

`\begin{titlepage}`

`\bfseries`

`\mdseries`

`\hrulefill`

`\small`

`\vfill`

`\footnotesize`

`\end{titlepage}`

레이텍

입문부터

활용까지

강우현

사용 환경 준비, 문서 작성, 수식 작성에서 비롯하여
표 작성, 상호 참조와 인용, 색인 생성과 매크로까지
LaTeX의 기본 기능과 주요 기능을 쉽게 익혀 보자!

레이텍 입문부터 활용까지

강우현

레이텍 입문부터 활용까지

작성 강우현 수리과학과 학사 21

편집 강우현 수리과학과 학사 21

발행일 2024년 4월 29일

Copyright 2024. 강우현

저작권법에 의해 보호를 받는 저작물이므로

무단 전재와 무단 복제를 금합니다.

들어가며

오늘날 자연과학 및 공학 분야의 많은 논문은 \LaTeX 이라는 문서 작성 시스템을 사용해 작성됩니다. \LaTeX 이 자주 쓰이는 이유는 이 프로그램의 두 가지 특징 때문일 것입니다.

- 다른 워드프로세서에 비해 미려한 출판물을 생성함
- 수식 작성 기능이 다른 프로그램에 비해 편리하고 강력함

이 특징은 논문을 작성할 때가 아니라 평범한 문서를 만들거나 책을 작성할 때에도 편리하게 작용합니다. 대학교에서는 수업 과제물을 \LaTeX 으로 작성할 것을 요구하는 교수님도 계십니다.

하지만, \LaTeX 을 처음 접하는 사람이라면 \LaTeX 을 사용하는 것이 직관적이지 않다고 느낄 것입니다. 이는 \LaTeX 으로 문서를 작성하는 방식이 우리가 흔히 접하던 아래아 한글이나 Word 등의 다른 워드프로세서와는 다르기 때문입니다. \LaTeX 을 사용하려면 문서의 실제 모습을 보지 못하는 채로 코드를 작성해야 하고, 이 코드를 컴파일 해야 실제 모습을 볼 수 있습니다. 프로그래밍이 익숙하거나 HTML 등으로 웹 디자인을 해 본 경험이 있다면 쉽게 적응하겠지만, 그런 경험이 없다면 \LaTeX 으로 문서를 작성하는데 꽤 애를 먹을 것입니다.

이 책의 첫 번째 목표는 **\LaTeX 을 처음 접하는 분들이 여기에 익숙해지고, 기본적인 문서를 손쉽게 작성할 수 있게 하는 것**입니다. 문서의 내용, 서식, 레이아웃 등을 모두 플레인 텍스트로 입력하고 컴파일 하여 PDF 파일을 얻는 과정이 익숙해지도록, 또 \LaTeX 의 필수 문법이 손에 익도록 많은 예제를 제시하였습니다.

《The Not-so-short Introduction to \LaTeX 2 ϵ 》[36] 등의 안내서가 이미 있고, 웹사이트에 검색하여 \LaTeX 의 사용법을 설명하는 많은 글을 찾을 수 있습니다. 하지만, 이는 어느 정도의 배경 지식이 있음을 전제하거나 대부분 영어로 쓰여 있기 때문에 아무것도 모르는 초보자가 이를 읽고 \LaTeX 을 배우기는 쉽지 않습니다(\LaTeX 의 많은 기능을 원활하게

사용하려면 결국에는 영어를 읽는 데 익숙해져야 합니다). 그래서, \LaTeX 에 대해 아무 배경 지식이 없는 분들도 예제를 따라하며 문서 작성 과정을 이해할 수 있도록, 또 한국어를 사용해 쉽게 읽을 수 있도록 이 책을 작성하는 데 집중하였습니다.

이 책의 두 번째 목표는 \LaTeX 을 어느 정도 다루어 보신 분들에게 참고 가이드가 되는 것입니다. \LaTeX 은 대표적인 수식 작성 기능 외에 목차 생성, 상호 참조, 표 작성, 색인 생성 등 수많은 기능을 가지고 있습니다. 여러 기능을 빠르게 사용할 수 있도록 중요한 것들을 일목요연하게 정리하여 소개할 것입니다. 특정 기능을 어떻게 사용하는지 알아보기 위해 웹 상의 수많은 글을 일일이 읽으면서 찾아낼 필요 없이, 이 책에서 간략히 정리해 둔 내용을 읽음으로써 그 기능을 쉽게 사용하실 수 있도록 하였습니다.

사실 \LaTeX 을 익히는 가장 좋은 방법은 직접 부딪혀 보는 것이라고 생각합니다. 아무리 글을 열심히 읽어도 직접 사용해 보지 않으면 배운 코드를 어떻게 활용하는지 잘 와닿지 않을 뿐 아니라 금방 잊어버리기 때문입니다. 반면, 여러 문서를 작성하다 보면 자연스럽게 코드가 머릿속에 남고, 매뉴얼을 찾아보지 않아도 문서를 빠르게 작성할 수 있게 됩니다. 따라서, \LaTeX 을 배우기 전 작성하고 싶은 문서를 생각해 보시기 바랍니다. 평소에 알고 있던 개념을 정리하면서 \LaTeX 을 배우도 좋고, 관심 분야의 글을 정리하거나 논문 또는 책의 내용을 따라서 써 보는 것도 좋습니다. 작성할 문서의 주제를 정하면 코드를 쉽게 익힐 수 있을 뿐 아니라, \LaTeX 의 기능을 배우는 데에도 동기부여가 될 것입니다.

이 책의 내용

이 책은 제I편 도입부, 제II편과 제III편의 기본 사용법, 제IV편부터 제VI편까지의 다양한 기능, 그리고 제VII편 부록으로 나눌 수 있습니다.

제I편부터 제III편까지는 기본적인 \LaTeX 사용법을 다룹니다. 제I편에서는 \LaTeX 에 대한 소개와 함께, 문서를 작성하는 과정과 기본 문법을 설명합니다. 제1장에서는 \LaTeX 의 간단한 역사와 배경 지식을 설명합니다. 여기에서는 \LaTeX 의 문법을 설명하지 않으므로, 간단하게 읽고 넘어가도 좋습니다. 제2장의 안내를 따르면 \LaTeX 으로 문서를 작성하는 과정을 익힐 수 있고, 제3장을 읽으면 \LaTeX 에서 명령을 내리는 방법을 알 수 있습니다.

제II편에서는 문서를 작성할 때 사용할 수 있는 여러 요소 중, 수식을 제외한 것의 문법을 설명합니다. \TeX 의 용어를 사용하면 ‘텍스트 모드’에서 사용할 수 있는 기능을 설명한다고 말할 수 있습니다. 문자를 입력하고, 표지와 장절 표제를 만들고, 특수한

형태의 문단을 만들고 글꼴을 바꾸는 등, 다른 워드프로세서에 비유했을 때 ‘기본 기능’이라고 말할 수 있는 기능을 다룹니다. 제 III 편에서는 마침내 \LaTeX 의 대표적인 기능인 수식을 작성하는 방법을 설명합니다. 수식 모드·텍스트 모드의 차이와 기본적인 수식 요소를 입력하는 방법을 설명하고, ‘큰 수식’은 어떻게 보기 좋게 입력하는지, 또 가환 도표는 어떻게 그리는지 설명합니다.

제 IV 편부터 제 VI 편까지는 \LaTeX 의 더 다양한 기능을 다룹니다. 먼저 제 IV 편에서는 표와 그림을 어떻게 문서에 넣고 배치하는지 설명합니다. 제 V 편에서는 \LaTeX 의 다양한 참조 기능을 다룹니다. 교차 참조 및 링크 삽입, 참고 문헌 인용 및 목록 출력, 하이퍼링크 및 북마크 생성을 어떻게 하는지 3개의 장에 걸쳐 설명합니다. 마지막으로, 제 VI 편에서는 \LaTeX 문서를 편리하게 작성하는 방법을 설명합니다. 매크로, 카운터와 길이 변수를 다루는 방법을 알아보고, 여러 파일에 소스 코드를 분리해 작성한 뒤 하나로 합쳐 컴파일 하는 방법을 설명합니다.

