle width \rangle$  by  $\langle height \rangle$ , which will not be split across pages. The default position of the origin in this environment is at the lower left hand corner of the box. For example,

\begin{picture}(80,160)

creates a picture box of width 80 and height 160 whose lower left hand corner is at (0,0). There is also an optional coordinate pair argument (which comes after the required argu-

ment) that specifies the coordinates of the lower left hand corner of the box if you do not want the default origin.

```
\begin{picture}(80,160)(10,20)
```

specifies a picture box of width 80 and height 160, as before, but with the bottom left hand corner having coordinates of (10,20). Thus, the top right hand corner will have coordinates (90,180). Note that the optional argument is enclosed in parentheses not square brackets as is ordinarily the case. Typically, the optional argument is used when you want to shift everything in the picture. LaTeX uses the required argument to determine the space required for typesetting the result.

You are not limited to drawing within the box, but if you do draw outside the box the result might run into other text on the page, or even run off the page altogether. LaTeX only reserves the space you specify for the picture and does not know if anything protrudes. In particular

```
\begin{picture}(0,0)
```

creates a zero-sized picture which takes no space at all. This can be very useful if you need to position things on the page.

Within the picture environment, LaTeX is in a special *picture* mode (which is a restriced form of LR mode). The only commands that can appear in picture mode are \put, \multiput and \quad \quad \text{pezier} commands, and a few declarations such as the type style and the thickness declarations. By the way, you should only change the value of \unitlength outside picture mode otherwise LaTeX will get confused about its measurements.

#### D.2 Picture objects

In a picture everything is placed and drawn by the \put (or its \multiput variant) command.

\put places  $\langle object \rangle$  in the picture with the object's *reference point* at position ( $\langle x, y \rangle$ ). The following sections describe the various picture objects.

#### D.2.1 Text

Text is the simplest kind of picture object. This is typeset in LR mode and the reference point is at the lower left hand corner of the text.

# Source for example D.1



In the diagram, and those following, the reference point is indicated by an arrow. Also, a box is drawn round the diagram at the same size as the picture environment.

#### D.2.2 Boxes

A box picture object is made with one of the box commands. When used in picture mode, the box commands have a slightly different form than when in normal text. The first argument of a box command is a size pair that specifies the width and height of the box. The last argument is the text to be placed in the box. The reference point of a box is the lower left hand corner.

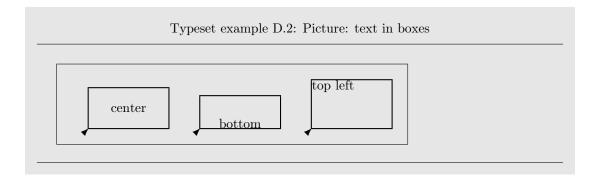
```
\label{lem:lemmakebox} $$ \ \ (\langle width, height \rangle) [\langle pos \rangle] {\langle text \rangle} $$ $$ \makebox(\langle width, height \rangle) [\langle pos \rangle] {\langle text \rangle} $$
```

The \framebox command draws a framed box of the specified ( $\langle width, height \rangle$ ) dimensions around the text.

```
Source for example D.2
\setlength{\unitlength}{1pc}
\begin{picture}(22,5)
\begin{array}{l} \begin{array}{l} \begin{array}{l} \begin{array}{l} \\ \end{array} \end{array} \end{array}
                                                       % empty box
\thicklines
\begin{array}{l} \begin{array}{l} \begin{array}{l} \begin{array}{l} \\ \end{array} \end{array} \end{array}
                                                       % centered text
\put(2,1){\vector(1,1){0}}
                                                       % ref point
\put(9,1){\framebox(5,2)[b]{bottom}}
                                                       % bottomed text
\put(9,1){\vector(1,1){0}}
                                                       % ref point
\put(16,1){\framebox(5,3)[tl]{top left}} % cornered text
\put(16,1){\vector(1,1){0}}
                                                       % ref point
\end{picture}
\setlength{\unitlength}{1pt}
```

The default position of the *text* is centered in the box. However, this can be changed via an optional argument (which is enclosed in square brackets), placed between the coordinate pair and the text argument. This argument consists of either one or two of the following letters.

I (left) Places the contents at the left of the box.

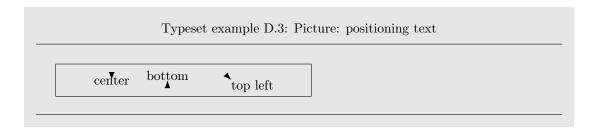


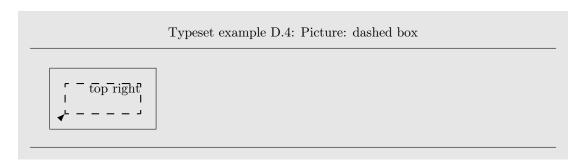
- r (right) Places the contents at the right of the box.
- t (top) Places the contents at the top of the box.
- b (bottom) Places the contents at the bottom of the box.

These place the text in the corresponding position in the box. In a two-letter argument the order of the letters is immaterial. For example, [tr] and [rt] will both result in the text being placed at the top right hand corner of the box. Unlike the normal \framebox command, a \framebox in a picture environment does not add any extra space around the text.

Corresponding to the \framebox there is a \makebox command which does not draw a frame around its contents. The \makebox command takes the same arguments as the \framebox. Particularly interesting is when you specify a zero sized \makebox. A \makebox(0,0){text} command will make the reference point the center of text. Similarly, the other positioning arguments which will adjust the reference point with respect to the box contents. This can be used for fine-tuning the position of text in a picture.

You can draw a dashed box with the \dashbox command.





```
\dshbox{\langle len \rangle}(\langle width, height \rangle) [\langle pos \rangle] {\langle text \rangle}
```

The first argument of this command specifies the length of each dash. The following arguments are the same as for the other box commands.

```
Source for example D.4

\setlength{\unitlength}{4mm}
\begin{picture}(7,4)
\put(0,0){\framebox(7,4){}}
\thicklines
\put(1,1){\dashbox{0.5}(5,2)[tr]{top right}}
\put(1,1){\vector(1,1){0}}
\end{picture}
\setlength{\unitlength}{1pt}
```

The appearance of the box is best when the width and height of the box are integer multiples of the dash length. In the example the dash length has been set to 0.5 with the width and height set as (5,2); thus the width and height are respectively ten and four times the dash length.

The \frame command draws a frame around the contents of the box that exactly fits the contents.

```
\frac{\langle contents \rangle}{}
```

It takes a single required argument which is the contents.

### Typeset example D.5: Picture: framing FRAME text

```
Source for example D.5

\setlength{\unitlength}{1pc}
\begin{picture}(10,3)
\put(0,0){\framebox(10,3){}}
\thicklines
\put(0.5,2){\frame{$\mathbb{FRAME}$ text}}
\put(0.5,2){\vector(1,1){0}}
\end{picture}
\setlength{\unitlength}{1pt}
```

The \shortstack command enables you to stack text vertically. It produces a box with a single column of text. As with the other boxes, the reference point is at the lower left hand corner, although no frame is drawn around the stack. The \shortstack command is an ordinary box making command, but it is not often used outside picture mode.

Each line of  $\langle text \rangle$ , except for the last, is ended by a \\ command. The default is to center each text line within the column. However, there is an optional positioning argument. A value of | for  $\langle pos \rangle$  will left align the text and a value of r will right align the text.

```
Source for example D.6

\setlength{\unitlength}{1mm}
\begin{picture}(75,25)
\put(0,0){\framebox(75,25){}}
\put(3,3){\shortstack{Default \\ short \\ Stack}}
\put(3,3){\vector(1,1){0}}
\put(23,3){\shortstack[1]{Left\\aligned\\short\\Stack}}
\put(23,3){\vector(1,1){0}}
\put(43,3){\shortstack[r]{Right\\aligned\\short\\Stack}}
\put(43,3){\vector(1,1){0}}
```

## Typeset example D.6: Picture: stacking Extra Left Right Default aligned aligned spaced short short short Stack Stack Stack Stack Stack Stack

```
\put(63,3){\shortstack{Extra \[4ex] spaced \[2ex] Stack}} \put(63,3){\vector(1,1){0}} \end{picture} \setlength{\unitlength}{1pt}
```

The rows in a stack are not evenly spaced. The spacing between two rows can be changed in one of two ways.

- 1. Add a strut to a row. A strut is a vertical rule with no width.
- 2. Use the optional argument to the \\ command. This optional argument is a length value.

```
\lceil \langle len \rangle \rceil
```

It has the effect of adding additional  $\langle len \rangle$  vertical space between the two lines that the \\ separates.

```
\label{local_box} $$ \operatorname{box}{\langle box \rangle} (\langle width, height \rangle) [\langle pos \rangle] {\langle text \rangle} $$ \\ \operatorname{box}{\langle box \rangle} {\langle text \rangle} $$ \\ \operatorname{box}{\langle box \rangle} $$
```

Just as in normal text you can save and reuse boxes. The \savebox macro in picture mode is a variant of the normal text version, but the other three commands are the same in both picture and paragraph modes, and are described in Chapter 15. In picture mode you have to specify the size of the storage box when saving it, via the ( $\langle width, height \rangle$ ) argument to \savebox.

A \savebox command can be used within a picture to store a picture object. The first argument of \savebox is the name of the storage bin to be used. The following arguments are the same as the \makebox command. The result is stored, not drawn. When you have saved something it can be drawn in either the same or other pictures via the \usebox command. This command takes one argument, which is the name of the storage bin.



# Source for example D.7 \setlength{\unitlength}{1pc} \begin{picture}(18,5) \put(0,0){\framebox(18,5){}} \newsavebox{\Mybox} \savebox{\Mybox}(6,3)[tr]{\\$\mathcal{SAVED}\\$} \thicklines \put(1,1){\frame{\usebox{\Mybox}}} \put(11,1){\frame{\usebox{\Mybox}}} \put(11,1){\vector(1,1){0}} \put(11,1){\vector(1,1){0}} \put(11,1){\vector(1,1){0}} \end{picture} \setlength{\unitlength}{1pt}

It can take LaTeX a long time to draw something. When a box is saved it actually contains the typeset contents, which then just get printed out when the box is used. It can save processing time if something which appears several times is saved and then used as and where required. On the other hand, a saved box can use up a significant amount of LaTeX's internal storage space. The \sbox command with an empty text argument can be used to delete the contents of a bin. For example,

\sbox{\Mybox}{}

will empty the \Mybox box. Note that this does not delete the storage box itself.

### D.2.3 Lines

LaTeX can draw straight lines, but the range of slopes for lines is somewhat restricted. Further, very short lines cannot be drawn.

```
\label{line} \label{line} \label{line} $$ \lim(\langle i,j\rangle) \{\langle distance\rangle\}$
```

The pair  $(\langle i, j \rangle)$  specifies the *slope* of the line, and  $\langle distance \rangle$  is a value that controls the length of the line. The line starts at its reference point (i.e., the place where it is \put). The

slope of the line is such that if a point on the line is slid along the line, then for every i units the point moves in the horizontal direction it will also have moved j units in the vertical direction. Negative values for i or j have the expected meaning. A move of -3 units in i means a move of 3 units to the left, and similarly a move of -4 units in j means a move of 4 units downwards. So, a line sloping up to the right will have positive values for i and j, while a line sloping up to the left will have a negative value for i and a positive value for j.

The  $\langle distance \rangle$  argument specifies the length of the line in the x (horizontal) direction. One problem with this may have occured to you: what if the line is vertical (i.e., i=0)? In this case only,  $\langle distance \rangle$  specifies the vertical length of the line. The  $\langle distance \rangle$  argument must be a non-negative value. For horizontal and vertical lines only, the actual length of the line is  $\langle distance \rangle$ . Figure D.2, which is produced from the code below, diagrams the line specification arguments.

```
\begin{figure}
\centering
\setlength{\unitlength}{1mm}
\begin{picture}(70,60)
\thicklines % draw line and ref point
  \begin{array}{l} \begin{array}{l} \begin{array}{l} \begin{array}{l} \\ \\ \end{array} \end{array} \end{array}
  \put(10,20){\vector(1,-1){0}}
\thinlines
                % draw axes
  \begin{array}{l} \begin{array}{l} \begin{array}{l} \begin{array}{l} (0,0) \\ \end{array} \end{array} \end{array} 
  % draw i and j vectors
  \put(20,25){\vector(1,0){20}}
  \put(30,22){\makebox(0,0)[t]{$i$}}
  \t(40,25) {\vector(0,1) {10}}
  \put(42,30){\makebox(0,0)[1]{\$j\$}}
                % draw distance vector
  \put(30,10){\vector(-1,0){20}}
  \put(30,10){\vector(1,0){20}}
  \put(30,8){\mbox(0,0)[t]{\textit{distance}}}
\end{picture}
\setlength{\unitlength}{1pt}
\caption{Specification of a line or arrow}
\label{flpic:spec}
\end{figure}
```

Only a fixed number of slopes are available. This is because LaTeX uses a special font for drawing lines — a line actually consists of little bits of angled rules joined together. Thus, there is only a limited number of values for i and j. They must both be integers and in the range  $-6 \le i, j \le 6$ . Also, they must have no common divisor other than 1. In other words, the ratio between i and j must be in its simplest form. You cannot, for example, have (3,6); instead it would have to be (1,2). The shortest line that LaTeX can draw is about ten points (1/7) inch approximately) in overall length. You can, though, draw lines that are too long to fit on the page.

Figure D.3 shows the lines and arrows slanting upwards and to the right that can be drawn in LaTeX. The slope (i, j) pair are shown to the right of the first set of lines and

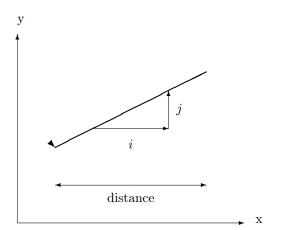


Figure D.2: Specification of a line or arrow

arrows, together with the j/i ratio which gives the slope of the line as a decimal number.

### D 2.4 Arrows

As shown in Figure D.3 you can also draw a line with an arrowhead on it. These are specified by the \vector command.

```
\vector(\langle i, j \rangle) \{\langle distance \rangle\}
```

This works exactly like the \line command and the arrowhead is put on the line at the end away from the reference point. That is, the arrow points away from the reference point. If the  $\langle distance \rangle$  argument is too small (zero, for instance) the arrowhead only is drawn, with its point at the position where it is \put.

LaTeX is even more restrictive in the number of slopes that it can draw with arrows than it is with lines. The (i,j) slope specification pair must lie in the range  $-4 \le i,j \le 4$ . Also, as with the \line command, they must have no common divisor.

### D.2.5 Circles

LaTeX can draw two kinds of circles. One is an open circle where only the perimeter is drawn, and the other is a solidly filled disk.

```
\circle{\langle diameter \rangle}
\circle*{\langle diameter \rangle}
```

The reference point for the open circle, drawn by the \circle command, and the disk, which is drawn by the \circle\* command, is at the center of the circle. The argument to the commands is the \( \lambda iameter \rangle \) of the circle.

### Source for example D.8

\setlength{\unitlength}{1pt}

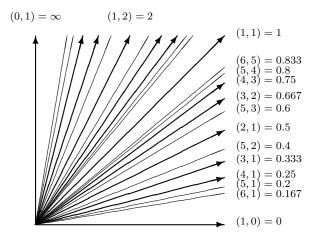
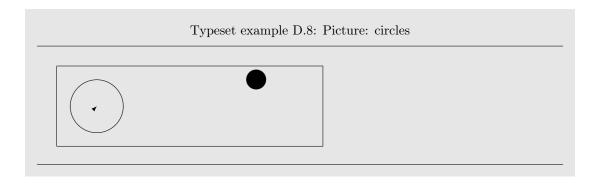


Figure D.3: Sloping lines and arrows



```
\begin{picture}(200,60)
\put(0,0){\framebox(200,60){}}
\put(30,30){\circle{40}}
\put(30,30){\vector(1,1){0}}
\put(150,50){\circle*{20}}
\end{picture}
\setlength{\unitlength}{1pt}
```

Just as with the \line and \vector commands, there is only a limited range of circles that can be drawn. Typically, the maximum diameter of a \circle is about 40 points, while for a \circle\* the maximum diameter is less, being about 15 points. LaTeX will choose the nearest sized circle to the one that you specify. Either consult your local guru to find what



sized circles you can draw on your system, or try some experiments by drawing a range of circles to see what happens.

Quarter circles and boxes

In LaTeX an \oval is a rectangular box with rounded corners.

The \oval command has one required argument which specifies the width and height of the box. The normally sharp corners of the box are replaced by quarter circles of the maximum possible radius (which LaTeX figures out for itself). Unlike the boxes discussed earlier, the reference point is at the 'center' of the oval.

```
Source for example D.9
\setlength{\unitlength}{1mm}
\begin{picture}(75,20)
\thicklines
\begin{array}{l} \begin{array}{l} \begin{array}{l} \begin{array}{l} \\ \\ \end{array} \end{array} \end{array} 
\put(15,10){\oval(15,10)}
                                         % complete oval
\begin{array}{l} \begin{array}{l} \begin{array}{l} \begin{array}{l} \\ \end{array} \end{array} \end{array} 
\put(30,10){\oval(5,5)}
                                         % small oval
\put(30,10){\vector(1,1){0}}
\put(45,10){\oval(15,10)[1]} % left half
\put(45,10){\vector(1,1){0}}
\put(60,10){\oval(15,10)[bl]} % bottom left quarter
\put(60,10){\vector(1,1){0}}
\end{picture}
\setlength{\unitlength}{1pt}
```

The \oval command also has one optional argument, \( \lambda portion \rangle \), which comes after the required argument. Use of the optional argument enables either half or a quarter of the complete rounded box to be drawn. The argument is a one or two letter code drawn from the following.

### Typeset example D.10: Picture: text in oval

Text in oval

- I (left) Draw the left of the oval.
- r (right) Draw the right of the oval.
- t (top) Draw the top of the oval.
- b (bottom) Draw the bottom of the oval.

These are similar to the optional positioning argument in the box commands. A one letter code will draw the designated half of the oval, while a two letter code results in the designated quarter of the oval being drawn. In all cases the reference point is at the center of the 'complete' oval.

### Source for example D.10

```
\setlength{\unitlength}{1mm}
\begin{picture}(30,10)
\thicklines
\put(15,5){\oval(30,10)}
\put(15,5){\makebox(0,0){Text in oval}}
\end{picture}
\setlength{\unitlength}{1pt}
```

Unlike the boxes described in §D.2.2 there is no  $\langle text \rangle$  argument for an \oval. If you want the rounded box to contain text, then you have to place the text inside the box yourself. The code in example D.10 shows one way of doing this; a zero-sized box is used to center the text at the center of the oval.

### D.3 Repetitions

The \multiput command is a convenient way to place regularly spaced copies of an object in a picture.

```
\multiput(\langle x, y \rangle)(\langle dx, dy \rangle){\langle num \rangle}{\langle object \rangle}
```

As you can see, this is similar to the syntax for the \put command, except that there are two more required arguments, namely  $(\langle dx, dy \rangle)$  and num.



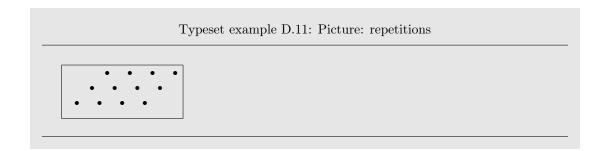
Figure D.4: Some measuring scales

The  $(\langle dx, dy \rangle)$  argument is a pair of (decimal) numbers that specify the amount that the  $\langle object \rangle$  shall be moved at each repetition. The first of this pair specifies the horizontal movement and the second the vertical movement. Positive values shift to the right or up, and negative numbers shift to the left or down. The  $\langle num \rangle$  argument specifies how many times the  $\langle object \rangle$  is to be drawn.

The code below produces Figure D.4. This example also shows that a picture can be placed inside another picture. Often it is useful to break a complex diagram up into pieces, with each piece being a separate picture. The pieces can then be individually positioned within the overall diagram.

```
\begin{figure}
\setlength{\unitlength}{1pc}
\centering
\begin{picture}(21,26)
% Draw Pica scale
\put(2,2){\begin{picture}(5,24)
\put(0,-0.5){\makebox(0,0)[t]{\textbf{Picas}}}
\thicklines \put(0,0){\line(0,1){24.0}}
```

```
\t \mathbb{1}
            \mathsf{Multiput}(0,0)(0,10){3}{\mathsf{line}(1,0){2}}
 \t(-1,0) {\bf 0},0) [br] {0}
 \put(-1,10){\makebox(0,0)[br]{10}}
 \put(-1,20){\makebox(0,0)[br]{20}}
  \end{picture}}
% Draw Points scale
\polinimes (7,2){\left\langle picture \right\rangle (5,24)}
 \put(0,-0.5){\makebox(0,0)[t]{\textbf{Points}}}
 \frac{0,0}{\ln(0,1){24.2}}
 \hat{0},0,0.8333,30{\line(1,0){1}}
            \mbox{multiput}(0,0)(0,8.333){3}{\line(1,0){2}}
 \t(-1,0) {\bf 0},0) [br] {0}
 \put(-1,8.333){\makebox(0,0)[br]{100}}
 \polinimes (0,0) [br] {200}
 \end{picture}}
% Draw Millimeter scale
\put(12,2){\begin{picture}(5,24)
 \put(0,-0.5){\mbox(0,0)[t]{\textbf{Millimeters}}}
 \frac{0,0}{\ln(0,1){24.2}}
 \t \mathbb{1}
            \mathsf{Multiput}(0,0)(0,2.3711)\{11\}\{\mathsf{line}(1,0)\{2\}\}\
 \polinizer (-1,0) {\makebox(0,0)[br]{0}}
 \put(-1,4.742){\makebox(0,0)[br]{20}}
 \put(-1,9.484){\mathbf box(0,0)[br]{40}}
 \put(-1,14,226){\mathbf br}{60}
 \put(-1,18.968){\makebox(0,0)[br]{80}}
  \t(-1,23.71){\mathbf br}{100}
 \end{picture}}
% Draw Inch scale
\put(17,2){\begin{picture}(5,24)
  \put(0,-0.5){\makebox(0,0)[t]{\textbf{Inches}}}
 \frac{0,0}{\ln(0,1)}
 \thinlines \multiput(0,0)(0,0.60225)\{41\}\{\line(1,0)\{1\}\}
            \mathsf{Multiput}(0,0)(0,6.0225){5}{\mathsf{line}(1,0){2}}
 \polinimiz (-1,0){\makebox(0,0)[br]{0}}
 \put(-1,6.0225){\makebox(0,0)[br]{1}}
 \put(-1,12.045){\makebox(0,0)[br]{2}}
 \poline{put(-1,18.0675){\mathbb{Q}}(0,0)[br]{3}}
 \polinimiz (-1,24.09){\makebox(0,0)[br]{4}}
 \end{picture}}
\end{picture}
\setlength{\unitlength}{1pt}
\caption{Some measuring scales} \label{flpic:scales}
\end{figure}
```



You can also make regular two-dimensional patterns by using a \multiput pattern inside another \multiput. As LaTeX will process each \multiput every time it is repeated it is often more convenient to store the results of the first \multiput in a bin and then use this as the argument to the second \multiput.

```
Source for example D.11 $$ \operatorname{lum}_{\operatorname{lum}}_{1mm} \Big( (32,14) \\ \operatorname{lun}_{0,0}_{\operatorname{mebox}(32,14)}_{1mm} \Big( (32,14) \\ \operatorname{lun}_{0,0}_{1mmebox}(32,14)_{1mm} \Big( (32,14)_{1mmebox}_{1mmebox}(32,14)_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox}_{1mmebox
```

Remember that a storage bin must have been declared via a \newsavebox command before it can be used. I originally declared and used the \Mybox bin in \SD.2.2. As the above example shows, you can change the contents of a storage bin by utilising it in another \savebox. Storage bins can use up a lot of LaTeX's memory. After you have finished with a storage bin empty it via the \sbox command with an empty last argument, as shown in the example.

### D.4 Bezier curves

Standard LaTeX provides one further drawing command — the \qubezier command. This can be used for drawing fairly arbitrary curves.

```
\qbezier[\langle num\rangle](\langle Xs, Ys\rangle)(\langle Xm, Ym\rangle)(\langle Xe, Ye\rangle)
```

The command will draw what geometers call a *quadratic Bezier curve* from the point ( $\langle Xs, Ys \rangle$ ) to the point ( $\langle Xe, Ye \rangle$ ). The curve will pass somewhere near to the point ( $\langle Xm, Ym \rangle$ ).

Bezier curves are named after Pierre Bezier who first used them in 1962. They are widely used in Computer Aided Design (CAD) programs and other graphics and font design systems. Descriptions, with varying degrees of mathematical complexity, can be found in many places: when I was a practicing geometer these included [FP80], [Mor85]

and [Far90]; no doubt there are more recent sources available and there is a brief review in [Wil04a].

Figure D.5 shows two of these curves. The figure was produced by the code below.

```
\begin{figure}
\setlength{\unitlength}{1mm}
\centering
\begin{picture}(100,100)
\thicklines % first curve
\qbezier(10,50)(50,90)(50,50)
\thinlines % draw lines joining control points
\begin{array}{l} \begin{array}{l} \begin{array}{l} \begin{array}{l} \\ \\ \end{array} \end{array} \end{array}
\put(50,90){\line(0,-1){40}}
% label control points
\put(10,45){\makebox(0,0)[t]{\texttt{(10,50)}}}
\put(50,95){\makebox(0,0)[b]{\texttt{(50,90)}}}
\t(55,50) \makebox(0,0) [1] {\texttt{(50,50)}}
\thicklines % second curve
\qbezier[25](50,50)(50,10)(90,50)
\thinlines % draw lines joining control points
% \put(50,10){\line(1,1){40}}
% label control points
\put(50,5){\makebox(0,0)[t]{\texttt{(50,10)}}}
\put(90,55){\mbox(0,0)[b]{\texttt{(90,50)}}}
\end{picture}
\setlength{\unitlength}{1pt}
\caption{Two Bezier curves}
\label{lpicf:bez}
\end{figure}
```

The three points used to specify the position and shape of the Bezier curve are called *control points*. The curve starts at the first control point and is tangent to the line joining the first and second control points. The curve stops at the last control point and is tangent to the line joining the last two control points.

In Figure D.5 the lines joining the control points for the first curve have been drawn in. The locations of all the control points for the two curves are labeled.

The second Bezier curve is the same shape as the first one, but rotated 180 degrees. The first control point of this curve is the same as the last control point of the first curve. This means that the two curves are joind at this point. The line, although it is not drawn, connecting the first two control points of the second curve is in the same direction as the line joining the last two control points of the first curve. This means that the two curves are also tangent at the point where they join. By stringing together several Bezier curves you can draw quite complex curved shapes.

```
\qbeziermax
```

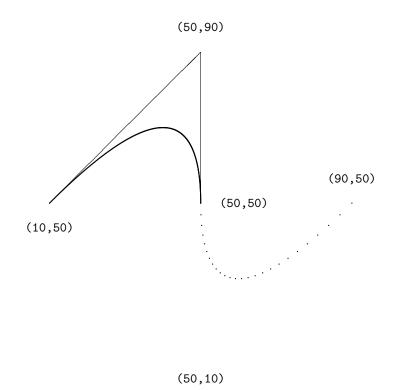


Figure D.5: Two Bezier curves

The Bezier curves are actually drawn as a linearized form using a series of rectangular blobs of ink. Left to itself, LaTeX will attempt to pick the number of blobs to give the smoothest looking curve, up to a maximum number. (Each blob takes up space in LaTeX's internal memory, and it may run out of space if too many are used in one picture.) The maximum number of blobs per Bezier curve is set by the \quad peziermax command. This can be adjusted with the \renewcommand command. For example:

\renewcommand{\qbeziermax}{250}

will set the maximum number of blobs to be 250.

Another method of controlling the number of blobs is by the optional  $\langle num \rangle$  argument to the \qbezier command. If used, it must be a positive integer number which tells LaTeX exactly how many blobs to use for the curve.

### LaTeX and TeX

Strictly speaking, LaTeX is a set of macros built on top of the TeX program originally developed by Donald Knuth [Knu86, Knu84] in the early 1980's. TeX is undoubtedly one of the most robust computer programs to date.

Leslie Lamport says that most TeX commands can be used with LaTeX and lists those that cannot be used [Lam94, Appendix E]. Apart from this he says nothing about any TeX commands. I have used some TeX macros in the code examples and so I need to talk a little bit about these.

I like to think of the commands and macros as falling into one of several groups.

- TeX primitives. These are the basic constructs of the TeX language.
- TeX commands or macros. These are part of the plain TeX system and are constructed from the TeX primitives.
- LaTeX kernel commands or macros. These are defined in the LaTeX kernel and are based on plain TeX primitives or commands. In turn, some higher level kernel macros are constructed from more basic aspects of the kernel. The kernel does redefine some of the plain TeX commands.
- Class command. These are mainly built up on the kernel commands but may use some basic TeX.
- Package commands. These are similar to the class commands but are less likely to directly use TeX macros.
- User commands. Typically these are limited to the commands provided by the class and any packages that might be called for, but more experienced users will employ some kernel commands, like \newcommand, to make their authoring more efficient.

Although TeX is designed as a language for typesetting it is also a 'Turing complete' language which means that it can perform any function that can be programmed in any familiar programming language. For example, an interpreter for the BASIC language has been written in TeX, but writing this kind of program using TeX is something that only an expert<sup>1</sup> might consider.

Nevertheless, you may have to, or wish to, write a typesetting function of your own. This chapter presents a few of the programming aspects that you may need for this, such

<sup>1</sup> Probably also a masochist with plenty of time.

as performing some simple arithmetic or comparing two lengths. For anything else you will have to read one or more of the TeX books or tutorials.

In England witnesses at a trial have to swear to 'Tell the truth, the whole truth, and nothing but the truth'. I will try and tell the truth about TeX but, to misquote Hamlet

There are more things in heaven and TeX, Horatio, Than are dreamt of in your philosophy.

### E.1 The TeX process

As we are delving deeper than normal and because at the bottom it is the TeX language that does all the work, it is useful to have an idea of how TeX processes a source file to produce a dvi file. It is all explained in detail by Knuth [Knu84] and also perhaps more accessibly by Eijkhout [Eij92]; the following is a simplified description. Basically there are four processes involved and the output from one process is the input to the following one.

Input The input process, which Knuth terms the 'eyes', reads the source file and converts what it sees into *tokens*. There are essentially two kinds of token. A token is either a single character such as a letter or a digit or a punctuation mark, or a token is a control sequence. A *control sequence* consists of a backslash and either all the alphabetic characters immediately following it, or the single non-alphabetic following it. Control sequence is the general term for what I have been calling a macro or a command.

Expansion The expansion processor is what Knuth calls 'TeX's mouth'. In this process some of the tokens from the input processor are expanded. Expansion replaces a token by other tokens or sometimes by no token. The expandible tokens include macros, conditionals, and a number of TeX primitives.

Execution The execution process is TeX's 'stomach'. This handles all the tokens output by the expansion processor. Control sequences that are not expandible are termed *executable*, and the execution processor executes the executable tokens. Character tokens are neither expandible nor executable. It handles any macro definitions and also builds horizontal, vertical and mathematical lists.

Layout The layout processor (TeX's 'bowels') breaks horizontal lists into paragraphs, mathematical lists into formulae, and vertical lists into pages. The final output is the dvi file.

In spite of the sequential nature implied by this description the overall process includes some feedback from a later process to an earlier one which may affect what that does.

It is probably the expansion processor that needs to be best understood. Its input is a sequence of tokens from the input processor and its output is a sequence of different tokens.

In outline, the expansion processor takes each input token in turn and sees if it is expandible; if it is not it simply passes it on to the output. If the token is expandible then it is replaced by its expansion. The most common expandible tokens are control sequences that have been defined as macros. If the macro takes no arguments then the macro's name is replaced by its definition. If the macro takes arguments, sufficient tokens are collected to get

the values of the arguments, and then the macro name is replaced by the definition. The expansion processor then looks at the first token in the replacement, and if that is expandible it expands that, and so on.

Nominally, the eventual output from the expansion processor is a stream of non-expandible tokens. There are ways, however of controlling whether or not the expansion processor will actually expand an expandible token, and to control the order in which things get expanded, but that is where things get rapidly complicated.

The layout processor works something like this. Ignoring maths, TeX stores what you type in two kinds of lists, vertical and horizontal. As it reads your words it puts them one after another in a horizontal list. At the end of a paragraph it stops the horizontal list and adds it to the vertical list. At the beginning of the next paragraph it starts a new horizontal list and adds the paragraph's words to it. And so on. This results in a vertical list of horizontal lists of words, where each horizontal list contains the words of a paragraph.

It then goes through each horizontal list in turn, breaking it up into shorter horizontal lists, one for each line in the paragraph. These are put into another vertical list, so conceptually there is a vertical list of paragraphs, and each paragraph is a vertical list of lines, and each line is a horizontal list of words, or alternatively one vertical list of lines. Lastly it chops up the vertical list of lines into page sized chunks and outputs them a page at a time.

TeX is designed to handle arbitrary sized inserts, like those for maths, tables, sectional divisions and so forth, in an elegant manner. It does this by allowing vertical spaces on a page to stretch and shrink a little so that the actual height of the typeblock is constant. If a page consists only of text with no widow or orphan then the vertical spacing is regular, otherwise it is likely to vary to some extent. Generally speaking, TeX is not designed to typeset on a fixed grid, but against this other systems are not designed to produce high quality typeset mathematics. Attempts have been made to tweak LaTeX to typeset on a fixed grid but as far as I know nobody has been completely successful.

TeX works somewhat more efficiently than I have described. Instead of reading the whole document before breaking paragraphs into lines, it does the line breaking at the end of each paragraph. After each paragraph it checks to see if it has enough material for a page, and outputs a page whenever it is full. However, TeX is also a bit lazy. Once it has broken a paragraph into lines it never looks at the paragraph again, except perhaps to split it at a page break. If you want to change, say, the width of the typeblock on a particular page, any paragraph that spills over from a previous page will not be reset to match the new measure. This asynchronous page breaking also has an unfortunate effect if you are trying to put a note in say, the outside margin, as the outside is unknown until after the paragraph has been set, and so the note may end up in the wrong margin.

### E.2 LaTeX files

The aux file is the way LaTeX transfers information from one run to the next and the process works roughly like this.

• The aux file is read at the start of the document environment. If \nofiles has not been specified a new empty aux file is then created which has the side effect of destroying the original aux file.

- Within the document environment there may be macros that write information to the aux file, such as the sectioning or captioning commands. However, these macros will not write their information if \nofiles has been specified.
- At the end of the document environment the contents of the aux file are read.

Under normal circumstances new output files are produced each time LaTeX is run, but when \nofiles is specified only the dvi and log files will be new — any other files are unchanged.

In the case of the sectioning commands these write macros into the aux file that in turn write information into a toc file, and the \tableof contents command reads the toc file which contains the information for the Table of Contents. To make this a bit more concrete, as LaTeX processes a new document through the first two runs, the following events occur.