마지막 부록에서는 \LaTeX 의 문법 외적인 부분을 설명합니다. 컴퓨터에 \TeX 을 설치해 사용하는 방법, 문제가 생겼을 때 해결하는 방법을 설명하고, 본문에서 못 다 한 이야기와 문서를 작성하는 데 도움이 될 만한 정보를 여기에 적어 두었습니다.

\LaTeX 을 처음 사용하는 분은 책을 처음부터 제 III 편까지 순서대로 읽는 것을 권장합니다. 제 I 편에서 기본적인 문법을 설명하고, 제 II 편과 제 III 편에서는 본문과 수식을 작성하는 기본적인 문법을 설명하고 있기 때문입니다. 뒤쪽의 내용은 \LaTeX 을 사용하면서 더 많은 기능이 필요할 때 읽어 나가면 됩니다.

궁금한 내용 빠르게 찾기

자주 사용하는 기능을 빠르게 찾기 위한 목록을 아래에 제공합니다.

- \LaTeX 의 기본 문법 → 제3장
- 전처리부, 클래스와 패키지 → 3.2.1절
- 텍스트 모드에서 사용할 수 있는 기호 → 4.3절
- 장과 절 표제 만들기 → 5.3.1절
- 목록 만들기 → 6.1.1절
- 본문 글꼴 바꾸기 → 6.2.2절
- 수학적 문단 만들기 → 제7장
- 수식 모드에서 사용할 수 있는 기호 → 8.2절
- 수식 문법 → 8.3절
- 수식 글꼴 바꾸기 → 9.5절

- 여러 줄짜리 수식 작성하기 → 제10장
- 표 그리기 → 12.1절
- 그림 넣기 → 13.2절
- 번호 붙은 대상 참조하기 → 15.1절
- 매크로 정의하기 → 18.1절

많은 분들께 이 책의 내용이 도움이 되기를 바랍니다.

2024년 4월 29일

강우현

약식 차례

제 I 편 L^AT_EX에 관한 기본 지식 · 1

- 제1장 T_EX과 L^AT_EX 소개 · 3
- 제2장 문서 작성 과정 · 11
- 제3장 문법과 코드의 구조 · 19

제 II 편 본문 작성하기 · 35

- 제4장 문자 입력하기 · 37
- 제5장 목차 구성하기 · 53
- 제6장 여러 가지 서식 · 65
- 제7장 수학적 문단 작성하기 · 81

제 III 편 수식 작성하기 · 93

- 제8장 기본적인 수식 작성하기 · 95
- 제9장 다양한 수식 작성하기 · 123
- 제10장 별행수식 작성하기 · 141
- 제11장 가환 도표 그리기 · 151

제 IV 편 표와 그림 다루기 · 165

- 제12장 표 작성하기 · 167
- 제13장 그림 넣기 · 185
- 제14장 표와 그림의 배치 · 193

제 V 편 참조와 인용 · 205

- 제15장 상호 참조 · 207

제16장 참고 문헌 · 229

제17장 PDF 파일의 링크와 속성 · 247

제 VI 편 편리하게 문서 작성하기 · 257

제18장 매크로 정의하기 · 259

제19장 카운터와 길이 변수 · 269

제20장 문서 파일 분리하기 · 283

제 VII 편 부록 · 287

부록 A \TeX 배포판 사용하기 · 289

부록 B 도움말과 추가 자료 · 295

차례

들어가며	iii
차례	ix
표 차례	xi
그림 차례	xiii

제 I 편 L^AT_EX에 관한 기본 지식 ————— 1

제1장 T_EX과 L^AT_EX 소개 3

- 1.1 간단한 역사와 특징 · 3
 - 1.1.1 T_EX의 개발 · 3
 - 1.1.2 L^AT_EX의 개발 · 4
 - 1.1.3 장점과 단점 · 6
- 1.2 사용해 보려면 · 7
 - 1.2.1 필요한 프로그램 · 7
 - 1.2.2 컴파일 엔진 · 8
- 1.3 관련 웹사이트 · 10

제2장 문서 작성 과정 11

- 2.1 두 가지 예시 · 11
 - 2.1.1 첫 번째 예시: 영어 문서 작성하기 · 12
 - 2.1.2 두 번째 예시: 한국어 문서 작성하기 · 12

2.2	자유롭게 문서 작성하기 · 14
2.2.1	알아두면 유용한 네 가지 명령어 · 16
2.2.2	컴파일 오류와 경고 · 17
2.2.3	추가 정보 · 18

제3장 문법과 코드의 구조 19

3.1	명령어 · 19
3.1.1	제어 문자열 · 19
3.1.2	환경 · 22
3.1.3	주석 · 22
3.2	소스 파일의 구조 · 23
3.2.1	전처리부 · 23
3.2.2	document 환경 · 27
3.3	기본적인 템플릿 · 31
	연습문제 · 33

제II편 본문 작성하기 35

제4장 문자 입력하기 37

4.1	로마자 입력하기 · 38
4.1.1	확장 로마자 · 38
4.1.2	대·소문자 전환 · 38
4.2	문장 부호 입력하기 · 39
4.2.1	유럽 언어의 문장 부호 · 39
4.2.2	단어 사이의 간격과 문장 사이의 간격 · 41
4.2.3	한국어 문장 부호 · 42
4.3	기호 입력하기 · 46
4.3.1	L ^A T _E X의 기본 기호 · 46
4.3.2	textcomp 패키지의 기호 · 48
4.3.3	문자 코드로 기호 직접 입력하기 · 51
	연습문제 · 51

제5장 목차 구성하기 53

5.1 문서의 3단 구성 · 53

5.1.1 클래스에 따른 문서의 구성 요소 · 53

5.1.2 book 클래스의 추가 명령어 · 54

5.2 처음 부분 · 55

5.2.1 표지 · 55

5.2.2 초록 · 56

5.2.3 목차 · 56

5.3 가운데 부분 · 57

5.3.1 표제 · 57

5.3.2 부록 · 60

5.3.3 주석 · 60

5.4 마지막 부분 · 62

5.4.1 참고 문헌 목록 · 62

5.4.2 색인 · 64

제6장 여러 가지 서식 65

6.1 다양한 형태의 문단 · 65

6.1.1 목록 · 65

6.1.2 운문과 인용문 · 67

6.1.3 문단 정렬 바꾸기 · 68

6.2 문자의 서식 · 70

6.2.1 특정 부분 강조하기 · 70

6.2.2 글꼴의 모양 바꾸기 · 71

6.2.3 글꼴의 크기 바꾸기 · 72

6.2.4 기타 서식 · 73

6.3 프로그램 코드 조판하기 · 75

연습문제 · 77

제7장 수학적 문단 작성하기 81

7.1 기본적인 수학적 문단 만들기 · 81

7.1.1 정리형 환경 정의하기 · 81

7.1.2	정의한 환경 사용하기	· 82
7.1.3	증명 작성하기	· 83
7.2	다양한 형태의 수학적 문단	· 84
7.2.1	환경의 번호 다루기	· 84
7.2.2	문단 스타일 바꾸기	· 86
7.3	예시 코드	· 88
	연습문제	· 89

제Ⅲ편 수식 작성하기 93

제8장 기본적인 수식 작성하기 95

8.1	LaTeX에서의 수식 입력	· 95
8.1.1	수식 모드	· 95
8.1.2	행중수식과 별행수식	· 97
8.1.3	수식 관련 패키지	· 98
8.2	수식 기호	· 98
8.2.1	비슷한 기호의 구분	· 110
8.2.2	모양은 같지만 앞뒤의 간격이 다른 기호	· 112
8.3	기본적인 수식 요소	· 114
8.3.1	분수와 이항 계수	· 114
8.3.2	근호	· 115
8.3.3	첨자	· 116
8.3.4	기타 요소	· 117
8.4	괄호의 높이 조절	· 118
8.4.1	자동 조절	· 118
8.4.2	수동 조절	· 119
	연습문제	· 121

제9장 다양한 수식 작성하기 123

9.1	표 형태의 구조	· 123
9.1.1	행렬	· 123
9.1.2	경우	· 126

9.2	분수 형태의 구조	127
9.2.1	스타일이 지정된 분수와 이항 계수	128
9.2.2	일반화된 분수	128
9.2.3	분자와 분모를 두 줄로 적기	129
9.3	복잡한 식의 첨자	129
9.3.1	첨자 여러 줄로 입력하기	129
9.3.2	대형 연산자의 계산 범위	130
9.3.3	기호 왼쪽에 첨자 입력하기	131
9.4	기호 위아래에 기호 입력하기	132
9.4.1	화살표 위아래에 기호 입력하기	132
9.4.2	기호 위아래에 기호 덧붙이기	133
9.4.3	식 위아래에 화살표·괄호 그리기	134
9.5	수식 글꼴	135
9.5.1	볼드체 수식	138
	연습문제	139

제10장 별행수식 작성하기 141

10.1	기본 환경	141
10.2	여러 줄의 수식	142
10.2.1	하나의 수식을 여러 줄로 조판하기	142
10.2.2	여러 수식을 세로로 나열하기	143
10.2.3	여러 수식을 여러 열과 행으로 조판하기	145
10.2.4	수식과 수식 사이의 일반 텍스트	146
10.3	수식의 번호	147
10.3.1	번호 지우거나 임의의 기호 붙이기	147
10.3.2	하위 번호 만들기	148
10.3.3	번호를 조판할 위치	149

제11장 가환 도표 그리기 151

11.1	기본적인 가환 도표 그리기	151
11.1.1	화살표의 대체 문법	155
11.2	각종 옵션	157
11.2.1	tikzcd 환경의 옵션	157

- 11.2.2 화살표의 옵션 · 159
- 11.2.3 레이블의 옵션 · 162
- 연습문제 · 163

제Ⅳ편 표와 그림 다루기 165

제12장 표 작성하기 167

- 12.1 기본적인 표 그리기 · 167
- 12.2 환경의 세부 문법 · 169
 - 12.2.1 열 속성 인자 · 169
 - 12.2.2 세로 정렬 인자 · 174
- 12.3 환경 안에서 사용할 수 있는 명령어 · 175
 - 12.3.1 칸 합치기 · 175
 - 12.3.2 가로선 그리기 · 179
- 12.4 표 전체 폭 조정하기 · 180
 - 12.4.1 tabular* 환경 · 180
 - 12.4.2 tabularx 환경 · 181
- 연습문제 · 182

제13장 그림 넣기 185

- 13.1 직접 그림 그리기 · 185
 - 13.1.1 picture 환경 · 186
 - 13.1.2 tikz 패키지 · 186
 - 13.1.3 다른 도구 · 187
- 13.2 외부 그림 파일 삽입하기 · 188
 - 13.2.1 그림 삽입 옵션 · 189
 - 13.2.2 기타 설정 · 190

제14장 표와 그림의 배치 193

- 14.1 유동 개체로 취급하여 배치 · 193
 - 14.1.1 기본 사용 방법 · 193
 - 14.1.2 배치 허용 위치 지정하기 · 195
 - 14.1.3 2단 편집 문서의 유동 개체 · 198

- 14.2 판면 좌우에 배치 · 198
 - 14.2.1 개체를 수평 방향으로 이동하기 · 200
 - 14.2.2 주의 사항 · 201
- 14.3 캡션과 개체의 번호 · 201
 - 14.3.1 하위 번호 만들기 · 201
- 14.4 새로운 종류의 개체 만들기 · 203
 - 14.4.1 기본 사용 방법 · 203
 - 14.4.2 개체 목차 만들기 · 204
 - 14.4.3 스타일 기능 · 204

제 V 편 참조와 인용 205

제15장 상호 참조 ······ 207

- 15.1 L^AT_EX의 상호 참조 기능 · 207
 - 15.1.1 자동 조사 · 210
 - 15.1.2 번호의 출력 형식 지정 · 210
- 15.2 참조 문구를 변경해 주는 패키지 · 212
 - 15.2.1 amsmath 패키지: 수식 참조 · 212
 - 15.2.2 varioref 패키지: 참조 대상의 위치에 따른 문구 · 214
 - 15.2.3 cleveref 패키지: 참조 대상의 종류에 따른 문구 · 219
 - 15.2.4 hyperref 패키지: 참조 대상의 종류에 따른 문구 · 226

제16장 참고 문헌 ······ 229

- 16.1 참고 문헌 관리 방법 · 229
 - 16.1.1 문헌 정보 파일 · 229
 - 16.1.2 정렬 프로그램 · 230
 - 16.1.3 파일 처리 과정 · 231
- 16.2 문헌 정보 파일 작성하기 · 232
 - 16.2.1 B_BT_EX의 문헌 정보 파일: 기본 기능 · 232
 - 16.2.2 B_BT_EX의 문헌 정보 파일: 심화 기능 · 236
 - 16.2.3 biber/biblatex의 문헌 정보 파일 · 240
- 16.3 L^AT_EX 문서 작성하기 · 240
 - 16.3.1 B_BT_EX을 사용하는 경우 · 244

16.3.2 biber/biblatex을 사용하는 경우 · 244

16.3.3 문헌 인용하기 · 245

제17장 PDF 파일의 링크와 속성 ······ 247

17.1 하이퍼링크 · 247

17.1.1 자동으로 생성되는 하이퍼링크 · 247

17.1.2 하이퍼링크 직접 만들기 · 248

17.2 책갈피 · 251

17.2.1 대체 문자열 지정 · 251

17.2.2 직접 책갈피 만들기 · 252

17.2.3 책갈피의 앵커 · 252

17.3 패키지의 옵션 · 253

17.3.1 자동 하이퍼링크 설정 범위 · 254

17.3.2 하이퍼링크 표시 방법 · 254

17.3.3 책갈피 관련 옵션 · 255

17.3.4 PDF 파일의 메타데이터 · 255

제 VI 편 편리하게 문서 작성하기 ————— 257

제18장 매크로 정의하기 ······ 259

18.1 제어 문자열 형태의 매크로 · 260

18.1.1 세 가지 종류의 매크로 · 260

18.1.2 기존 제어 문자열 재정의하기 · 262

18.1.3 세부 사항 · 262

18.2 환경 형태의 매크로 · 264

18.3 유용한 매크로 · 265

제19장 카운터와 길이 변수 ······ 269

19.1 정수 값과 카운터 · 269

19.1.1 카운터 기본 사용 방법 · 269

19.1.2 카운터 번호의 출력 형식 · 272

19.1.3 카운터의 종속 · 274

19.1.4 표준 클래스의 기본 카운터 · 275

19.2 길이 값과 길이 변수 · 277	
19.2.1 길이 변수 기본 사용 방법 · 277	
19.3 값 지정하기 · 278	
19.3.1 정수 값 · 278	
19.3.2 길이 값 · 280	
19.3.3 값 사이의 연산 · 282	
제20장 문서 파일 분리하기 ······ 283	
20.1 \input 제어 문자열 · 283	
20.2 \include 제어 문자열 · 285	
제 VII 편 부록 ····· 287	
부록 A T_EX 배포판 사용하기 ······ 289	
A.1 T _E X 배포판 설치 · 289	
A.1.1 T _E X Live와 MacT _E X · 289	
A.1.2 MikT _E X · 290	
A.1.3 편집기 및 뷰어 · 290	
A.2 T _E X 배포판 관리 · 291	
A.2.1 T _E X Live · 291	
A.2.2 MacT _E X · 292	
A.2.3 MiK _T _E X · 293	
A.3 T _E X의 명령줄 인터페이스 · 294	
부록 B 도움말과 추가 자료 ······ 295	
B.1 문제 해결 · 295	
B.1.1 오류 및 경고 메시지 · 295	
B.1.2 커뮤니티의 도움 받기 · 302	
B.2 용어의 한국어 표기 · 303	
후기 ······ 307	
참고 문헌 ······ 309	