- 1. Initially there is neither an aux nor a toc file. At the start of the document environment a new empty aux file is created.
- 2. During the first run the \tableofcontents typesets the Contents heading and creates a new empty toc file.
  - During the run sectional commands write information into the new aux file. At the end of the document environment the aux file is read. Contents information in the aux file is written to the toc file. Lastly all the output files are closed.
- 3. For the second run the aux file from the previous run is read at the start of the document environment; no information can be written to a toc file because the toc file is only made available by the \tableofcontents command. The aux file from the previous run is closed and the new one for this run is created.

This time the \tableofcontents reads toc file that was created during the previous run which contains the typesetting instructions for the contents, and then starts a new toc file.

And so the process repeats itself.

The aux file mechanism means that, except for the simplest of documents, LaTeX has to be run at least twice in order to have all the information to hand for typesetting. If sections are added or deleted, two runs are necessary afterwards to ensure that everything is up to date. Sometimes three, or even more, runs are necessary to guarantee that things are settled.

### E.3 Syntax

The LaTeX syntax that you normally see is pretty regular. Mandatory arguments are enclosed in curly braces and optional arguments are enclosed in square brackets. One exception to this rule is in the picture environment where coordinate and direction pairs are enclosed in parentheses.

The TeX syntax is not regular in the above sense. For example, if in LaTeX you said

```
\newcommand*{\cmd}[2]{#1 is no. #2 of}
\cmd{M}{13} the alphabet. % prints: M is no. 13 of the alphabet
```

Then in TeX you would say

```
\def\cmd#1#2{#1 is no. #2 of}
```

and you could then use either of the following calls:

```
\cmd M{13} the alphabet. \% prints: M is no. 13 of the alphabet \cmd{M}{13} the alphabet. \% prints: M is no. 13 of the alphabet
```

A simplistic explanation of the first TeX call of \cmd is as follows. A control sequence starts with a backslash, followed by either a single character, or one or more of what TeX thinks of as letters (normally the 52 lower- and upper-case alphabetic characters); a space or any non-letter, therefore, ends a multiletter control sequence. TeX and LaTeX discard any spaces after a macro name. If the macro takes any arguments, and \cmd takes two, TeX will then start looking for the first argument. An argument is either something enclosed in braces or a single token. In the example the first token is the character 'M', so that is the value of the first argument. TeX then looks for the second argument, which is the '13' enclosed in the braces. In the second example, both arguments are enclosed in braces.

Here are some TeX variations.

```
\cmd B{2} the alphabet. % prints: B is no. 2 of the alphabet.
\cmd B2 the alphabet. % prints: B is no. 2 of the alphabet.
\cmd N14 the alphabet. % prints: N is no. 1 of4 the alphabet.
```

The result of \cmd B{2} is as expected. The results of \cmd B2 and \cmd N14 should also be expected, and if not take a moment to ponder why. The 'B' and 'N' are the first arguments to \cmd in the two cases because a single character is a token. Having found the first argument TeX looks for the second one, which again will be a token as there are no braces. It will find '2' and '1' as the second arguments and will then expand the \cmd macro. In the case of \cmd B2 this gives a reasonable result. In the case of \cmd N14, TeX expands \cmd N1 to produce 'N is in position 1 of', then continues printing the rest of the text, which is '4 the alphabet', hence the odd looking result.

### E.4 (La)TeX commands

I have used some TeX commands in the example code and it is now time to describe these. Only enough explanation is given to cover my use of them. Full explanations would require a doubling in the size of the book and a concomitant increase in the price, so for full details consult the *TeXbook* which is the definitive source, or one of the TeX manuals listed in the Bibliography. I find *TeX by Topic* particularly helpful.

I have also used LaTeX commands that are not mentioned by Lamport. LaTeX uses a convention for command names; any name that includes the @ character is an 'internal' command and may be subject to change, or even deletion. Normal commands are meant to be stable — the code implementing them may change but their effect will remain unaltered. In the LaTeX kernel, and in class and package files the character @ is automatically treated as a letter so it may be used as part of a command name. Anywhere else you have to use \makeatletter to make @ be treated as a letter and \makeatother to make @ revert to its other meaning. So, if you are defining or modifying or directly using any command that includes an @ sign then this must be done in either a .sty file or if in the document itself it must be surrounded by \makeatletter and \makeatother.

The implication is 'don't use internal commands as they may be dangerous'. Climbing rocks is also dangerous but there are rock climbers; the live ones though don't try climbing Half Dome in Yosemite or the North Face of the Eiger without having first gained experience on friendlier rocks.

The LaTeX kernel is full of internal commands and a few are mentioned in Lamport. There is no place where you can go to get explanations of all the LaTeX commands, but if you run LaTeX on the <code>source2e.tex</code> file which is in the standard LaTeX distribution you will get the commented kernel code. The index of the commands runs to about 40 double column pages. Each class and package introduce new commands over and above those in the kernel.

LaTeX includes \newcommand, \providecommand and \renewcommand as means of (re)defining a command, but TeX provides only one method.

```
\def\langle cmd\rangle\langle arg\text{-}spec\rangle\{\langle text\rangle\}
```

\def specifies that within the local group the command \cmd is defined as  $\langle text \rangle$ , and any previous definitions of  $\langle cmd \rangle$  within the group are overwritten. Neither the  $\langle text \rangle$  nor any arguments can include an end-of-paragraph. The LaTeX equivalent to \def is the pair of commands \providecommand\* followed by \renewcommand\*.

The  $\langle arg\text{-}spec \rangle$  is a list of the argument numbers (e.g., #1#2) in sequential order, the list ending at the '{' starting the  $\langle text \rangle$ . Any spaces or other characters in the argument list are significant. These must appear in the actual argument list when the macro is used.

```
\label{long global} $$ \left( \frac{cmd}{arg-spec} \right) \left( \frac{text}{s} \right) $$ \left( \frac{cmd}{arg-spec} \right) \left( \frac{text}{s} \right) $$ \left( \frac{cmd}{arg-spec} \right) \left( \frac{text}{s} \right) $$ $$
```

If you use the  $\log qualifier before \def (as \long \def...)$  then the  $\langle text \rangle$  and arguments may include paragraphs. The LaTeX version of this is the unstarred  $\providecommand$  followed by  $\providecommand$ .

To make a command global instead of local to the current group, the  $\global\qualifier$  can be used with  $\def$  (as  $\global\def...$ ) when defining it;  $\gdef$  is provided as a shorthand for this common case.

Normally any macros within the replacement  $\langle text \rangle$  of a command defined by \def are expanded when the command is called. The macro \edef also defines a command but in this case any macros in the replacement  $\langle text \rangle$  are expanded when the command is defined. Both \long and \global may be used to qualify \edef, and like \gdef being shorthand for \global\\def, \xdef is short for \global\\edef.

There is much more to the \def family of commands than I have given; consult elsewhere for all the gory details.

The \let macro gives \( \lambda cmda \rangle \) the same definition as \( \lambda cmdb \rangle \) at the time the \let is called. The = sign is optional. \let is often used when you want to save the definition of a command. Here is a short example of how some of \def and \let work.

```
\def\name{Alf}
\let\fred = \name
\name, \fred. % prints Alf, Alf.
```

```
\label{eq:csname} $$ \operatorname{string} \rightarrow \
```

If you have ever tried to define commands like  $\cmd1$ ,  $\cmd2$  you will have found that it does not work. TeX command names consists of either a single character or a name composed solely of what TeX thinks of as alphabetic characters. However, the  $\cmd2$  you will have found that it does not work. TeX command names consists of either a single character or a name composed solely of what TeX thinks of as alphabetic characters. However, the  $\cmd2$  you will have found that it does not work. TeX to make the composed solely of what TeX thinks of as alphabetic characters. However, the  $\cmd2$  you will have found that it does not work. TeX to make the composed solely of what TeX thinks of as alphabetic characters. However, the  $\cmd2$  you will have found that it does not work.

\csname cmd1\endcsname

Note that the resulting \cmd1 is not defined (as a macro).

```
\label{eq:constraint} $$ \operatorname{\mathcal{S}} \ \Omega_{nameuse}(string) $$
```

The kernel \@namedef macro expands to \def\<string>, where \string> can contain any characters. You can use this to define commands that include non-alphabetic characters. There is the matching \@nameuse macro which expands to \<string> which then lets you use command names that include non-alphabetic characters. For example:

```
\@namedef{fred2}{Frederick~II}
...
\makeatletter\@nameuse{fred2}\makeatother reigned from ...
```

At any point in its processing TeX is in one of six *modes* which can be categorized into three groups:

- 1. horizontal and restricted horizontal;
- 2. vertical and internal vertical;
- 3. math and display math.

More simply, TeX is in either horizontal, or vertical, or math mode. In horizontal mode TeX is typically building lines of text while in vertical mode it is typically stacking things on top of each other, like the lines making up a paragraph. Math gets complicated, and who can do with more complications at this stage of the game?

With \hbox,  $\langle text \rangle$  is put into a horizontal box, and similarly \vbox puts  $\langle text \rangle$  into a vertical box. The sizes of the boxes depend on the size of the  $\langle text \rangle$ . The optional to  $\langle dimen \rangle$  phrase sets the size of the box to the fixed  $\langle dimen \rangle$  value. If the  $\langle text \rangle$  does not fit neatly inside a fixed size box then TeX will report overfull or underfull warnings. LaTeX supplies the \hb@xt@ command as a shorthand for \hbox to.

Inside a horizontal box TeX is in restricted horizontal mode which means that everything in the box is aligned horizontally. Inside a vertical box TeX is in internal vertical mode and the contents are stacked up and aligned vertically.

```
\dp\langle box\rangle \ht\langle box\rangle \wd\langle box\rangle
```

The depth, height and width of a box are returned by the macros \dp, \ht and \wd respectively.

```
\leavevmode
```

TeX may be in either vertical or horizontal mode and there are things that can be done in one mode while TeX reports an eror if they are attempted in the other mode. When typesetting a paragraph TeX is in horizontal mode. If TeX is in vertical mode, \leavevmode makes it switch to horizontal mode, but does nothing if TeX is already in horizontal mode. It is often used to make sure that TeX is in horizontal mode when it is unclear what state it might be in.

### E.5 Calculation

LaTeX provides some methods for manipulating numbers and these, of course, are composed from TeX's more basic methods. Sometimes it is useful to know what TeX itself provides. We have met most, if not all, of LaTeX's macros earlier but I'll collect them all here for ease of reference.

### E.5.1 Numbers

In LaTeX a counter is used for storing an integer number.

A new counter called  $\langle counter \rangle$ , without a backslash, is created using \newcounter. Its value can be set to a  $\langle number \rangle$  by the \setcounter command and \stepcounter increases its value by one. If the counter is to be used as the basis for a \label, its value must be set using \refstepcounter, neither \stepcounter nor \setcounter will work as expected in this case.

Internally, a LaTeX *counter* is represented by a TeX *count* — the \newcounter macro creates a TeX count named  $\c @\langle counter \rangle$ , and the other \...counter macros similarly operate on the  $\c @\langle counter \rangle$  count.

```
\newcount\langle count \rangle
```

The TeX \newcount command creates a new count,  $\langle count \rangle$ , which does include an initial backslash. For example

\newcount\mycount

TeX's method of assigning a number to a count uses nothing like \setcounter.

```
\langle count \rangle [ = ] \langle number \rangle
```

The [ and ] enclosing the = sign are there only to indicate that the = sign is optional. For example:

```
\mycount = -24\relax % \mycount has the value -24 \mycount 36\relax % now \mycount has the value 36
```

\m@ne	-1	\z@	0	\@ne	1
\tw@	2	\thr@@	3	\sixt@@n	16
\@xxxii	32	\@cclv	255	\@cclvi	256
∖@m	1000	\@Mi	10001	\@Mii	10002
\@Miii	10003	\@Miv	10004	\@MM	20000

Table E.1: Some internal macros for numbers

I have added \relax after the digits forming the number for safety and efficiency. When TeX is reading a number it keeps on looking until it comes across something that is not part of a number. There are things that TeX will treat as part of a number which you might not think of, but \relax is definitely not part of a number. See, for example, [Eij92, chapter 7] for all the intricate details if you need them.

There are some numbers that are used many times in the LaTeX kernel and class codes. To save having to use \relax after such numbers, and for other reasons of efficiency, there are commands that can be used instead of typing the digits. These are listed in Table E.1. The command \z@ can be used both for the number zero and for a length of 0pt. Do not use the commands to print a number.

TeX has a limited vocabulary for arithmetic. It can add to a count, and can multiply and divide a count, but only by integers. The result is always an integer. This may be disconcerting after a division where any remainder is discarded. The syntax for these operations is:

```
\label{local_count} $$ \left| by \right| \langle number \rangle $$ \\ \ \ \left| by \right| \langle number \rangle $$ \\ \ \ \left| by \right| \langle number \rangle $$ \\ \ \ \ \left| by \right| \langle number \rangle $$ \\
```

The by is a TeX keyword and the brackets are just there to indicate that it can be missed out. Some examples:

The value of a count can be typeset by prepending the count by the  $\t$  command, e.g.,  $\t$ 

### E.5.2 Lengths

Every length has an associated unit. For convenience I'll use 'dimension' as shorthand for a number and a length unit.

```
| dimension: \langle number \rangle \langle length-unit \rangle
```

For example, a *dimension* may be 10pt, or 23mm, or 1.3pc.

Unlike LaTeX, TeX distinguishes two kinds of lengths. A TeX \dimen is a length that is fixed; in LaTeX's terms it is a *rigid* length. On the other hand a TeX \skip is a length that may stretch or shrink a little; it is what LaTeX calls a *rubber* length.

The TeX macros \newdimen and \newskip are used for creating a new  $\langle dimen \rangle$  or a new  $\langle skip \rangle$ . For instance:

```
\newdimen\mydimen
\newskip\myskip
```

The value of a \dimen is a *dimension* and the value of a \skip is what TeX calls *glue*. It so happens that LaTeX's \newlength always creates a new skip — all LaTeX lengths are created as rubber lengths. Glue has at least one and possibly as many as three parts.

```
glue: dimension [ plus dimension ] [ minus dimension ]
```

The optional plus part is the amount that the glue can stretch from its normal size and the optional minus part is the amount the glue can shrink below its normal size. Both plus and minus are TeX keywords. Glue can never shrink more than the minus *dimension* and it normally does not stretch more than the plus *dimension*.

```
\@plus \@minus
```

LaTeX supplies \@plus and \@minus which expand to plus and minus respectively. Writing \@plus instead of plus uses one instead of four tokens, saving three tokens, and \@minus in place of minus saves four tokens — remember that a TeX token is either a control sequence (e.g. \@minus) or a single character (e.g., m). TeX's memory is not infinite — it can only hold so many tokens — and it makes sense for kernel and class or package writers to use fewer rather than more to leave sufficient space for any that authors might want to create.

In TeX, assigning a value to a length (\dimen or \skip) is rather different from the way it would be done in LaTeX.

```
\langle dimen \rangle [ = ] \langle dimension \rangle
\langle skip \rangle [ = ] \langle glue \rangle
```

The [ and ] enclosing the = sign are there only to indicate that the = sign is optional. For example:

Like counts, the value of a length can be typeset by prepending the length by the \the command, e.g., \the\myskip.

TeX's lengths can be manipulated in the same way as a count, using the \advance, \multiply and \divide macros. Ignoring some details, lengths can be added together but may only be multiplied or divided by an integer number.

A length can be multiplied by a fractional number by prepending the length with the number. For example:

```
\triangleright \Wdimen = 0.5\Wdimen \Rightarrow Wdimen = 12.64705pt \triangleright \Wskip = 0.5\Wskip \Rightarrow Wskip = 7.64705pt
```

When \multiply or \divide is applied to a \skip all its parts are modified, both the fixed part and any elastic components. However, if a \skip is multiplied by a fractional number then it loses any elasticity it might have had. In the same vein, if a \skip is added to a \dimen any elasticity is lost. A \skip can be coerced into behaving like a \dimen but a \dimen is always rigid. For example, typing

'\Wdimen = 10pt plus 2pt minus 1pt' results in: 'plus 2pt minus 1pt'.

```
\label{lem:lemma} \verb|\newlength{\{\langle len\rangle\}}|
```

LaTeX's \newlength macro creates a new rubber length (internally it uses \newskip); there is no LaTeX specific macro to create a rigid length (i.e., a \dimen).

LaTeX has a variety of macros for setting or changing its length values.

```
\left( \left( len \right) \right) \left( \left( len \right) \right)
```

The LaTeX \setlength macro assigns the value  $\langle glue \rangle$  to the rubber length  $\langle len \rangle$ . Some examples of this are:

As shown in the last example above where both mm and pt are used as a length unit, the \the applied to a length always prints the value in pt units.

```
\label{eq:local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_local_
```

These put the  $\langle text \rangle$  into a box and then set the  $\langle len \rangle$  to the width, height and depth respectively of the box.

```
\label{lem:lemma} $$ \addtolength{\langle len\rangle}{\langle glue\rangle}$
```

LaTeX's \addtolength macro is the equivalent of TeX's \advance command. There are no equivalents to TeX's \multiply or \divide but in any case a length can still be multiplied by prepending it with a fractional number.

```
\z@
fil fill filll
```

\z@ is a very useful LaTeX command when specifying lengths. Depending on the context it either stands for the number 0 (zero) or 0pt (zero length). TeX has three kinds of infinitely stretchy length units that can be used in the plus or minus parts of a skip. fil is infinitely more flexible than any fixed amount, but fill is infittly more flexible than fil and fill1 is infinitely more flexible than anything else at all. These infinite glues can be used to push things around.

```
\hskip\langle skip
angle \ \norm{\label{eq:linear_skip}}
```

The TeX command \hskip inserts  $\langle skip \rangle$  horizontal space and likewise \vskip inserts  $\langle skip \rangle$  vertical space.

```
\hfil \hfill \hfilneg \hss
```

These commands are all TeX primitives and are equivalent to horizontal skips with some kind of infinite glue, as indicated below (note the use of fil as a length unit, it being preceded by a number):

```
\vfil\vfill\vfilneg\vss
```

These commands are all TeX primitives and are equivalent to vertical skips with some kind of infinite glue, as indicated below:

### E.6 Programming

One of the commonest programming operations is to possibly do one thing if something is true and to possibly do another thing if it is not true. Generally speaking, this is called an 'if-then-else' or *conditional* statement.

```
\if... \langle test \rangle \text \langle [\else \langle false-text \rangle ]\fi
```

TeX has several kinds of 'if-then-else' statements which have the general form shown above. The statement starts with an \if... and is finished by a matching \fi. As usual, the brackets enclose optional elements, so there need be no \else portion. The  $\langle true-text \rangle$ , it it exists, is processed if the  $\langle test \rangle$  is true otherwise the  $\langle false-text \rangle$ , if both the \else clause and  $\langle false-text \rangle$  are present, is processed.

The simplest kind of  $\setminus if...$  is defined by the  $\setminus newif$  macro.

```
\newif\if\(name\)
```

\newif creates three new commands, the \ifname and the two declarations, \nametrue and \namefalse, for setting the value of \ifname to true or false respectively. In this case the  $\langle test \rangle$  is embedded in the \ifname. For example:

```
\newif\ifpeter
...
\ifpeter
My name is Peter.
\else
Call me Ishmael.
\fi
or a more likely scenario is
\newif\ifmine
\minetrue % or \minefalse
\newcommand{\whose}{%
\ifmine It's mine. \else I don't know whose it is. \fi}
Here are some of the other more commonly used kinds of ifs.
```

The  $\langle rel \rangle$  in \ifnum and \ifdim is one of the three characters: < (less than), = (equals), or > (greater than). \ifdim results in true if the two lengths are in the stated relationship otherwise it results in false. Similarly \ifnum is for the comparison of two integers. The \ifodd test is true if the integer  $\langle number \rangle$  is an odd number, otherwise it results in false.

Among other things, the LaTeX class code that organizes the page layout checks if the length values are sensible. The following code is a snippet from the layout algorithm. It checks that the sum of the margins and the width of the typeblock is the same as the width of the page after trimming. \Otempdima and \Otempdimb are two 'scratch' lengths used in many calculations.

```
\@tempdimb= -1pt % allow a difference of 1pt \@tempdima=\paperwidth % paperwidth
```

Changing the subject, on the offchance that you might want to see how the Fibonacci sequence progresses, the first thirty numbers in the sequence are: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, 514229 and 832040. I got LaTeX to calculate those numbers for me, and it could have calculated many more. They were produced by just saying \fibseries{30}. The French mathematician Édouard Lucas (1842–1891) studied sequences like this and was the one to give it the name Fibonacci. Lucas also invented the game called the Tower of Hanoi with Henri de Parville (1838–1909), supplying the accompanying fable [dP84, RBC74]:

In the great temple at Benares beneath the dome that marks the center of the world, rests a brass plate in which are fixed three diamond needles, each a cubit high and as thick as the body of a bee. On one of these needles, at the creation, God placed sixty-four discs of pure gold, the largest disc resting on the brass plate, and the others getting smaller and smaller up to the top one. This is the tower of Bramah. Day and night unceasingly the priests transfer the discs from one diamond needle to another according to the fixed and immutable laws of Bramah, which require that the priest on duty must not move more than one disc at a time and that he must place this disc on a needle so that there is no smaller disc below. When the sixty-four discs shall have been thus transferred from the needle which at creation God placed them, to one of the other needles, tower, temple, and Brahmins alike will crumble into dust and with a thunderclap the world will vanish.

The number of separate transfers of single discs is  $2^{64}-1$  or just under eighteen and a half million million moves, give or take a few, to move the pile. At the rate of one disc per second, with no mistakes, it would take more than 58 million million years before we would have to start being concerned.

In his turn, Lucas has a number sequence named after him. There are many relationships between the Fibonacci numbers  $F_n$  and the Lucas numbers  $L_n$ , the simplest, perhaps, being

$$L_n = F_{n-1} + F_{n+1} (E.1)$$

$$5F_n = L_{n-1} + L_{n+1} (E.2)$$

The first 15 numbers in the Lucas sequence are: 2, 1, 3, 4, 7, 11, 18, 29, 47, 76, 123, 199, 322, 521 and 843. These were produced by saying \gfibseries{2}{1}{15}. The Lucas numbers are produced in the same manner as the Fibonacci numbers, it's just the starting pairs that differ.

However, it is the definition of the \fibseries and \gfibseries macros that might be more interesting in this context.

First, create four new counts. \fibtogo is the number of terms to be calculated, \fib is the current term, and \fibprev and \fibprevprev are the two prior terms.

```
\newcount\fib
\newcount\fibprev
\newcount\fibprevprev
\newcount\fibtogo
```

The argument to \fibseries is the number of terms. The counts \fibprevprev and \fibprev are set to the starting pair in the sequence. Provided the number of terms requested is one or more the macro \Ofibseries is called to do the work.

```
\newcommand*{\fibseries}[1]{%
 \fibprevprev=1\relax
 \fibprev=1\relax
 \ifnum #1>0\relax
   \@fibseries{#1}%
 fi
```

The macro \Ofibseries calculates and prints the terms.

```
\newcommand*{\@fibseries}[1]{%
 \fibtogo=#1\relax
```

It's simple if no more than two terms have been asked for — just print them out.

```
\ifnum \fibtogo=\@ne
  \the\fibprevprev
\else
  \ifnum \fibtogo=\tw@
    \the\fibprevprev{} and \the\fibprev
 \else
```

Three or more terms have to be calculated. We reduce the number to be calculated by 2, and print the first two terms.

```
\advance\fibtogo by -\tw@
\the\fibprevprev, \the\fibprev
```

We now have to calculate the rest of the terms, where each term is the sum of the two previous terms. The macro \@fibnext calculates the next term, prints it out and reduces the number of terms left to be calculated (\fibtogo) by one. If there are terms left to be done then the process is repeated until they have all been printed.

```
\loop
      \@fibnext
    \ifnum \fibtogo>\z@
    \repeat
  \fi
\fi}
```

The \@fibnext macro calculates a term in the series, uses \printfibterm to print it, and decrements the \fibtogo count.

```
\newcommand*{\@fibnext}{%
```

```
\fib=\fibprev
\advance\fib by \fibprevprev
\fibprevprev=\fibprev
\fibprev=\fib
\printfibterm
\advance\fibtogo \m@ne}
```

The last of the macros, \printfibterm, typesets a term in the sequence. If the term is the last one print an 'and' otherwise print a ',', then a space and the term.

```
\newcommand*{\printfibterm}{%
  \ifnum \fibtogo=\@ne \space and \else , \fi
  \the\fib}
```

You have met all of the macros used in this code except for TeX's \loop construct. I find the syntax for this a little unusual.

```
\langle text1 \rangle  \if... \langle text2 \rangle  \repeat
```

The construct starts with \loop and is ended by \repeat; the \if... is any conditional test, but without the closing \fi. TeX processes  $\langle text1 \rangle$ , then if the \if... is true it processes  $\langle text2 \rangle$  and repeats the sequence again starting with  $\langle text1 \rangle$ . On the other hand, as soon as the result of the \if... is false the loop stops (i.e., TeX jumps over  $\langle text2 \rangle$  and goes on to do whatever is after the \repeat).

The \gfibseries macro that I used for the Lucas numbers is a generalisation of \fibseries, where the first two arguments are the starting pair for the sequence and the third argument is the number of terms; so \gfibseries{1}{1}\{...\} is equivalent to \fibseries{...\}.

```
\newcommand*{\gfibseries}[3]{%
  \fibprevprev=#1\relax
  \fibprev=#2\relax
  \ifnum #3>0\relax
  \@fibseries{#3}%
  \fi}
```

The calculation of the terms in the Fibonacci and in the generalised sequences is the same so \Ofibseries can be used again.

I used the TeX  $\loop$  construct in the  $\ensuremath{\texttt{Qfibseries}}$  macro but LaTeX has a similar construct.

As long as the appropriate  $\langle \textit{test} \rangle$  is true the  $\langle \textit{body} \rangle$  is processed.

In  $\ensuremath{\texttt{Ofibseries}}$  I used  $\ensuremath{\texttt{Inms}}$  to check for 3 possible values. There is another  $\ensuremath{\texttt{If}}\dots$  form that can be used for this type of work.

```
\if case \langle number \rangle \langle text \ for \ 0 \rangle \setminus (text \ for \ 1 \rangle \setminus (text \ for \ 2 \rangle 
...
\or \langle text \ for \ N \rangle  [\else \langle text \ for \ anything \ else \rangle]\fi
```

If the  $\langle number \rangle$  is 0 then  $\langle text \ for \ 0 \rangle$  is processed, but if  $\langle number \rangle$  is 1 then  $\langle text \ for \ 1 \rangle$  is processed, but if  $\langle number \rangle$  is ... Each  $\langle text \ for \ ... \rangle$  is separated by an  $\backslash or$ . If  $\langle number \rangle$  is anything

other than the specified cases (i.e., less than zero or greater than N) then if the  $\ensuremath{\mbox{\mbox{\mbox{less}}}}$  is present  $\ensuremath{\mbox{\s\m\s$ 

Here's another version of the \@fibseries macro using \ifcase and \@whilenum.

```
\renewcommand*{\@fibseries}[1]{%
  \fibtogo=#1\relax
  \ifcase \fibtogo % ignore 0
  \or % \fibtogo=1
   \the\fibprevprev
  \or % \fibtogo=2
   \the\fibprevprev{} and \the\fibprev
  \else % fibtogo > 2
   \advance\fibtogo by -\tw@
   \the\fibprevprev, \the\fibprev
   \@whilenum \fibtogo > \z@ \do {% must kill space after the {
    \@fibnext}%
  \fi}
```

TeX has more programing constructs than I have shown here and these will be explained in any good TeX book. LaTeX also has more than I have shown but in this case the best place to look for further information is in the LaTeX kernel code, for example in ltcntrl.dtx.

### The terrors of errors

No matter how conscientious you are a mistake or two will occasionally creep into your document source. The good news is that whatever happens TeX will not destroy your files — it may produce some odd looking output, or even no output at all, but your work is safe. The bad news is that you have to correct any errors that TeX finds. To assist you in this TeX stops whenever it comes across what it thinks is an error and tells you about it. If you're not sure what to do it will also provide some possibly helpful advice.

TeX underlies LaTeX which underlies classes and packages. You may get messages than originate from TeX, or from LaTeX, or from the class and any packages you may be using. I'll describe the TeX, LaTeX, and class messages below.

In general, you will see a message on your terminal and LaTeX will stop and wait for you to respond. It prints a question mark and is expecting you to type one of the following:

- ⟨*return*⟩ (or ⟨*enter*⟩ or what is the equivalent on your keyboard): LaTeX will continue processing the document.
- H (help): the help message is output and LaTeX waits for you to respond again.
- S (scroll): Continue processing, outputting any further error messages, but not stopping.
- Q (quiet): Continue processing without stopping and with no further messages.
- R (run): Like the Q option but not even stopping if your document requires some user input.
- I (insert): To insert some material for TeX to read but no changes are made to the source file.
- E (edit): This may return you to an editor so you can change the file. What actually happens is system dependent.
- X (exit): Stop this LaTeX run.

On the system I am used to the case of the characters does not matter. I must admit that the only ones I have used are  $\langle return \rangle$ , q, h and x, in approximately that order of frequency. All messages are output to the log file so you can study them later if you need to.

### F.1 TeX messages

The following is an alphabetical list of some of TeX's messages, abbreviated in some cases, together with their corresponding remarks. As an example of how these appear on your terminal, if you had a line in your source that read:

```
resulting in $x^3^4$. then TeX would output this:
```

```
! Double superscript
1.102 resulting in $x^3^
4^$.
```

If you typed h in response to this you would then see:

```
I treat 'x^1^2' essentially like 'x^1{}^2'.
```

TeX's messages start with ! followed by the particular message text. The second line starts 1. and a number, which is the number of the line in your file where the error is. This is followed by the text of the line itself up to the point where the error was detected, and the next line in the report shows the rest of the erroneous line. The last line of the report is a ? and TeX awaits your response.

In the listing I have used this font for the error message and this font for the comment message.

### ! A box was supposed to be here.

I was expecting to see \hbox or \vbox or \copy or \box or something like that. So you might find something missing in your output. But keep trying; you can fix this later.

```
! Argument of ... has an extra }.
```

I've run across a '}' that doesn't seem to match anything. For example, '\def\a#1{...}' and '\a}' would produce this error. If you simply proceed now, the \par that I've just inserted will cause me to report a runaway argument that might be the root of the problem. But if your '}' was spurious, just type '2' and it will go away.

```
In LaTeX terms, the example can be translated into \new command \{a\}[1]\{...\}, and \adjustarrow above the command and <math>\adjustarrow above the command and th
```

If you can't find the extra } it might be that you have used a fragile command in a moving argument. Footnotes or math in division titles or captions are a fruitful source for this kind of error. You shouldn't be putting footnotes into titles that will get listed in the ToC. For maths, put \protect before each fragile command.

### ! Arithmetic overflow.

I can't carry out that multiplication or division, since the result is out of range.

The maximum number that TeX can deal with is 2,147,483,647 and it balks at dividing by zero.

### ! Dimension too large.

I can't work with sizes bigger than about 19 feet. Continue and I'll use the largest value I can.

! Display math should end with \$\$.

The '\$' that I just saw supposedly matches a previous '\$\$'. So I shall assume that you typed '\$\$' both times

Although \$\$ is one of TeX's methods for starting and ending display math, do *not* use it in LaTeX.

! Double subscript.

```
I treat 'x_1_2' essentially like 'x_1{}_2'.
```

This would produce  $x_{12}$ . If you were after say,  $x_{23}$  instead, type  $x_{23}$ .

! Double superscript.

```
I treat 'x^1^2' essentially like 'x^1{}^2'.
```

This would produce  $x^{12}$ . If you were after say,  $x^{2^3}$  instead, type  $x^{2^3}$ .

! (\end occurred inside a group at level  $\dots$ ).

This is message is output at the end of a run. It means that you have not ended all the groups that you started; a group can be started by a simple open brace  $(\{)$ , but there are other starting mechanisms as well, such as  $\lceil \{ \dots \} \rceil$ . If the problem is a missing  $\lceil \{ \dots \} \rceil$ , LaTeX is kind enough to tell you what the mismatch is.

- ! (\end occurred when ... was incomplete).
- ! Extra \fi. or Extra \else. or Extra \or.

I'm ignoring this; it doesn't match any \if.

! Extra \endcsname.

I'm ignoring this, since I wasn't doing a \csname.

! Extra \right.

I'm ignoring a \right that had no matching \left.

! Extra }, or forgotten \endgroup, \$, or \right.

I've deleted a group closing symbol because it seems to be spurious, as in 'x'. But perhas the } is legitimate and you forgot something else, as in ' $hbox{$x}$ '. In such cases the way to recover is to insert both the forgotten and the deleted material, e.g., by typing 'x'.

The braces or math mode delimeters didn't match. You might have forgotten a  $\{, \ \ \ \ \ \ \}$ .

! Extra ...

Things are pretty mixed up, but I think the worst is over.

! Extra alignment tab has been changed to \cr.

You have given more \span or & marks than there were in the preamble to the \halign or \valign now in progress. So I'll asume that you meant to type \cr instead.

Internally, LaTeX uses \halign for its array and tabular environments. The message means that you have too many column entries in a row (i.e., too many & before the end of the row). Perhaps you have forgotten to put \\ at the end of the preceding row.

! File ended while scanning ... or Forbidden control sequence found while scanning ...

I suspect you have forgotten a '}', causing me to read past where you wanted me to stop. I'll try to recover; but if the error is serious you'd better type 'E' or 'X' now and fix your file.

- ! Font "not loadable: Metric (TFM) file not found.
- ! Font "not loadable: Bad metric (TFM) file.

I wasn't able to read the size data for this font, so I will ignore the font specification. [Wizards can fix TFM files using TFtoPL/PLtoTF.] You might try inserting a different font spec; e.g., type 'I\font<same font id>=<substitute font name>'.

LaTeX can't find a font you have asked for.

! Huge page cannot be shipped out.

The page just created is more than 18 feet tall or more than 18 feet wide, so I suspect something went wrong.

! I can't find file `...', please type another.

TeX couldn't find the file you asked it to read. You can also get this message with LaTeX if you have missed the braces around the argument to \input.

! I can't go on meeting you like this.

One of your faux pas seems to have wounded me deeply... in fact, I'm barely conscious. Plase fix it and try again.

! I can't write on file `...', please type another.

TeX couldn't write on a file, you might have mispelled the name or not have permission to use it.

! Illegal parameter number in definition of  $\cdots\!.$ 

You meant to type ## instead of #, right? Or maybe a } was forgotten somewhere earlier, and things are all screwed up? I'm going to assume that you meant ##.

This is probably due to a command defining command like \newcommand or \renewcommand or \providecommand, or an environment defining command like \newenvironment or \renewenvironment, where a # has been used incorrectly. Apart from the command \#, a # can only be used to indicate an argument parameter, like #3 which denotes the third argument. You cannot use an argument parameter, like the #3 in the last argument of either the \newenvironment or the \renewenvironment commands.

You get the same error if you try to include any of the above defining commands inside another one.

! Illegal unit of measure (replaced by fill1).  $I \, dddon't \, go \, any \, higher \, than \, filll.$ 

You have tried to use a fill1 with more than 3 '1's.