용어 찾아보기	315
명령어 찾아보기	319
클래스·패키지 찾아보기	331

표 차례

표 3.1	자주 사용되는 문서 클래스 · 24
표 3.2	표준 문서 클래스의 주요 옵션 · 25
표 3.3	자주 사용되는 패키지 · 26
표 3.4	\TeX 의 특수 문자 · 30
표 4.1	확장 로마자 · 38
표 4.2	문장 부호를 입력하는 문자열 · 40
표 4.3	한국어 문장 부호 입력 명령어 · 42
표 4.4	kopunct 패키지: 한국어 문장 부호를 입력하는 제어 문자열 · 44
표 4.5	\LaTeX 의 기호 · 47
표 4.6	텍스트 모드와 수식 모드에서 모두 사용 가능한 기호 · 47
표 4.7	textcomp 패키지: 악센트 · 48
표 4.8	textcomp 패키지: 기호 · 49
표 5.1	장절 표제 명령어 · 58
표 6.1	텍스트 모드의 글꼴 · 72
표 6.2	글꼴 크기를 바꾸는 명령어 · 73
표 6.3	기본 글꼴 크기에 따라 명령어가 설정하는 실제 글꼴 크기 · 74
표 8.1	이항 연산자 · 99
표 8.2	대형 연산자와 적분 기호 · 100
표 8.3	로그형 함수 · 101
표 8.4	관계 기호 · 103

표 8.5	mathtools 패키지: 관계 기호	· 105
표 8.6	화살표	· 106
표 8.7	괄호	· 107
표 8.8	문장 부호	· 107
표 8.9	일반 기호	· 108
표 8.10	그리스 문자	· 109
표 8.11	수식 악센트	· 109
표 8.12	높이를 조절할 수 있는 기호	· 118
표 9.1	amsmath 패키지: 행렬 작성 환경	· 123
표 9.2	화살표 위아래에 기호를 입력하는 제어 문자열	· 133
표 9.3	식 위아래에 그리는 화살표와 괄호	· 135
표 9.4	수식 모드의 글꼴	· 136
표 11.1	tikz-cd 패키지: 화살표의 모양	· 159
표 11.2	tikz-cd 패키지: 화살표의 부분별 모양	· 160
표 12.1	table 및 array 환경의 열 속성 인자	· 170
표 13.1	graphicx 패키지: 그림을 삽입할 때의 주요 옵션	· 190
표 14.1	유동 개체의 배치 허용 위치 인자	· 195
표 14.2	wrapfig 패키지: 개체의 배치 방향	· 199
표 15.1	varioref 패키지: 참조 대상의 위치에 따른 문구	· 215
표 15.2	kotex-varioref 패키지: 참조 대상의 위치에 따른 문구	· 217
표 16.1	BibTeX: 문헌의 종류	· 233
표 16.2	BibTeX: 정보 필드의 종류	· 234
표 16.3	biber/biblatex: 문헌의 종류	· 241
표 19.1	카운터 값 표시 형식	· 272
표 19.2	kotex 패키지: 카운터 값 표시 형식	· 273
표 19.3	L ^A T _E X의 번호 카운터	· 276
표 19.4	T _E X에서 사용할 수 있는 길이의 단위	· 280

그림 차례

그림 2.1	\LaTeX 문서 작성 과정 · 11
그림 2.2	예시 문서 2.1의 컴파일 결과. · 13
그림 2.3	예시 문서 2.2의 컴파일 결과. · 15
그림 5.1	클래스에 따른 문서 구성 · 54
그림 7.1	7장의 연습문제, 1페이지 · 90
그림 7.2	7장의 연습문제, 2페이지 · 91
그림 12.1	표를 작성하는 코드의 시각화 · 168
그림 14.1	$\langle lines \rangle$ 인자를 지정하지 않았을 때 그림이 잘못 배치된 모습 · 200
그림 16.1	문헌 목록 생성 시의 파일 처리 과정 · 231
그림 17.1	Adobe Acrobat의 책갈피 창 · 251
그림 A.1	\TeX Live와 MikTeX 의 차이점 · 290



L^AT_EX에 관한 기본 지식

제1편에서는 문서를 작성하는 데 쓰이는 다양한 명령어가 아닌, L^AT_EX에 관한 기본적인 지식을 배웁니다. L^AT_EX으로 어떤 문서를 작성하든 알고 있어야 하는 내용을 다루므로, L^AT_EX에 익숙하더라도 잘 모르는 내용이 있는 분은 이 부분을 잘 읽어 두기를 권장합니다.

제1장 T_EX과 L^AT_EX 소개 · 3

제2장 문서 작성 과정 · 11

제3장 문법과 코드의 구조 · 19

제 1 장

TeX과 LaTeX 소개

이번 장에서는 \LaTeX 이 무엇인지 소개하며, \LaTeX 코드와 문법을 설명하기에 앞서 여러분이 알고 있어야 할 내용을 짚어봅니다.

1.1 간단한 역사와 특징

\LaTeX 이 오늘날까지 발전되어 온 과정과 그 특징을 간략하게 짚어보겠습니다. \LaTeX 은 \TeX 이라는 프로그램 위에서 작동하므로, 이에 대한 역사와 특징도 함께 알아봅니다.

1.1.1 \TeX 의 개발

\TeX 은 1978년 도널드 어빈 커누스(Donald Ervin Knuth)가 개발한 문서 조판 프로그램입니다. 자신의 저서인 <The Art of Computer Programming> [25]을 미려하게 조판하기 위해 처음 만들어진 이 프로그램은 1982년 안정화를 거친 두 번째 버전이 나왔고, 1989년에 나온 세 번째 버전이 지금까지 이어지고 있습니다. 1989년 발표된 버전인 \TeX 3은 매우 안정하며, 이 이후로는 큰 변경 사항 없이 버그 수정만 이루어지고 있습니다. \TeX 의 버전은 원주율 π 에 수렴하며, 2024년 현재 최신 버전은 3.141592653입니다.

아래아한글이나 Word, 또는 InDesign 등의 워드프로세서 및 출판물 편집 도구와는 다르게, \TeX 은 ‘입력 파일’과 ‘출력 파일’이 구분되어 있습니다. ‘입력 파일’은 \LaTeX 의 확장자를 가지며, 원고의 내용과 함께 어떤 서식을 지정하고 어디에서 페이지를 나눌지 등의 정보가 명령어로서 작성되어 있습니다. \TeX 은 이 입력 파일을 읽어들인 뒤 컴파일 과정을 통해 \LaTeX 확장자의 출력 파일을 만듭니다. 이 파일은 오늘날의 PDF 파일과 비슷하게, 프린터를 통해 인쇄할 때와 동일한 형태의 문서를 담고 있습니다. 이처럼 코드를 사용하여 문서를 작성하는 방식은 비직관적이지만, \TeX 의 매크로 기능을 사

용한다면 논리적인 구조가 잘 짜여 있고 일관된 서식이 사용된 문서를 쉽게 만들 수 있습니다. (이 매크로 기능을 사용하여 만들어진 것 중 가장 대표적인 것이 바로 \LaTeX 입니다.)

\TeX 의 최대 장점은 강력한 수식 작성 기능입니다. 직관적으로 문서를 작성할 수 있는 Word가 1980년대에 개발되었음에도 불구하고, 자연과학 및 일부 공학 분야에서는 지금까지도 \TeX 을 사용하여 문서를 작성합니다. 학회지의 논문을 비롯하여, 각종 도서와 조각 문서, 대학교의 강의 자료와 과제물까지도 \TeX 을 사용해 만들어집니다. 분수나 첨자처럼 비선형적인 구조가 많은 수식을 명령어의 형태로 풀어냄으로써 선형적으로 작성할 수 있게 되었고, 이러한 입력 상의 이점에 더불어 미려한 결과물, 자유로운 확장성이 더해져 \TeX 은 강력한 문서 작성 도구로서의 자리를 차지할 수 있게 되었습니다.

이름과 로고

이 프로그램의 이름 \TeX 은 ‘기술’을 뜻하는 그리스어 접두사 ‘ $\text{\textit{tex}}$ ’¹에서 유래한 것으로, ‘텍(*tek*)’이라고 읽습니다. 이름의 세 글자 ‘T’, ‘E’, ‘X’는 로마자가 아니라 그리스 문자 타우, 엡실론, 카이이므로, ‘텍스’ 대신 카이의 발음을 살려 ‘테흐’라고 발음하는 것이 옳습니다. 다만 한국어에는 카이에 대응하는 발음이 없으므로 한국어 화자는 비슷한 발음인 ‘텍’이라고 관용적으로 발음합니다.

\TeX 시스템에서 \TeX 로고를 조판할 때에는 $\backslash\text{\TeX}$ 이라고 입력하면 됩니다. 만약 이 로고를 조판할 수 없는 환경(다른 프로그램을 사용할 때나 이메일을 작성할 때 등)에서는 ‘TeX’이라고 입력합니다.

1.1.2 \LaTeX 의 개발

커누스가 개발한 \TeX 은 저수준 명령어가 대부분이었고, 다양한 문서를 원활하게 작성하기 위해서는 많은 매크로를 사용자가 직접 작성해야 했습니다. \TeX 을 더 쉽게 사용하기 위해서는 각종 편의성 매크로를 정의해 둔 포맷 파일이 필요했으며, \TeX 공개 이후 여러 포맷 파일이 작성되었습니다. 그중 가장 널리 사용되는 것이 바로 레슬리 램포트 Leslie Lamport가 개발한 \LaTeX 입니다.

1984년에 \LaTeX 의 첫 버전이 개발되었으며, 이를 ‘ $\text{\LaTeX}2.09$ ’라고 부릅니다. 여기에 확장성과 글꼴 선택 기능을 강화한 두 번째 버전인 $\text{\LaTeX}2_{\epsilon}$ 가 1994년에 개발되었으며, 이 이래로 $\text{\LaTeX}2.09$ 는 더 이상 사용되지 않습니다. 우리가 ‘ \LaTeX ’이라고 부르는 것은 엄밀히는 $\text{\LaTeX}2_{\epsilon}$ 인 것입니다. 현재 \LaTeX 의 개발은 \LaTeX 프로젝트 팀 \LaTeX Project Team이 맡

¹ 영어 단어 ‘technique’, ‘technical’ 등의 접두사 ‘tech-’가 여기에서 유래한 것입니다.

고 있으며, 프로그래밍 기능을 강화한 **L^AT_EX3 프로그래밍 레이어** L^AT_EX3 Programming Layer를 개발하고 있습니다.

엄밀히 말하면 L^AT_EX은 하나의 독립된 프로그램이 아니고, T_EX과 함께 사용하기 위한 포맷 파일을 뜻합니다. 그러나 오늘날 많은 사람들은 L^AT_EX의 포맷을 바탕으로 T_EX을 사용하며, 소스 파일을 컴파일 할 때에도 L^AT_EX을 하나의 프로그램처럼 호출할 수 있습니다. 따라서, 특별한 상황이 아니라면 L^AT_EX을 하나의 프로그램이라고 생각해도 무방합니다.

명령어 디자인 철학

L^AT_EX의 매크로는 ‘편집자’와 ‘작가’를 분리하도록 디자인 되어 있습니다. 여기에서 편집자는 L^AT_EX이 맡고, 작가는 문서를 작성하는 사람, 즉 여러분입니다. 다시 말해, 문서의 레이아웃이나 디자인에 관한 설정은 L^AT_EX이 전담하고, 문서를 작성하는 사람은 조판된 문서의 모습에 신경쓰지 않고 문서의 내용에 집중하여 내용을 작성하게 되어 있습니다. 이때 저자와 편집자를 연결하는 것은 클래스 파일입니다. 클래스 파일은 원시적인 템플릿 파일이라고 생각할 수 있으며, 클래스 파일에서 제공하는 논리적 마크업을 사용해 작가가 ‘이 부분은 제목/표제/인용문/수식/그림의 캡션을 뜻하는 부분임’이라고 L^AT_EX에게 알려 주면, L^AT_EX은 클래스 파일에서 그 논리적 마크업의 정의를 읽고 올바른 서식을 적용해 줍니다.

문서의 클래스 파일이 원하는 논리적 마크업을 제공하지 않는 경우, 사용자는 다른 패키지를 사용하거나 직접 마크업을 만들 수 있습니다. 또, 문서의 클래스 파일에 정의된 서식이 마음에 들지 않는 경우 사용자가 이를 직접 바꿔 줄 수도 있습니다. 이러한 작업은 문서의 본문을 작성하는 곳과 분리된 곳에서 진행하므로, 여전히 편집자와 작가가 분리되어 있다고 생각할 수 있습니다. 이러한 설정 작업을 하는 곳을 ‘전처리부’, 문서의 본문을 작성하는 곳을 ‘document 환경’이라고 부르며, 이는 제3장에서 설명합니다.

이처럼, L^AT_EX에서 편집자와 작가를 엄격하게 분리하는 데에는 여러 이유가 있습니다. 작가가 편집자의 역할까지 겸하는 경우에는 다음의 문제가 생길 수 있습니다.

- 예뻐 보이는 문서가 항상 읽기 좋은 것은 아닙니다. 즉, 심미성을 중요시하더라도 가독성이 나빠질 수 있습니다. 따라서 문서를 작성할 때에는 심미성이 가독성을 해치지 않도록 디자인 측면에서 올바른 결정을 내려야 합니다.
- 그러나, 이러한 부분에서 올바른 결정을 내려야 한다는 점은 오히려 작가에게 부담이 됩니다. 문서를 작성할 때 제목에는 어떤 디자인을, 표제에는 어떤 디자인을 적용해야 할지 문서를 작성할 때마다 고민해야 한다면 작가는 원고의 내용을

작성하고 다듬는 것보다 상대적으로 덜 중요한 부분에 시간을 낭비하게 됩니다. 따라서, 편집자가 해야 할 작업을 LaTeX에게 맡기고, 대신 문서 작성자의 의도를 LaTeX이 알 수 있도록 명령어가 만들어진 것입니다. LaTeX이 사용하는 클래스 파일은 가독성이 보장되도록 서식이 설정되어 있습니다.

이름과 로고

LaTeX은 일반적으로 ‘레이텍(*lay-tek*)’이라고 읽습니다. LaTeX 2_ε와 LaTeX3는 각각 ‘레이 텍 투 이(LaTeX two E)’, ‘레이텍 쓰리(LaTeX three)’라고 읽습니다.

LaTeX 시스템에서 LaTeX과 LaTeX 2_ε 로고를 조판할 때에는 각각 \LaTeX, \LaTeXe라고 입력하면 됩니다. 이 로고를 조판할 수 없는 환경에서는 각각 ‘LaTeX’, ‘LaTeXe’라고 입력합니다.

1.1.3 장점과 단점

LaTeX이 강력한 수식을 작성할 수 있다는 것은 잘 알려져 있지만, 여기에 장점만 있는 것은 아닙니다. 아래아한글이나 Word 등 흔히 쓰이는 다른 워드프로세서와 비교했을 때 흔히 언급되는 몇 가지 단점을 꼽아 보자면…….

- 비직관적입니다. 입력 파일과 출력 파일이 구분되어 있고, 입력 파일을 컴파일 해야 출력 파일을 확인할 수 있습니다. 게다가, 입력 파일은 직관적인 형태가 아니라 코드의 형태로 작성해야 합니다.
- 무슨 기능이 있는지 알기 어렵습니다. 입력 파일을 작성하는 편집기에는 LaTeX의 기능과 관련하여 무언가 클릭해 볼 수 있는 메뉴 버튼이 거의 없으며, 아주 기초적인 기능만을 보조해 줍니다. 무슨 기능이 있는지 알아보려면 온라인 검색을 하거나 매뉴얼을 찾아 읽어야 합니다.

그러나, (일단 익숙해진다면) 많은 장점을 누릴 수 있습니다.

- 강력한 수식 작성 기능. 이에 대해서는 더 설명하지 않겠습니다. 제 III 편의 목차를 확인해 보시기 바랍니다.
- 미려한 타이포그래피 기능. 다른 워드프로세서에 비해 더 세부적인 조정을 할 수 있으며, 패키지를 사용하면 이 작업을 손쉽게 처리할 수도 있습니다.
- 높은 이식성. 문서를 작성할 때 필요한 파일만 모두 갖춰져 있다면 어떤 운영 체제에서든 문서를 편집할 수 있고, 동일한 컴파일 결과를 얻을 수 있습니다.

- 자유로운 확장성. \LaTeX 에는 함께 사용할 수 있는 클래스와 패키지가 수천 개 있으며, 이들은 편의성을 높여 주거나 부가 기능을 추가해 줍니다.