#### ! Illegal unit of measure (mu inserted).

The unit of measurement in math glue must be mu. To recover gracefully from this error it's best to delete the erroneous units; e.g., type '2' to delete two letters. (See Chapter 27 of The TeXbook.)

TeX was in math mode and expecting a length, which must be in mu units.

#### ! Illegal unit of measure (pt inserted).

Dimensions can be in units of em, ex, in, pt, pc, cm, mm, dd, cc, bp, or sp; but yours is a new one! I'll assume you meant to say pt, for printers' points. To recover gracefully from this error it's best to delete the erroneous units; e.g., type '2' to delete two letters. (See Chapter 27 of The TeXbook.)

TeX was expecting a length but it found just a number without a known length unit. For example you wrote 2ib instead of 2in.

# ! Improper \hyphenation will be flushed.

Hyphenation exceptions must contain only letters and hyphens. But continue; I'll forgive and forget.

# ! Incomplete "all text was ignored after line "...

A forbidden control sequence occurred in skipped text. This kind of error happens when you say ' $\in f$ ..' and forget the matching ' $\fi$ '. I've inserted a ' $\fi$ '; this might work.

# ! Infinite glue shrinkage found in a paragraph.

The paragraph just ended includes some glue that has infinite shrinkability, e.g., '\hskip Opt minus 1fil'. Such glue doesn't belong there—it allows a paragraph of any length to fit on one line. But it's safe to proceed, since the offensive shrinkability has been made finite.

# ! Limit controls must follow a math operator.

*I'm ignoring this misplaced* \limits or \nolimits command.

#### ! Misplaced &. or Misplaced \cr. or Misplaced \span.

I can't figure out why you would want to use a tab mark or \cr or \span here. If you just want an ampersand the remedy is simple: Just type 'I\&' now. But if some right brace up above has ended a previous alignment prematurely, you're probably due for more error messages, and you might try typing 'S' now just to see what is salvageable.

In LaTeX the most likely of these messages is the Misplaced &. You can only use a naked & in environments like array and tabular as column separators. Anywhere else you have to use \&.

#### ! Misplaced \noalign.

*I expect to see* \noalign *only after the* \cr *of an alignment. Proceed, and I'll ignore this case.* 

#### ! Misplaced \omit.

I expect to see \omit only after the tab marks or the \cr of an alignment. Proceed, and I'll ignore this case.

#### ! Missing \cr inserted.

I'm guessing that you meant to end an alignment here.

You might have missed a \\ at the end of the last row of a tabular or array.

# ! Missing = inserted for ....

I was expecting to see '<', '=', or '>'. Didn't.

## ! Missing # inserted in alignment preamble.

There should be exactly one # between &'s, when an \halign or \valign is being set up. In this case you had none, so I've put one in; maybe that will work.

If you get this in LaTeX then there are problems with the argument to an array or tabular.

# ! Missing \$ inserted. or Missing \endgroup inserted. or Missing \right inserted.

I've inserted something that you may have forgotten. (See the <inserted text> above.) With luck, this will get me unwedged, But if you really didn't forget anything, try typing '2' now; then my insertion and my current dilemma will both disappear.

This is a general response to the above messages. There is also a more specific response for each of the messages, as listed below.

## ! Missing \$ inserted.

I've inserted a begin-math/end-math symbol since I think you left one out. Proceed with fingers crossed.

Certain commands can only be executed in math mode and there are others that cannot be used in math mode. TeX has come across a command that cannot be used in the current mode, so it switches into, or out of, math mode on the assumption that that was what you had forgotten to do.

#### ! Missing \endcsname inserted.

The control sequence marked <to be read again> should not appear between \csname and \endcsname.

#### ! Missing { inserted.

A left brace was mandatory here, so I've put one in. You might want to delete and/or insert some corrections so that I will find a matching right brace soon. If you're confused by all this, try typing 'I}' now.

#### ! Missing { inserted.

Where was the left brace? You said something like  $\def\a$ ', which I'm going to interpret as  $\def\a$ '.

In LaTeX terms, the example wrongdoing would be \newcommand{\a}}

# ! Missing { inserted.

I've put in what seems becessary to fix the current column of the current alignment. Try to go on, since this might almost work.

It seems that a { might have been missing in a tabular or array entry.

#### ! Missing control sequence inserted.

Please don't say '\def cs{...}', say '\def\cs{...}'. I've inserted an inaccessible control sequence so that your definition will be completed without mixing me up too badly. You can recover graciously from this error, if you're careful; see exercise 27.2 in The TeXbook.

#### ! Missing delimeter(. inserted).

I was expecting to see something like '(' or '\{' or '\}' here. If you typed, e.g., '{' instead of '\{' you should probably delete the '{' by typing '1' now, so that braces don't get unbalanced. Otherwise just proceed. Acceptable delimeters are characters whose \delcode is nonnegative, or you can use '\delimeter <delimeter code>'.

# ! Missing number, treated as zero.

A number should have been here; I inserted '0'. (If you can't figure out why I needed to see a number, look up 'weird error' in the index to The TeXbook.)

In LaTeX this is often caused by a command expecting a number or a length argument but not finding it. You might have forgotten the argument or an opening square bracket in the text might have been taken as the start of an optional argument. For example, the \\ (newline) command takes an optional length argument, so the following will produce this error:

```
... next line\\
[Horatio:] ...
```

#### ! Not a letter.

Letters in \hyphenation words must have \lccode>0.

One or more characters in the argument to the  $\protect\operatorname{hyphenation}$  command should not be there.

#### ! Number too big.

These all represent the same value, firstly in decimal, secondly in octal, and lastly in hexadecimal notations.

## ! Output loop--- "consecutive dead cycles.

I've concluded that your \output is awry; it never does a \shipout, so I'm shipping \box255 out myself. Next time increase \maxdeadcycles if you want me to be more patient!

TeX appears to be spinning its wheels, doing nothing.

#### ! Overfull \hbox ("pt too wide).

This is a warning that TeX couldn't cram some text into the alloted horizontal space.

### ! Overfull \vbox (...pt too high).

This is a warning that TeX couldn't find a good place for a pagebreak, so it has put too much onto the current page.

#### ! Paragraph ended before ...was complete.

I suspect you've forgotten a '}', causing me to apply this control sequence to too much text. How can we recover? My plan is to forget the whole thing and hope for the best.

Either a blank line or a \par command appeared in the argument to a macro that cannot handle paragraphs (e.g., a macro that was defined using \newcommand\*).

! Please type a command or say `\end'.

This is the message that causes me the most trouble. My computer always ignores whatever I say to it and even typing \end has no effect. What I usually do, after having tried a few variations like \end{document}, is to kill the program by whatever means the operating system provides. Some other possible responses include:

- Type \stop
- Type \csname @Qend\endcsname (LaTeX stores TeX's version of \end as \@Qend)
- Type some macro that you think is unknown, perhaps \qwertyuiod, then respond to the error message: Undefined control sequence.
- Sometimes nothing works except killing the program. If you are are sure you know how to kill a program, try the following highly contrived code:

```
\documentclass{article}
  \newif\ifland
  \newif\ifprint
  \newcommand{\Xor}[2]{\ifx #1 #2}
\begin{document}
% \Xor{\ifland}{\ifprint}% try uncommenting this
  \iffalse
\end{document}
```

- ! Runaway argument. or Runaway definition. or Runaway preamble. or Runaway text.
- ! Sorry, but I'm not programmed to handle this case.

I'll just pretend that you didn't ask for it. If you're in the wrong mode, you might be able to return to the right one by typing 'I}' or 'I\par'.

! TeX capacity exceeded, sorry [...].

If you absolutely need more capacity, you can ask a wizard to enlarge me.

This is dealt with in more detail below.

! Text line contains an invalid character.

A funny symbol that I can't read has just been input. Continue, and I'll forget that it ever happened.

The input file contains a nonprinting (control) character; only printing characters should be in the file. Some programs, like word processors, insert invisible characters into their output file. If you have used one of these to prepare your input file, make sure you save it as a plain text file (also known as an ASCII file).

- ! That makes 100 errors; please try again.
- ! This can't happen (...).

I'm broken. Please show this to someone who can fix can fix

This is the message you should never see!

# ! Too many }'s.

You've closed more groups than you opened. Such booboos are generally harmless, so keep going.

There are more closing braces (}) than there are opening braces ({).

#### ! Unbalanced output routine.

Your sneaky output routine has fewer real {'s than }'s. I can't handle that very well; good luck.

A package or class has done nasty things to one of LaTeX's most delicate parts — the output routine.

#### ! Unbalanced write command.

On this page there's a \write with fewer real {'s than }'s. I can't handle that very well; good luck.

# ! Undefined control sequence.

The control sequence at the end of the top line of your error message was never  $\def'ed$ . If you have misspelled it (e.g., ' $\del{hobx}$ '), type 'I' and the correct spelling (e.g., ' $\del{lhobx}$ '). Otherwise just continue, and I'll forget whatever was undefined.

TeX has come across a macro name that it does not know about. Perhaps you mispelled it, or it is defined in a package you did not include. Another possibility is that you used a macro name that included the @ character without enclosing it between \makeatletter and \makeother (see §E.4). In this case TeX would think that the name was just the portion up to the @.

#### ! Underfull \hbox (badness ...).

This is a warning. There might be some extra horizontal space. It could be caused by trying to use two \newline or \\ commands in succession with nothing intervening, or by using a \linebreak command or typesetting with the \sloppy declaration.

# ! Underfull \vbox (badness ...).

This is a warning that TeX couldn't find a good place for a pagebreak, so it produced a page with too much whitespace on it.

#### ! Use of "doesn't match its definition.

If you say, e.g., ' $\ensuremath{\mbox{def}\al{...}}$ ', then you must always put '1' after ' $\ensuremath{\mbox{a}}$ ', since the control sequence names are made up of letters only. The macro here has not been followed by the required stuff, so I'm ignoring it.

# ! You can't use `...' in `...'.

This often manifests itself in the form

You can't use `\spacefactor' in vertical mode

the cause is usually trying to use a macro with @ in its name, typically in the preamble (see §E.4). The solution is to enclose the macro within \makeatletter and \makeatother.

### Another version is

You can't use `macro parameter character #' in ... mode.

In this case you have used a naked # in ordinary text; it can only be used in the definition of a macro. In ordinary text you have to use \#.

#### F.1.1 TeX capacity exceeded

TeX has run out of computer space before it finished processing your document. The most likely cause is an error in the input file rather than there really not being enough space — I have processed documents consisting of more than 1400 pages without any capacity problems.

You can very easily make TeX run out of space. Try inputting this:

The offending code above tries to define \fred in terms of itself, and TeX just keeps chasing round and round trying to pin down \fred until it is exhausted.

At the end of the log file for a run, TeX prints the memory space it has used. For example:

```
Here is how much of TeX's memory you used:
2432 strings out of 60985
29447 string characters out of 4940048
106416 words of memory out of 8000001
5453 multiletter control sequences out of 10000+65535
8933 words of font info for 31 fonts out of 1000000 for 1000
276 hyphenation exceptions out of 1000
26i,11n,21p,210b,380s stack positions out of
15000i,4000n,6000p,200000b,40000s
```

The error message says what kind of space it exhausted (input stack size in the example above). The most common are:

buffer size Can be caused by too long a section or caption title appearing in the ToC, LoF, etc. Use the optional argument to produce a shorter entry.

exception dictionary There are too many words listed in \hyphenation commands. Remove any that are not actually used and if that doesn't work, remove the less common ones and insert \- in the words in the text.

hash size The document defines too many command names and/or uses too many cross-referencing \labels.

input stack size Typically caused by a self-referencing macro definition.

main memory size There are three main things that cause TeX to run out of main memory:

- Defining a lot of very long complicated macros.
- Having too many \index or \glossary commands on a page.
- Creating such a complicated page that TeX cannot hold all it needs to process it.

The solution to the first two problems is to simplify and eliminate. The third is more problematic.

Large tabulars, arrays and pictures (the \quad qbezier command is a memory hog) can gobble up memory. A queue of floats also demands memory space. Try putting a \clearpage just before the place where the error occurs and if it still runs out of room then there may be an error in your file, otherwise you did exceed the capacity.

If you have a long paragraph or a long verbatim environment try breaking it up, as TeX keeps these in memory until it is ready to typeset them. If you have a queue of floats make sure that you have done your best to help LaTeX find a way to output them (see §10.4) and try adding \clearpage at appropriate places to flush the queue.

pool size Typically caused by having too many characters in command names and label names.

It can also be caused by omitting the right brace that ends the argument of a counter command (\setcounter or \addtocounter) or of a \newenvironment or \newtheorem command.

save stack size This happens if commands or environments are nested too deeply. For instance a picture that contains a picture that includes a \multiput that includes a picture that includes a ···

# F.2 LaTeX errors

LaTeX errors introduce themselves differently from those that TeX finds. For example, if you ever happended to use the \caption command outside a float, like:

```
\caption{Naked}
```

you would get the message:

?

If you then typed H in response you would get the following helpful message:

```
You're in trouble here. Try typing <return> to proceed. If that doesn't work, type X <return> to quit.
```

The majority of LaTeX's help messages follow this formula, so I have not noted them in the alphabetical listing below.

```
\< in mid line</pre>
```

A \< appears in the middle of a line in a tabbing environment; it should only come at the start of a line.

```
... allowed only in math mode
```

You have tried to use a math command in a non-math mode.

```
Bad \line or \vector argument
```

A \line or \vector has a negative length argument or the slope is not within the allowed range.

```
Bad math environment delimeter
```

If in math mode there is a start math mode command like  $\ \$  or  $\$  or if in LR or paragraph mode there is an end math mode command like  $\ \$ ) or  $\$ ]. The basic problem is unmatched math mode delimeters or unbalanced braces.

```
\begin{...} ended by \end{...}
```

The name of the \begin argument is not the same as the name of the \end argument. This could be caused by a typo or a missing \end.

```
Can only be used in the preamble
```

Some commands can only be used in the preamble, such as \usepackage, but there was one of these after the \begin{document}.

```
\caption outside float
```

You have used the \caption command outside a float, such as a figure or table environment.

```
Command \... already defined or name \end... illegal
```

This is normally because you have used one of the <code>\new...</code> commands to define a command or environment or counter name that has already been used; remember also that defining an environment foo automatically defines the macro <code>\foo</code>. Either choose a new name or use the appropriate <code>\renew...</code>; also, see §19.1. In the unlikely event that you have tried to define something beginning with <code>\end...</code>, choose another name.

```
{\tt Command} \ \dots \ {\tt invalid} \ {\tt in} \ {\tt math} \ {\tt mode}
```

You have used a non-math command in math mode.

# Command ... not provided in base LaTeX2e

You have tried to use a symbol that is not part of basic LaTeX. Try loading the latexsym or amsfonts package which might define the symbol.

#### Counter too large

You are using a non-numeric counter representation, such as letters or footnote symbols, and the counter has exceeded the allowed number (for example there are only 26 alphabetic characters).

#### Environment ... undefined

LaTeX does not know the name of the argument of a \begin. You have probably misspelled it.

File not found. Type X to quit or <RETURN> to proceed or enter new name (Default extension: ...)

LaTeX cannot find the file you requested. The extension tex results from a problematic \input or \include; the extension sty from a \usepackage and an extension cls f

#### Float(s) lost

Usually caused by having too many \marginpars on a page.

## Illegal character in array argument

There is an illegal character in the argument of an array or tabular environment, or in the second argument of a \multicolumn command.

#### \include cannot be nested

A file that is \included cannot \include any other files.

#### \LoadClass in package file

This is an error in a package file you are using (you can only use \LoadClass in a class file). Complain to the author.

# Lonely \item --- perhaps a missing list environment

An \item command appears to be outside any list environment.

#### Missing \begin{document}

If you haven't forgotten \begin{document} then there is something wrong in the preamble as LaTeX is trying to typeset something before the document starts. This is often caused by missing the backslash from a command, misplaced braces round an argument, a stray character, or suchlike.

#### Missing @-exp in array argument

The @ character is not followed by an @-expression in the argument of an array or tabular environment, or in the second argument of a \multicolumn command.

#### Missing p-arg in array argument

There is a p not followed by braces in the argument of an array or tabular environment, or in the second argument of a \multicolumn command.

#### No counter ... defined

The argument to a \setcounter or \addtocounter command, or in the optional argument to \newcounter or \newtheorem is not the name of a counter. Perhaps you misspelled the name. However, if the error occured while an aux file was being read then you might well have used a \newcounter in an \included file.

#### No room for a new ...

TeX is limited in the numbers of different things it can handle. You might not recognize the thing that the message mentions as some of them are hidden in LaTeX. The LaTeX counter uses a TeX \count for example, and a length is a TeX \skip. Most things are limited to a maximum of 256 but there can be no more than 16 files open for reading and 16 for writing.

### No \title given

You did not put a \title command before using \maketitle.

#### Not in outer par mode

There is a float (e.g., a figure or a \marginpar) in math mode or in a parbox (e.g., in another float).

#### Option clash for ...

The same package was used twice but with different options. It is possible for one package to use another package which might be the cause if you can't see anything obvious.

#### Page height already too large

You are trying to use \enlargethispage when the page is already too large.

# \pushtabs and \poptabs don't match

There are unmatched \pushtabs and \poptabs in a tabbing environment.

#### \RequirePackage or \LoadClass in Options Section

This is a problem in a class or package file. Complain to the author.

# Something's wrong --- perhaps a missing \item

This can be caused by not starting a list environment, such as itemize with a \item command, or by omitting the argument to the thebibliography environment. There are many other non-obvious causes, such as calling some macro that ends up using \addvspace or \addpenalty when not in vmode.

# Suggested extra height (...) dangerously large

LaTeX is concerned that you a trying to increase the page size too much with the \enlargthispage command.

#### Tab overflow

In the tabbing environment a \= has exceeded LaTeX's maximum number of tab stops.

```
The file needs format ... but this is ...
```

The document uses a document class or package that is not compatible with the version of LaTeX you are using. If you are using only standard files then there is a problem with your LaTeX installation.

#### There's no line to end here

A \newline or \\ appears in vertical mode, for example between paragraphs. Or perhaps you have tried to put \\ immediately after an \item to start the text on a new line. If this is the case, then try this:

```
\item \mbox{} \\
...
```

### This may be a LaTeX bug

This is a message you don't want to see as it is produced by the output routine — perhaps the most obscure part of LaTeX. It is probably due to an earlier error. If it is the first error, though, and you can't see anything wrong, ask for somebody's help.

#### Too deeply nested

There are too many list environments nested within each other. At least four levels are usually available but some list environments are not obvious (for example the quotation environment is actually a list).

#### Too many columns in eqnarray environment

An eqnarray environment has three & column separators with no \\ between.

#### Too many unprocessed floats

There may be too many \marginpars to fit on a page, but it's more likely that LaTeX hasn't been able to find locations for printing all the figures or tables. If one float cannot be placed, all later ones are saved until LaTeX runs out of storage space. See §10.4 for details on how LaTeX decides to place a float.

#### Two \documentclass commands

Your document has two \documentclass commands; only one is permitted.

# Two \LoadClass commands

This is an error in the class file. Complain to the author.

# Undefined tab position

A  $\$ ,  $\$ -, or  $\$  tabbing command is trying to move to a tab position that has not been defined by a  $\$ - command.

Unknown option ... for class/package ...

You have asked for an option that the class or package does not know about. Perhaps you have mispelled something, or omitted a comma.

\usepackage before \documentclass

In general, the \usepackage command can only be used in the preamble.

\verb ended by end of line

The argument of a \verb command runs past the end of the line. Perhaps you forgot to put in the correct ending character.

\verb illegal in command argument

A \verb cannot be part of the argument to another command.

## F.3 LaTeX warnings

Most warnings are given at the point in the document where a potential problem is discovered, while others are output after the document has been processed.

For example, the following code

```
... \ref{joe}... \cite{FRED96} ...
```

may produce warnings like

Latex Warning: Reference 'joe' on page 12 undefined

on input line 881.

Latex Warning: Citation 'FRED96' on page 12 undefined

at lines 890--897.

during the document processing, and then at the end there will also be the warning:

LaTeX Warning: There were undefined references.

Some warning messages pinpoint where a problem might lie, as in the citation warning above, while others make no attempt to do so. In the alphabetical listing that follows I have not included such information, even if it is supplied.

```
Citation ... on page ... undefined
```

The key in a \cite command was not defined by any \bibitem.

Citation ... undefined

The key in a \cite command was not defined by any \bibitem.

```
Command ... invalid in math mode
```

The command is not permitted in math mode but was used there anyway. Remember that font size commands and \boldmath or \unboldmath cannot be used in math mode.

Float too large for page by ...

A float (table or figure) is too tall to fit properly on a page by the given amount. It is put on a page by itself.

Font shape ... in size ... not available size ... substituted

You asked for a font size that was not available. The message also says what font is being used instead.

Font shape ... undefined using ... instead

You asked for a font shape that was not available. The message also says what font is being used instead.

h float specifier changed to ht or !h float specifier changed to !ht

A float has an optional h or !h argument but as it wouldn't fit on the curent page it has been moved to the top pf the next page.

Label ... multiply defined

Two \label or \bibitem commands have the same argument (at least during the previous LaTeX run).

Label(s) may have changed. Rerun to get cross-references right

This is only output at the end of the run.

One of the numbers printed by \cite, \ref or \pageref commands might be incorrect because the correct values have changed since the preceding LaTeX run.

Marginpar on page ... moved

A \marginpar was moved down the page to avoid overwriting an earlier one. The result will not be aligned with the \marginpar call.

No \author given

There is no \author command before calling \maketitle.

No positions in optional float specifier. Default added (so using `tbp')

You have used an empty optional argument to a float, for example:

\begin{figure}[]

so it has used

\begin{figure}[tbp]

instead.

Optional argument of \twocolumn too tall on page ...

The contents of the optional argument to \twocolumn was too long to fit on the page.

\oval, \circle, or \line size unavailable

You have asked for too large (or too small) an oval or circle, or too short a line, in a picture.

Reference ... on page ... undefined

The argument of a \ref or \pageref has not been defined on the preceding run by a \label command.

Size substitutions with differences up to ... have occured. Please check the transcript file carefully and redo the format generation if necessary!

This is only output at the end of the run.

Some fonts have had to be used as substitutes for requested ones and they are a different size.

Some shapes were not available, defaults substituted

This is only output at the end of the run.

At least one font had to be substituted.

Text page ... contains only floats

The page should have included some textual material but there was no room for it.

There were multiply defined labels

This is only output at the end of the run.

Two or more \label or \cite commands had the same argument.

There were undefined references

This is only output at the end of the run.

There was at least one \ref or \pageref or \cite whose argument had not been defined on the preceding run by a \label or \biblabel command.

Unused global option(s) [...]

The listed options were not known to the document class or any packages you used.

You have requested release ... of LaTeX but only release ... is available You are using a class or package that requires a later release of LaTeX than the one you are using. You should get the latest release.

You have requested version ... of class/package ... but only version ... is available

You (or the class or one of the packages you are using) needs a later release of a class or package than the one you are using. You should get the latest release.

#### F.4 Class errors

The class errors introduce themselves differently from those that LaTeX finds. Instead of starting with

! LaTeX Error:

the class errors start with

! Class memoir Error:

After that, it is indistinguishable from a LaTeX error. For example, if you ever happened to input the next line as line 954 in your document you would get the error message that follows

\sidecapmargin{either}

! Class memoir Error: Unrecognized argument for \sidecapmargin.

See the memoir class documentation for explanation.

Type H <return> for immediate help.

. . .

1.954  $\sidecapmargin{either}$ 

?

If you then typed H (or h) in response you would get the following helpful message:

```
Try typing <return> to proceed. If that doesn't work, type X < \text{return> to quit.}
```

The majority of the help messages follow this formula, so I have not noted them in the alphabetical listing below.

... is negative

The value is negative. It should be at least zero.

... is not a counter

An argument that should be the name of a counter is not.

... is zero or negative

The value must be greater than zero.

>{...} at wrong position: token ignored

A >{...} in the argument to an array or tabular is incorrectly placed and is being ignored.

<{...} at wrong position: changed to !{...}

A  $\{\ldots\}$  in the argument to an array or tabular is incorrectly placed. It has been changed to  $!\{\ldots\}$  instead.

A pattern has not been specified

You are trying to use the patverse or patverse\* environment without having first defined a pattern.

Argument to \setsidecappos is not t or c or b

The argument will be assumed to be c.

Argument to \overridesidecapmargin neither left nor right

The argument to \overridesidecapmargin must be either left or right. The attempted override will be ignored.

Cannot change a macro that has delimited arguments

You are using patchcmd on a macro that has delimted arguments.

Empty preamble: 'l' used

The argument to an array or tabular is empty. The specification {1} is being used instead.

Font command ... is not supported

You have tried to use a deprecated font command. Either replace it with the current font command or declaration or use the oldfontcommands class option.

\footskip is too large for \lowermargin by  $\dots$ 

The \footskip is too large for the \lowermargin. Either increase the \lowermargin or decrease the \footskip.

\headheight and/or \headsep are too large for \uppermargin by ...

The sum of the \headheight and the \headsep is larger than the \uppermargin. Either increase the \uppermargin or reduce the others.

Illegal pream-token (...): `c' used

An illegal character is used in the argument to an array or tabular. The 'c' specifier is being used instead (which centers the column).

Index ... outside limits for array ...

Trying to access an index for the array data structure that is not between the low and high indices.

Limits for array  $\dots$  are in reverse order

The low index is not less than the high index in \newarray.

Missing arg: token ignored

The argument to a column specifier for a array or tabular is missing.

No array called ...

You have tried to access an unknown array data structure.

Not defined: ...

You are using \patchcmd on a macro that is not defined.

Not redefinable: ...

You are using \patchcmd on a macro that it is unable to modify.

Only one column-spec. allowed

There can only be one column specifier in a \multicolumn.

Optional argument is not one of: classic, fixed, lines, or nearest. I will assume the default.

You have provided an unknown name for the optional argument to \checkthelayout. The default classic will be used instead.

\paperheight and/or \trimtop are too large for \stockheight by ...

The sum of the \paperheight and the \trimtop is larger than the \stockheight. Either increase the \stockheight or reduce the others.

\paperwidth and/or \trimedge are too large for \stockwidth by ...

The sum of the \paperwidth and the \trimedge is larger than the \stockwidth. Either increase the \stockwidth or reduce the others.

\spinemargin and/or \textwidth and/or \foremargin are too large for \paperwidth by ...

The sum of the \spinemargin and the \textwidth and the \foremargin is larger than the \paperwidth. Either increase the \paperwidth or reduce the others.

The combination of argument values is ambiguous. The lengths will be set to zero

The combination of values in the arguments to one of the commands for page layout does not make sense.

The `extrafontsizes' option is required to use the `...pt' option

If you want to use a '...pt' class option greater than 25pt you also have to use the extrafontsizes option. The class will use the 17pt option.

Unknown document division name (...)

You have used an unknown division name in the argument to \settocdepth or \setsecnumdepth and friends. If you haven't mistyped it you will have to use \setcounter instead.

Unknown mark setting type `...' for ...mark

In \createmark or \createplainmark the mark setting type should have been left or both or right. The class will use both.

Unknown numbering type ... for ...mark

In \createmark the class expected either shownumber or nonumber for displaying the number. It will use shownumber.

Unrecognized argument for \sidecapmargin

The argument to \sidecaption should be left or right or inner or outer.

\uppermargin and/or \textheight and/or \lowermargin are too large for \paperheight by ...

The sum of the \uppermargin and the \textheight and the \lowermargin is larger than the \paperheight. Either increase the \paperheight or reduce the others.

You have used the `\*pt' option but file ... can't be found

You have used the \*pt option but the corresponding clo file can't be found. Check your definitions of \anyptfilebase and \anyptsize. The mem10.clo file will be used instead.

XeTeX is required to process this document

The document needs to be processed via XeTeX. Try using xelatex instead of (pdf)latex, or try removing any XeTeX packages from the document.

#### F.5 Class warnings

These are introduced by Class memoir Warning:

For example \addtodef{alf}{\joe}{fred} will produce a message along the lines of:

Class memoir Warning: 'alf' is not a macro on input line 91.

while \addtodef{\joe}{alf}{fred} might produce:

Class memoir Warning: '\joe' is not a macro on input line 97.

The following is an alphabeticised list of the class warnings.

... at index ... in pattern ... is not a digit

The character at the given position in the verse pattern is not a digit.

... is not a macro

Using \addtodef or \addtoiargdef you have tried to extend the definition of an unknown macro.

... is not an input stream

You are trying to access a non-existent input stream.

... is not an output stream

You are trying to access a non-existent output stream.

Bad \sidebarmargin argument

The argument to \sidebarmargin is not recognized. The class will use right.

Characters dropped after \end{...}

At the end of a verbatim environment there should be no characters after the  $\end{...}$  on the same line.

Column ... is already defined

The column type has been defined by a previous \newcolumntype.

Counter ... already defined

For information only, the counter in \providecounter is already defined.

Do not use \footnote in \maketitle. Use \thanks instead

You cannot use \footnote in any of the \maketitle elements (i.e., \title or \author or \date) but you can use \thanks.

Empty `thebibliography' environment

There are no \bibitems in the thebibliography environment.

Environment ... already defined

For information only, the environment in \provideenvironment is already defined.

Index ... for pattern ... is out of bounds

The index for the verse pattern is either too low or too high.

Input stream ... is already defined

You are trying to use \newinputstream to create an already existing input stream.

Input stream ... is not open

You are trying to access or close an input stream that is closed.

Input stream ... is open

You are trying to open an input stream that is already open.

Length ... already defined

For information only, the length in \providelength is already defined.

Marginpar on page  $\dots$  moved by  $\dots$ 

A marginal note has been lowered by the given amount to avoid overwriting a previous note; the moved note will not be aligned with its \marginpar. (This is a more informative message than the normal LaTeX one.)

No more to read from stream ...

There is nothing left in the stream to be read.

Optional argument of \twocolumn too tall on page ...

The contents of the optional argument to \twocolumn was too long to fit on the page.

Output stream ... is already defined

You are trying to use \newoutputstream to create an already existing output stream.

Output stream ... is not open

You are trying to access or close an output stream that is closed.

Output stream ... is open

You are trying to open an output stream that already open.

Redefining primitive column ...

The argument to \newcolumntype is one of the basic column types.

Stream ... is not open

You are trying to access a stream, either input or output, that is closed.

The ... font command is deprecated. Use ... or ... instead

You are using a deprecated font command. Consider using one of the alternatives.

The counter will not be printed. The label is: ...

The optional  $\langle style \rangle$  argument to the enumerate environment does not include one of the special characters.

Undefined index file ...

You are trying to add an index entry to an unknown idx file.

Unknown toclevel for ...

The division name you have used for \settocdepth is not recognized.

\verb may be unreliable inside tabularx

A \verb in a tabularx may work, but may not.

X columns too narrow (table too wide)

The width of the X columns in a tabularx had to be made too narrow.

# Comments

# G.1 Algorithms

Over time we may use this section to explain, or list some of the algorithms for some of the macros in the class. The information may be useful to some.

# G.1.1 Autoadjusting \marginparwidth

This algorithm is used within \fixthelayout unless the user have used \setmarginnotes.

```
if twocolumn then
  marginparwidth = min{inner margin,outer margin}
  if twoside then
    if marginpar always left or always right then
      marginparwidth = min{inner margin,outer margin}
    else if marginpar in outer margin then
      marginparwidth = outer margin
    else if marginpar in inner margin then
      marginparmargin = inner margin
    end if
  else
    if marginpar in left margin then
      marginparwidth = inner margin
      marginparwidth = outer margin
    end if
  end if
end if
marginparwidth = marginparwidth - 2marginparsep
if marginparwidth < 1pt then
  marginparwidth = 1pt
end if
```

# Notes

# Chapter 3 텍스트와 글꼴

['work in progress'] (page 35) XeTeX enables you to use Opentype fonts with LaTeX, and supports both left-to-right and right-to-left typesetting. It has become very popular with those involved in linguistics and non-Latin scripts.

[using the appropriate package] (page 35) I have found Christopher League's *TeX support for the FontSite 500 CD*, obtainable from http:contrapunctus.net/fs500tex, extremely useful in providing packages for a wide range of PostScript fonts for me to use. You do have to buy a CD containing the sources of the fonts from FontSite (http://www.fontsite.com); it cost me a total of \$37.12, including taxes and shipping, in 2002 for 512 PostScript and TrueType professional quality fonts that are legal and very reasonably priced.

Many of the fonts fall into the Decorative/Display category but the book fonts include:

Blackletter Alte Schwabacher, Engravers Old English, Fette Fraktur, Fette Gotisch, and Olde English.

Uncial/Mediaeval American UncialXXAmerican Uncial, Linden, and Rosslaire.

Geralde/Venetian Bergamo (also known as Bembo), Caslon, Garamond, Goudy Old Style, Jenson Recut (also known as Centaur), URW Palladio (also known as Palatino), Savoy (also known as Sabon), Schnittger, University Old Style, and Vendome.

Transitional URW Antiqua, Baskerville, Century Old Style, ATF Clearface, English Serif, Jessica (also known as Joanna), Lanston Bell, New Baskerville, and Nicholas Cochin.

Modern/Didone Basel (also known as Basilia), Bodoni, Modern, and Walbaum.

Free Form Barbedour, Bernhard Modern, Della Robbia, Engravers Litho, Flanders, and Lydian.

Sans Serif There are over 20 in this category but some of the ones I am most familiar with are: Chantilly (also known as Gill Sans), Franklin Gothic, Function (also known as Futura), Lanston Koch, News Gothic, Opus (also known as Optima), Struktor (also known as Syntax), and Unitus (also known as Univers).

Slab Serif Cheltenham, Clarendon, Egyptian, Glytus (also known as Glypha), URW Latino (also known as Melior), Litho Antique, Serific (also known as Serifa).

Script There are some sixteen Script fonts.

Decorative There are over fifty Decorative fonts.

Symbol There are a dozen miscellaneous symbol fonts which include, among others, arrows, borders, fleurons and various icons.

Chapter 12 Page notes

[Put no mark …finally printed] (page 243) This manual uses both footnotes and endnotes. For identifying the endnotes I have used the 'words' method for identifying the parent location of an endnote, so as not to start confusing the reader with two sets of note marks in the body of the text. Typically either footnotes or endnotes are used, not both, so the question of distinguishing them does not normally arise.

# 문헌목록

CTAN is the Comprehensive TeX Archive Network. Information on how to access CTAN is available at http://www.tug.org.