…… 이 책을 읽고 있는 여러분은 (아마도) \LaTeX 을 꼭 배워야 할 테니, 장점만 기억하고 넘어갑시다!

1.2 사용해 보려면

\LaTeX 은 크게 두 가지 방법으로 사용할 수 있습니다.

- 컴퓨터에 \TeX 배포판 \TeX distribution을 설치하여 문서를 작성할 수 있습니다. \TeX 배포판은 \TeX 시스템을 사용해 문서를 작성할 때 필요한 프로그램과 파일을 컴퓨터에 설치해 주는 프로그램입니다. 이를 컴퓨터에 설치하면 \LaTeX 의 모든 기능을 사용할 수 있지만, 설치 과정이 복잡합니다.
- 컴퓨터에 프로그램을 설치하지 않고, Overleaf 등의 온라인 플랫폼을 사용할 수 있습니다. 필요한 여러 프로그램이 통합되어 있고 기본적인 작업을 편리하게 할 수 있도록 컴파일 과정이 일부 자동화 되어 있으나, 명령줄 인터페이스를 사용하는 등 모든 기능을 사용하는 데에는 제약이 있습니다.

컴퓨터에 \TeX 배포판을 설치하여 사용하는 방법은 부록 A에서 읽을 수 있으며, 온라인 환경을 사용해 문서를 작성하는 방법은 제2장에서 읽을 수 있습니다.

1.2.1 필요한 프로그램

\LaTeX 을 사용해 문서를 작성하는 데에는 소스 파일 편집기, 컴파일 엔진, PDF 파일 뷰어 세 가지 프로그램이 필요합니다.

- 소스 파일 편집기 \LaTeX 소스 코드를 작성하는 데 필요한 프로그램입니다. 운영 체제의 기본 텍스트 파일 편집기를 사용할 수도 있고, \TeX 문서 작성에 편리한 기능을 갖춘 \TeX works나 \TeX Shop 등을 사용할 수도 있습니다.
- 컴파일 엔진 작성한 \LaTeX 문서를 컴파일 하는 데 사용하는 프로그램, 즉 \TeX 을 가리킵니다. 컴파일 엔진에는 여러 가지가 있으므로 원하는 것을 선택하여 사용할 수 있습니다. 기본적으로는 운영 체제의 명령줄 인터페이스를 통해 사용하지만, \TeX 전용 편집기에서 제공하는 컴파일 버튼을 통해 사용할 수도 있습니다.
- PDF 파일 뷰어 소스 파일을 컴파일 하여 얻은 PDF 문서를 읽거나 출력하기 위한 프로그램입니다.

\TeX 배포판을 설치하면 컴퓨터에 소스 파일 편집기와 컴파일 엔진이 설치됩니다. \TeX 배포판과 함께 설치되는 소스 파일 편집기에는 컴파일 엔진과 PDF 파일 뷰어 기능이 통합되어 있으며, Overleaf를 사용할 때에도 세 가지 프로그램이 하나의 화면에 통합되어 표시됩니다.

이외에도, 작성하는 문서에 따라 여러 보조 프로그램을 추가로 사용할 수도 있습니다. 이들도 위의 프로그램과 함께 \TeX 배포판을 통해 설치됩니다.

- **파일 변환 프로그램** 커누스가 개발한 \TeX 엔진은 소스 파일을 컴파일 한 후 PDF 문서를 생성하지 않고, DVI(device independent, 장치 독립) 파일을 생성합니다. 이 파일을 더 일반적으로 사용하는 PS 또는 PDF 파일로 변환할 때 사용하는 프로그램이 파일 변환기입니다. dvipdfm, dvipdfmx, dvi2ps, ps2pdf 등이 있습니다.
- **그래픽 생성 프로그램** \TeX 문서에 삽입할 그림도 명령어를 통해 그릴 수 있습니다. \TeX 배포판에서는 METAPOST와 Asymptote를 제공합니다. 자세한 내용은 13.1절에서 확인할 수 있습니다.
- **참고 문헌·색인 정렬 프로그램** 참고 문헌 목록이나 색인을 만들 때에는 각 항목을 적절한 순서로 정렬해 주어야 합니다. 이러한 작업을 해 주는 프로그램으로 Bib \TeX , biber, *MakeIndex*, xindy가 있습니다. 자세한 내용은 제16장에서 확인할 수 있습니다.
- **컴파일 자동화 프로그램** 복잡한 문서를 컴파일 할 때에는 \TeX 엔진 외에도 위의 보조 프로그램을 적절한 순서로 사용해야 합니다. latexmk는 소스 파일을 읽고 필요한 프로그램을 판단하여 컴파일 과정을 자동으로 처리해 주는 프로그램입니다. Overleaf와 몇 가지 \TeX 전용 편집기에서는 이 프로그램을 통해 컴파일 과정을 처리합니다.

이들은 모두 명령줄 인터페이스를 통해 사용합니다. 앞의 설명과 마찬가지로, \TeX 전용 편집기에서 이들을 편리하게 사용할 수 있는 컴파일 버튼을 제공하기도 합니다.

1.2.2 컴파일 엔진

커누스가 개발한 \TeX 에는 세 가지 문제가 있습니다. 첫째는 외국 문자를 자유롭게 처리할 수 없다는 점이고, 둘째는 PDF 문서를 바로 생성하지 않는다는 점입니다. 셋째는 프로그램이 오래전 개발된 만큼 컴퓨터의 자원을 효과적으로 사용할 수 없다는 점입니다. 이러한 문제를 해결하기 위해 \TeX 을 발전시킨 다양한 컴파일 엔진이 등장하고 사라졌으며, 오늘날에는 네 가지 엔진이 널리 사용됩니다.

- **PDF_{TeX}** 플레인 **TeX**에서 생성하는 DVI 파일을 외부 프로그램을 사용해 PDF 파일로 변환하는 과정을 거치지 않고, 컴파일 과정에서 PDF 파일을 바로 생성하도록 합니다.
- **Lua_{TeX}** 스크립트 언어인 Lua를 사용할 수 있는 엔진입니다. UTF-8 인코딩으로 저장된 **TeX** 파일을 사용하고, TTF 및 OTF 글꼴을 사용하며 PDF 파일을 생성합니다.
- **X₃TeX** 다국어 지원을 위해 만들어진 **TeX** 엔진입니다. UTF-8 인코딩을 사용한 **TeX** 파일을 사용하고, TTF 및 OTF 글꼴 파일을 사용하며 PDF 파일을 생성합니다. 이 엔진은 ‘지텍(*zee-tekh*)’이라고 읽으며, 로고를 조판할 수 없는 경우 ‘Xe_{TeX}’이라고 적습니다.
- **up_{TeX}** 일본에서는 두 번째 버전의 **TeX** 엔진을 바탕으로 ‘p_{TeX}(publishing **TeX**)’이라는 엔진을 개발하였습니다. 이 엔진을 발전시켜 유니코드를 지원하도록 만든 엔진이 up_{TeX}입니다. 일본 외에서는 사용되지 않습니다.

컴파일 엔진과 L^A_{TeX} 포맷

위에서 설명한 네 가지 엔진을 L^A_{TeX} 포맷과 함께 사용할 수 있습니다. 이때에는 다음의 여섯 가지 포맷으로 컴파일 할 수 있습니다. 포맷에 따라 컴파일 과정에서 사용되는 엔진이 자동으로 결정됩니다.

- **L^A_{TeX}** PDF_{TeX} 엔진을 사용해 소스 파일을 컴파일 하며, DVI 파일을 생성하는 호환 모드로 문서가 처리됩니다.
- **PDFL^A_{TeX}** PDF_{TeX} 엔진으로 소스 파일을 컴파일 하여 PDF 파일을 생성합니다.
- **LuaL^A_{TeX}** Lua_{TeX} 엔진을 사용해 소스 파일을 처리합니다.
- **X₃L^A_{TeX}** X₃TeX 엔진을 사용해 소스 파일을 처리합니다.
- **pL^A_{TeX}** up_{TeX} 엔진을 사용해 소스 파일을 처리하며, 문자 처리를 p_{TeX}의 방식으로 하는 호환 모드로 문서가 처리됩니다.
- **upL^A_{TeX}** up_{TeX} 엔진을 사용해 소스 파일을 처리합니다.