- [AHK90] Paul W. Abrahams, Kathryn Hargreaves and Karl Berry. *TeX for the Impatient*. Addison-Wesley, 1990. (Available at ftp://tug.org/tex/impatient)
- [Ars99] Donald Arseneau. *The url package*. February, 1999. (Available from CTAN via /macros/latex/contrib/url/)
- [Ars01a] Donald Arseneau. *The titleref package*. April, 2001. (Available from CTAN via /macros/latex/contrib/titleref/)
- [Ars01b] Donald Arseneau. *The chapterbib package*. September, 2001. (Available from CTAN via /macros/latex/contrib/cite/)
- [Ars07] Donald Arseneau. *The framed package* v0.95. October, 2007. (Available from CTAN via /macros/latex/contrib/framed/)
- [Ber02] Jens Berger. *The titlesec and titletoc packages*. September, 2002. (Available from CTAN via /macros/latex/contrib/titlesec/)
- [Bez99] Javier Bezos. *The titlesec and titletoc packages*. February, 1999. (Available from CTAN via /macros/latex/contrib/titlesec/)
- [Bir04] Derek Birdsall. notes on book design. Yale University Press, 2004. ISBN 0-300-10347-6.
- [Bra97] Johannes Braams. *The alltt environment*. June, 1997. (Available as alltt.dtx and alltt.ins from CTAN via /macros/latex/base/)
- [Bri99] Robert Bringhurst. *The Elements of Typographic Style*. Hartley & Marks, second edition, 1999. ISBN 0-88179-033-8.
- [Car14] David Carlisle. *The delarray package*. October, 2014. (Available from CTAN via /macros/latex/required/tools/)
- [Car95] David Carlisle. *The afterpage package*. October, 1995. (Available from CTAN via /macros/latex/required/tools/)
- [Car98b] David Carlisle. *The longtable package*. May, 1998. (Available from CTAN via /macros/latex/required/tools/)
- [Car98c] David Carlisle. *The enumerate package*. August, 1998. (Available from CTAN via /macros/latex/required/tools/)

- [Car16] David Carlisle. *The tabularx package*. February, 2016. (Available from CTAN via /macros/latex/required/tools/)
- [CR99] David Carlisle and Sebastian Rahtz. *The graphicx package*. February, 1999. (Available from CTAN via /macros/latex/required/graphics/)
- [Car14] David Carlisle. *The dcolumn package*. May, 2001. (Available from CTAN via /macros/latex/required/tools/)
- [Car04] David Carlisle. *The textcase package*. October, 2004. (Available from CTAN in /macros/latex/contrib/textcase)
- [Car05] David Carlisle. *Packages in the graphics bundle* (includes the color package). November, 2005. (Available from CTAN via /macros/latex/required/graphics/)
- [CB99] Warren Chappell and Robert Bringhurst. *A Short History of the Printed Word*. Hartley & Marks, 1999. ISBN 0-88179-154-7.
- [CH88] Pehong Chen and Michael A. Harrison. 'Index Preparation and Processing'. Software: Practice and Experience, 19:8, pp. 897–915, September, 1988. (Available from CTAN via /indexing/makeindex/paper/)
- [Chi93] The Chicago Manual of Style, Fourteenth Edition. The University of Chicago, 1993. ISBN 0-226-10389-7.
- [Dal99a] Patrick W. Daly. Natural Sciences Citations and References. May, 1999. (Available from CTAN via /macros/latex/contrib/natbib/)
- [Dal99b] Patrick W. Daly. Customizing Bibliographic Style Files. August, 1999. (Available from CTAN via /macros/latex/contrib/custom-bib)
- [Dow96] Geoffrey Dowding. Finer Points in the Spacing & Arrangement of Type. Hartley & Marks, 1996. ISBN 0-88179-119-9.
- [Dow00] Michael J. Downes. *The patchcmd package*. July, 2000. (Available from CTAN via /macros/latex/contrib/patchcmd/)
- [Eij92] Victor Eijkhout. *TeX by Topic*. Addison-Wesley, 1992. ISBN 0-201-56882-9. (Available from http://www.eijkhout.net/tbt/).
- [Eij99] Victor Eijkhout. comment.sty October, 1999. (Available from CTAN via /macros/latex/contrib/comment/)
- [Fai00] Robin Fairbairns. footmisc a portmanteau package for customising footnotes in LaTeX2e. March, 2000. (Available from CTAN via /macros/latex/contrib/footmisc/)
- [FAQ] Robin Fairbairns. *The UK TeX FAQ*. (Available from CTAN via http://faq.tug.org/)
- [Far90] Gerald Farin. Curves and Surfaces for Computer Aided Geometric Design A Practical Guide. Academic Press, 2 edition, 1990.
- [FP80] I. D. Faux and M. J. Pratt. Computational Geometry for Design and Manufacture. Ellis Horwood, 1980.
- [Fea16] Simon Fear. Publication quality tables in LaTeX. April, 2016. (Available from CTAN via /macros/latex/contrib/booktabs)

- [Fly98] Peter Flynn. Formatting Information: A Beginner's Introduction to Typesetting with LaTeX2. 2002. (Available from CTAN via /info/beginlatex/)
- [GM<sup>+</sup>07] Michel Goossens, Frank Mittelbach, et al. *The LaTeX Graphics Companion: Second edition*. Addison-Wesley, 2007. ISBN 0-321-50892-0.
- [GR99] Michel Goossens and Sebastian Rahtz (with Eitan Gurari, Ross Moore and Robert Sutor). *The LaTeX Web Companion: Integrating TeX, HTML and XML*. Addison-Wesley, 1999. ISBN 0-201-43311-7.
- [Hoe98] Alan Hoenig. *TeX Unbound: LaTeX and TeX strategies for fonts, graphics, and more.* Oxford University Press, 1998. ISBN 0–19–509686–X.
- [Jon95] David M. Jones. *A new implementation of LaTeX's indexing commands*. September, 1995. (Available from CTAN via /macros/latex/contrib/camel)
- [Keh98] Roger Kehr. xindy: A flexible indexing system. February, 1998. (Available from CTAN via /indexing/xindy/)
- [Ker07] Uwe Kern. Extending LaTeX's color facilities: the xcolor package. January, 2007. (Available from CTAN via /macros/latex/contrib/xcolor/)
- [Kha10] Vafa Khalighi. The Bidi package. 2010. (Available from CTAN via /macros/latex/contrib/bidi/)
- [Knu84] Donald E. Knuth. *The TeXbook*. Addison-Wesley, 1984. ISBN 0-201-13448-9.
- [Knu86] Donald E. Knuth. TeX: The Program. Addison-Wesley, 1986. ISBN 0-201-13437-3.
- [Knu87] Donald E. Knuth. Computer Modern Typefaces. Addison-Wesley, 1987. ISBN 0-201-134446-2.
- [Knu92] Donald E. Knuth. The METAFONT Book. Addison-Wesley, 1992. ISBN 0-201-13444-6.
- [Lam94] Leslie Lamport. *LaTeX: A Document Preparation System*. Addison-Wesley, 1994. ISBN 0-201-52983-1.
- [LEB04] Leslie Lamport, Victor Eijkhout and Johannes Braams. NTG document classes for LaTeX version 2e. June, 2004. (Available from CTAN via /macros/latex/contrib/ntgclass/)
- [Lea03] Christopher League. TeX support for the FontSite 500 CD. May 2003. (Available from http://contrapunctus.net/fs500tex)
- [Leh04] Philipp Lehman. *The Font Installation Guide*. December 2004. (Available from CTAN via /info/Type1fonts/fontinstallationguide)
- [Leu92] Mary-Claire van Leunen. *A Handbook for Scholars*. Oxford University Press, 1992. ISBN 0-19-506954-4.
- [Lon91] F. W. Long. multind. August, 1991. (Available from CTAN as /macros/latex209/contrib/misc/multind.sty)
- [Mad06] Lars Madsen. Various chapter styles for the memoir class. July, 2006. (Available from CTAN via /info/latex-samples/MemoirChapStyles/)
- [Mad07] Lars Madsen. The Memoir Experimental Support Package. 2007. (Available from CTAN via /macros/latex/contrib/memexsupp/)

- [McD98] Rowland McDonnell. *The sectsty package*. November, 1998. (Available from CTAN via /macros/latex/contrib/sectsty/)
- [McL80] Ruari McLean. *The Thames & Hudson Manual of Typography*. Thames & Hudson, 1980. ISBN 0-500-68022-1.
- [Mit18] Frank Mittelbach. An environment for multicolumn output. April, 2018. (Available from CTAN (the multicol package) via /macros/latex/required/tools/)
- [MC18] Frank Mittelbach and David Carlisle. *A new implementation of LaTeX's tabular and array environment*. May, 2018. (Available from CTAN (the array package) via /macros/latex/required/tools/)
- [MC00] Frank Mittelbach and David Carlisle. *The fixltx2e package*. December, 2016. As of 2015 the functionality of this package has been merged with the LaTeX kernel. (Available from CTAN via /macros/latex/base/)
- [MG<sup>+</sup>04] Frank Mittelbach, Michael Goossens, et al. *The LaTeX Companion: Second Edition*. Addison-Wesley, 2004. ISBN 0-201-36299-6.
- [Mor85] Michael E. Mortenson. *Geometric Modeling*. John Wiley & Sons, 1985.
- [NG98] Rolf Niespraschk and Hubert Gäßlein. *The sidecap package*. June, 1998. (Available from CTAN via /macros/latex/contrib/sidecap/)
- [Oet] Tobias Oetiker. *The Not So Short Introduction to LaTeX2e.* (Available from CTAN via /info/lshort/)
- [Oos96] Piet van Oostrum. *Page Layout in LaTeX*. June, 1996. (Available from CTAN via /macros/latex/contrib/fancyhdr/)
- [Pak01] Scott Pakin. *The Comprehensive LaTeX Symbol List*. July, 2001. (Available from CTAN via /info/symbols/comprehensive/)
- [dP84] H. de Parville. Recreations mathematique: La Tour d'Hanoi et la question du Tonkin. *La Nature*, part I:285–286, Paris 1884.
- [Pat88a] Oren Patashnik. *BibTeXing*. February, 1988. (Available from CTAN as /bibliography/bibtex/distribs/doc/btxdoc.tex)
- [Pat88b] Oren Patashnik. Designing BibTeX Styles. February, 1988. (Available from CTAN as /bibliography/bibtex/distribs/doc/btxhak.tex)
- [Pug02] Diego Puga. The Pazo Math fonts for mathematical typesetting with the Palatino fonts. May, 2002. (Available from CTAN via /fonts/mathpazo/)
- [Rahtz01] Sebastian Rahtz. Section name references in LaTeX. January, 2001. (Available from CTAN (the nameref package) via /macros/latex/contrib/hyperref/)
- [Rahtz02] Sebastian Rahtz. *Hypertext marks in LaTeX*. May, 2002. Now maintained and frequently updated by Heiko Oberdiek. (Available from CTAN via /macros/latex/contrib/hyperref/)
- [Rec97] Keith Reckdahl. *Using Imported Graphics in LaTeX2e*. December, 1997. Updated in 2006. (Available from CTAN via /info/epslatex.pdf)

- [Rei07] Edward M. Reingold. 'Writing numbers in words in TeX'. TUGboat, 28, 2 pp 256–259, 2007.
- [RBC74] W. W. Rouse Ball and H. S. M. Coxeter. *Mathematical Recreations and Essays*. University of Toronto Press, twelfth edition, 1974.
- [SW94] Douglas Schenck and Peter Wilson. *Information Modeling the EXPRESS Way*. Oxford University Press, 1994. ISBN 0-19-508714-3.
- [SRR99] Rainer Schöpf, Bernd Raichle and Chris Rowley. *A New Implementation of LaTeX's verbatim and verbatim\* Environments*. December, 1999. (Available from CTAN via /macros/latex/required/tools/)
- [Sch07] Martin Scharrer, Version Control of LaTeX Documents with svn-multi. The PracT<sub>E</sub>X Journal, 3, 2007. ISSN 1556-6994.
- [Sch09] Martin Scharrer, *The svn-multi package*, 2009. (Available from CTAN via /macros/latex/contrib/svn-multi/)
- [Sne04] Maarten Sneep. *The atmosphere in the laboratory: cavity ring-down measurements on scattering and absorption.* Phd thesis, Vrijie Universiteit, Amsterdam, 2004.
- [Tal06] Nicola L. C. Talbot. datetime.sty: Formatting Current Date and Time. December, 2006. (Available from CTAN via /macros/latex/contrib/datetime/)
- [Thi98] Christina Thiele. 'Hey it works: Ornamental rules'. *TUGboat*, vol. 19, no. 4, p 427, December 1998.
- [Thi99] Christina Thiele. 'The Treasure Chest: Package tours from CTAN', *TUGboat*, vol. 20, no. 1, pp 53–58, March 1999.
- [TJ05] Kresten Krab Thorup, Frank Jensen (and Chris Rowley). The calc package Infix notation arithmetic in LaTeX. August, 2005. (Available from CTAN via /macros/latex/required/tools/)
- [Tob00] Geoffrey Tobin. setspace.sty. December, 2000. (Available from CTAN via /macros/latex/contrib/setspace/)
- [Tsc91] Jan Tschichold. The Form of the Book. Lund Humphries, 1991. ISBN 0-85331-623-6.
- [Ume99] Hideo Umeki. *The geometry package*. November, 1999. (Available from CTAN in /macros/latex/contrib/geometry/)
- [Wil00] Graham Williams. *The TeX Catalogue*. (Latest version on CTAN as /help/Catalogue/catalogue.html)
- [Wil93] Adrian Wilson. *The Design of Books*. Chronicle Books, 1993. ISBN 0-8118-0304-X.
- [Wil99b] Peter Wilson. *The tocvsec2 package*. January, 1999. (Available from CTAN via /macros/latex/contrib/tocvsec2/)
- [Wil00a] Peter Wilson. *The epigraph package*. February, 2000. (Available from CTAN via /macros/latex/contrib/epigraph/)
- [Wil00b] Peter Wilson. LaTeX files for typesetting ISO standards. February, 2000. (Available from CTAN via /macros/latex/contrib/isostds/iso/)

- [Wil00c] Peter Wilson. *The nextpage package*. February, 2000. (Available from CTAN as /macros/latex/contrib/misc/nextpage.sty)
- [Wil00e] Peter Wilson. *The xtab package*. April 2000. (Available from CTAN via /macros/latex/contrib/xtab)
- [Wil01a] Peter Wilson. *The abstract package*. February, 2001. (Available from CTAN via /macros/latex/contrib/abstract/)
- [Wil01d] Peter Wilson. *The ccaption package*. March, 2001. (Available from CTAN via /macros/latex/contrib/ccaption/)
- [Wil01e] Peter Wilson. *The chngcntr package*. April, 2001. (Available from CTAN via /macros/latex/contrib/chngcntr/)
- [Wil01f] Peter Wilson. *The hanging package*. March, 2001. (Available from CTAN via /macros/latex/contrib/hanging/)
- [Wil01g] Peter Wilson. The titling package. March, 2001. (Available from CTAN via /macros/latex/contrib/titling/)
- [Wil01h] Peter Wilson. *The tocbibind package*. April, 2001. (Available from CTAN via /macros/latex/contrib/tocbibind/)
- [Wil01i] Peter Wilson. *The tocloft package*. April, 2001. (Available from CTAN via /macros/latex/contrib/tocloft/)
- [Wil03a] Peter Wilson. *The layouts package* November, 2003. (Available from CTAN in /macros/latex/contrib/layouts/)
- [Wil03b] Peter Wilson. ledmac: A presumptuous attempt to port EDMAC and TABMAC to LaTeX November, 2003. (Available from CTAN via /macros/latex/contrib/ledmac/)
- [Wil04a] Peter Wilson. *The bez123 and multiply packages*, April 2004. (Available from CTAN in /macros/latex/contrib/bez123/)
- [Wil04b] Peter Wilson. *The pagenote package* September, 2004. (Available from CTAN via /macros/latex/contrib/pagenote/)
- [Wil07a] Peter Wilson. Some Examples of Title Pages. Herries Press, 2007. (Available from CTAN via /info/latex-samples/TitlePages/)
- [Wil07e] Peter Wilson. 'Glisterings', TUGboat, 28(2):229–232, 2007.
- [Wil08a] Peter Wilson. *The changepage package*. March, 2008. (Available from CTAN via /macros/latex/contrib/changepage/)
- [Wil08b] Peter Wilson. 'Glisterings', TUGboat, 29(2):324–327, 2008.
- [Wil09a] Peter Wilson. *The fonttable package* April, 2009. (Available from CTAN via /macros/latex/contrib/fonttable/)
- [Wil09b] Peter Wilson (with the assistance of Lars Madsen). The LaTeX memoir class for configurable book typesetting: source code July, 2009. (Available from CTAN via /macros/latex/contrib/memoir/)

- [Wil07c] Peter Wilson (with the assistance of Lars Madsen). The Memoir Class for Configurable Typesetting User Guide August, 2009. Regularly updated. (Available from CTAN via /macros/latex/contrib/memoir/)
- [Wil09d] Peter Wilson. A Few Notes on Book Design August, 2009. (Available from CTAN via /info/memdesign/)
- [Wil??] Peter Wilson. A Rumour of Humour: A scientist's commonplace book. To be published.
- [Wri18] Joseph Wright. Siunitx A comprehensive (SI) units package May, 2018. (Available from CTAN via /macros/latex/contrib/siunitx)
- [Zan98] Timothy Van Zandt. Documentation for fancybox.sty: Box tips and tricks for LaTeX, November, 1998. (Available from CTAN via /macros/latex/contrib/fancybox/)

# 색인

The first page number is usually, but not always, the primary reference to the indexed topic.

```
가
                                        바닥글, 106-108, 110, 114, 115
                                        방주, 114
각주, 67-69
                                                         사
   표지, 67
감사의 말, 67
                                        삽화, 105
   양식화, 67
                                         선
구현, xxiii
                                            두께, 110
기본값
   용지, 1
                                                         아
   인쇄 옵션, 4
   활자 크기, 3
                                        알파벳 번호, 105
                                        여백
                나
                                            바깥, 114
                                            안쪽, 114
뉴스그룹, xxiii
                                            왼쪽, 121
재단, 114
                다
                                            재단 여백, 106
                                        외부 패키지, xxiii
떠다니는 개체, 5
                                        요약, 66
                라
                                        용지, 4
                                            ,
기본값, 1
                                            크기, 1
로마 숫자, 105
                                              A3, 1
                마
                                              A4, 1
                                              A5, 1
머릿글, 106-110, 114, 115
                                              A6, 1
문단, 5, 121
                                              B3, 1
   B6, 1
   단위로 조판, 122
                                              broadsheet, 2
   들여쓰기, 121
                                              crown octavo, 2
   블록, 121
                                              demy octavo, 2
     반대 이유, 121
                                              dollar bill, 2
                                              ebook, 1
                바
                                              executive, 2
```

인용문 페이지 양식

ledger, 2		
legal, 2		
letterpaper, 1, 2		
medium octavo, 2		
metric crown octavo, 1		
metric demy octavo, 1		
metric large crown octavo, 1		
metric small royal octavo, 1		
old, 2		
post octavo, 2		
pott octavo, 2		
royal octavo, 2		
small demy octavo, 2		
small royal octavo, 2		
statement, 2		
super royal octavo, 2		
차		
71		
초고 문서, 111		
111 to 111		
카		
•		
카운터 표시, 105		
컬럼		
double, 4		
코드, xxiii		
크기		
용지, 1		
종이, 1		
클래스, xxiii		
크네스, xxiii 클래스 옵션, 1		
· · · · · · · · · · · · · · · · · · ·		
article, 5		
bibliography, 5		
fonts, 5		
layout truncation, 5		
math, 5		
크기, 1		
활자 크기, 2		
파		
~		
페기기		
패키지, xxiii		
페이지 매김, 105		
변경, 105		
중단, 105		
페이지 양식		

페이지 컬러 \@startsection

지정, 106	in macro code, 143, 163, 185, 201, 352,
클래스, 106	357, 429, 451
페이지 컬러, 5	0 (actual specifier), 306
폴리오, 106	@-expression, 455
표시 변경, 105	\@@end, 450
표제, 65	\@@wrglom@m, 312
양식화, 67	\@MM, 433
표지, 57, 63, 66, 107, 114	\@Mi, 433
필자, xxiii	\@Mii, 433
2/1, XXIII	
ठॅ}-	\@Miii,433
•	\@Miv, 433
활자 크기, 2	\@bsphack, 290
기본값, 3	\@caption, 200, 201
1 C W, 0	\@captype, 199, 200
Analphabetics	\@cclv, 433
<b>F</b>	\@cclvi, 433
\!, 333	\Qcftasnum, 157
! (level specifier), 304	\@cftasnumb, 157
!h float specifier, 459	\Qcftbsnum, 157
\#, 446, 452	\@chapapp, 88
\&, 447	\@dblfpbot (length), 179, 180
\(,445,454	\@dblfpsep (length), 179
\), 454	\@dblfptop (length), 179, 180
*pt (option), 3, 464	\@dblftop (length), 180
\+, 457	\@dotsep, 140, 143, 150
,333	\@dottedtocline, 142
\-, 452, 457	\@esphack, 290
allowed only in math mode, 454	\@fibnext, 439
· · · · · · · · · · · · · · · · · · ·	\@fibseries, 439-441
at index in pattern, 464	\@fnsymbol, 235
is negative, 461	\@fpbot (length), 179, 180
is not a counter, 461	\@fpsep (length), 179, 180
is not a macro, 464	\@fptop (length), 179, 180
is not an input stream, 464	\@hangfrom, 96, 122, 349
is not an output stream, 464	\@m, 433
is zero or negative, 461	\@makecaption, 200-202
\:,333	\@makefnmark,232
\;, 333 \: 454 457	\@minus, 434
\<, 454, 457	$\ensuremath{\texttt{Qnamedef,431}}$
\< in mid line, 454	\@nameuse, 431
<{}at wrong position, 461	\@ne, 433
\=, 457	\@plus, 434
=, 430	\@pnumwidth, 140, 147, 154
\>, 457	\@pnumwidth (length), 151
>{}at wrong position, 461	\@seccntformat, 96, 97
©,	\@setref, 291
	\@startsection 163

\@tempdima, 437	\absrightindent (length), 72
\@tempdimb, 437	\abstitleskip (length), 72, 73
\@thefnmark, 232, 233	abstract, 66, 71
\@tocrmarg, 140, 142, 147, 154	heading, 71
\@whiledim, 440	one column, 73
$\ensuremath{\verb \displayslayer }$	styling, 71–73
\@xfloat, 200	abstract (environment), xxiii, 71-73,
\0xxxii,433	335
\@zeroseps, 283, 378	abstract (package), 71, 336
\[, 205, 212, 445, 454	\abstractcol, 71, 72
\ (escape specifier), 307	\abstractintoc, 71, 72
\\*, 183, 257, 258	\abstractname, 71, 72, 335
<b>\\&gt;,</b> 259, 267	\abstractnamefont,72
\\!,258	\abstractnum, 71, 72
\], 205, 212, 454	\abstractrunin,71-73
l (encap specifier), 306	\abstracttextfont,72
10pt (option), 2, 3, 5, 46	Adams, William, 90, 319
11pt (option), 2, 46	add to a macro, 323
12pt (option), 2, 5, 46	add to contents, 142
14pt (option), 2, 3, 46, 110	\addappheadtotoc, 78
17pt (option), 2, 3, 46, 463	\addcontentsline, 140, 142, 143, 145,
20pt (option), 2, 3, 46	165, 187
25pt (option), 2, 46, 463	\added, 317, 318
30pt (option), 3, 46	\addlinespace, 217
36pt (option), 3, 46	\addpenalty, 456
48pt (option), 3, 46	\addperiod, 334
60pt (option), 3, 46	\addtocontents, 142, 143, 155, 165
9pt (option), 2, 46	\addtocounter, 105, 453, 456
	\addtodef, 162, 323, 360, 464
Α	\addtodef*, 324
	\addtoiargdef, 323, 324, 464
A box was supposed to be here, 444	\addtoiargdef*,324
A pattern has not been specified, 461	\addtolength, 22, 72, 86, 134, 258, 368,
a3paper (option), 1	436
A4 paper, 8	\addtonotes, 247
a4paper (option), 1, 27, 29	\addtopsmarks, 115
a5paper (option), 1	\addtostream, 285
aбpaper (option), 1	$\addvspace, 87, 456$
\abnormalparskip, 49	adjustwidth (environment), 90, 129,
\abovecaptionskip, 204	130, 134, 276
\abovecaptionskip (length), 201	adjustwidth* (environment), 66, 129
\aboverulesep (length), 216, 217	\advance, 433, 434, 436
\abslabeldelim,72	\afterbookskip, $80$ , $82$
\absleftindent (length), 72	\afterchapskip (length), 85, 86
\absnamepos, 72	\afterchapternum, 85, 89
\absparindent (length), 72	\afterchaptertitle, 85, 146, 348
\absparsep (length), 72	\afterepigraphskip (length), 251

\afternextrecto \AtEndOfClass

\afternextrecto, 164 \afternextverso, 164 \afternextverso, 164 \afternextverso, 164 \afterpage, 164 \afterpage (package), 164, 189 \afterpage (package), 164, 189 \afterpartskip, 80, 83, 254 \afterPoemTitle, 263, 264 \afterPoemTitlenum, 263, 264 \afterPoemTitleskip (length), 264 \afterYxitle, 146 \appendix Pagenote, 246 \makeheadposition, 110 \makeTextUppercase, 107 \aliangagestyle, 109 \alialtt (penvironment), 255, 256 \alltt (package), 255, 375 \alpha, 105 \alphabet length, 15 \alphabet length, 15 \alphabet cnumbering, 105 \alsoname, 306, 335 \Alte Schwabacher, 469 \altverse (environment), 259, 261, 265 \amname, 321, 335 \amsfonts (package), 455 \amsmath (package), 455 \amsmath (package), 375 \and, 66 \andnext, 66 \andnext
\afterpage, 164 afterpage (package), 164, 189 afterPoemTitle, 263, 264 \afterPoemTitlenum, 263, 264 \afterPoemTitleskip (length), 264 \afterPoemTitleskip (length), 264 \afterPoemTitleskip (length), 264 \afterVentTitle, 146 makeheadposition, 110 MakeTextUppercase, 107 \aliaspagestyle, 109 alltt (pervironment), 255, 256 alltt (package), 255, 375 Alph, 105 alphabet length, 15 alphabetic numbering, 105 \alsoname, 306, 335 Alte Schwabacher, 469 altverse (environment), 259, 261, 265 amsmath (package), 375 \amsmath (package), 375 \amsmath (package), 375 \and, 66 anonymous division, 99 styling, 100-101 ans (file), 290 answer (environment), 290, 291 answer (package), 292 \anyptfilebase, 3, 464 \appendix, 78, 362 appendix, 78, 82, 88, 144, 153, 154  marginparmargin, 169 argument, optional, 292 Argument to \overridesidecapmargin neither, 462 Arg
afterpage (package), 164, 189 \afterpartskip, 80, 83, 254 \afterPoemTitle, 263, 264 \afterPoemTitleskip (length), 264 \afterPoemTitleskip (length), 264 \afterPoemTitleskip (length), 264 \afterXtitle, 146 pagenote, 246 makeheadposition, 110 MakeTextUppercase, 107 \aliaspagestyle, 109 alltt (environment), 255, 256 alltt (package), 255, 375 Alph, 105 alph, 105 alphabet length, 15 alphabetic numbering, 105 \aliasname, 306, 335 Alte Schwabacher, 469 altverse (environment), 259, 261, 265 \amsmath (package), 375 \amsmath (package), 375 \amsmath (package), 375 \amsmath (package), 375 \and, 66 \anonymous division, 99 styling, 100–101 ans (file), 290 answer (environment), 290, 291 answer (package), 292 \anyptfilebase, 3, 464 \anyptsize, 3, 4, 464 appendix, 78, 362 appendix, 78, 82, 88, 144, 153, 154  argument to \overridesidecapmargin neither, 462 Arial, 353, 369 Arithmetic overflow, 444 array, 445 intercolumn space, 224 array (environment), 205,
\afterpartskip, 80, 83, 254 \afterPoemTitle, 263, 264 \afterPoemTitlenum, 263, 264 \afterPoemTitleskip (length), 264 \afterXtitle, 146 pagenote, 246 makeheadposition, 110 MakeTextUppercase, 107 alliaspagestyle, 109 alltt (environment), 255, 256 alltt (package), 255, 375 Alph, 105 alph, 105 alphabet length, 15 alphabetic numbering, 105 \alsoname, 306, 335 Alte Schwabacher, 469 altverse (environment), 259, 261, 265 \amsoname, 321, 335 amsfonts (package), 455 amsmath (package), 455 amsmath (package), 375 \and, 66 \anonymous division, 99 styling, 100-101 ans (file), 290 answer (package), 292 anyptfilebase, 3, 464 appendics (environment), 78 appendix, 78, 362 appendix, 78, 82, 88, 144, 153, 154  appendix, 78, 82, 88, 144, 153, 154  appendix, 78, 82, 88, 144, 153, 154  Argument of has an extra }, 444 Argument to \overridesidecapmargin neither, 462 Argument to \overridesidecappos is not, 462 Arial, 353, 369 Arithmetic overflow, 444 array, 445 intercolumn space, 224 row fill, 218-219 row spacing, 224 array (package), 205, 206, 336
\afterPoemTitle, 263, 264 \afterPoemTitlenum, 263, 264 \afterPoemTitleskip (length), 264 \afterPoemTitleskip (length), 264 \afterXtitle, 146 \agenote, 246 \makeheadposition, 110 \makeTextUppercase, 107 \aliaspagestyle, 109 \alltt (package), 255, 375 \alph, 105 \alphabet length, 15 \alphabet length, 15 \alphabetic numbering, 105 \alsoname, 306, 335 \Alte Schwabacher, 469 \altverse (environment), 259, 261, 265 \amsmath (package), 455 \amsmath (package), 375 \amsforts (package), 455 \amsmath (package), 375 \and, 66 \annonymous division, 99 \styling, 100-101 \answer (package), 292 \anyptfilebase, 3, 464 \anypendix, 78, 862 \appendix, 78, 862 \appendix, 78, 82, 88, 144, 153, 154 \argument to \overridesidecapmargin neither, 462 \Argument to \overridesideapproprimer, 445 \argument to \overridesideapproprimer, 445 \argument of \argument of \argument of \argument of \argument of \argument of \argument
\afterPoemTitlenum, 263, 264 \afterPoemTitleskip (length), 264 \afterXtitle, 146 agenote, 246  makeheadposition, 110  MakeTextUppercase, 107 \aliaspagestyle, 109 alltt (environment), 255, 256 alltt (package), 255, 375  Alph, 105 alphabet length, 15 alphabetic numbering, 105 \alsoname, 306, 335  Alte Schwabacher, 469 altverse (environment), 259, 261, 265 \amsamfonts (package), 455 amsfonts (package), 455 amsmath (package), 375 \ams(file), 290 answer (environment), 290, 291 answer (package), 292 anyptfilebase, 3, 464 appendics, 78, 82, 88, 144, 153, 154  Argument to \overridesidecapmargin neither, 462 Argument to \overridesideappea page and tale 2 Argument to \overridesideappea page and tale 2 Argument to \overridesideappea page, 292 Arithmetic overflow, 444 array, 445 array (environment), 295, 207, 211, 212, 218, 222, 224, 225, 445, 447, 448, 453, 455, 456, 461, 462 array (package), 205, 206, 336 Array (package), 205,
\afterPoemTitleskip (length), 264 \afterXtitle, 146 pagenote, 246 makeheadposition, 110 MakeTextUppercase, 107 \aliaspagestyle, 109 alltt (environment), 255, 256 alltt (package), 255, 375 Alph, 105 alph, 105 alphabet length, 15 alphabetic numbering, 105 \aliasoname, 306, 335 Alte Schwabacher, 469 altverse (environment), 259, 261, 265 amsmath (package), 375 amsmath (package), 375 and, 66 \andnext, 66 \andn
\afterXtitle, 146
According to the page of the
makeheadposition, 110  MakeTextUppercase, 107  \aliaspagestyle, 109  alltt (environment), 255, 256  alltt (package), 255, 375  Alph, 105  alph, 105  alphabet length, 15  alphabetic numbering, 105  \alsoname, 306, 335  Alte Schwabacher, 469  altverse (environment), 259, 261, 265  \ammame, 321, 335  amsfonts (package), 455  amsmath (package), 375  \and, 66  \andnext, 66  \andnext, 66  \andnext, 66  \anonymous division, 99  styling, 100-101  ans (file), 290  answer (environment), 290, 291  answer (package), 292  \anyptfilebase, 3, 464  \anyptsize, 3, 4, 464  \appendix, 78, 862  appendix, 78, 82, 88, 144, 153, 154  Arial, 353, 369  Arithmetic overflow, 444  array, 445  intercolumn space, 224  row fill, 218-219  row spacing, 224  array (environment), 205, 207, 211, 212,  218, 222, 224, 225, 445, 447, 448,  453, 455, 456, 461, 462  array (package), 205, 206, 336  \arraybackslash, 222  \arraytarraytlewidth (length), 224  \arraytostring, 331  \arraytostring, 331  \arraytostring, 331  \arraytolee (chapterstyle), 87  \arricle (class), 71, 87, 142  \arricle (chapterstyle), 87  \arricle (class), 71, 87, 142  \arricle (chapterstyle), 361  \asu(pagestyle), 360  \asuappendix (chapterstyle), 362  \arpendix (chapterstyle), 362  \arraypendix, 78, 82, 88, 144, 153, 154
MakeTextUppercase, 107       Arithmetic overflow, 444         \aliaspagestyle, 109       array, 445         alltt (environment), 255, 256       intercolumn space, 224         alltt (package), 255, 375       row fill, 218–219         Alph, 105       row spacing, 224         alph, 105       array (environment), 205, 207, 211, 212,         alphabet length, 15       218, 222, 224, 225, 445, 447, 448,         alphabetic numbering, 105       453, 455, 456, 461, 462         \alsoname, 306, 335       array (package), 205, 206, 336         Alte Schwabacher, 469       \arraycolsep (length), 224         \ammame, 321, 335       \arraycolsep (length), 224         \amsfonts (package), 455       \arraystretch, 223, 224         amsmath (package), 375       \arraytostring, 331         \and, 66       \arraytostring, 331         \and, 66       \arraytostring, 331         \anonymous division, 99       336         \styling, 100-101       article (chapterstyle), 87         \answer (environment), 290, 291       article (chapterstyle), 361         \anypendizes, 3, 464       asu (pagestyle), 360         \anypendizes, 3, 4, 464       asu (pagestyle), 360         \anypendizes, 78, 362       \attended (asumpendix (chapterstyle)), 362         \appendix, 78, 82, 88, 144, 153, 154
\aliaspagestyle, 109 alltt (environment), 255, 256 alltt (package), 255, 375 Alph, 105 alph, 105 alphabet length, 15 alphabetic numbering, 105 Alsoname, 306, 335 Alte Schwabacher, 469 altverse (environment), 259, 261, 265 \amname, 321, 335 amsath (package), 375 amsath (package), 375 ans (file), 290 answer (package), 292 Arraytostring, 331 arrow, see vector Arseneau, Donald, 53, 234, 238, 272, 274, arricle (class), 71, 87, 142 article (class), 71, 87, 142 article (class), 71, 87, 142 article (option), 5, 77, 84, 87, 93 asu (chapterstyle), 361 asu (pagestyle), 360 asuappendix (chapterstyle), 362 appendices (environment), 78 \attraction \text{AtBeginDocument}, 337 \attraction \text{AtBeginDocument}, 337 \attraction \text{AtBeginEnvironment}, 133
alltt (environment), 255, 256 alltt (package), 255, 375 Alph, 105 alph, 105 alph, 105 alph, 105 alphabet length, 15 alphabetic numbering, 105 Alte Schwabacher, 469 altverse (environment), 259, 261, 265 amsamt, (package), 455 amsamth (package), 375 and, 66 andnext, 66 andnext, 66 ans (file), 290 answer (environment), 290, 291 answer (package), 292 answer (environment), 78 appendize, 78, 362 appendix, 78, 82, 88, 144, 153, 154  intercolumn space, 224 row fill, 218–219 row spacing, 224 array (environment), 205, 207, 211, 212, 218, 222, 224, 225, 445, 447, 448, 453, 455, 456, 461, 462 array (package), 205, 206, 336 array (package), 202 arrayolsep (length), 224 arrayoulewidth (length), 224 arraystretch, 223, 224 arraytostring, 331 arraytostring, 361 arraytostring, 362 Arraytostring, 3
alltt (package), 255, 375  Alph, 105  alph, 224, 224, 225, 445, 447, 448, 453, 455, 456, 461, 462  alsoname, 306, 335  Alte Schwabacher, 469  altverse (environment), 259, 261, 265  altraybackslash, 222  altverse (length), 224  arraycolsep (length), 224  arraycolsep (length), 224  arraystretch, 223, 224  arraytostring, 331  arrow, see vector  Arseneau, Donald, 53, 234, 238, 272, 274,  anonymous division, 99  styling, 100–101  article (chapterstyle), 87  article (chapterstyle), 87  article (option), 5, 77, 84, 87, 93  asu (chapterstyle), 361  asu (pagestyle), 360  asuappendix (chapterstyle), 362  appendices (environment), 78  appendix, 78, 362  appendix, 78, 362  appendix, 78, 82, 88, 144, 153, 154  AtbeginEnvironment, 133
Alph, 105 alph, 105 alph, 105 alph, 105 alph, 105 alphabet length, 15 alphabet length, 15 alphabetic numbering, 105 alsoname, 306, 335 Alte Schwabacher, 469 altverse (environment), 259, 261, 265 amsfonts (package), 455 amsmath (package), 375 and, 66 anonymous division, 99 styling, 100–101 answer (environment), 290, 291 answer (package), 292 answer (package), 205, 206, 336 array(package), 205, 206, 336 arraybackslash, 222 arraybackslash, 222 arraycolsep (length), 224 arraystretch, 223, 224 arraystretch, 223, 224 arraytostring, 331 arrow, see vector Arseneau, Donald, 53, 234, 238, 272, 274, anonymous division, 99 styling, 100–101 article (chapterstyle), 87 article (chapterstyle), 87 article (chapterstyle), 361 aricle (chapterstyle), 361 asu (pagestyle), 360 asuappendix (chapterstyle), 362 appendices (environment), 78 appendix, 78, 362 appendix, 78, 82, 88, 144, 153, 154 AtBeginDocument, 133
alph, 105 alphabet length, 15 alphabet length, 15 alphabetic numbering, 105 alsoname, 306, 335 Alte Schwabacher, 469 altverse (environment), 259, 261, 265 almsame, 321, 335 amsamth (package), 455 and, 66 anraytostring, 331 and, 66 anraytostring, 331 ans (file), 290 answer (environment), 290, 291 answer (package), 292 answer (package), 292 answer (package), 292 answer (environment), 290, 291 answer (environment), 290, 291 answer (environment), 290, 291 answer (package), 375 answer (package), 292 answer (package), 360 answer (package), 362 appendices (environment), 78 answer (package), 362 appendix, 78, 362 appendix, 78, 362 AtbeginDocument, 337 AtbeginEnvironment, 133
alphabet length, 15
alphabetic numbering, 105
\alsoname, 306, 335 Alte Schwabacher, 469 altverse (environment), 259, 261, 265 \amname, 321, 335 \amsfonts (package), 455 \amsmath (package), 375 \and, 66 \annonymous division, 99 \styling, 100-101 \ans (file), 290 \answer (environment), 290, 291 \answer (package), 292 \answer (package), 360 \answer (package), 362 \ans
Alte Schwabacher, 469 altverse (environment), 259, 261, 265 \amname, 321, 335 amsfonts (package), 455 amsmath (package), 375 \and, 66 \andnext, 66 anonymous division, 99 styling, 100–101 ans (file), 290 answer (environment), 290, 291 answer (package), 292 answer (package), 292 anyptfilebase, 3, 464 anyptsize, 3, 4, 464 appendices (environment), 78 \appendix, 78, 362 appendix, 78, 82, 88, 144, 153, 154 \arraybackslash, 222 \arraybackslash, 224 \arraycolsep (length), 224 \array
altverse (environment), 259, 261, 265  \amname, 321, 335  amsfonts (package), 455  amsmath (package), 375  \and, 66  \annonymous division, 99  styling, 100-101  ans (file), 290  answer (environment), 290, 291  answer (package), 292  \anyptfilebase, 3, 464  anyptsize, 3, 4, 464  appendices (environment), 78  \appendix, 78, 362  appendix, 78, 82, 88, 144, 153, 154  \arraycolsep (length), 224  \arraycolsep (length), 31  \arraycolsep (length), 32  \arraycolsep (length), 224  \arraycolsep (length), 32  \arraycolsep (length), 224  \arraycolsep (length), 36  \arraycolsep (length), 23  \arraycolsep (length), 22  \array
\amname, 321, 335 \amsonts (package), 455 \amsonts (package), 375 \and, 66 \and arrow, see vector \and anonymous division, 99 \styling, 100-101 \ans (file), 290 \answer (environment), 290, 291 \answer (package), 292 \answer (package), 292 \answer (package), 292 \answer (package), 292 \answer (environment), 290, 291 \anyptfilebase, 3, 464 \anyptsize, 3, 4, 464 \appendices (environment), 78 \appendices (environment), 78 \appendix, 78, 362 \appendix, 78, 82, 88, 144, 153, 154 \arraystretch, 223, 224 \arraytostring, 331 \arraytostring, 321 \arr
amsfonts (package), 455
amsmath (package), 375 \and, 66 arrow, see vector \andnext, 66 anonymous division, 99 styling, 100-101 ans (file), 290 answer (environment), 290, 291 answer (package), 292 answer (package), 292 answer (package), 292 anyptfilebase, 3, 464 anyptfilebase, 3, 464 anyptsize, 3, 4, 464 appendices (environment), 78 \appendix, 78, 362 appendix, 78, 362 appendix, 78, 82, 88, 144, 153, 154 \arraytostring, 331 arrow, see vector Arseneau, Donald, 53, 234, 238, 272, 274, 336 article (chapterstyle), 87 article (class), 71, 87, 142 article (option), 5, 77, 84, 87, 93 asu (chapterstyle), 361 asu (pagestyle), 360 Asuappendix (chapterstyle), 362 AtBeginClass, 337 \AtBeginDocument, 337 \AtBeginEnvironment, 133
\and, 66 arrow, see vector \andnext, 66 Arseneau, Donald, 53, 234, 238, 272, 274, anonymous division, 99 336 styling, 100-101 article (chapterstyle), 87 ans (file), 290 article (class), 71, 87, 142 answer (environment), 290, 291 article (option), 5, 77, 84, 87, 93 answer (package), 292 asu (chapterstyle), 361 \anyptfilebase, 3, 464 asu (pagestyle), 360 \anyptsize, 3, 4, 464 asuappendix (chapterstyle), 362 appendices (environment), 78 \AtBeginClass, 337 \appendix, 78, 362 \AtBeginDocument, 337 appendix, 78, 82, 88, 144, 153, 154 \AtBeginEnvironment, 133
\and, 66 arrow, see vector \andnext, 66 Arseneau, Donald, 53, 234, 238, 272, 274, anonymous division, 99 336 styling, 100-101 article (chapterstyle), 87 ans (file), 290 article (class), 71, 87, 142 answer (environment), 290, 291 article (option), 5, 77, 84, 87, 93 answer (package), 292 asu (chapterstyle), 361 \anyptfilebase, 3, 464 asu (pagestyle), 360 \anyptsize, 3, 4, 464 asuappendix (chapterstyle), 362 appendices (environment), 78 \AtBeginClass, 337 \appendix, 78, 362 \AtBeginDocument, 337 appendix, 78, 82, 88, 144, 153, 154 \AtBeginEnvironment, 133
\andnext, 66
anonymous division, 99
styling, 100–101  article (chapterstyle), 87  ans (file), 290  answer (environment), 290, 291  answer (package), 292  answer (package), 292  asu (chapterstyle), 361  anyptfilebase, 3, 464  asu (pagestyle), 360  anyptsize, 3, 4, 464  asuappendix (chapterstyle), 362  appendices (environment), 78  AtBeginClass, 337  AtBeginDocument, 337  appendix, 78, 82, 88, 144, 153, 154  AtBeginEnvironment, 133
ans (file), 290 article (class), 71, 87, 142 answer (environment), 290, 291 article (option), 5, 77, 84, 87, 93 answer (package), 292 asu (chapterstyle), 361 anyptfilebase, 3, 464 asu (pagestyle), 360 anyptsize, 3, 4, 464 asuappendix (chapterstyle), 362 appendices (environment), 78 \appendix, 78, 362 appendix, 78, 362 appendix, 78, 82, 88, 144, 153, 154 \attrice{AtBeginEnvironment}, 133
answer (environment), 290, 291  answer (package), 292  asu (chapterstyle), 361  anyptfilebase, 3, 464  asu (pagestyle), 360  anyptsize, 3, 4, 464  asuappendix (chapterstyle), 362  appendices (environment), 78  AtBeginClass, 337  AtBeginDocument, 337  appendix, 78, 82, 88, 144, 153, 154  AtBeginEnvironment, 133
answer (package), 292  asu (chapterstyle), 361  asu (pagestyle), 360  asu appendix (chapterstyle), 362  asuappendix (chapterstyle), 362  appendices (environment), 78  AtBeginDocument, 337  appendix, 78, 82, 88, 144, 153, 154  AtBeginEnvironment, 133
\anyptfilebase, 3, 464
\anyptsize, 3, 4, 464
appendices (environment), 78 \AtBeginClass, 337 \AtBeginDocument, 337 appendix, 78, 82, 88, 144, 153, 154 \AtBeginEnvironment, 133
\appendix, 78, 362 \AtBeginDocument, 337 appendix, 78, 82, 88, 144, 153, 154 \AtBeginEnvironment, 133
appendix, 78, 82, 88, 144, 153, 154 \AtBeginEnvironment, 133
appendix (package), 336 \AtBeginPackage, 337
\appendixpage, 78, 82, 155, 335, 401, 404 \atcentercr, 125
\appendixpage*, 78 patchcmd, 462
\appendixpagename, 78, 335 date, 66
\appendixrefname, 294 \AtEndClass, 337, 338
\appendixtableofcontents, 404 \AtEndDocument, 337
\appendixtocname, 78, 335 \AtEndFile, 337
arabic, 105 \AtEndOfClass, 337, 338

\AtEndOfPackage blank space

\AtEndOfPackage, 337, 338	Bernhard Modern, 469
\AtEndPackage, 337, 338	Bezier curve, see also picture object
\atendtheglossaryhook, 314	control points, 422
ATF Clearface, 469	linearized rendition, 423
\author, 57, 64, 66, 459, 465	Bezier, Pierre, 421
\autocols, 227, 228	\bfseries, 42, 81, 82, 86, 96, 133
\autorows, 226-228	bianchi (chapterstyle), 90
aux (file), 284, 293, 427, 428	Bianchi, Stefano, 90
Avant Garde Gothic, 16, 38	bib (file), 299, 300
	\bibindent,5
В	\bibintoc, 297
	\bibitem, 297, 299, 458, 459, 465
b (position argument), 168, 175, 180, 409,	bibitemlist (environment), 297-299
418	\bibitemsep (length), 299
b3paper (option), 1	\biblabel, 460
b4paper (option), 1	bibliographic database, 299
b5paper (option), 1	\bibliography, 299
b6paper (option), 1	bibliography, 154, 164, 254, 297–300
back matter, 75, 297	explanatory text, 298
\backmatter, 75, 77, 78	flushleft entries, 298
Bad \line or \vector argument, 454	heading, 298
Bad math environment delimeter, 454	label styling, 299
Bad \sidebarmargin argument, 464	list styling, 298–299
Barbedour, 469	title in ToC, 297
Basel, 469	\bibliographystyle, 299
$\begin{tabular}{l} \begin{tabular}{l} tabu$	\biblistextra, 298, 363
\baselineskip (length), 21, 23, 25, 50,	\bibmark, 298
51, 121, 122, 201, 251, 333, 378	\bibname, 297, 335
Basilia, 469	\bibsection, 297, 298, 363
Baskerville, 15, 469	BibTeX database, 300
bch (fontfamily), 39	BibTeX (program), 299, 300
\beforebookskip, 80	BibTeX style, 299, 300
\beforechapskip, 86	abbrv, 300
\beforechapskip (length), 85, 86, 395	alpha, 300
\beforeepigraphskip (length), 251	changing, 300
\beforepartskip, 80, 254	plain, 300
\beforePoemTitleskip (length), 264	unsrt, 300
\begin, 157, 282, 454, 455	\bicaption, 191
ended by , 454	\bicontcaption, 192
\begintheglossaryhook, 314	bidi (package), 342, 343
\belowcaptionskip, 204	bilingual captions, 190–192
\belowcaptionskip (length), 201	binding allowance, 19, 23, 25
\belowrulesep (length), 216, 217	\bionenumcaption, 190
Bembo, 469	bitmap, 35
Benguiat, Ed, 39	font, 35, 49
Benton, Morris, 40	\bitwonumcaption, 190
Bergamo, 469	blank space, 259

blockdescription (environment), 131	\box255, 449
\blockdescriptionlabel, 131	boxed verbatim, 279
Bodoni, 469	boxedverbatim (environment), 279-281,
body font, 41, 44	286, 287
boek (class), 91, 104	<pre>boxedverbatim* (environment), 279</pre>
\boldmath, 458	\boxedverbatiminput, 286
\book, 76, 77, 79, 80, 82, 84, 107, 147, 148,	\boxedverbatiminput*, 286
150, 155, 156, 253, 254, 335, 339,	\break, 125
401	\Bref, 294, 335
book, 76	bringhurst (chapterstyle), 90, 104
number, 77, 79	bringhurst (headstyles), 104
book (class), 71, 87, 88, 102, 104, 111, 142	bringhurst (package), 352
book (pagestyle), 80, 107	Bringhurst, Robert, 13
\book*, 339	\bringpicl, 350
\bookblankpage, 82	\bringpicr, 350
Bookman, 15, 16, 39	broadsheetpaper (option), 2
bookmark (package), 165	brotherton (chapterstyle), 90
\bookname, 81, 335	\bs, 340
\booknamefont, 81, 102	bst (file), 300
\booknamenum, 81	buffer size, 452
\booknumberline, 150	\bvbox, 279-281
\booknumberlinebox, 150	\bvboxsep (length), 279, 280
\booknumberlinehook, 150	\bvendofpage, 280
\booknumfont, 81, 102	\bvendrulehook, 280
\bookpageend, 82	\bvleftsidehook, 280
\bookpagemark, 82	\bvnumberinside, 281
\bookrefname, 294, 335	\bvnumbersinside, 281
booktabs (package), 205, 216, 336	\bvnumbersoutside, 281
\booktitlefont, 81, 102	\bvperpagefalse, 280
\bottomfraction, 179-181	\bvperpagetrue, 280
bottomnumber (counter), 179	\bvrightsidehook, 280
\bottomrule, 216, 217	\bvsides, 279-281
\bottomsectionpenalty, 77, 78	\bvtopandtail, 279-281
\bottomsectionskip (length), 77, 78	\bvtopmidhook, 280
\box, 444	\bvtopofpage, 280
box, 167, 225, 271, 408	\bvtoprulehook, 280
framed, 272–276, 408	by (keyword), 433
include pagebreak, 272	C
picture, 406	С
reference point, 408	- (ition organizat) 22( 227
rounded, see also oval	c (position argument), 226, 227
half, 417	calc (package), 25, 27, 239, 399, 400
quarter, 417	\calcentering, 66, 130, 131
saved object, 412	Can only be used in the preamble, 454
shaded background, 272	\cancelthanksrule, 69
unbreakable, 406	Cannot change a macro that has delimited arguments 462
using saved object, 412	ited arguments, 462

\caption \cftpartbreak

\caption, 140, 148, 154, 165, 167, 185, 186,	\cases, 213
188–190, 192–194, 199, 200, 203,	Caslon, 469
226, 293, 295, 296, 315, 335, 339,	ccaption (package), 181, 336
351, 453, 454	\cdot, 208
caption, 75, 154, 167, 181–204	Centaur, 469
LaTeX methods, 199–202	\center, 123
anonymous, 181, 186–188	center (environment), 72, 123, 130, 137,
bilingual, 190–192	138, 201, 377
in list of, 190	\centerfloat, 123
styling, 192	\centering, 122-125, 182, 184, 195, 199,
continuation, 181	201, 222, 264, 377
continued, 185–186	centering,
outside a float, 189	float, 123
delimeter, 182, 202	vertical, 336
fixed, 172, 188–190, 226	\centerlastline, 182, 185, 195
font, 182, 184, 201, 202	Century, 353
footnote, 202–203	Century Old Style, 63, 469
multiline, 182, 185	\cftaddnumtitleline, 154
multiple, 172	\cftaddtitleline, 154
new subcaption, 174	\cftbeforeKskip (length), 148
on opposite page, 189	\cftbookbreak, 148
outside a float, 188–190	\cftchapterbreak, 148
paragraph style, 182	\cftchapterleader, 151
ruled, 183	\cftdot, 147
short style, 182	\cftdotsep, 147, 150
side, 196–199	\cftinsertcode, 155
side caption, 183	\cftinserthook, 155
space above, 201	\cftKafterpnum, 151, 152, 156
style, 167, 181–185	\cftKaftersnum, 149, 150, 152, 157
subcaption, 192–196	\cftKaftersnumb, 149, 150, 152, 157
continued, 194	\cftKdotsep, 150
in list of, 193	\cftKfont, 149, 152, 156
referencing, 194	\cftKformatpnum, 151
styling, 195	\cftKformatpnumhook, 151
width, 183	\cftKindent (length), 148
\caption outside float, 454	\cftKleader, 150, 152, 153
caption (package), 203, 204	CftKname, 149
\captiondelim, 181, 182	\cftKnumwidth (length), 148, 149
\captionnamefont, 182, 197	\cftKpagefont, 151
\captionstyle, 182	\cftKpresnum, 149, 150, 157
\captiontitlefinal, 183	\cftlocalchange, 154
\captiontitlefont, 182	\cftnodots, 147, 150
\captionwidth, 183	\cftpagenumbersoff, 153, 164
\cardinal,327	\cftpagenumberson, 153
cardinal,	\cftparfillskip,153
Carnase, Tom, 38	\cftparskip (length), 147
Carter, Matthew, 39	\cftpartbreak, $1\overline{48}$

\cftsetindents, 151, 163	crosshead, 90, 104
\cftwhatismyname, 150, 400	culver, 90
\changecaptionwidth, 183	dash, 91
\changed, 317, 318	default, 87–90, 252
\changeglossactual, 313	demo, 393
\changeglossnum, 313, 314	demo2, 91, 393
\changeglossnumformat, 313	demo3, 91, 104, 393
\changeglossref,313	dowding, 91, 104
\changemarks, 317	ell, 91
changepage (package), 325	fred, 87
Chantilly, 469	ger, 91
\chapindent (length), 89	hangnum, 87, 89
\chapnamefont, 86, 102	komalike, 91, 104
\chapnumfont, 86, 102	lyhne, 91
chappell (chapterstyle), 90, 392	madsen, 91
\chapter, 5, 75-80, 84, 86, 87, 93, 107, 108,	ntglike, 91, 104
115, 142, 144, 148, 150, 155, 156,	pedersen, 75, 91, 92, 104, 393
162, 251–253, 263, 293, 296, 335,	reparticle, 87
339, 400, 401	section, 87, 89
chapter, 4, 5, 75, 76, 78, 140	southall, 92, 394
number, 77, 78	tandh, 92, 104
precis, 92–94	thatcher, 92
chapter (counter), 288	veelo, 92, 395, 396
chapter (pagestyle), 77, 84, 107, 146	verville, 92
\chapter*, 77, 86, 251, 252, 339	wilsondob, 92, 104
chapterbib (package), 298	\chaptertoc, 403
\chapterheadstart, 85, 86	\chaptitlefont, 86, 102
\chaptermark, 108, 114	Characters dropped after , 465
\chaptername, 78, 247, 335	Charter, 16, 39
\chapternamenum, 86	\checkandfixthelayout, 22, 23, 25, 26,
\chapternumberline, 150	29, 33, 53, 196, 240, 346
\chapternumberlinebox, 150	\checkarrayindex, 331
\chapternumberlinehook, 150	\checkifinteger, 332
\chapterprecis, 92, 93, 317, 403	\checkoddpage, 325
\chapterprecishere, 93	\checkthelayout, 23, 24, 463
\chapterprecistoc, 93, 156, 403	Cheltenham, 469
\chapterrefname, 294, 335	chngcntr (package), 322, 336
\chapterstyle, 87, 88	chngpage (package), 129, 325
chapterstyle, 87–92, 379–396	\circle, 415, 416, 459
article, 87	circle, see also picture object
asu, 361	disk, 415
asuappendix, 362	open, 415
bianchi, 90	reference point, 415
bringhurst, 90, 104	restricted diameter, 416
brotherton, 90	\circle*, 415, 416
chappell, 90, 392	Citation on page, 458
companion, 87, 89, 377	Citation undefined, 458

\cite \copy

\cite, 54, 297, 299, 458-460	\cmidrulewidth (length), 217
cite bibliographic item, 297	colophon, 130
\citeindexfile, 310	color (package), 91, 272, 375
\citeindextrue, 310	\colorchapnum, 394
Clarendon, 469	\colorchaptitle, 394
class	
	column, 13
article, 71, 87, 142	double, 4, 13, 20, 71, 73, 116, 375
boek, 91, 104	multiple, 14, 175
book, 71, 87, 88, 102, 104, 111, 142	single, 14, 71, 73
iso, 317	Column is already defined, 465
memoir, 1, 101, 142, 144, 145, 163, 257,	\columnsep (length), 20, 24
295, 333	\columnseprule (length), 20, 24
report, 64, 71, 88, 102, 142	Command \ already defined, 341,
	454
scrbook, 91, 104	
class option, 44	Command invalid, 454, 458
class options, 1	Command not provided, 455
article, 5	comment (environment), 276, 277
bibliography, 5	comment out text, 276
fonts, 5	comment (package), 277
layout truncation, 5	\commentsoff, 276
math, 5	\commentson, 276
stock size, 1	companion (chapterstyle), 87, 89, 377
type size, 2	companion (pagestyle), 107, 114, 115, 117,
\cleaders, 101	377
· · · · · · · · · · · · · · · · · · ·	
\cleardoublepage, 106, 107, 175, 190,	Computer Modern, 3, 15, 16
253, 326	Computer Sans, 15, 16
cleared (pagestyle), 107	Computer Typewriter, 15, 16
\clearforchapter, 84	concordance, 301
\clearmark, 113	Concrete Roman, 15, 16
\clearpage, 84, 175, 176, 189, 326, 453	conditional, 437
\clearplainmark, 113	configuration file
\cleartoevenpage, 190, 253, 326	MakeIndex, 304
\cleartooddpage, 326	Connes, Frederic, 304
\cleartorecto, 84, 107, 326	\contcaption, 185
\cleartoverso, 84, 107, 326	\contentsline, 140, 142, 154
\cline, 224	\contentsname, 145, 146, 335
clo (file), 464	\continuousmarks, 67
\closeinputstream, 285	\continuousnotenums, 245, 247
\closeoutputstream, 285	control sequence, 426
\club, 101	\contsubbottom, 194
\cmd, 340	\contsubcaption, 194
\cmdname, 83	\contsubtop, 194
\cmdname*, 83	\contsuptop, 194
\cmdprint, 340	coordinate, 405
\cmidrule, 217, 218	coordinate pair, 405
\cmidrulekern (length), 217, 218	coordinate system, 405
\cmidrulesep (length), 217	\copy, 444
= . •	

copyfitting, 13	crosshead (headstyles), 104
\copypagestyle, 109	crownvopaper (option), 2
\count, 456	\cs, 340
count, 432	\csname, 431, 445, 448
counter, 167, 432	\cstyle, 377
bottomnumber, 179	ctabular (environment), 225, 226
chapter, 288	CTAN, 35
dbltopnumber, 179	стт, 342, 395
footnote, 67, 229	culver (chapterstyle), 90
increment, 290	Culver, Christopher, 90
lastpage, 320	\currenttitle, 296
lastsheet, 320	
maxsecnumdepth, 77	D
page, 105	
pagenote, 245, 247	dash (chapterstyle), 91
poemline, 262	\dashbox, 409, 410
pseudo, 288, 289	\date, 57, 64, 66, 465
question, 292	datetime (package), 321
secnumdepth, 77, 79	dbillpaper (option), 2
sheetsequence, 320	$\d$ oblfloatpagefraction, $179$
tocdepth, 139, 145, 158, 162, 168	\dblfloatsep (length), 179
topnumber, 179	\dbltextfloatsep (length), 179
totalnumber, 179	\dbltopfraction, 179
Xdepth, 168	dbltopnumber (counter), 179
Counter already defined, 465	dcolumn (package), 205, 208, 336
counter representation, 105	\def, 430, 451
Alph, 105	default,
alph, 105	page layout, 7
arabic, $105$	printing options, 4
Roman, 105	stock, 1
roman, 105	type size, 3
Counter too large, 455	default (chapterstyle), 87–90, 252
\counterwithin, 322	default (headstyles), 102–104
\counterwithin*, 322	\defaultaddspace, 217
\counterwithout, 322	\defaultaddspace (length), 217, 218
\counterwithout*,322	\defaultlists, 133, 134
Courier, 15, 16, 39	\defaultsecnum, 97
\cplabel, 325	delarray (package), 205, 336
\cr, 445, 447	\delcode, 449
$\createmark, 112, 113, 463$	\deleted, 317, 318
\createplainmark, 112, 113, 463	\DeleteShortVerb, 277, 278
\Cref, 294, 335	\delimeter, 449
crop (package), 336, 397	Della Robbia, 469
cross reference, 293	demo (chapterstyle), 393
automatic, 293	demo2 (chapterstyle), 91, 393
specified, 293	demo3 (chapterstyle), 91, 104, 393
crosshead (chapterstyle), 90, 104	demyvopaper (option), 2

description (environment), 131, 134	ebook (option), 1, 5
\descriptionlabel, 131	secmark, 108
diagram, 405	sectionname, 79
\diamond, 101	\edef, 430
\dimen, 433-435	seename, 306
dimension, 433	leftmargini (length), 258
Dimension too large, 444	Egyptian, 469
\ding, 100	ell (chapterstyle), 91
\DisemulatePackage, 336	Els, Danie, 83, 149
disk, see circle	\else, 437, 440, 441
Display math should end with \$\$, 445	\eminnershape, 44
\divide, 433-436	\emph, 43, 44
divide by zero, 444	emphasis, 43
division,	Empty 'thebibliography' environment,
anonymous, 99	465
sectional, see also subhead, 79	empty (pagestyle), 65, 83, 106, 107, 109,
\do, 440	111, 146, 360, 361
Do not use \footnote, 465	Empty preamble: 'I' used, 462
\doccoltocetc, 144, 145	\emptythanks, 67
document (environment), 5, 337, 427, 428	\EmulatedPackage, 336, 337
\documentclass, 1, 3, 5, 16, 41, 455, 457,	\EmulatedPackageWithOptions, 336, 337
458	\end, 157, 282, 445, 450, 454
double column,	end
index, 300, 301	line, 411
double spacing, 50	end floats, 286-289
Double subscript, 445	\end occurred inside a group, 445
Double superscript, 445	\end occurred when, 445
\doublerulesep (length), 216, 217, 224	\endcsname, 431, 445, 448
\DoubleSpacing, 50, 51, 358	\endgroup, 445, 448
\DoubleSpacing*, $50$	\endinput, 352
dowding (chapterstyle), 91, 104	\endlinechar, 160
dowding (headstyles), 104	\endlist, 136
\downbracefill, 218	\endMakeFramed, 273
Downes, Michael, 53, 324	\endnote, 243
\dp, 432	endnote
draft document, 111	mark, see also reference mark
draft (option), 4, 111, 290, 317	endnotes, 286, 292, 364
draw, 405	endnotes (package), 243
\dropchapter, 252	\endwriteverbatim, 289
\droptitle (length), 64	English Serif, 469
dvi (file), 284, 426, 428	Engravers Litho, 469
Dye, Thomas, 92, 394	Engravers Old English, 469
	\enlargethispage, 52, 365, 456
E	\enlargthispage, 456
	ent (file), 243-245, 247
headstyles, 102	enumerate (environment), 122, 131-134,
\easypagecheck, 129, 325	293, 466

enumerate (package), 132, 336	margintable, 169
enumitem (package), 133, 134	midsloppypar,52
environment,	minipage, 52, 172, 188, 193, 198, 199,
abstract, xxiii, 71-73, 335	229, 271, 350
adjustwidth, 90, 129, 130, 134, 276	multicols, 157
adjustwidth*, 66, 129	new, 290
alltt, 255, 256	onecolabstract,73
altverse, 259, 261, 265	patverse, 259, 261, 267, 331, 461
answer, 290, 291	patverse*, 261, 262, 461
appendices, 78	picture, 405-409, 419, 428, 453, 459
array, 205, 207, 211, 212, 218, 222,	qfame, 276
224, 225, 445, 447, 448, 453, 455,	qframe, 276
456, 461, 462	qshade, 276
bibitemlist, 297-299	quotation, 71, 124, 129, 131, 134,
blockdescription, 131	276, 457
boxedverbatim, 279-281, 286, 287	quote, 93, 124, 129, 131, 134
boxedverbatim*, 279	shaded, 272, 273, 276
center, 72, 123, 130, 137, 138, 201,	sidecaption, 196, 197, 199
377	sidecontcaption, 197
comment, 276, 277	sidelegend, 197
ctabular, 225, 226	sidenamedlegend, 197
description, 131, 134	sloppypar,52
document, 5, 337, 427, 428	snugshade, 272, 273
enumerate, 122, 131-134, 293, 466	snugshaded, 272
epigraphs, 249-252	subappendices, 78, 79
eqnarray, 457	symbols, 136
equation, 293	syntax, 377
fboxverbatim, 279, 289	tabbing, 454, 456, 457
figure, 167, 169, 187, 189, 200, 287,	table, 167, 169, 187, 188, 226, 454
288, 405, 454, 456	tabular, 172, 199, 205, 207, 211, 220-
flexlabelled, 132	222, 224, 225, 229, 377, 445, 447,
flushleft, 72, 123, 138	448, 453, 455, 456, 461, 462
flushright, 72, 123, 138	tabular*, 205, 220-222, 224
framed, 169, 272-274, 276	tabularx, 205, 220-223, 466
framedminipage, 271, 272	thebibliography, 107, 297, 298, 314,
frametitle,275	456, 465
framewithtitle, 274, 275	theglossary, 311, 314
hangparas, 122	theindex, 107, 301, 303, 314
itemize, 131, 132, 134, 239, 321, 456	tightcenter, 377
KeepFromToc, $144$	titledframe, 275
labelled, 131, 132	titlepage,65
lcode, 282, 283, 378	titlingpage, 65, 66, 69, 106, 107
leftbar, 273	trivlist, 136-138, 283, 377
list, 72, 131, 134, 136, 258, 378	verbatim, 138, 256, 277-279, 282,
lrbox, 271	283, 286, 378, 453, 465
MakeFramed, 273	verbatim*,277-279
marginfigure.169	verbatimoutput, 285

verse, 255-258, 260-262	\ext@type, 200
vminipage, 52	extend a macro, 323
vplace, 336	Extra, 445
writefigure, 288, 289	Extra or forgotten, 445
writeverbatim, 285, 289	Extra alignment tab, 445
Environment already defined, 465	Extra \else, 445
Environment undefined, 455	Extra \endcsname, 445
\epigraph, 249-252	Extra \fi, 445
epigraph, 85, 249–254	Extra \or, 445
epigraph (package), 249, 336	Extra \right, 445
epigraph (pagestyle), 107, 253	\extracolsep, 211
\epigraphfontsize, 251	extrafontsizes (option), 2, 3, 49, 463
\epigraphforheader, 252	\extrarowheight (length), 224
\epigraphhead, 107, 251, 252	\extratabsurround, 225
\epigraphpicture, 252	\extratabsurround (length), 225
\epigraphposition, 250	textwidth, 241
\epigraphrule (length), 251	textwidth (length), 110
epigraphs (environment), 249–252	
\epigraphsourceposition, 250	F
\epigraphtextposition, 250	
\epigraphwidth (length), 250, 252	fancybox (package), 273
eqnarray (environment), 457	\fancybreak, $99$ , $100$
equation (environment), 293	\fancybreak*,99
error, 443	fancyhdr (package), 106, 336
LaTeX, 453–458	\fbox, 271-273, 279
memoir class, 461–464	\fboxrule (length), 292
TeX, 444–453	\fboxsep (length), 292
error message,	fboxverbatim (environment), 279, 289
response, 443	\fcardinal,327
continue, 443	\feetabovefloat, 230
edit, 443	\feetbelowfloat, 230
exit, 443	Fette Fraktur, 469
help, 443	Fette Gotisch, 469
insert, 443	\fi, 437, 440
quiet, 443	\fibseries, 438-440
run, 443	\fibtogo, 439
scroll, 443	figure, 117, 129, 142, 151, 154, 163, 167,
descriptionlabel, 131	168, 171, 181, 189, 190, 192, 193,
eso-pic (package), 399	201, 203, 253, 345, 351, 405
etex, 332	float definition, 167
etex (package), 375	reference, 294
etoolbox (package), 133, 399	sub-, 163
\evensidemargin (length), 24	subfigure, 192–196
\everydisplay, 51	figure (environment), 167, 169, 187, 189,
\everypar, 128	200, 287, 288, 405, 454, 456
exception dictionary, 452	\figurename, 167, 202, 335
executivepaper (option), 2	\figurerefname, 294, 335

figures \fnsymbol

figures,	\flegtable, 188
file, 284–292	\flegtocfigure, 188
ans, 290	$\floor 188$
aux, 284, 293, 427, 428	\flegtoctable, $188$
bib, 299, 300	fleqn (option), 5
bst, 300	fleuron, 100, 470
clo, 464	flexlabelled (environment), 132
close, 285	float, 5, 75, 167, 181, 183, 185–189, 192,
dvi, 284, 426, 428	194, 200, 203, 229, 326, 351, 375
ent, 243-245, 247	bottom, 175, 176, 179
idx, 300-302, 304, 310	centering, 123
ind, 300, 301, 304, 309	double column, 167, 175, 179, 181
ist, 304, 307	flush, 175
jobname.idx, 302	framed, 169
jobname.ind,301	here, 175, 176, 179
lof, 140	multiple, 171–175
log, 284, 336, 428, 443	table and figure, 172
lot, 140	new, 167–169
mem10.clo, 464	new diagram, 168
open, 285	new subfloat, 174
pdf, 284	page, 117–119, 175, 176
read, 271, 284–286, 289	parameters, 176
single line, 286	placement, 175–181
verbatim, 286	position, 168
source2e.tex,430	ruled, 170
tex, 284	set off, 169–171
toc, 93, 139, 140, 156, 243, 284, 428	single column, 167, 175
write, 271, 284–285, 288, 290	subfloat, 192–196
verbatim, 285	suppress, 176
File ended while scanning, 446	suppress bottom, 176
File not found, 455	suppress top, 176
fill1,436	top, 175, 176, 179
fill, 436	Float too large, 459
fil,436	Float(s) lost, 455
final (option), 4, 290, 317	\FloatBlock, 176
\finishwritefigure, 289	\FloatBlockAllowAbove, 176
\firmlist, 134	\FloatBlockAllowBelow, 176
\firmlists, 133, 134	floatcomp (pagestyle), 117
\firmlists*, 133, 134	\floatpagefraction, 179-181
\firsthline, 225	\floatsep (length), 179, 181
fixltx2e (package), 116, 375	\flushbottom, 5, 23, 24, 77, 326
\fixthelayout, 23, 25, 467	flushleft, 123
flafter (package), 176	flushleft (environment), 72, 123, 138
\flagverse, 262, 268	\flushleftright, 125
Flanders, 469	flushright, 123
\flegfigure, 188	flushright (environment), 72, 123, 138
\flegfloat, 188	\fnsymbol, 235