현재는 DVI 파일을 생성하는 L^A_{TeX} 포맷은 거의 사용되지 않으며, PDFL^A_{TeX}이 일반적으로 사용됩니다. 다양한 글꼴을 사용할 수 있는 LuaL^A_{TeX}과 X₃L^A_{TeX}도 자주 사용되며, 한국어 문서를 작성할 때에는 특히 이 두 포맷을 사용하기를 권장합니다. pL^A_{TeX}과 upL^A_{TeX}은 일본어 문서를 작성할 때에만 사용됩니다.

이 책에서 \TeX 엔진을 직접적으로 다룰 일은 없으므로, 위의 여섯 가지 포맷을 ‘엔진’이라고 부르겠습니다. 설명의 편의를 위해, \LaTeX 과 \PDFLaTeX 을 레거시 엔진, \LuaLaTeX 과 \XeLaTeX 을 유니코드 엔진, \pLaTeX 과 \upLaTeX 은 일본어 엔진이라고 묶어 부르겠습니다.

참고 | 이 책에서 일본어 엔진에 대해서는 특별히 다루지 않습니다. 일본어 엔진은 많은 상황에서 레거시 엔진과 비슷하게 동작하지만, 완벽하게 호환되는 것은 아닙니다.

1.3 관련 웹사이트

이번 절에서는 \TeX 과 관련된 웹사이트를 소개합니다. \TeX 배포판이나 여기에 포함된 파일, 또는 공식적으로 공개된 기술적 문서를 확인할 수 있습니다.

- **TUG**(\TeX Users Group) \TeX 을 배포하는 공식 홈페이지입니다. 이곳에서 \TeX 배포판을 내려받거나, \TeX 배포판과 관련된 기술적 문서를 읽을 수 있습니다.
URL: <https://tug.org/>
- **The \LaTeX Project** 현재 \LaTeX 의 개발을 진행하고 있는 \LaTeX 프로젝트 팀의 공식 홈페이지입니다. \LaTeX 의 업데이트 이력이나 기술적 문서를 확인할 수 있습니다.
URL: <https://www.latex-project.org/>
- **CTAN**(Comprehensive \TeX Archive Network) \TeX 과 관련된 각종 파일이 보관되어 있는 웹사이트입니다. 패키지나 클래스, 글꼴 파일을 직접 내려받을 수도 있고, 그에 관한 안내서를 읽을 수도 있습니다.
URL: <https://ctan.org/>
- **KTUG**(Korean \TeX Users Group, 한글 \TeX 사용자 그룹) \TeX 에서의 한국어 사용에 관한 내용을 다루는 웹사이트입니다. \TeX 으로 한국어 문서를 작성할 수 있도록 하는 *kotex* 패키지[56]를 이곳에서 개발하였습니다.
URL: <https://www.ktug.org/>

제 2 장

문서 작성 과정

\LaTeX 의 문법은 어떻게 되고 원하는 내용을 어떻게 작성하는지 알아보기에 앞서, 이번 장에서는 \LaTeX 에서 문서를 작성하는 과정을 알아볼 것입니다. 그림 2.1에서 확인할 수 있듯, 소스 파일을 작성한 뒤 이를 \TeX 으로 컴파일하면 보기 좋게 조판된 PDF 문서를 얻습니다.

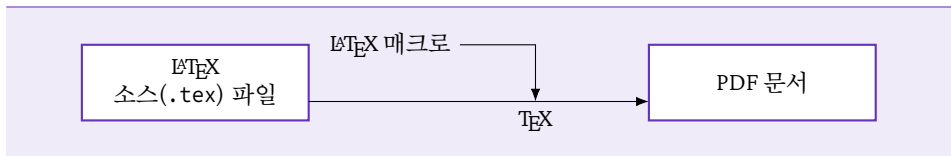


그림 2.1 | \LaTeX 문서 작성 과정

2.1 두 가지 예시

두 가지 예시 문서를 직접 작성해 보면서 \LaTeX 으로 문서를 작성하는 과정을 이해해 봅시다. 첫 번째 예시는 영어 문서, 두 번째 예시는 한국어 문서입니다.

문서를 작성하려면 먼저 소스 코드를 작성하기 위한 편집기를 준비해야 합니다. 편집기에는 여러 가지가 있지만, 여기에서는 복잡한 준비 과정이 필요 없는 온라인 환경인 Overleaf를 사용하겠습니다(회원 가입이 필요합니다). 웹 브라우저를 사용하여 <https://overleaf.com>에 접속한 후 로그인 하면 파일 목록이 나타납니다.

로그인 후 나타나는 파일 목록 화면에서는 기존에 생성했던 프로젝트를 열어 수정할 수 있습니다. ‘New Project’를 누른 후에는 ‘Upload Project’를 클릭해 가지고 있는 프로젝트 파일(.tex 파일, .aux 파일, .pdf 파일 등이 포함된 .zip 파일)을 올려 편집할 수도

있고, ‘Example Project’를 클릭해 여러 가이드가 포함된 파일로부터 새로운 문서를 만들 수도 있습니다. Overleaf에서 제공하는 템플릿을 사용할 수도 있습니다.

2.1.1 첫 번째 예시: 영어 문서 작성하기

첫째로 그림 2.2의 영어 문서를 작성해 봅시다. 먼저, 문서를 만들기 위해 새로운 파일을 만들고 편집기를 열어야 합니다.

1 | 파일 목록 화면의 좌상단에 있는 ‘New Project’ → ‘Blank Project’를 클릭합니다.

2 | 프로젝트 이름으로 ‘First Document’를 입력한 후 ‘Create’를 클릭합니다.

이 과정을 통해 편집기가 열리면 문서를 작성할 준비가 끝납니다.

편집기가 준비되었다면, 다음 과정을 따라서 편집기에 소스 파일을 작성합니다.

1 | 기존에 작성되어 있던 코드를 삭제합니다.

2 | 예시 문서 2.1의 코드를 입력합니다. 제6행 `\author` 옆의 ‘Woohyun Kang’은 자신의 이름으로 바꾸십시오. 이름은 로마자로 적어야 합니다.

3 | 입력한 코드가 자동으로 컴파일 되어 오른쪽 화면에 나타날 것입니다. 자동으로 컴파일이 되지 않으면 화면 위쪽의 Recompile 버튼을 클릭하거나 단축키 `command + return`(Windows에서는 `Ctrl + Enter`)을 누릅니다.

이로써 여러분은 \LaTeX 으로 첫 번째 문서를 작성하였습니다. 컴파일이 끝나면 그림 2.2와 동일한 문서를 오른쪽 화면에서 확인할 수 있을 것입니다.

생성된 PDF 파일은 Recompile 버튼의 오른쪽에 있는 ‘↓’ 모양의 버튼을 클릭해 내려받을 수 있습니다. 화면 왼쪽 위의 Menu 버튼을 클릭하면 프로젝트 파일을 내려받을 수 있으며, 컴파일 할 때 사용할 엔진을 선택하는 등 각종 옵션을 변경할 수 있습니다.

2.1.2 두 번째 예시: 한국어 문서 작성하기

앞선 예시에서는 영어 문서를 작성하였습니다. 이번에는 그림 2.3의 한국어 문서를 작성한 후, 앞과 비슷한 과정을 통해 컴파일 해 봅시다.

1 | 앞과 같은 방법으로 ‘New Project’ → ‘Blank Project’를 클릭한 뒤 ‘First Korean Document’라는 새로운 프로젝트를 만듭니다.

2 | 화면 좌상단의 Menu 버튼을 클릭한 뒤, ‘Settings’ 부분에서 ‘Compiler’를 ‘Xe-LaTeX’으로 설정합니다.

3 | 편집기에 예시 문서 2.2의 코드를 입력한 뒤 컴파일을 진행합니다.

First L^AT_EX Document

Woohyun Kang

January 7, 2024

Hello, L^AT_EX! You have just created your *first* L^AT_EX document! You can type out most of the characters you want here freely.

To start a new paragraph, insert empty line(s) between paragraphs in the source code. Each paragraph will be indented automatically.