\fnum@figure \footruleheight

\fnum@figure, 201, 202	\footfudgefiddle, 231
\fnum@type, 200, 201	\footmarksep (length), 232-234
\fnumbersep, 327	\footmarkstyle, 233
folio, 21, 75, 106, 116, 164, 165, 345, 349	\footmarkwidth (length), 232-234
changing representation, 105	footmisc (package), 230, 232
font	-
	\footnote, 102, 203, 229-232, 240, 241,
bitmap, 35, 49	243, 248, 465
Computer Modern, 35	footnote, 44, 67–69, 102, 236
Knuthian, 15	bottom float, 230
measuring, 13	fragile, 203
Metafont, 15	in caption, 202–203, 444
Metafont, 35	in float, 203
monospaced, 15	in heading, 101–102, 444
Opentype, 35	mark, see also reference mark, 67, 102,
outline, 35, 44, 49	203
PostScript, 15, 35, 469	marker, 232
TrueType, 35, 469	multiple, 234
Font not loadable, 446	styling, 232
font characteristic,	marker separator, 230
family, 41, 42	new series, 231
series, 41, 42	reference, 229
shape, 41, 42	style, 232
Font command is not supported, 462	styles, 231
font commands, 41	styling, 232–236
font declarations, 42	symbol, 235
Font shape, 459	order, 235
font size, 44	text, 203, 232, 233
fontenc (package), 375	font, 234
fontfamily, 38	too long, 231
bch, 39	verbatim text, 230
pag, 38	footnote (counter), 67, 229
pbk, 39	footnote, 각주, 69
<del>-</del>	•
pcr, 39	\footnotemark, 68, 203, 229, 232, 248
phv, 39	\footnoterule, 69, 232, 235
pnc, 40	footnotes,
ppl, 40	as a paragraph, 231
psy, 41	as paragraphs, 231
ptm, 40	as three columns, 231
put, 40	as two columns, 231
pzc, 41	\footnotesatfoot, 240
pzd, 41	\footnotesinmargin, 240
FontSite, 63	\footnotesize, 44-46, 69, 134, 234, 242,
fonttable (package), 376	251
foolscapvopaper (option), 2	\footnotetext, 203, 229, 240, 248
footer, 7, 21, 106–108, 110, 114–117, 317,	\footparindent (length), 233
345	\footref, 229
\footfootmark, 233	\footruleheight, 110

\footruleskip gutter

\footruleskip, 110	fred (chapterstyle), 87
\footskip is too large, 462	\freetabcaption, 226
\footskip (length), 21, 23, 24, 27, 346,	\fref, 294, 335
462	front matter, 75
\foottextfont, 234, 358	frontispiece, 154
\foottopagenote, 248	\frontmatter, 72, 75, 80, 86, 339
Forbidden control sequence found, 446	\frontmatter*,75
Ford, Matthew, 260	Frutiger, Adrian, 39
\fordinal, 327	full stop, see also period
fore-edge, 21	fullptlayout (option), 5
\foremargin (length), 23, 24, 238, 399,	Function, 469
463	\fussy,52
\frac, 330	Futura, 469
fragile, 157, 296, 444	\futurelet, 324
fragile command, 101	
\frame, 410	G
frame,	
box, 272–276	Garamond, 353, 369, 469
styling, 273	Garcia, Gerardo, 91
minipage, 271	gathering, 320
verbatim, 271	\gdef,430
narrow box, 273	geometry (package), 7, 336
rounded corners, 273	ger (chapterstyle), 91
title, 274–275	\getarrayelement, 331
verbatim, 279, 286	$\gray \gray \gra$
styling, 280	Gill Sans, 469
\framebox, 408, 409	\global, 430
\FrameCommand, 273	\glossary, 310-312, 316, 453
framed,	\glossarycolsep (length), 314
box, 408	\glossaryentry,313
float, 169	\glossaryintoc,314
verbatim, 289	\glossarymark, 314
framed (environment), 169, 272-274, 276	\glossaryname, 314, 335
framed (package), 272, 275, 336	\glossaryrule (length), 314
framedminipage (environment), 271, 272	\glossitem, 311-313, 315
\FrameHeightAdjust, 273	glue, 434
\FrameHeightAdjust (length), 272	Glypha, 469
\framepichead, 119	Glytus, 469
\framepichook, 119	golden section, 27
\framepictextfoot, 119	Goudy Old Style, 469
\FrameRestore, 273	\gparindent (length), 283, 284, 378
\FrameRule (length), 272	graphicx (package), 91, 92, 375, 387, 388
\FrameSep (length), 272	391, 396
\FrameTitle, 274	group, 430
frametitle (environment), 275	gutter, 20
framewithtitle (environment), 274, 275	width, 20
Franklin Gothic, 469	

Н	dowding, 104
	komalike, 104
h (position argument), 175, 180	memman, 103, 104
h float specifier, 459	ntglike, 104
haeder, 107	tandh, 104
\halign, 445, 448	wilsondob, 104
\hangcaption, 182	\headwidth (length), 110
\hangfrom, 122	\heavyrulewidth (length), 216, 217
hanging,	Helvetica, 15, 16, 39
hanging (package), 122	\hfil, 101, 149, 153, 436
hangnum (chapterstyle), 87, 89	\hfill, 153, 172, 185, 436
\hangpara, 122	\hfilneg, 436
hangparas (environment), 122	Høgholm, Morten, 14, 15
\hangsecnum, 97	\hhrule, 172
\hangsubcaption, 195, 196	\hideindexmarks, 303
chapter, 80	\hline, 224
hash size, 453	\hmpunct, 321
\hb@xt@, 431	horizontal mode, 431
\hbox, 101, 431, 444, 449, 451	\hrule, 172, 280
\headdrop (length), 21	\hrulefill, 218
header, 7, 21, 27, 76, 101, 102, 106–110,	\hsize (length), 273
114–117, 251, 317, 345, 349	\hskip, 436
specifying size, 21, 23	\hss, 436
\headheight and/or \headsep are too	\ht, 432
large, 462	\HUGE, 45, 46, 49
\headheight (length), 19, 21, 23, 24, 27,	\Huge, 45, 46, 81, 82, 86
462	\huge, 45, 46, 81, 86
heading, 5, 144, 176, 250	Huge page cannot be shipped out, 446
abstract, 71	Hurenkinder, see widow
book, 80–83	hyperindex, 304
chapter, 5, 84–87, 92, 154, 251, 253	hyperref (package), 145, 149, 156, 159,
part, 5, 80–83	160, 165, 194, 246, 294, 303, 304,
sections, 5, 94–99	338, 340
장, 5	\hyphenation, $447$ , $449$ , $452$
절, 5	hyphenation, 239
편, 5	
heading (pagestyle), 106	I
headings (pagestyle), 7, 106, 107, 111, 114	
\headmargin (length), 23, 24, 27	I can't find file, 446
\headnameref, 295	I can't go on meeting you like this, 446
\headsep (length), 19, 21, 24, 27, 349, 462	I can't write on file, 446
\headskip, 23	sidecapfloatwidth, 198
\headstyles, 102	\idtextinnotes, 246
headstyles,	idx (file), 300–302, 304, 310
bringhurst, 104	\idxmark, 116
crosshead, 104	if-then-else, 437
default, 102–104	\ifbounderror, 331

\ifcase \itshape

\ifcase, 440, 441	main entry, 303
\ifdim, 437	multiple, 257, 300, 302
\ifdraftdoc, 111, 290, 317	name, 301
\ifetex, 332	preparation, 301
ifetex (package), 332	print, 301
\ifinteger, 332	printing, 300–301
\ifluatex, 333	see also reference, 302, 306
ifmtarg (package), 336	see reference, 302, 306
\ifnum, 437	show indexed items, 303
\ifodd, 437	single column, 300, 301
\ifoddpage, 325	subentry, 303
\ifonlyfloats, 117	subsubentry, 303
\ifpdf, 332	title in ToC, 301
ifpdf (package), 332, 336	Index for pattern, 465
\ifsamename, 324	Index outside limits, 462
\ifscapmargleft, 196	index (package), 257, 336, 337
\IfStreamOpen, 285, 291	index (pagestyle), 116
\ifxetex, 333	\indexcolsep (length), 301
ifxetex (package), 333	indexing, 116
\ignorenoidxfile, 301	\indexintoc, 301
\iiirdstring, 328	\indexmark, 301
\iindstring, 328	\indexmarkstyle, 303
Illegal character, 455	\indexname, 301, 335
Illegal parameter number, 446	\indexrule (length), 301
Illegal pream-token, 462	Infinite glue shrinkage, 447
Illegal unit of measure, 446, 447	\input, 286, 337, 446, 455
illustration, 85, 105, 154, 164, 171, 172,	input stack size, 453
188, 201	Input stream is already defined, 465
multiple, 171	Input stream is not open, 465
imperialvopaper (option), 2	Input stream is open, 465
Improper \hyphenation, 447	\InputIfFileExists, 337, 338
\include, 303, 337, 455, 456	insert in contents, 142
\include cannot be nested, 455	\insertchapterspace, 87, 162, 361
\includegraphics, 172	interlinear space, see leading
\includeonly, 303	\intextsep (length), 179
Incomplete, 447	iso (class), 317
ind (file), 300, 301, 304, 309	\isopage, 29, 32
\indentafterchapter, 87	ist (file), 304, 307
\indentcaption, 182	\iststring, 328
vindent (length), 258	\item, 131-134, 250, 293, 299, 303, 455-
\indentpattern, 261, 267, 269	457
\index, 302–304, 306, 453	itemize (environment), 131, 132, 134,
index, 116, 154, 164, 254, 257, 376, 377	239, 321, 456
double column, 300, 301	\itemsep (length), 299
gutter, 301	\itshape, 42, 94
entry levels, 303	T
explanatory text, 301	J

Jenson Recut length

Jenson Recut, 469 Jessica, 469 Joanna, 469 \jobname, 284 jobname.idx (file), 302 jobname.ind (file), 301	lastsheet (counter), 320 LaTeX, error, 453–458 warning, 458–460 latexsym (package), 375, 455 Latin Modern, 3
jurabib (package), 300 \justlastraggedleft, 128	layouts (package), 29, 32, 147, 376 \lccode, 449
K	\lcminusname, 329, 335 lcode (environment), 282, 283, 378 leading, 50, 333
KeepFromToc (environment), 144 \keepthetitle, 67 Kettler, Howard, 39 keyword, by, 433 minus, 434 plus, 434 killtitle, 67 komalike (chapterstyle), 91, 104 komalike (headstyles), 104	\leadpagetoclevel, 83 League, Christopher, 63, 469 \leavespergathering, 321 \leavevmode, 101, 432 ledgerpaper (option), 2 ledmac (package), 230, 232 \left, 212, 213, 445 leftbar (environment), 273 \leftcenteright, 128 \leftcenterright, 128
L	\leftmargini (length), 258 \leftmark, 107, 108, 111, 113, 114, 116
I (position argument), 226, 227, 408, 411, 418 \l@book, 148 \l@kind, 140 \l@section, 142 \label, 54, 164, 172, 189, 194, 229, 289, 290, 292–296, 432, 453, 459, 460 label, 293 Label multiply defined, 459 Label(s) may have changed, 459 labelled (environment), 131, 132 landscape (option), 1 Lanston Bell, 469 Lanston Koch, 469 \LARGE, 45, 46 \Large, 45, 46, 348 \large, 45, 46, 96, 121, 263 largecrownvopaper (option), 2	\leftskip, 342 \leftspringright, 128 legalpaper (option), 2 \legend, 154, 186, 188, 296, 339, 351 legend, 186     in list of, 187     named, 188 length, 433     \@dblfpbot, 179, 180     \@dblfpsep, 179     \@dblfptop, 179, 180     \@dblfptop, 180     \@fpbot, 179, 180     \@fpbot, 179, 180     \@fpsep, 179, 180     \@fptop, 179, 180     \@fpsep, 179, 180
largepostvopaper (option), 2 \lasthline, 225 \lastlineparrule, 127 \lastlinerulefill, 127 lastpage (counter), 320	\absleftindent, 72 \absparindent, 72 \absparsep, 72 \absrightindent, 72 \abstitleskip, 72, 73 \afterchapskip, 85, 86

length

\afterepigraphskip, 251	\glossarycolsep,314
\afterPoemTitleskip, 264	\glossaryrule, 314
\arraycolsep, 224	\gparindent, 283, 284, 378
\arrayrulewidth, 224	\headdrop, 21
\baselineskip, 21, 23, 25, 50, 51,	\headheight, 19, 21, 23, 24, 27, 462
121, 122, 201, 251, 333, 378	\headmargin, 23, 24, 27
\beforechapskip, 85, 86, 395	\headsep, 19, 21, 24, 27, 349, 462
\beforeepigraphskip, 251	\headwidth, 110
\beforePoemTitleskip, 264	\heavyrulewidth, 216, 217
\belowcaptionskip, 201	\hsize, 273
\belowrulesep, 216, 217	vindent, 258
\bibitemsep, 299	\indexcolsep, 301
\bottomsectionskip, 77, 78	\indexrule, 301
\bvboxsep, 279, 280	\intextsep, 179
\cftbeforeKskip, 148	\itemsep, 299
\cftKindent, 148	\leftmargini, 258
\cftKnumwidth, 148, 149	\lightrulewidth, 216, 217
\cftparskip, 147	$\$ linewidth, 142, 198
\chapindent, 89	$\label{eq:lowermargin} \$ lowermargin, 462, 464
\cmidrulekern, 217, 218	\lxvchars, 15, 16
\cmidrulesep, 217	\marginparpush, 22-24
\cmidrulewidth, 217	\marginparsep, 22-24, 196, 237, 349,
\columnsep, 20, 24	350
\columnseprule, 20, 24	\marginparwidth, 22-24, 196, 237,
\dblfloatsep, 179	349, 350
\dbltextfloatsep, 179	\midchapskip, 85, 86, 395
\defaultaddspace, 217, 218	\midPoemTitleskip, 264
\doublerulesep, 216, 217, 224	\normalrulethickness, 110
\droptitle, 64	\oddsidemargin, 24
leftmargini, 258	\onelineskip, 50, 333, 378
\epigraphrule, 251	\paperheight, 12, 22, 24, 463, 464
\epigraphwidth, 250, 252	\paperwidth, 12, 22, 24, 130, 346, 463
\evensidemargin, 24	\parindent, 16, 50, 121, 124, 126, 207
\extrarowheight, 224	\parsep, 299
\extratabsurround, 225	\parskip, 49, 50, 121, 147, 201
textwidth, 110	\partopsep, 137
\fboxrule, 292	\pfbreakskip, 100
\fboxsep, 292	\prechapterprecisshift, 93
\floatsep, 179, 181	\pwlayii, 349, 350
\footmarksep, 232-234	\pwlayi, 349, 350
\footmarkwidth, 232-234	\ragrparindent, 123, 124
\footparindent, 233	rigid, 433, 435
\footskip, 21, 23, 24, 27, 346, 462	rubber, 433–435
\foremargin, 23, 24, 238, 399, 463	\sidebarhsep, 239, 240
\FrameHeightAdjust, 272	\sidebartopsep, 239, 240
\FrameRule, 272	\sidebarvsep, 239, 240
\FrameSep, 272	\sidebarwidth, 239, 240

\sidecapraise, 198	$\$ limits, $447$
\sidecapsep, 196, 198	Limits for array, 462
\sidecapwidth, 196	Linden, 469
\sidefootadjust, 241	\line, 413, 415, 416, 454, 459
\sidefoothsep, 241	line, , see also picture object
\sidefootmarksep, 242	arrowhead, see vector
\sidefootmarkwidth, 242	end, 411
\sidefootparindent, 242	length, 413
\sidefootvsep, 241	orphan, 52–54
\sidefootwidth, 241	reference point, 413
\sideparvshift, 237	restricted slope, 413, 414
\spinemargin, 23, 24, 130, 463	short, 413
\stanzaskip, 258	slope, 413
\stockheight, 22, 24, 463	thickness, 406
\stockwidth, 22, 24, 346, 463	widow, 52–54
\tabcolsep, 224	line number, 258, 259, 268, 279, 287
\textfloatsep, 179, 181	font, 262, 281
\textheight, 16, 22-25, 239, 240, 464	frequency, 262, 281
\textwidth, 16-18, 22-24, 110, 130,	position, 262, 281
225, 463	reset, 281
\thanksmarksep, 68	line too long, see overfull lines
\thanksmarkwidth, 68	\linebreak, 451
\topmargin, 19, 24	\linenottooshort, 126
\topsep, 137	\linenumberfont, 262, 281
\topskip, 23, 25, 53	\linenumberfrequency, 262, 281
\trimedge, 22, 24, 27, 346, 463	\linethickness, 406
\trimtop, 22, 24, 27, 463	\linewidth (length), 142, 198
\unitlength, 131, 161, 252, 349, 405,	lining,
406	Linotype, 39
\uppermargin, 19, 23, 24, 462, 464	\list, 136
\verbatimindent, 279	list, 131–138
\versewidth, 257, 258	new, 134–136
\vgap, 258, 259, 261	new list of, 161–165
\vindent, 258	spaces, 283
\vleftmargin, 258	tight, 134
\vleftskip, 262	list (environment), 72, 131, 134, 136,
\vrightskip, 262	258, 378
\width, 273	\listanswername, 161
\xlvchars, 15, 16	\listfigurename, 145, 146, 335
Length already defined, 465	\listofanswers, 161
leqno (option), 5	\listoffigugres, 335
\let, 157, 430	\listoffigures, 107, 139, 144, 147
\letcountercounter, 322	\listoffigures*, 139, 144
letterpaper, 8	\listofplates, 164
letterpaper (option), 1, 2, 5, 27, 29	\listoftables, 107, 139, 142, 144, 147,
\lightrulewidth (length), 216, 217	335
Limit controls, 447	\listoftables*, 139, 144

\listplatename \MakeUppercase

\listplatename, 164	\mainmatter*,75,80
\listtablename, 145, 146, 335	\makeatletter, 143, 185, 201, 235, 352,
Litho Antique, 469	357, 429, 451
\llap, 89	\makeatother, 143, 185, 201, 235, 352,
Imodern (package), 3	357, 429, 451
\LoadClass, 337, 455-457, 520	\makebox, 408, 409, 412
\LoadClass in package file, 455	makebst (package), 300
LoF, 139, 140, 143–147, 149, 154, 162, 164,	\makechapterstyle, 88
345, 351, 360, 361, 452	\makeevenfoot, 109
lof (file), 140	\makeevenhead, 109
\lofmark, 145	\makefootrule, 110
log (file), 284, 336, 428, 443	\MakeFramed, 273
Lonely \item, 455	MakeFramed (environment), 273
\long, 430	\makeglossary, 310, 314
longtable (package), 225	\makeheadfootruleprefix, 110
\loop, 440	\makeheadfootstrut, 111
\loosesubcaptions, 195	\makeheadposition, 110, 324
LoT, 139, 140, 143–147, 162, 164, 345, 351,	\makeheadrule, 110
360, 361	\makeheadstyles, 102, 103
lot (file), 140	makeidx (package), 336
\lotmark, 145	MakeIndex,
lowercase,	$\mbox{\tt makeindex}, 300, 301$
\lowermargin, 23	MakeIndex (program), 304, 306, 307, 309-
\lowermargin (length), 462, 464	313, 316, 377
LR	configuration file, 304
mode, 407	output styling, 307–309
lrbox (environment), 271	raw data, 304–307
alsoname, 306	\MakeLowercase, 54, 348
LTR, 342	$\mbox{\tt makeoddfoot}, 109$
Lubalin, Herb, 38	\makeoddhead, 109
Lucas, Édouard, 438	\makeother, 451
Lucida Bright, 353, 369	\makepagenote, 243
Luecking, Daniel, 151	$\mbox{\tt makepagestyle}, 109-111$
\lxvchars (length), 15, 16	\makepsmarks, 110, 114, 117
Lydian, 469	\makerunningwidth, 110
lyhne (chapterstyle), 91	\MakeShortVerb, 277, 278
Lyhne, Anders, 91	\MakeTextLowercase, 54
M	\MakeTextUppercase, 54, 156
M	\makethanksmark, 69
\a 422	\makethanksmarkhook, 69
\m@ne, 433	\maketitle, 57, 63-67, 69, 73, 107, 231,
\moth, 101	456, 459, 465
madsen (chapterstyle), 91	\maketitlehooka, 65
Madsen, Lars, 83, 90, 91, 112, 154, 394	\maketitlehookb, 65
main matter, 75, 297	\maketitlehookc, 65
main memory size, 453	\maketitlehookd, 65
\mainmatter, 75, 77, 78, 80, 111, 364	\MakeUppercase, 54, 55, 111, 112, 149

\marg, 340	math unit, 140
margin, 7, 17, 18, 23, 72, 129–131, 143, 154,	$\mbox{\mbox{\tt mathindent}}, 5$
318, 349	mathpazo (package), 16
fore-edge, <i>see also</i> outer, 8, 17, 18, 27,	\max, 323
345	\maxdeadcycles, 449
fore-edge, 17, 106	maximum length, 444
foredge, 114	maximum number, 444
inner, see also spine, 17, 27, 114	\maxsecnumdepth, 79, 80
left, 87, 94, 95, 97, 121, 123, 129, 140,	maxsecnumdepth (counter), 77
142, 148, 258	\maxtocdepth, 145
lower, 17, 19, 27	mcrownvopaper (option), 1
outer, see also fore-edge, 17, 21, 27,	mdemyvopaper (option), 1
114, 116, 129	\mdseries, 42
right, 123, 129, 140, 143, 147, 153, 251	measure
specify width, 17	45 characters, 15
specifying size, 17–20, 23	65 characters, 15
spine, see also inner, 8, 17, 18, 23, 27,	\medievalpage, 29, 32
129	mediumvopaper (option), 2
top, see also upper, 346	\medspace, 333
upper, see also top, 17, 19, 21, 23, 27,	Melior, 469
345	mem10.clo (file), 464
margin note, 236	\memappchapinfo,339
moved, 237	\memappchapstarinfo, 339
wrong margin, 236	\memapppageinfo, 339
marginalia, 7, 21–23, 27, 114, 187, 345,	\memapppagestarinfo,339
346, 349	\membicaptioninfo, 340
\marginbox, 99	\membionenumcaptioninfo, 340
marginfigure (environment), 169	\membitwonumcaptioninfo, 340
\marginhead, 99	\membookinfo,339
\marginpar, 169, 229, 236-238, 240, 455-	\membookstarinfo, 339
457, 459, 465	\memcaptioninfo, 340
Marginpar on page, 459, 465	\memchapinfo, 339
\marginparmargin, 22, 236	\memchapstarinfo, 339
\marginparmargin, 22, 250 \marginparpush (length), 22-24	\memchaps darinio, 339
\marginparsep, 241	-
	\memdskipstretch, 51
\marginparsep (length), 22-24, 196, 237,	\memendofchapterhook, 84
349, 350	\memfontenc, 3
\marginparwidth, 241	\memfontfamily, 3
\marginparwidth (length), 22-24, 196,	\memfontpack, 3
237, 349, 350	\memglodesc, 313
margintable (environment), 169	\memglonum, 313
\markboth, 107, 108	\memgloref, 313
markers, 107, 114	\memgloterm, 313
\markright, 107, 108	\memgobble, 336
math,	memhfixc (package), 338
in caption or title, 444	\memjustarg,336
math mode, 431	\memleadpageinfo, 339

\memleadpagestarinfo,339	Miller & Richard, 39
\memlegendinfo, 340	Minion, 345
memman (headstyles), 103, 104	minipage,
\memnamedlegendinfo, 340	in float, 172
memoir (class), 1, 101, 142, 144, 145, 163,	minipage (environment), 52, 172, 188
257, 295, 333	193, 198, 199, 229, 271, 350
error, 461–464	\miniscule, 45, 46, 49
warning, 464–466	minus (keyword), 434
\memorigdbs, 125	\minusname, 329, 335
\memorigpar, 125	minus, 434, 436
\mempartinfo, 339	Misplaced &, 447
\mempartstarinfo,339	Misplaced \cr, 447
\mempoeminfo, 339	Misplaced \noalign, 447
\mempoemstarinfo,339	Misplaced \omit, 447
\memPoemTitleinfo, 339	Misplaced \span, 447
\memPoemTitlestarinfo, 339	Missing { inserted, 448
\mempostaddapppagetotochook, 155	Missing = inserted, 448
\mempostaddbooktotochook, 155	Missing @-exp, 455
\mempostaddchaptertotochook, 155	Missing # inserted, 448
\mempostaddparttotochook, 155	Missing \$ inserted, 448
\mempreaddapppagetotochook, 155	Missing arg: token ignored, 462
\mempreaddbooktotochook, 155	Missing control sequence inserted, 448
\mempreaddchaptertotochook, 155	Missing \cr inserted, 447
\mempreaddparttotochook, 155	Missing delimeter(. inserted)., 449
\memRTLleftskip, 342	Missing \begin{document}, 455
\memRTLraggedleft, 342	Missing \endcsname inserted, 448
\memRTLraggedright, 342	Missing \endgroup inserted, 448
\memRTLrightskip, 342	Missing }inserted, 448
\memRTLvleftskip, 342	Missing number, 449
\memRTLvrightskip, 342	Missing p-arg, 456
\memsecinfo, 339	Missing \right inserted, 448
\memsecstarinfo, 339	mixing fonts, 42
memsty (package), 158, 312, 376	\mkern, 101
\memUChead, 112	mlargecrownvopaper (option), 1
\mergepagefloatstyle, 117, 118	mode, 431
$\text{\temp}$	display math, 431
Metafont, 35	horizontal, 431
\midbicaption, 192	internal vertical, 431
\midbookskip, 81	LR, 407
\midchapskip (length), 85, 86, 395	math, 431
midpage (package), 189	paragraph, 412
\midpartskip, 81	picture, 407, 408, 411, 412
\midPoemTitleskip (length), 264	restricted horizontal, 431
\midrule, 216	vertical, 431
\midsloppy, 52	Modern, 469
midsloppypar (environment), 52	monospaced font, 15
Miedinger, Max, 39	\morecmidrules, 217

moreverb (package), 336	\newcommand, 210, 323, 425, 430, 446
Morison, Stanley, 40	\newcommand*, 324, 449
\movetoevenpage, 326	\newcomment, 276
\movetooddpage, 326	\newcount, 432
moving argument, 296, 444	\newcounter, 162, 322, 432, 456
\mpjustification, 169	\newdimen, 433, 434
ms (option), 4	\newenvironment, 446, 453
msmallroyalvopaper (option), 1	newfile (package), 336
\multfootsep, 230	\newfixedcaption, 189
multicol (package), 157, 175, 272	\newfloat, 167, 168, 175
multicols (environment), 157	\newfootnoteseries, 231, 232
\multicolumn, 211, 223, 455, 456, 463	\newfootseries, 231
multind (package), 257	\newif, 437
\multiply, 433-436	\newinputstream, 284, 465
\multiput, 407, 418, 421, 453	\newleadpage, 83
myheading (pagestyle), 113	$\newlength, 434, 435$
myheadings (pagestyle), 106–108	\newline, $77$ , $451$ , $457$
	\newlistentry, 162, 163
N	\newlistof, 77, 161, 162
	\newloglike, 323
\n@me@number, 330	\newloglike*, 323
name,	\newoutputstream, 284, 466
index, 301	
named,	\newpage, 326
,	News Gothic, 469
number, 327, 328	\newsavebox, 271, 412, 421
\namedlegend, 187-189	newsgroup, xxiii
\namedsubappendices, 78, 79	\newskip, 433-435
\namenumberand, 329, 330, 335	\newsubfloat, 174, 192
\namenumbercomma, 329, 335	\newtheorem, 453, 456
nameref (package), 294	nextpage (package), 326, 336
\namerefoff, 296	\nextpageref, 165
\namerefon, 296	Nicholas Cochin, 469
\nametest, 324, 325	
natbib (package), 298, 300, 310	No array called, 462
•	No \author given, 459
Needspace, 326	No counter defined, 456
\needspace, 257, 268, 326	No more to read from stream, 465
needspace (package), 336	No positions in optional float specifier,
\Needspace*, 326	459
new,	No room for a new, 456
counter, 288	No room for a new read, 284
environment, 288	No room for a new write, 284
New Baskerville, 469	No \title given, 456
New Century Schoolbook, 3, 16, 40, 63	\noalign, 447
\newarray, 331, 462	=
\newblock, 299	\nobibintoc, 297
	\nobookblankpage, 82
\newcolumntype, 208, 210, 211, 222, 465,	\nobvbox, 279–281
466	\NoCaseChange, 54, 55

\nochangemarks option

\nochangemarks 317	\NumToNome 328
\nochangemarks, 317	NumToName, 328
\nocite, 299	\numtoName, 328
\noDisplayskipStretch, 51 \nofiles, 427, 428	Numan Patrick 122
	Nyman, Patrick, 122
\noglossaryintoc, 314	O
\noindent, 94, 250	Ğ
\noindentafterchapter, 87	\oarg, 340
\noindexintoc, 301 \nolimits, 447	Oberdiek, Heiko, 338
\nonzeroparskip, 49	\oddsidemargin (length), 24
\nopartblankpage, 82	old-style,
\noprelistbreak, 321	Olde English, 469
\normalbottomsection, 77, 78	oldfontcommands (option), 5, 462
\normalcaption, 182	oldpaper (option), 2
\normalcaption, 182	colorchapnum, 394
\normalfont, 15, 16, 42, 237	colorchaptitle, 394
\normalmarginpar, 236	\omit, 447
\normalrulethickness, 117	onecolabstract (environment), 73
\normalrulethickness (length), 110	\onecolglossary, 314
\normalsize, 44-46, 121, 237	\onecolindex, 301
\normalsubcaption, 195, 196	\onecoltocetc, 144, 145
Not a letter, 449	onecolumn (option), 4, 145, 167
Not defined:, 462	\OnehalfSpacing, 50, 358
Not in outer par mode, 456	\OnehalfSpacing*, 50
Not redefinable:, 463	\onelineskip, 50, 204, 241
\notedivision, 245, 247, 335	\onelineskip (length), 50, 333, 378
\noteidinnotes, 246	oneside (option), 4, 5, 26, 84
\notenuminnotes, 246	Only one column-spec. allowed, 463
\notenumintext, 245	footnotesinmargin, 240
\notepageref, 245, 246	footnotesize, 69
\notesname, 245, 335	\openany, 84
\nouppercaseheads, 107, 112	openany (option), 4, 66, 84
ntglike (chapterstyle), 91, 104	openbib (option), 5, 299
ntglike (headstyles), 104	\openinputfile, 285
\nthstring, 328	\openleft, 84
number, 327	openleft (option), 4, 84
named, 327, 328	\openoutputfile, 285
numeric, 327	\openright, 84
representation, 327	openright (option), 4, 84
Number too big, 449	Opentype, 35
numbered lines, 279	Optima, 469
\numberline, 140, 150, 154, 156, 157	option,
\numberlinebox, 150	*pt, 3, 464
\numberlinehook, 150	10pt, 2, 3, 5, 46
\NumberPoemTitle, 262, 263	11pt, 2, 46
numeric,	12pt, 2, 5, 46
number, 327	14pt, 2, 3, 46, 110

Option clash for ... Overfull \vbox ...

17pt, 2, 3, 46, 463	openany, 4, 66, 84
20pt, 2, 3, 46	openbib, 5, 299
25pt, 2, 46, 463	openleft, 4, 84
30pt, 3, 46	openright, 4, 84
36pt, 3, 46	postvopaper, 2
48pt, 3, 46	pottvopaper, 2
60pt, 3, 46	royalvopaper, 2
9pt, 2, 46	sectionbib, 298
a3paper, 1	showtrims, 4, 318
a4paper, 1, 27, 29	smalldemyvopaper, 2
a5paper, 1	smallroyalvopaper, 2
aбpaper, 1	statementpaper, 2
article, 5, 77, 84, 87, 93	superroyalvopaper, 2
b3paper, 1	titlepage, $65$ , $71$
b4paper, 1	twocolumn, 4, 5, 71–73, 145, 167
b5paper, 1	twoside, 4, 5, 26, 34, 84, 238
b6paper, 1	Option clash for, 456
broadsheetpaper, 2	Optional argument is not one of:, 463
crownvopaper, 2	Optional argument of \twocolumn, 459,
	465
dbillpaper, 2	
demyvopaper, 2	Opus, 469
draft, 4, 111, 290, 317	\or, 440
ebook, 1, 5	\ordinal, 327
executivepaper, 2	ordinal,
extrafontsizes, 2, 3, 49, 463	\OrdinalToName, 328
final, 4, 290, 317	\ordinaltoName, 328
fleqn, 5	\ordinaltoname, 328
foolscapvopaper, 2	\ordscript, 327, 328
fullptlayout, 5	orphan
imperialvopaper, 2	line, 52–54
landscape, 1	postbibhook, 298
largecrownvopaper, 2	postnotetext, 247
largepostvopaper, 2	outline,
ledgerpaper, 2	font, 35, 44, 49
legalpaper, 2	\output, 449
leqno, 5	Output loop, 449
letterpaper, 1, 2, 5, 27, 29	Output stream is already defined, 466
mcrownvopaper, 1	Output stream is not open, 466
mdemyvopaper, 1	Output stream is open, 466
mediumvopaper, 2	\oval, 417, 418, 459
mlargecrownvopaper, 1	oval, see also picture object
	1 /
ms, 4	reference point, 417
msmallroyalvopaper, 1	\oval, \circle, or \line size unavail-
oldfontcommands, 5, 462	able, 459
oldpaper, 2	Overfull \hbox, 449
onecolumn, 4, 145, 167	overfull lines, 52
oneside, 4, 5, 26, 84	Overfull \vbox, 449
	•

overfull package

overfull, 431	framed, 272, 275, 336
\overridescapmargin, 198	geometry, 7, 336
\overridesidecapmargin, 462	graphicx, 91, 92, 375, 387, 388, 391, 396
(overradoradoapmargin, 102	hanging, 122
P	hyperref, 145, 149, 156, 159, 160, 165,
	194, 246, 294, 303, 304, 338, 340
p (position argument), 168, 175, 180	ifetex, 332
package, , xxiii, 72, 288, 351, 352	ifmtarg, 336
abstract, 71, 336	ifpdf, 332, 336
afterpage, 164, 189	ifxetex, 333
alltt, 255, 375	index, 257, 336, 337
amsfonts, 455	jurabib, 300
amsmath, 375	latexsym, 375, 455
answer, 292	layouts, 29, 32, 147, 376
appendix, 336	ledmac, 230, 232
array, 205, 206, 336	Imodern, 3
bidi, 342, 343	longtable, 225
bookmark, 165	makebst, 300
booktabs, 205, 216, 336	makeidx, 336
bringhurst, 352	mathpazo, 16
calc, 25, 27, 239, 399, 400	memhfixc, 338
caption, 203, 204	memsty, 158, 312, 376
ccaption, 181, 336	midpage, 189
changepage, 325	moreverb, 336
chapterbib, 298	multicol, 157, 175, 272
chngcntr, 322, 336	multind, 257
chngpage, 129, 325	nameref, 294
color, 91, 272, 375	natbib, 298, 300, 310
comment, 277	needspace, 336
crop, 336, 397	newfile, 336
datetime, 321	nextpage, 326, 336
dcolumn, 205, 208, 336	pagenote, 336
delarray, 205, 336	parskip, 336
endnotes, 243	patchcmd, 324, 336
enumerate, 132, 336	pifont, 41, 100, 375
enumitem, 133, 134	placeins, 176
epigraph, 249, 336	ragged2e, 169
eso-pic, 399	sectsty, 75
etex, 375	setspace, 50, 336
etoolbox, 133, 399	shortvrb, 278, 336
fancybox, 273	showidx, 336
fancyhdr, 106, 336	sidecap, 183, 336
fixltx2e, 116, 375	siunitx, 208
flafter, 176	subfigure, 192, 336
fontenc, 375	tabularx, 205, 336
fonttable, 376	tcolorbox, 272, 281
footmisc, 230, 232	textcase, 54, 472

titleref, 294, 296, 336	\pagedemyvo, 2
titlesec, 75, 336	\pageexecutive, 2
tocbibind, 139, 336	\pagefoolscapvo, 2
tocloft, 139, 336	\pageimperialvo, 2
tocvsec2, 80, 145	\pageinnotes, 246
url, 358, 375	\pagelargecrownvo, 2
verbatim, 278, 336	\pagelargepostvo, 2
verse, 336	\pageledger, 2
wrapfig, 199	\pagelegal, 2
xcolor, 272, 375	\pageletter, 2
xtab, 225	\pagemdemyvo, 1
packages before class, 16	\pagemediumvo, 2
pag (fontfamily), 38	$\page metric crown vo, 1$
page, 13	$\pagemlargecrownvo, 1$
break, 225	\pagemsmallroyalvo,1
current number, 293	\pagename, 335
height, 12, 17, 19	\pagenote, 243, 245, 247, 248
location, 22	pagenote (counter), 245, 247
of floats, 117	pagenote (package), 336
reference, 294	\pagenotesubhead, 247
size, 8, 12, 22	= =
	\pagenotesubheadstarred, 247
specifying size, 12, 20	\pagenumbering, 105
width, 12, 17, 18	\pagenumbering*, 105
page break,	\pageold, 2
asynchronous, 427	\pagepostvo, 2
page color, 5	\pagepottvo, 2
page (counter), 105	\pageref, 164, 165, 293, 459, 460
Page height already too large, 456	\pagerefname, 246, 294, 335
page layout, 7, 22	\pageroyalvo, 2
change, 360	\pagesmalldemyvo, 2
check parameters, 23	\pagesmallroyalvo, 2
class parameters, 8, 22, 26	\pagestatement, 2
default, 7	\pagestyle, 87, 106, 110, 146, 162
design, 8	
Č	pagestyle, 7
LaTeX parameters, 8, 22, 25, 26	asu, 360
parameters, 23	book, 80, 107
\pageaiii, 1	chapter, 77, 84, 107, 146
\pageaiv, 1	class, 106
\pageav, 1	cleared, 107
\pageavi, 1	combining, 117
\pagebiii, 1	companion, 107, 114, 115, 117, 377
\pagebiv, 1	empty, 65, 83, 106, 107, 109, 111, 146,
\pagebroadsheet, 2	360, 361
\pagebv, 1	epigraph, 107, 253
\pagebvi, 1	float pages, 117–119
\pagecrownvo, 2	floatcomp, 117
\pagediumvo, 2	heading, 106
rpagoabiti, 2	neumitz, 100

headings, 7, 106, 107, 111, 114	motric domy octavo 1
index, 116	metric large grown estavo 1
	metric argel royal actave, 1
index pages, 116	metric small royal octavo, 1
myheading, 113	old, 2
myheadings, 106–108	post octavo, 2
part, 80, 107	pott octavo, 2
plain, 64, 66, 106–108, 111, 117, 252,	royal octavo, 2
360, 361	small demy octavo, 2
Ruled, 106, 107	small royal octavo, 2
ruled, 7, 106	statement, 2
section, 87	super royal octavo, 2
showlocs, 119, 318	stock, 8
simple, 106	\paperheight and/or \trimtop are too
specifying, 116	large, 463
title, 64, 107	\paperheight (length), 12, 22, 24, 463,
titlingpage, 65, 107	464
\pagesuperroyalvo, 2	\paperwidth, 17, 18
\pagetofootnote, 248	\paperwidth and/or \trimedge are too
pagination, 105	large, 463
changing, 105	\paperwidth (length), 12, 22, 24, 130,
interrupt, 105	346, 463
Palatino, 15, 16, 35, 40, 469	\par, 160, 444, 449
paper, 1, 8, 29, 164	\paragraph, 76, 79, 98, 99, 148, 334
end, 154	paragraph, 5, 121–124, 130, 153, 255, 345,
size, 1	348
A3, 1	block, 96, 121, 182, 349
A4, 1, 26, 27, 130	reasons against, 121
A5, 1	hanging, 96, 122–123, 130, 349
A6, 1	in list, 122
B3, 1	indentation, 72, 99, 121, 124, 182, 250
B4, 1	mode, 412
B5, 1	outdentation, 121
B6, 1	typeset as a unit, 121, 122, 124
broadsheet, 2	paragraph break,
crown octavo, 2	invisible, 238
demy octavo, 2	Paragraph ended before, 449
dollar bill, 2	paragraph indentation, 50
executive, 2	\paragraphfootnotes, 231
foolscap octavo, 2	\paragraphfootstyle, 231
imperial octavo, 2	\paraheadstyle, 102
large crown octavo, 2	parallel texts, 226
large post octavo, 2	\parbox, 99, 206, 222
ledger, 2	\parfillskip, 153
legal, 2	\parg, 340
letterpaper, 1, 2, 26, 27, 130, 346	\parindent (length), 16, 50, 121, 124,
medium octavo, 2	126, 207
metric crown octavo, 1	\parnopar, 238
metric crown octavo, i	(parnopar, 200

\parsep (length) poem

\parsep (length), 299	box, 406
\parshape, 130, 257	mode, 407, 408, 411, 412
parskip (package), 336	zero-sized, 407
\parskip (length), 49, 50, 121, 147, 201	picture (environment), 405-409, 419,
\part, 5, 76, 77, 79-84, 107, 147, 148, 150,	428, 453, 459
151, 155, 156, 253, 335, 339, 401	picture object, 407
part, 76, 140, 142, 143	Bezier curve, see also Bezier curve
number, 77	box, 408
part (pagestyle), 80, 107	circle, see also circle
\part*, 339	dashed box, 409
\partblankpage, 82	frame, 410
\partmark, 82, 83	framed box, 408
\partname, 81, 335	text position, 408
\partnamefont, 81, 102	line, see also line
\partnamenum, 81	oval, see also box, rounded
\partnumberline, 150	picture, 419
\partnumberlinebox, 150	regular pattern, 418
\partnumberlinehook, 150	stack, 411
\partnumfont, 81, 102	text, 407
\partopsep (length), 137	two-dimensional pattern, 421
\partpageend, 82, 83	unframed box, 409
\partrefname, 294, 335	text position, 409
\parttitlefont, 81, 82, 102	zero size, 409
de Parville, Henri, 438	vector, see also vector
\patchcmd, 462, 463	pifont (package), 41, 100, 375
patchcmd (package), 324, 336	pilcrow (¶), 133
\patchcommand, 324	placeins (package), 176
patverse (environment), 259, 261, 267,	plain (pagestyle), 64, 66, 106-108, 111,
331, 461	117, 252, 360, 361
patverse* (environment), 261, 262, 461	\plainbreak, 99
pbk (fontfamily), 39	\plainbreak*,99
pcr (fontfamily), 39	\plainfancybreak, 99, 100
\Pcstyle,377	\plainfancybreak*,99
pdf (file), 284	\plainfootnotes, 231
pdfLaTeX, 332, 333	\plainfootstyle, 231
pedersen (chapterstyle), 75, 91, 92, 104,	\PlainPoemTitle, 262, 263, 265
393	plates, 164
Pedersen, Troels, 91	Please type a command, 450
period, see also full stop	plus (keyword), 434
\pfbreak, 100	plus, 434, 436
\pfbreak*, 100	\pmname, 321
\pfbreakdisplay, 100	\pmnane, 321, 335
\pfbreakskip (length), 100	pnc (fontfamily), 40
\phantomsection, 145	\pnchap, 247
Phemister, Alexander, 39	\pnschap, 247
phv (fontfamily), 39	poem,
picture, 405	author, 265

poem title, 263–265	\preindexhook, 254, 301
in ToC, 263, 265	\prenoteinnotes, 246
styling, 267	\prenotetext, 246, 247
poemline (counter), 262	\pretitle,64
\PoemTitle, 262, 263, 339	\printanswer, 290
\poemtitle, 265, 294, 339	\printanswers, 290
\PoemTitle*, 263	\printbookname, 81
\PoemTitlefont, 264	\printbooknum, 81
\PoemTitleheadstart, 263, 264	\printbooktitle,81
\poemtitlemark, 263	\printchaptername, 86, 89
\PoemTitlenumfont, 264	\printchapternonum, 86, 89
\poemtitlepstyle, 263	\printchapternum, 86
\poemtitlestarmark, 263	\printchaptertitle, 86, 146, 348
\poemtitlestarpstyle,263	\printfibterm, 439, 440
\poemtoc, 263, 265	\printfigures, 289
pool size, 453	\printglossary, 311, 316
\poptabs, 456	\printindex, 300, 301
\postauthor, 64	\printpageinnotes, 246
\postbibhook, 298	\printpageinnoteshyperref, 246
\postcaption, 183, 192	\printpagenotes, 243, 245
\postchapterprecis,93	\printpagenotes*, 243-245, 247
\postdate,64	\printpartname, 81
\postnoteinnotes, 246	\printpartnum, 81
\postnotetext, 246	\printparttitle, 81, 82
PostScript, 35, 469	\printPoemTitlenonum, 263, 264
\posttitle,64	\printPoemTitlenum, 263, 264
postvopaper (option), 2	\printPoemTitletitle, 263, 264
pottvopaper (option), 2	\printtime, 321
uppercaseheads, 107	\printtime*, 321
ppl (fontfamily), 40	\printXtitle,146
\Ppstyle, 376, 377	\priorpageref, 165
preamble, 5, 27, 69, 80, 116, 143, 147, 151,	program,
183, 235, 288, 300, 351, 352, 376,	BibTeX, 299, 300
454, 458	MakeIndex, 304, 306, 307, 309-313,
\preauthor, 64	316, 377
\prebibhook, 254, 298	xindy, 304
\precaption, 183, 192	proportion
\prechapterprecis,93	book, 12
\prechapterprecisshift (length), 93	margin, 17
\precisfont,93	typeblock, 13
\precistocfont, 93, 94, 156	\protect, 55, 101, 142, 143, 157, 203, 247,
\precistocformat, 93, 94, 156	285, 287, 296, 444
\precistoctext, 93, 94, 156	protect, 101, 291
\predate, 64	\providecommand, 323, 430, 446
\Pref, 294, 335	\providecommand*,430
\pref, 294, 335	\providecounter, 323, 465
\preglossaryhook,314	\provideenvironment, 323, 465

\providefixedcaption, 189	raggedleft, 123
\providelength, 323, 465	\raggedright, 123, 124, 182, 195, 199,
\provideloglike,323	222, 342
\provideloglike*,323	raggedright, 123, 239
pseudo (counter), 288, 289	\raggedrightthenleft, 128
psnfss, 15, 35	\raggedyright, 123, 124
\pstyle, 376, 377	\ragrparindent (length), 123, 124
psy (fontfamily), 41	ranging,
ptm (fontfamily), 40	\readaline, 286
\published, 65	\readboxedverbatim, 286
\pushtabs, 456	\readboxedverbatim*, 286
\pushtabs and \poptabs don't match,	\readstream, 286
456	\readverbatim, 286
\put, 407, 413, 415, 418	\readverbatim*, 286
put (fontfamily), 40	Redefining primitive column, 466
\pwlayii (length), 349, 350	\ref, 54, 55, 194, 289, 291-293, 459, 460
\pwlayi (length), 349, 350	reference,
pzc (fontfamily), 41	by name, 294–296
pzd (fontfamily), 41	by number, 293–294
_	current value, 293
Q	set current value, 293, 294
107 404 400 450	to appendix, 294
\qbezier, 407, 421, 423, 453	to book, 294
\qbeziermax, 422, 423	to chapter, 294
qfame (environment), 276	to counter, 303
qframe (environment), 276	to figure, 294
\qitem, 249, 250	to label, 293
qshade (environment), 276	to page, 293, 294, 303
\quarkmarks, 319, 321	to part, 294
question (counter), 292	to section, 294
questions and answers, 289–292	to table, 294
\quietref, 291	unexpected result, 293, 295
quotation, 121, 124, 249, 251, 252	Reference on page, 460
quotation (environment), 71, 124, 129,	reference mark, see also endnote mark,
131, 134, 276, 457 quote (environment), 93, 124, 129, 131,	footnote mark, 235
134	reference point, 407
104	box, 408
R	circle, 415
	line, 413
r (position argument), 226, 227, 409, 411,	oval, 417
418	stack, 411
ragged2e (package), 169	text, 407
\raggedbottom, 5, 52, 78, 230, 326	vector, 415
\raggedbottomsection, 77, 78	\refstepcounter, 290, 293, 294, 432
\raggedleft, 123, 124, 182, 195, 199, 222,	registered trademark, 334
342	\registrationColour, 319
	\relax, 112, 433

\renewcommand, 89, 110, 140, 149-151,	\russianpar, 126
188, 201, 285, 322, 324, 325, 423, 430, 446	S
\renewcommand*, 198, 324, 430	
\renewed vironment, 446	Sabon, 469
\renewfixedcaption, 189	save stack size, 453
\renewleadpage, 83	\savebox, 271, 412, 421
reparticle (chapterstyle), 87	\savepagenumber, 105
\repeat, 440	\savetrivseps, 137
report (class), 64, 71, 88, 102, 142	Savoy, 469
\reportnoidxfile, 301	\saythanks, 73
representation,	\sbox, 271, 412, 413, 421
number, 327	\scapmargleftfalse, 196
\RequireAtEndClass, 337, 338	\scapmarglefttrue, 196
\RequireAtEndPackage, 337	Schnittger, 469
\RequirePackage, 16, 456	Schusterjungen, see orphan
\RequirePackage or \LoadClass in Op-	scrbook (class), 91, 104
tions Section, 456	Script
\RequireXeTeX, 333	type, 41
\resetbvlinenumber, 281	\scriptsize, 45, 46
\resizebox, 396	\scshape, 42
\restorepagenumber, 105	\secheadstyle, 102
\restoretrivseps, 137	secnumdepth (counter), 77, 79
\reversemarginpar, 236	\section, 5, 76-79, 87, 95-97, 99, 115, 142,
\right, 212, 213, 445, 448	148, 157, 239, 293, 295, 296, 339,
\rightmark, 107, 108, 111, 113, 114, 116	349, 400
\rightskip, 342	section, see also subhead
\rmfamily,42	section (chapterstyle), 87, 89
roman numerals, 105	section (pagestyle), 87
Roman, 105	\section*,76
roman, 105	sectional division, see also subhead
Rosslaire, 469	level number, 79
royalvopaper (option), 2	sectionbib (option), 298
RTL, 342	\sectionmark, 108, 113, 114
rule,	\sectionrefname, 294, 335
in margin, 273	sectsty (package), 75
invisible, 117	\see, 306, 335
thickness, 20, 110, 117	\seealso, 306, 335
ruled,	\seename, 306, 335
float, 170	\semiisopage, 29, 32
Ruled (pagestyle), 106, 107	Serifa, 469
ruled (pagestyle), 7, 106	Serific, 469
Runaway argument, 450	\setaftersecskip, 348
Runaway definition, 450	\setafterSskip, 96
Runaway preamble, 450	\setarrayelement, 331
Runaway text, 450	\setbeforesecskip), 348
running BibTeX, 300	\setbeforeSskip, 95

\setbiblabel \showtrimsoff

\setbiblabel, 299	\setsidebarheight, 239, 240
\setbinding, 19, 23	\setsidebars, 240
\setbvlinenums, 281	\setsidecappos, 196, 197, 462
\setcolsepandrule, 20	\setsidecaps, 196
\setcounter, 105, 139, 287, 322, 432, 453,	\setsidefeet, 241
456, 463	\setsidefootheight, 241
\setDisplayskipStretch, 51	\setSindent, 95
\setfloatadjustment, 168, 169	\setSingleSpace, 51
\setFloatBlockFor, 176	setspace (package), 50, 336
\setfloatlocations, 175	\setstocksize, 8, 12, 22
\setFloatSpacing, 51	\setsubsecheadstyle, 349
\setFloatspacing, 31 \setfootins, 22, 236	\setsubsecheaustyle, 347
	•
\sethangfrom, 96, 97, 349	\settocdepth, 139, 145, 158, 351, 404,
\setheaderspaces, 21, 23	463, 466
\setheadfoot, 21, 23	\settocpreprocessor, 156
\setlength, 8, 22, 72, 86, 110, 134, 148,	\settodepth, 436
149, 198, 201, 250, 251, 258, 368,	\settoheight, 436
400, 405, 435	\settowidth, 436
\setlist, 133	\settrimmedsize, 12, 13, 16, 22, 32
\setlrmargins, 17, 19, 23	\settrims, 12, 13, 22
\setlrmarginsandblock, 16, 18, 23	\settypeblocksize, 16, 17, 22, 324
\setlxvchars, 15	\settypeoutlayoutunit, 26
\setmarginfloatcaptionadjustment,	\setulmargins, 17, 19, 21, 23
169	\setulmarginsandblock, 16, 19, 20, 23
\setmarginnotes, 22, 23, 196, 236, 237,	\setupmaintoc, 160
240, 467	\setupparasubsecs, 159, 161
\setmpjustification, 169	\setupshorttoc, 158
\setpagebl, 32	\setverbatimfont, 278, 358
\setpagebm, 33	\setverselinenums, 262
\setpagebr, 33	\setxlvchars, 15
\setpagecc, 13, 33	\sffamily, 42, 202
\setpageml, 32	\sfseries, 345
\setpagemr, 33	shaded (environment), 272, 273, 276
\setPagenoteSpacing, 51	sheet, 8
\setraget1,32	sheetsequence (counter), 320
\setpagetm, 33	\Shook, 97
\setpagetr, 33	\shortstack, 411
\setpageXX, 33	\shortsubcaption, 195, 196
\setparahook, 99	shortvrb (package), 278, 336
\setpnumwidth, 147, 151	\showcols, 211
\setrmarg, 147	\showheadfootlocoff, 119
\setsecheadstyle, 348	showidx (package), 336
\setsecnumdepth, 79, 80, 145, 463	\showindexmarks, 303
$\strut_{79,96,97,99}$	showlocs (pagestyle), 119, 318
\setSheadstyle, 95	\showtextblockoff, 119
\setShook, 97	showtrims (option), 4, 318
\setsidcaps, 196	\showtrimsoff, 318

\showtrimson space

\showtrimson, 318	sidelegend (environment), 197
side footnote	sidenamedlegend (environment), 197
text, 242	\sidepar, 237, 238, 240
side note, 237	\sideparfont, 237
adjust position, 237	\sideparform, 237
text for particular margin, 238	\sideparmargin, 237-239
\sidebar, 238-240	\sideparvshift (length), 237
sidebar, 238, 394	simple (pagestyle), 106
styling, 239	single column,
\sidebarfont, 239, 240	index, 300, 301
\sidebarform, 239	\SingleSpacing, 51
\sidebarhsep (length), 239, 240	\SingleSpacing*, 51
\sidebarmargin, 238, 239	siunitx (package), 208
\sidebartopsep (length), 239, 240	\sixt@@n, 433
\sidebarvsep (length), 239, 240	size,
\sidebarwidth (length), 239, 240	page, 8, 12, 22
sidecap (package), 183, 336	paper, 1
\sidecapfloatwidth, 198	stock, 1, 8, 12, 22
\sidecapmargin, 196, 198, 464	typeblock, 22
\sidecapraise (length), 198	Size substitutions, 460
\sidecapsep (length), 196, 198	\skip, 433–435, 456
\sidecapstyle, 197	\slashfrac, 330
\sidecaption, 464	\slashfracstyle, 330
sidecaption (environment), 196, 197,	Slimbach, Robert, 40
199	slope, 413, 414
\sidecapwidth (length), 196	line, 413
sidecontcaption (environment), 197	vector, 415
\sidefootadjust, 241	\sloppy, 52, 451
\sidefootadjust (length), 241	\sloppybottom, 53
\sidefootenote, 241	sloppypar (environment), 52
\sidefootfont, 241	\slshape, 42
\sidefootform, 242	\small, 45, 46, 121, 134, 168, 222, 251, 282,
\sidefoothsep, 241	345, 378
\sidefoothsep (length), 241	smalldemyvopaper (option), 2
\sidefootmargin, 241	smallroyalvopaper (option), 2
\sidefootmarksep (length), 242	snugshade (environment), 272, 273
\sidefootmarkstyle, 242	snugshaded (environment), 272
\sidefootmarkwidth (length), 242	Some shapes, 460
\sidefootnote, 241	Something's wrong, 456
\sidefootnotemark, 241	Sorry, but I'm not, 450
\sidefootnotetext, 241	source2e.tex (file), 430
\sidefootparindent (length), 242	\sourceatright, 124
\sidefoottextfont, 242	southall (chapterstyle), 92, 394
\sidefootvsep, 241	space,
\sidefootvsep (length), 241	at start of paragraph, see paragraph
\sidefootwidth, 241	indentation
\sidefootwidth (length), 241	between lines, see leading

double, see (ouble spacing)i	broadsheet, 2
gobble, 290	crown octavo, 2
inter-paragraph, 49–50	demy octavo, 2
interword, 52	dollar bill, 2
\spacefactor, 451	ebook, 1
\span, 445, 447	executive, 2
\specialindex, 303	foolscap octavo, 2
\specialrule, 217	imperial octavo, 2
\spinemargin and/or \textwidth	large crown octavo, 2
and/or \foremargin are too	large post octavo, 2
large, 463	ledger, 2
\spinemargin (length), 23, 24, 130, 463	legal, 2
spread, 12, 17	letterpaper, 1, 2, 8
\Sref, 294, 335	medium octavo, 2
stack	metric crown octavo, 1
reference point, 411	metric demy octavo, 1
stacking	metric large crown octavo, 1
text, 411	metric small royal octavo, 1
vertical spacing, 412	old, 2
stanza, end, 255	post octavo, 2
indent alternate lines, 261, 265	pott octavo, 2 royal octavo, 2
_	The state of the s
indent pattern, 261, 269 last line, 258	small demy octavo, 2
	small royal octavo, 2
numbering, 258	statement, 2
line break, 259, 267	super royal octavo, 2
indent, 259	specifying size, 8, 22
long line, 266	specifying trimming, 13
number, 262, 268	trimmed, 8, 12
prevent page break, 258	width, 8
space, 258	stock paper size option, 8, 12
\stanzaskip (length), 258	\stockaiii,1
start new page, 189	\stockaiv, 1
statementpaper (option), 2	\stockav, 1
\stepcounter, 432	\stockavi,1
stock, 4, 7, 8, 12, 13, 22, 26, 27, 346	\stockbiii,1
commercial printing, 8	\stockbiv, 1
default, 1	\stockbroadsheet, 2
height, 8	\stockbv, 1
paper, 8	$\stockbvi, 1$
size, 1, 8, 12, 22	\stockcrownvo, 2
A3, 1	\stockdbill, 2
A4, 1, 8	\stockdemyvo, 2
A5, 1	\stockexecutive, 2
A6, 1	\stockfoolscapvo,2
B3, 1	\stockheight (length), 22, 24, 463
B6, 1	\stockimperialvo, 2

\stockimperialvo

\stocklargecrownvo, 2	subhead, 75
\stocklargepostvo, 2	moved, 77
\stockledger, 2	near bottom of page, 77
\stocklegal, 2	\subitem, 303
\stockletter, 2	\subparagaph, 334
\stockmdemyvo, 1	\subparagraph, 76, 79, 96, 148
\stockmediumvo, 2	\subparaheadstyle, 102
\stockmetriccrownvo,1	\subsecheadstyle, 102
\stockmlargecrownvo, 1	\subsection, 76, 79, 95, 97, 99, 148, 159,
\stockmsmallroyalvo,1	339, 400
\stockold, 2	\subsubitem, 303
\stockpostvo, 2	\subsubsecheadstyle, 102
\stockpottvo, 2	\subsubsection, 76, 79, 96, 148
\stockroyalvo,2	\subtop, 174, 193
\stocksmalldemyvo, 2	Suggested extra height, 456
\stocksmallroyalvo, 2	superroyalvopaper (option), 2
\stockstatement, 2	superscript, 334
\stocksuperroyalvo, 2	\suppressfloats, 176
\stockwidth (length), 22, 24, 346, 463	Symbol, 41, 375
\stop, 450	symbol, 334
stream, 284–292	symbols (environment), 136
check open, 285	\symbolthanksmark, 67
close input, 285	Syntax, 345, 469
close output, 285	syntax (environment), 377
input, 284, 285	<u>_</u>
limited number, 284	T
new input, 284	. /
new output, 284	t (position argument), 168, 175, 180, 409,
output, 284, 285, 289, 290	418
Stream is not open, 466	Tab overflow, 457
\strictpagecheck, 129, 236, 325	tabbing (environment), 454, 456, 457
\strictpagecheck, 198	\tabcolsep (length), 224
\stringtoarray,331	table, 117, 142, 163, 167, 171, 181, 185,
\strip@pt, 349	187–189, 192, 345, 351
Struktor, 469	column
\strut, 111	cut-in head, 214
strut, 412	decked head, 214
subappendices (environment), 78, 79	head, 214
\subbottom, 174, 193	spanner rule, 214
\subcaption, 174, 193	spanner rule, 214
\subcaptionfont, 195	stub, 214
\subcaptionlabelfont, 195	half and half, 216
\subcaptionref, 194	long, 225
\subcaptionsize, 195	reference, 294
\subcaptionstyle, 195	row fill, 218–219
\subconcluded, 194	rule, 215–218
subfigure (package), 192, 336	horizontal, 214

vertical, 214	TeX,
subtable, 192–196	error, 444–453
table (environment), 167, 169, 187, 188,	warning, 444–452
226, 454	TeX capacity exceeded, 450, 452–453
\tablename, 167, 188, 335	tex (file), 284
\tableofcontents, 77, 107, 139, 140, 142,	text,
144, 147, 160, 335, 428	Text line contains, 450
\tableofcontents*, 139, 144, 403	Text page contains only floats, 460
\tablerefname, 294, 335	\textasteriskcentered, 133
\tabsoff, 278, 279	\textbf, 41
\tabson, 278	\textbullet, 132
tabular, 220–223, 445	textcase (package), 54, 472
automatic, 226–228	\textendash, 133
by column, 227	\textfloatsep (length), 179, 181
by row, 226	\textfraction, 179, 180
column style, 227	\textheight (length), 16, 22-25, 239,
column width, 227	240, 464
position, 227	\textit, 41
table width, 227	\textmd, 41
column width, 220, 221	\textperiodcentered, 133
continuous, 225	\textregistered, 334
position, 225	\textrm, 41
controlling width, 220	\textsc, 41
free, 225–228	\textsf,41
caption, 226	\textsl,41
intercolumn space, 220, 221, 224	\textsubscript, 330, 331
new column type, 222	\textsuperscipt, 334
raggedright, 222	\textsuperscript, 330, 331
row spacing, 224	\texttrademark, 334
rule, 220	\texttt, 41
vertical position, 224	\textup, 41
X column, 220, 222	\textwidth, 5
tabular (environment), 172, 199, 205,	textwidth, 15
207, 211, 220–222, 224, 225, 229,	\textwidth (length), 16-18, 22-24, 110,
377, 445, 447, 448, 453, 455, 456,	130, 225, 463
461, 462	\thanks, 57, 65, 67-69, 73, 231, 232, 465
tabular* (environment), 205, 220-222,	thanks, 67
224	styling, 67–69
tabularx (environment), 205, 220-223,	\thanksfootmark, 68, 69
466	\thanksheadextra, 67
tabularx (package), 205, 336	\thanksmark, 68
\tabularxcolumn, 222	\thanksmarksep (length), 68
Tahoma, 353, 369	\thanksmarkseries,67
tandh (chapterstyle), 92, 104	\thanksmarkstyle,68
tandh (headstyles), 104	\thanksmarkwidth (length), 68
tcolorbox (package), 272, 281	\thanksrule,69
\tensunitsep, 329	\thanksscript, 69

That makes 100 errors, 450	\tightsubcaptions, 195
thatcher (chapterstyle), 92	Times, 353
Thatcher, Scott, 92	Times New Roman, 40
\the, 433, 434, 436	Times Roman, 15, 16, 40, 369
The font command is deprecated,	\tiny, 45, 46
466	tip in, 164
The 'extrafontsizes' option, 463	\title, 57, 64, 66, 456, 465
The combination of argument values,	title
463	styling, 57–67
The counter will not be printed, 466	title page, 57, 63, 65–66, 336
The file needs format, 457	title (pagestyle), 64, 107
\theauthor, 66, 67	titledframe (environment), 275
\thebibliography, 335	\TitleFrame, 275
thebibliography (environment), 107,	titlepage (environment), 65
297, 298, 314, 456, 465	titlepage (option), 65, 71
\thechapter, 287	\titleref, 294-296
\thectr, 322	titleref (package), 294, 296, 336
\thedate, 66, 67	titlesec (package), 75, 336
\theglossary, 335	titling,
theglossary (environment), 311, 314	titlingpage (environment), 65, 66, 69,
\theindex, 335	106, 107
theindex (environment), 107, 301, 303,	titlingpage (pagestyle), 65, 107
314	\titlingpageend, 65
\thepage, 105, 320	\tmarkbl, 318, 319
\thepoem, 262	\tmarkbm, 318, 319
\thepoemline, 262	\tmarkbr, 318, 319
There were multiply defined labels, 460	\tmarkml, 318, 319
There were undefined references, 460	\tmarkmr, 318, 319
There's no line to end here, 457	\tmarktl, 318, 319
\thesection, 99	\tmarktm, 318, 319
\thesheetsequence, 320	\tmarktr, 318, 319
\thetitle, 66, 67	ToC, 72, 76-78, 83, 92-94, 101, 102, 114,
\theTitleReference, 296	139, 140, 142–151, 153–162, 164,
\thicklines, 406	263, 265, 271, 293, 295, 297, 301,
thickness	326, 345, 348, 351, 360, 361, 363,
line, 406	364, 369, 399, 401–404, 444, 452
\thinlines, $406$	controlling entries, 168
\thinspace, 333	toc (file), 93, 139, 140, 156, 243, 284, 428
This can't happen, 450	tocbibind (package), 139, 336
This may be a LaTeX bug, 457	tocdepth (counter), 139, 145, 158, 162,
\thispagestyle, 106	168
\thr@@, 433	tocloft (package), 139, 336
\threecolumnfootnotes, 231	\tocmark, 145
\threecolumnfootstyle, 231	\tocnameref, 295
tightcenter (environment), 377	tocvsec2 (package), 80, 145
\tightlist, 134	token, 426
\tightlists, 133, 134	Too deeply nested, 457

Too many }'s typeface

Too many columns, 457 Too many unprocessed floats, 457 \toppargin (length), 19, 24 topnumber (counter), 179 \topsep (length), 137 \topsep (length), 137 \topsep (length), 23, 25, 53 totalnumber (counter), 179 to, 431 \tracingtabularx, 222 trademark, 334 \traditionalparskip, 49, 50 Transitional type, 40 \treft, 294, 335 trim,	T	Laura a
Too many unprocessed floats, 457 \topfraction, 179-181 \topmargin (length), 19, 24  topnumber (counter), 179 \topsep (length), 137 \topsep (length), 137 \topsep (length), 23, 25, 53  total number (counter), 179  to, 431 \tracingtabularx, 222  trademark, 334 \tracingtabularx, 222  trademark, 334 \tracingtabularx, 222  trademark, 334 \tracingtabularx, 222  trademark, 335 \trimm, specifying, 22  \trimedge (length), 22, 24, 27, 346, 463 \trimmarks, 318 \trimmarks, 318, 319 \trimm	Too many 3's, 451	type
ttopfraction, 179-181         Didone, 469           ttoppargin (length), 19, 24         Free Form, 469           toppumber (counter), 179         Mediaeval, 469           ttopskip, 86, 349         Mediaeval, 469           ttopskip, 86, 349         Modern, 469           ttopskip (length), 23, 25, 53         Script, 41, 470           totalnumber (counter), 179         Slab Serif, 469           vo, 431         Symbol, 470           tracingtabularx, 222         Transitional, 40, 469           traditional parskip, 49, 50         Venetian, 469           transitional         type, 40           trim,         specifying, 22           trimmark, 335         tyrimframe, 318           trimmarks, 318         trimmarks, 319           trimmarks, 319         trimmarks, 318, 319           trimmone, 318         trimmone, 318         trimmone, 318         typeface, 15           Alter Schwabacher, see Alte         Schwabacher, see Alte           Schwabacher         Alte Schwabacher, see Alte           Schwabacher, see Basel         Basel, see Basel           Basilia, see Basel         Basilia, see Basel           Basilia, see Basel         Basilia, see Basel           Basilia, see Basel         Basel, see Basel           Basilia, see Ba		
\topmargin (length), 19, 24 topnumber (counter), 179 \topsep (length), 137 \topsep (length), 23, 25, 53 totalnumber (counter), 179 to, 431 ttracingtabularx, 222 trademark, 334 \traditionalparskip, 49, 50 Transitional type, 40 \tref, 294, 335 trim, specifying, 22 \trimedge (length), 22, 24, 27, 346, 463 \trimframe, 318 \trimframe, 318 \trimframe, 318 \trimflamarks, 318 \trimflone,		
topnumber (counter), 179 \topsup (length), 216 \topsep (length), 137 \topskip, 86, 349 \topsep (length), 23, 25, 53 \totalnumber (counter), 179  to, 431 \tracingtabularx, 222 trademark, 334 \tracingtabularx, 222 trademark, 334 \traditionalparskip, 49, 50  Transitional		
\topsule, 216 \topsep (length), 137 \topskip, 86, 349 \topskip, 86, 349 \topskip, 86, 349 \topskip (length), 23, 25, 53  totalnumber (counter), 179  to, 431 \tracingtabularx, 222  trademark, 334 \traditionalparskip, 49, 50  Transitional		
\topsep (length), 137 \topsep (length), 86, 349 \topskip (length), 23, 25, 53 \totalnumber (counter), 179 \to, 431 \tracingtabularx, 222 \trademark, 334 \traditionalparskip, 49, 50  Transitional \type, 40 \tref, 294, 335 \trim,  specifying, 22 \trimdage (length), 22, 24, 27, 346, 463 \trimTarame, 318 \trimTarame, 318 \trimTarame, 318 \trimMarks, 318, 319 \trimmarks, 318, 319 \trimMarkscolor, 318 \trimNone, 318 \trimTop (length), 22, 24, 27, 463 \trimXarrisy, 318, 319 \trimTrist (environment), 136–138, 283, 377  TrueType, 35, 469 \trimtage compelet language, 425 \trimtg complete language, 425 \trimtg, 318 \trimtg complete language, 425 \trimtg, 341, 425 \two, 433  Two \documentclass commands, 457 \two colloaclass commands, 457 \two colloctect, 144, 145 \twocollome, 301 \twocollome, 34, 5, 26, 34, 84, 238  Modern, 469  Script, 41, 470  Slab Serif, 469  Schude, 9  Venetian, 469  Ve		
\topskip, 86, 349 \topskip, 123, 25, 53 \totalnumber (counter), 179 \to, 431 \tracingtabularx, 222 \trademark, 334 \traditionalparskip, 49, 50  Transitional \type, 40 \tref, 294, 335 \trimframe, 318 \trimframe, 318 \trimframe, 318 \trimmarks, 318, 319 \trimmarks, 318, 319 \trimmarks, 318, 319 \trimfmarks, 318, 319 \trimfmarks, 15 (environment), 136–138, 283, 377  TrueType, 35, 469 \ttfamily, 42 Turing complete language, 425 Turing, Alan, 425 \two \documentclass commands, 457 \two \coloumdocute, 144, 145 \twocolumn (option), 4, 5, 71–73, 145, 167 \twocolumn (option), 4, 5, 26, 34, 84, 238  Sans Serif, 469 Script, 41, 470  Slab Serif, 469 Symbol, 470  Transitional, 40, 469 Uncial, 469 \tvenetian, 469 \type size, 2 default, 3  typeslock, 5, 7, 8, 17–20, 22, 26, 27, 29, 66, 110, 114, 130, 249, 250, 258, 345, 346, 348–350 height, 16, 17, 19 location, 17, 20, 23 size, 22 specifying size, 16, 20, 22 width, 16–18 typeface, 15  Alte Schwabacher, see Alte Schwabacher, see Alte Schwabacher ATF Clearface, see AAVANT Garde Gothic, see Avant Garde Gothic, see Baselia Baskerville, see Basilia Baskerville, see Bernhard Modern Bodoni, see Bodoni Bookman, see Bookman Caslon, see Caslon Centaur, see Centaur Century Old Style, see Century Old Style Chantilly, see Chantilly		
\topskip (length), 23, 25, 53 totalnumber (counter), 179 to, 431 \tracingtabularx, 222 trademark, 334 \traditionalparskip, 49, 50 Transitional type, 40 \tref, 294, 335 trim, specifying, 22 \trimedge (length), 22, 24, 27, 346, 463 \trimTramark, 318 \trimTrame, 318 \trimTramarks, 318, 319 \trimMarks, 318, 319 \trimMarks, 318, 319 \trimTrimtop (length), 22, 24, 27, 463 \trimType, 35, 469 \trimIty, 42 Turing complete language, 425 Turing, Alan, 425 \trimQ, 433 Two \LoadClass commands, 457 Two \LoadClass comm		
totalnumber (counter), 179 to, 431 type, 40 \traditionalparskip, 49, 50 Transitional type, 40 \tref, 294, 335 trim, specifying, 22 \trimedge (length), 22, 24, 27, 346, 463 \trimTrame, 318 \trimLmarks, 318 \trimLmarks, 318, 319 \trimtone, 318 \trimtone, 319 \trimtone, 318 \tri	\topskip, 86, 349	•
to, 431 \tracingtabularx, 222 trademark, 334 \traditionalparskip, 49, 50  Transitional type, 40 \treft, 294, 335 trim, specifying, 22 \trimedge (length), 22, 24, 27, 346, 463 \trimFrame, 318 \trimLmarks, 318 \trimMarks, 318, 319 \trimmarks, 318, 319 \trimtop (length), 22, 24, 27, 463 \trimXmarks, 318, 319  trivilist (environment), 136–138, 283, 377  TrueType, 35, 469 \trimg complete language, 425 Turing, Alan, 425 \two, 433  Two \LoadClass commands, 457 Two \coolumn (option), 4, 5, 71–73, 145, 167 \twocolumn (option), 4, 5, 71–73, 145, 167 \twocolumnfoottstyle, 231 twoside (option), 4, 5, 26, 34, 84, 238  Symbol, 470 Transitional, 40, 469 \trianitional, 40, 469 \type size, 2 \default, 3 \typeblock, 5, 7, 8, 17–20, 22, 26, 27, 29, 66,	\topskip (length), 23, 25, 53	<b>.</b>
\tracingtabularx, 222 trademark, 334 \traditionalparskip, 49, 50 Transitional     type, 40 \treft, 294, 335 trim,     specifying, 22 \trimedge (length), 22, 24, 27, 346, 463 \trimFrame, 318 \trimLmarks, 318 \trimmarks, 319 \trimmarks, 318 \trimNone, 318 \trimNone, 318 \trimNone, 318 \trimVollength), 22, 24, 27, 463 \trimVist (environment), 136–138, 283, 377 TrueType, 35, 469 \trimg complete language, 425 Turing, Alan, 425 \two_loadClass commands, 457 Two \LoadClass commands,	totalnumber (counter), 179	•
trademark, 334 \traditionalparskip, 49, 50  Transitional type, 40 \tref, 294, 335 trim, specifying, 22 \trimedge (length), 22, 24, 27, 346, 463 \trimTrame, 318 \trimLmarks, 318 \trimmarks, 318, 319 \trimmarks color, 318 \trimtone, 318 \trimTrimymarks (length), 22, 24, 27, 463 \trimTimymarks, 318, 319 trivlist (environment), 136–138, 283, 377  TrueType, 35, 469 \trimg, Alan, 425 \two, 433  Two \documentclass commands, 457 Two \LoadClass commands, 457 Two \coolumn (option), 4, 5, 71–73, 145, 167 \twocolumn foottotete, 231, 236 \twoside (option), 4, 5, 26, 34, 84, 238  Uncial, 469 Venetian, 469 type size, 2 default, 3 typeblock, 5, 7, 8, 17–20, 22, 26, 27, 29, 66, 110, 114, 130, 249, 250, 258, 345, 346, 348–350 height, 16, 17, 19 location, 17, 20, 23 size, 22 specifying size, 16, 20, 22 width, 16–18 typeface, 15  Alte Schwabacher, see Alte Schwabacher ATF Clearface, see ATF Clearface Avant Garde Gothic, see Avant Garde Gothic Barbedour, see Barbedour Basel, see Basel Basilia, see Basilia Baskerville, see Baskerville Bembo, see Bembo Bergamo, see Bergamo Bernhard Modern, see Bernhard Modern Modern Bodoni, see Bodoni Bookman, see Bookman Caslon, see Caslon Centaur, see Centaur Century Old Style, see Century Old Style	to, 431	Symbol, 470
\traditionalparskip, 49, 50 Transitional type, 40 \tref, 294, 335 trim, specifying, 22 \trimedge (length), 22, 24, 27, 346, 463 \trimFrame, 318 \trimLmarks, 318 \trimLmarks, 318 \trimmark, 319 \trimmarks, 318, 319 \trimtop (length), 22, 24, 27, 463 \trimtop (length), 22, 24, 27, 463 \trimtop (length), 22, 24, 27, 463 \trimtintop (length), 22, 24, 27, 463 \trimlist (environment), 136–138, 283, 377  TrueType, 35, 469 \trimg complete language, 425 Turing complete language, 425 Turing, Alan, 425 \txim 301 \txim 0documentclass commands, 457 Two \LoadClass commands, 457 Two \LoadClass commands, 457 Two \LoadClass commands, 457 Two \LoadClass commands, 457 Two \cloadclass commands, 457 Two \clo	\tracingtabularx, 222	Transitional, 40, 469
type, 40 \tref, 294, 335  trimedge (length), 22, 24, 27, 346, 463 \trimTrame, 318 \trimLmarks, 318 \trimMarks, 318 \trimTrame, 319 \trimtone, 318 \trimtone, 328 \trimtone,	trademark, 334	Uncial, 469
type, 40 \tref, 294, 335 trim,	\traditionalparskip, 49, 50	Venetian, 469
\tref, 294, 335  \trim,	Transitional	type size, 2
\tref, 294, 335  trim,	type, 40	default, 3
trim, specifying, 22 \trimedge (length), 22, 24, 27, 346, 463 \trimFrame, 318 \trimLmarks, 318 \trimLmarks, 318 \trimmarks, 319 \trimmarks, 318, 319 \trimNone, 318 \trimtypeface, 15 \trimtypeface, 26 \trimtypeface, 15 \trimtypeface, 15 \trimtypeface, 15 \trimtypeface, 26 \trimtypef	* =	typeblock, 5, 7, 8, 17–20, 22, 26, 27, 29, 66,
specifying, 22 \trimedge (length), 22, 24, 27, 346, 463 \trimeTrame, 318 \trimeTrame, 318 \trimeTrame, 318 \trimeTrame, 318 \trimeTrame, 318 \trimeTrame, 319 \trimeTrame, 318 \trimeTrame, 318 \trimeTrame, 319 \trimeTrame, 318, 319 \trimeTrame, 318, 319 \trimeTrame, 318 \trimeTrame, 318, 319 \trimeTrame, 318, 319 \trimeTrame, 318 \trimeTrame, 318, 319 \trimeTrame, 318 \trimeTrame, 319 \trimeTrame, 318 \trimeTrame, 320 \trimeTram		7.1
\trimedge (length), 22, 24, 27, 346, 463 \trimFrame, 318 \trimFrame, 318 \trimImarks, 318 \trimmarks, 319 \trimmarks, 318, 319 \trimmarkscolor, 318 \trimNone, 318 \trimXmarks, 318, 319 \trimXmarks, 318, 319 \trimXmarks, 318, 319 \trimXmarks, 318, 319 \trimImarks, 318, 319 \trimXmarks, 318, 319 \trimImarks, 318 \trimImarks, 425 \trimImarks, 425 \trimImarks, 425 \trimImarks, 425 \trimImarks, 425	•	
\trimFrame, 318 \trimLmarks, 318 \trimLmarks, 319 \trimmarks, 318, 319 \trimtone, 318 \trimtone, 32, 469 \trimtone, 32, 469 \trimtone, 32, 459 \trimtone, 32, 459 \trimtone, 32, 459 \trimtone, 32, 459 \trimtone, 45, 45, 465 \trimtone, 32, 459 \trimtone, 45, 45, 465 \trimtone, 45, 45, 465 \trimtone, 45, 45, 465 \trimtone, 45, 45, 465 \trimtone, 47, 45, 465 \trimtone, 45, 45, 465 \trimtone, 47, 45, 465 \trimton		
\trimLmarks, 318 \trimmarks, 319 \trimmarks, 318, 319 \trimmarks, 318, 319 \trimmarks, 318, 319 \trimmarks, 318 \trimmarks, 318, 319 \trimmarks, 318 \trimmone, 318 \trimtop (length), 22, 24, 27, 463 \trimtop (length), 22, 24, 27, 463 \trimtop (lenyth), 136–138, 283, 377 \trimtop (lenyth), 136–138, 283, 377 \trimtop (lenyth), 136–138, 283, 377 \trimtop (lenyth), 136–138, 283, 360 \trimtop (lenyth), 136–138, 283, 377 \trimtop (lenyth), 288 \trimmarks, 318, 319 \trimtop (length), 22, 24, 27, 463 \trimtop (length), 34, 459 \trimtop (length), 22, 24, 27, 463 \trimtop (length), 34, 459 \trimtop (length), 34, 5, 26, 34, 84, 238 \trimtop (length), 22, 24, 27, 463 \trimtop (length), 22, 24, 27, 463 \trimtop (length), 34, 5, 26, 34, 84, 238 \trimtop (length), 22, 24, 27, 463 \trimtop (length), 22, 24, 27, 463 \trimtop (length), 34, 5, 26, 34, 84, 238 \trimtop (length), 36, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20		0
\trimmark, 319 \trimmarks, 318, 319 \trimmarks, 318, 319 \trimmarkscolor, 318 \trimmone, 318 \trimtop (length), 22, 24, 27, 463 \trimtop (length), 22, 24, 27, 463 \trimtop (length), 136–138, 283, 377 \trimtop (length), 22, 24, 27, 463 \trimtop (length), 24, 24, 27, 463 \trimtop (length), 26, 24, 28, 28, 28, 28, 28, 28, 28, 28, 28, 28		
\trimmarks, 318, 319 \trimmarkscolor, 318 \trimNone, 318 \trimNone, 318 \trimtop (length), 22, 24, 27, 463 \trimtop (length), 22, 24, 27, 463 \trimtop (length), 22, 24, 27, 463 \trimtop (length), 136–138, 283, 377  TrueType, 35, 469 \trimtop (length), 42  Turing complete language, 425  Turing complete language, 425  Turing, Alan, 425  \trimtop (length), 22, 24, 27, 463  \trimtop (length), 24  \trimtop (length), 32, 26  \trimtop (length), 34, 5, 26, 34, 84, 238  \trimtop (length), 32, 26, 34, 84, 238  \trimtop (length), 34, 5, 26, 34, 84, 238  \trimtop (length), 36, 26, 34, 84, 238  \trimtop (length), 36, 26, 34, 84, 238  \trimtop (length), 36, 26, 34, 84, 238  \trimtop (length), 26, 26		
\trimmarkscolor, 318 \trimNone, 318 \trimNone, 318 \trimtop (length), 22, 24, 27, 463 \trimtop (length), 22, 24, 27, 463 \trimXmarks, 318, 319 \trimXmarks, 318, 319 \trivlist (environment), 136–138, 283, 377 \trivlist (environment), 36–138, 283, 377 \trivlist (environment), 136–138, 283, 377 \trivlist (environment), 136–138, 283, 377 \trivlist (environment), 269 \trivlist (environment),		
\trimNone, 318 \trimtop (length), 22, 24, 27, 463 \trimtop (length), 26, 34, 84, 238 Alte Schwabacher, see ATF Clearface \trimtop (ATF Clearface, see Baslia \trimtop (ATF Clearface, see Baslia \trimtop (ATF Clearface, see Baslia and see Baslia,		
\trimtop (length), 22, 24, 27, 463 \trimXmarks, 318, 319 \trimXmarks, 318, 319 \trivlist (environment), 136–138, 283, 377 \trimtop (length), 42 \trivlist (environment), 136–138, 283, 377 \trimtop (length), 42 \trivlist (environment), 136–138, 283, 377 \trimtop (length), 429 \trivlist (environment), 136–138, 283, 377 \trivlist (gothic Gothic see Avant Garde Gothic see Avant Garde Gothic see Barbedour, see Barbedour, see Barbedour, see Basel Basel, see Basel Basel, see Basel Basilia, see Basilia, see Basilia Baskerville, see Baskerville see Baskerville see Baskerville see Bembo see Bembo Bergamo, see Bergamo Bernhard Modern, see Bernhard Modern see Bernhard Modern Modern Bodoni, see Bodoni Bookman, see Bookman See Bookman See Bookman Caslon, see Caslon Caslon, see Caslon Caslon, see Caslon Centaur, see Centaur Century Old Style, see Century Old Style, see Century Old Style Style Chantilly, see Chantilly		
\trimXmarks, 318, 319  \triwlist (environment), 136–138, 283, 377  \text{TrueType, 35, 469}  \text{\text{tfamily, 42}}  \text{Turing complete language, 425}  \text{Turing, Alan, 425}  \text{\text{two, 433}}  \text{Two \localClass commands, 457}  \text{Two \localClass commands, 457}  \text{Two colindex, 301}  \text{\text{twocolumn, 73, 459, 465}}  \text{twocolumn (option), 4, 5, 71–73, 145, 167}  \text{\text{twocolumnfootstyle, 231}}  \text{twoside (option), 4, 5, 26, 34, 84, 238}  \text{ATF Clearface, see ATF Clearface}  Avant Garde Gothic, see Avant Garde  Avant Garde Gothic, see Barbedour  Basel, see Basel  Basilia, see Basilia  Baskerville, see Baskerville  Bembo, see Baskerville  Bergamo, see Bergamo  Bernhard Modern, see Bernhard  Modern  Bodoni, see Bodoni  Caslon, see Caslon  Centaur, see Centaur  Century Old Style, see Century Old  Style  Chantilly, see Chantilly		•
trivlist (environment), 136–138, 283, 377 Gothic  TrueType, 35, 469 \ttfamily, 42 Basel, see Basel  Turing complete language, 425 Turing, Alan, 425 \two.\documentclass commands, 457 Two \LoadClass commands, 457 Bergamo, see Bergamo Two \LoadClass commands, 457 Bergamo, see Bergamo Bernhard Modern, see Bernhard \twocolglossary, 314 \twocolindex, 301 \twocolindex, 301 \twocolumn, 73, 459, 465  twocolumn (option), 4, 5, 71–73, 145, 167 \twocolumnfootnotes, 231, 236 \twocolumnfootstyle, 231 twoside (option), 4, 5, 26, 34, 84, 238  Avant Garde Gothic, see Avant Garde Gothic Gothic  Barbedour, see Barbedour  Basel, see Baskerville Baskerville, see Baskerville Bembo, see Bembo Bergamo, see Bergamo Bernhard Modern, see Bernhard Modern  Caslon, see Bodoni Bookman, see Bookman Caslon, see Caslon Centaur, see Centaur Century Old Style, see Century Old Style Chantilly, see Chantilly		
TrueType, 35, 469  \text{tfamily, 42}  Turing complete language, 425  Turing, Alan, 425  \text{tw0, 433}  Two \documentclass commands, 457  Two \LoadClass commands, 457  \text{twocolindex, 301}  \text{twocolindex, 301}  \text{twocolumn, 73, 459, 465}  \text{twocolumn (option), 4, 5, 71–73, 145, 167}  \text{twocolumnfoottstyle, 231}  \text{twoside (option), 4, 5, 26, 34, 84, 238}   Basebedour, see Barbedour  Basel, see Basel  Basslia, see Basslia  Basskerville, see Baskerville  Baskerville, see Baskerville  Basslia, see Basslia  Basslia, see Bassle  Basslia, see Basslia  Basslia, see Basslia  Basslia, see Bassle  Basslia, see Basslia  Basslia, see Bassle  Basslia, see Bassle  Caslania		
TrueType, 35, 469  \ttfamily, 42  Turing complete language, 425  Turing, Alan, 425  \two.\documentclass commands, 457  Two \LoadClass commands, 457  Two \LoadClass commands, 457  Twocolindex, 301  \twocolindex, 301  \twocolumn, 73, 459, 465  twocolumn (option), 4, 5, 71–73, 145, 167  \twocolumnfootnotes, 231, 236  \twocolumnfootnotestyle, 231  twoside (option), 4, 5, 26, 34, 84, 238  Barbedour, see Barbedour  Basel, see Basel  Basilia, see Basilia  Baskerville, see Baskerville  Bembo, see Bembo  Bergamo, see Bergamo  Bernhard Modern, see Bernhard  Modern  Bodoni, see Bodoni  Caslon, see Bookman  Centaur, see Caslon  Centaur, see Centaur  Century Old Style, see Century Old  Style  Chantilly, see Chantilly		
\ttfamily, 42Basel, see BaselTuring complete language, 425Basilia, see BasiliaTuring, Alan, 425Baskerville, see Baskerville\tw@, 433Bembo, see BemboTwo \documentclass commands, 457Bergamo, see BergamoTwo \LoadClass commands, 457Bernhard Modern, see Bernhard\twocolglossary, 314Modern\twocolindex, 301Bodoni, see Bodoni\twocolumn, 73, 459, 465Caslon, see Bookman\twocolumn (option), 4, 5, 71-73, 145, 167Centaur, see Centaur\twocolumnfootnotes, 231, 236Century Old Style, see Century Old\twocolumnfootstyle, 231Styletwoside (option), 4, 5, 26, 34, 84, 238Chantilly, see Chantilly		
Turing complete language, 425  Turing, Alan, 425  \tw0, 433  Bembo, see Bembo  Bergamo, see Bergamo  Bernhard Modern, see Bernhard  \twocolglossary, 314  \twocolindex, 301  \twocoltocetc, 144, 145  \twocolumn, 73, 459, 465  twocolumn (option), 4, 5, 71–73, 145, 167  \twocolumnfootnotes, 231, 236  \twocolumnfootstyle, 231  twoside (option), 4, 5, 26, 34, 84, 238  Basilia, see Basilia  Baskerville, see Baskerville  Bermbo, see Bergamo  Bernhard Modern, see Bernhard  Modern  Bodoni, see Bodoni  Caslon, see Bookman  Caslon, see Caslon  Centaur, see Centaur  Century Old Style, see Century Old  Style  Chantilly, see Chantilly	* -	
Turing, Alan, 425  \tw0, 433  Bembo, see Bembo  Bergamo, see Bergamo  Bernhard Modern, see Bernhard  \twocolglossary, 314  \twocolindex, 301  \twocoltocetc, 144, 145  Bookman, see Bookman  \twocolumn, 73, 459, 465  twocolumn (option), 4, 5, 71–73, 145, 167  \twocolumnfootnotes, 231, 236  \twocolumnfootstyle, 231  twoside (option), 4, 5, 26, 34, 84, 238  Baskerville, see Baskerville  Bembo, see Bembo  Bergamo, see Bergamo  Bernhard Modern  Bodoni, see Bodoni  Caslon, see Caslon  Centaur, see Centaur  Centaur, see Centaur  Century Old Style, see Century Old  Style  Chantilly, see Chantilly	•	
\tw@, 433  Two \documentclass commands, 457  Bergamo, see Bergamo  Bergamo, see Bergamo  Bernhard Modern, see Bernhard  Modern  Modern  Bodoni, see Bodoni  Bookman, see Bookman  Caslon, see Caslon  Centaur, see Centaur  Century Old Style, see Century Old  Twocolumnfootstyle, 231  twoside (option), 4, 5, 26, 34, 84, 238  Bembo, see Bembo  Bergamo, see Bergamo  Bernhard Modern, see Bernhard  Modern  Casloni, see Caslon  Centaur, see Caslon  Centaur, see Centaur  Centaur, see Centaur  Century Old Style, see Century Old  Chantilly, see Chantilly		
Two \documentclass commands, 457  Two \LoadClass commands, 457  Bergamo, see Bergamo  Bernhard Modern, see Bernhard  Modern  Modern  Bodoni, see Bodoni  Bookman, see Bookman  Caslon, see Caslon  Centaur, see Centaur  Century Old Style, see Century Old  Twocolumnfootstyle, 231  twoside (option), 4, 5, 26, 34, 84, 238  Bergamo, see Bergamo  Bernhard Modern, see Bernhard  Modern  Caslon, see Caslon  Centaur, see Centaur  Centaur, see Centaur  Century Old Style, see Century Old  Style  Chantilly, see Chantilly		
Two \LoadClass commands, 457  \twocolglossary, 314  \twocolindex, 301  \twocoltocetc, 144, 145  \twocolumn, 73, 459, 465  \twocolumn (option), 4, 5, 71–73, 145, 167  \twocolumnfootnotes, 231, 236  \twocolumnfootstyle, 231  \twoside (option), 4, 5, 26, 34, 84, 238  Bernhard Modern, see Bernhard  Modern  Bodoni, see Bodoni  Bookman, see Bookman  Caslon, see Caslon  Centaur, see Centaur  Century Old Style, see Century Old  Style  Chantilly, see Chantilly		
\twocolglossary, 314 \twocolindex, 301 \twocolindex, 301 \twocoltocetc, 144, 145 \twocolumn, 73, 459, 465 \twocolumn (option), 4, 5, 71–73, 145, 167 \twocolumnfootnotes, 231, 236 \twocolumnfootstyle, 231 \twoside (option), 4, 5, 26, 34, 84, 238  Modern  Bodoni, see Bodoni  Bodoni, see Bookman  Caslon, see Caslon  Centaur, see Centaur  Century Old Style, see Century Old  Style  Chantilly, see Chantilly		
\twocolindex, 301 Bodoni, see Bodoni Bookman, see Bookman See Bookman Bookman, see Bookman Caslon, see Caslon Centaur, see Centaur Century Old Style, see Century Old Style Style Twoside (option), 4, 5, 26, 34, 84, 238 Bodoni, see Bodoni Bookman, see Bookman Caslon, see Caslon Centaur, see Centaur Century Old Style, see Century Old Style, see Century Old Chantilly, see Chantilly		
\twocoltocetc, 144, 145 Bookman, see Bookman \twocolumn, 73, 459, 465 Caslon, see Caslon twocolumn (option), 4, 5, 71–73, 145, 167 \twocolumnfootnotes, 231, 236 Century Old Style, see Century Old \twocolumnfootstyle, 231 Style twoside (option), 4, 5, 26, 34, 84, 238 Chantilly, see Chantilly	<u> </u>	
\twocolumn, 73, 459, 465  twocolumn (option), 4, 5, 71–73, 145, 167  \twocolumnfootnotes, 231, 236  \twocolumnfootstyle, 231  twoside (option), 4, 5, 26, 34, 84, 238  Caslon, see Caslon  Centaur, see Centaur  Century Old Style, see Century Old  Style  Chantilly, see Chantilly		
twocolumn (option), 4, 5, 71–73, 145, 167  \twocolumnfootnotes, 231, 236  \twocolumnfootstyle, 231  twoside (option), 4, 5, 26, 34, 84, 238  Centaur, see Centaur  Century Old Style, see Century Old  Style  Chantilly, see Chantilly		
\twocolumnfootnotes, 231, 236 \twocolumnfootstyle, 231 twoside (option), 4, 5, 26, 34, 84, 238  Century Old Style, see Century Old Style Chantilly, see Chantilly		
\twocolumnfootstyle, 231 Style twoside (option), 4, 5, 26, 34, 84, 238 Chantilly, see Chantilly		
twoside (option), 4, 5, 26, 34, 84, 238 Chantilly, see Chantilly		
	· ·	•
\TX@verb, 223 Charter, see Charter	` <b>1</b>	
	\TX@verb, 223	Charter, see Charter

Cheltenham, see Cheltenham	Nicholas Cochin, see Nicholas
Clarendon, see Clarendon	Cochin
Computer Modern, see Computer	Olde English, see Olde English
Modern	Optima, see Optima
Computer Sans, see Computer Sans	Opus, see Opus
Computer Typewriter, see Computer	Palatino, see Palatino
Typewriter	Rosslaire, see Rosslaire
Concrete Roman, see Concrete Ro-	Sabon, see Sabon
man	Savoy, see Savoy
Courier, see Courier	Schnittger, see Schnittger
Della Robbia, see Della Robbia	Serifa, see Serifa
Egyptian, see Egyptian	Serific, see Serific
English Serif, see English Serif	Struktor, see Struktor
Engravers Litho, see Engravers Litho	Symbol, see Symbol
Engravers Old English, see Engravers	Syntax, see Syntax
Old English	Times New Roman, see Times New
Fette Fraktur, see Fette Fraktur	Roman
Fette Gotisch, see Fette Gotisch	Times Roman, see Times Roman
Flanders, see Flanders	Typewriter, see Typewriter
Franklin Gothic, see Franklin Gothic	Unitus, see Unitus
Function, see Function	Univers, see Univers
Futura, see Futura	University Old Style, see University
Garamond, see Garamond	Old Style
Gill Sans, see Gill Sans	URW Antiqua, see URW Antiqua
Glypha, see Glypha	URW Latino, see URW Latino
Glytus, see Glytus	URW Palladio, see URW Palladio
Goudy Old Style, see Goudy Old	Utopia, see Utopia
Style	Vendome, see Vendome
Helvetica, see Helvetica	Walbaum, see Walbaum
Jenson Recut, see Jenson Recut	Zapf Chancery, see Zapf Chancery
Jessica, see Jessica	Zapf Dingbats, see Zapf Dingbats
Joanna, see Joanna	\typeout, 15
Lanston Bell, see Lanston Bell	\typeoutlayout, 26
Lanston Koch, see Lanston Koch	\typeoutstandardlayout, 26
Latin Modern, see Latin Modern	Typewriter, 15, 223
Linden, see Linden	typewriter, 371
Litho Antique, see Litho Antique	typist, 371
Lydian, see Lydian	U
Melior, see Melior	O
Minion, see Minion	Vucminuanomo 320 335
Modern, see Modern	\ucminusname, 329, 335
New Baskerville, see New	numberline, 400 Unbalanced output routine, 451
Baskerville	Unbalanced write command, 451
New Century Schoolbook, see New	\unboldmath, 458
Century Schoolbook	Undefined control sequence, 451
News Gothic, see News Gothic	Undefined index file, 466
	Officialities mack inc, 400

Undefined tab position, 457	\vector, 415, 416, 454
Underfull \hbox, 451	vector, see also picture object
Underfull \vbox, 451	reference point, 415
underfull, 431	restricted slope, 415
\undodrop, 252	slope, 415
Unicode, 35	veelo (chapterstyle), 92, 395, 396
unit length, 405	Veelo, Bastiaan, 92, 238, 398
\unitlength, 406, 407	Vendome, 469
\unitlength (length), 131, 161, 252, 349,	\verb, 222, 223, 277, 303
405, 406	\verb, 458
Unitus, 469	\verb ended by end of line, 458
Univers, 469	\verb illegal in command argument, 271,
University Old Style, 469	458
Unknown document division, 463	\verb may be unreliable, 466
Unknown mark setting type, 463	\verb*, 223, 277, 278
Unknown numbering type, 463	\verbatim, 278, 283, 284
Unknown option, 458	verbatim, 271, 277–284
Unknown toclevel for, 466	changing font, 278
\unletcounter, 322	font, 282
\unnamedsubappendices, 78, 79	frame, 279, 286
Unrecognized argument for	styling, 280
\sidecapmargin, 464	in argument, 271
Unused global option(s), 460	new, 281–284
\upbracefill, 218	short, 277
\uppercaseheads, 107, 112, 113	with tab spaces, 278
\uppermargin and/or \textheight	wrap long lines, 279
and/or \lowermargin are too	write, 286, 289
large, 464	verbatim (environment), 138, 256, 277-
\uppermargin (length), 19, 23, 24, 462,	279, 282, 283, 286, 378, 453, 465
464	verbatim (package), 278, 336
\upshape, 42	verbatim* (environment), 277-279
url (package), 358, 375	\verbatim@font, 283
URW Antiqua, 469	\verbatim@line, 283
URW Latino, 469	\verbatim@processline, 283
URW Palladio, 469	\verbatim@startline, 283, 284
Use of doesn't match, 451	\verbatimbreakchar, 279
\usebox, 271, 412	\verbatimindent (length), 279
\usepackage, 336, 337, 454, 455, 458	\verbatiminput, 286
\usepackage before \documentclass,	\verbatiminput*, 286
458	verbatimoutput (environment), 285
\usethanksrule, 69	\verbfootnote, 230
Utopia, 16, 40	Verdana, 353, 369
•	verse, 255–270
V	centering, 257, 258
	end a line, 255
\valign, 445, 448	end a stanza, 255
\vbox, 431, 444, 449, 451	indent line, 256, 257

verse (environment)

indent pattern, 257, 267	widow
indent space, 258, 259	line, 52–54
long line, 257, 258	\width (length), 273
long lines, 256	wilsondob (chapterstyle), 92, 104
multiple indexes, 257	wilsondob (headstyles), 104
prevent page break, 257	wrapfig (package), 199
typesetting environments, 255–257	\wrappingoff, 279
wrapped line indent, 258	\wrappingon, 279
verse (environment), 255–258, 260–262	\write, 451
verse (package), 336	write,
\verselinebreak, 259, 267	verbatim, 291
\verselinenumbersleft, 262	writefigure (environment), 288, 289
\verselinenumbersright, 262	writeverbatim (environment), 285, 289
\versewidth (length), 257, 258	,,,,,
vertical,	X
centering, 336	
vertical mode, 431	X columns too narrow (table too wide),
verville (chapterstyle), 92	223
Verville, Guy, 92	X columns too narrow, 466
\vfil,436	xcolor (package), 272, 375
\vfill, 436	\xdef, 430
\vfilneg, 436	Xdepth (counter), 168
\vgap (length), 258, 259, 261	XeTeX, 35, 333, 464, 469
\vin, 258	XeTeX is required to process this docu-
\vindent (length), 258	ment, 464
\vinphantom, 259	\Xheadstart, 146
\vleftmargin (length), 258	xindy (program), 304
\vleftofline, 259	$\xim 304$
\vleftskip, 342	\xlvchars (length), 15, 16
\vleftskip (length), 262	\Xmark, 146
\vline, 224	xtab (package), 225
vminipage (environment), 52	•
vplace (environment), 336	Y
\vrightskip,342	
\vrightskip (length), 262	You can't use in, 451
\vskip, 51, 436	You can't use 'macro parameter character
\vspace*, 86	#' in mode, 451
\vss, 436	You can't use '\spacefactor' in vertical mode, 451
W	You have requested release, 460
	You have requested version, 460
Walbaum, 469	You have used the '*pt' option but file,
warning,	464
LaTeX, 458–460	7
memoir class, 464–466	Z
TeX, 444–452	\=@ 278 422 426
\wd, 432	\z@, 378, 433, 436

Zapf Chancery \zerotrivseps

Zapf Chancery, 41 Zapf Dingbats, 41, 375 Zapf, Hermann, 40, 41 zero-sized picture, 119 \zerotrivseps, 137

## Colophon

This manual was typeset using the LaTeX typesetting system created by Leslie Lamport and the memoir class. The body text is set 10/12pt on a 33pc measure with Palatino designed by Hermann Zapf, which includes italics and small caps. Other fonts include Sans, Slanted and Typewriter from Donald Knuth's Computer Modern family.