You can also directly enter in math formulae like $a^2+b^2=c^2$ or $e^{i\pi}+1=0$. For important formulae, use display math mode as below:

$$i\hbar\frac{\partial}{\partial t}\Psi = H\Psi.$$

| 예시 문서 2.1 | First Document

```

1 \documentclass[a4paper, 12pt]{article}
2
3 \usepackage[T1]{fontenc}
4
5 \title{First \LaTeX\ Document}
6 \author{Woohyun Kang}
7 \date{\today}
8
9 \begin{document}
10 \maketitle
11
12 \textbf{Hello, \LaTeX !} You have just created your \emph{first} \LaTeX\
    document! You can type out most of the characters you want here freely.
13
14 To start a new paragraph, insert empty line(s) between paragraphs in the
    source code. Each paragraph will be indented automatically.
15
16 % In each line, any letters at the right side of the percent symbol will be
    considered as comments.
17
18 You can also directly enter in math formulae like  $(a^2 + b^2 = c^2)$  or
     $(e^{i\pi} + 1 = 0)$ , or
19  $\left[ i\hbar \frac{\partial}{\partial t} \right] \Psi = H \Psi.$  \]
20
21 \end{document}

```

컴파일이 완료되면 그림 2.3과 동일한 문서가 나타날 것입니다.

한국어 문서를 작성할 때에는 kotex 패키지가 필요합니다. 예시 문서 2.2의 제3행이 이 패키지를 불러오는 코드입니다. 또, 제4~6행은 한글 글꼴을 설정해 주는 코드입니다. kotex 패키지를 불러오고 글꼴 설정을 함으로써 문서의 한글을 올바른 글꼴로 조판하고, 낱자 등 각종 문구가 한국어로 바뀌어 문서가 만들어지게 됩니다.

2.2 자유롭게 문서 작성하기

이제, 문서를 자유롭게 작성해 봅시다. 앞서 제시했던 두 가지 예시 문서의 코드를 원하는 대로 수정하면서 자신만의 짧은 문서를 만들어 보시기 바랍니다. Overleaf에서 새로운 프로젝트를 만들 때 선택할 수 있는 ‘Example Project’도 좋은 출발점이 될 수 있습니다.

첫 번째 한국어 L^AT_EX 문서

강우현

2024년 1월 7일

Hello, L^AT_EX! 직접 첫 번째 한국어 L^AT_EX 문서를 만들어 보았습니다.
여기에 대부분의 문자들을 자유롭게 입력해 조판할 수 있습니다.

새로운 문단을 시작하려면, 소스 코드에서 각 문단 사이에 빈 줄(들)을 입력하세요. 각 문단은 자동으로 들여쓰기 처리됩니다.

$$a^2 + b^2 = c^2, e^{i\pi} + 1 = 0, \text{ 또는}$$

$$i\hbar \frac{\partial}{\partial t} \Psi = H\Psi$$

처럼 다양한 수식을 자유롭게 조판할 수도 있습니다.

그림 2.3 | 예시 문서 2.2의 컴파일 결과.

| 예시 문서 2.2 | First Korean Document

```

1 \documentclass[a4paper, 12pt]{article}
2
3 \usepackage[hangul]{kotex}
4 \setmainhangulfont[AutoFakeSlant]{나눔명조}
5 \setmainhangulfont[AutoFakeSlant]{나눔고딕}
6 \setmainhangulfont[AutoFakeSlant]{D2Coding}
7
8 \title{첫 번째 한국어 \LaTeX\ 문서}
9 \author{강우현}
10 \date{\today}
11
12 \begin{document}
13 \maketitle
14
15 \textbf{Hello, \LaTeX !} 직접 \dotemph{첫 번째} 한국어 \LaTeX\ 문서를 만들어 보았습니
    다. 여기에 대부분의 문자들을 자유롭게 입력해 조판할 수 있습니다.
16
17 새로운 문단을 시작하려면, 소스 코드에서 각 문단 사이에 빈 줄(들)을 입력하세요. 각 문단은 자동으로
    들여쓰기 처리됩니다.
18
19 % 각 줄에서, 퍼센트 기호의 오른쪽에 있는 문자들은 주석으로 처리됩니다.
20
21 \((a^2 + b^2 = c^2)\), \((e^{i\pi} + 1 = 0)\), 또는
22 \[ i\hbar \frac{\partial}{\partial t} \Psi = H \Psi \]
23 처럼 다양한 수식을 자유롭게 조판할 수도 있습니다.
24
25 \end{document}

```

2.2.1 알아두면 유용한 네 가지 명령어

\LaTeX 의 다양한 명령어를 알아보기 전에, 입력한 내용이 어떤 모습으로 문서에 조판되고 어떤 코드가 오류를 일으키는지 직접 체험해 보십시오. 이때 다음 네 가지 사항은 아주 기본적인 내용으로, 알아두는 것이 좋습니다.

- `\begin{document} ... \end{document}` 사이에 작성한 모든 내용이 PDF 파일에 조판됩니다. 조판하려는 내용을 `\begin{document}`의 앞이나 위에 입력하면 컴파일 과정에서 오류가 발생합니다.
- 다음 열 가지 문자는 특수한 역할로 예약되어 있으므로, 사용하면 안 됩니다.

\$ % & { } _ ^ ~ \

마지막 세 개를 제외한 문자는 앞에 역슬래시를 붙여서(예: \\$) 문서에 입력할 수 있습니다.

- 짧은 수식은 `\(... \)`로, 중요하거나 긴 수식은 `\[... \]`로 입력합니다.
- 문단을 나눌 때에는 `return` 키(Windows에서는 `Enter` 키)를 두 번 (이상) 눌러 빈 줄을 만들면 됩니다.

L^AT_EX의 문법은 다음 장부터 자세히 다룰 것이므로, 문서 작성을 체험해 보는 데에는 위의 네 가지 사항만 알아두어도 충분할 것입니다. 다음은 컴파일할 수 있는 짧은 예시 L^AT_EX 코드입니다.

```

1 \documentclass{article}
2 \usepackage[hangul]{kotex}
3 \setmainhangulfont{나눔명조}
4 \begin{document}
5 이차방정식 \((ax^2+bx+c=0)\) (단, \((a\neq
0\))의 근의 공식은 \([ x =
\frac{-b\pm\sqrt{b^2-4ac}}{2a} \)]이다.
6 \end{document}

```

이차방정식 $ax^2 + bx + c = 0$ (단, $a \neq 0$)
의 근의 공식은

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

이다.

2.2.2 컴파일 오류와 경고

소스 파일에 문제가 있다면, 컴파일 도중 오류(error)와 경고(warning) 메시지가 나타납니다. 오류는 컴파일이 불가능할 때 발생하고, 경고는 컴파일은 진행할 수 있지만 문서 작성자의 의도와 다른 결과물을 얻을 수 있거나 T_EX의 입장에서 최적의 결과물을 만들 수 없을 때 발생합니다.

앞서 작성했던 First Document의 제10행에 있는 `\maketitle`을 `\Maketitle`로 바꾸어 의도적으로 오류를 일으켜 봅시다. 컴파일을 진행하면 Recompile 버튼 오른쪽의 ‘Show logs and messages’ 버튼에 숫자 ‘1’이 표시될 것입니다. 버튼을 클릭하면 PDF 파일 대신 오류 메시지가 표시됩니다.

오류 메시지는 문제가 발생한 파일이 무엇인지, 문제가 몇 번째 줄에서 발생했는지, 문제의 원인은 무엇인지 알려줍니다. 여기에서는 `main.tex` 파일의 제10행에서 오류가 발생했으며, `\Maketitle`이 정의되지 않은 제어 문자열(control sequence)이기 때문에 오류가 발생했음을 알 수 있습니다. 이처럼 소스 파일에 문제가 있으면 컴파일 과정에서도 문제가 발생하며, 이때 발생했던 문제에 대한 정보는 오류 메시지를 통해 알 수 있습니다.

Overleaf에서 오류 메시지는 빨간색으로, 경고 메시지는 노란색으로 표시됩니다. \LaTeX 으로 문서를 작성하다 보면 여러 메시지를 보게 될 것입니다. 어떤 상황에서 어떤 메시지가 발생하는지, 또 어떻게 해결할 수 있는지는 부록 B.1.1에서 확인할 수 있습니다.

2.2.3 추가 정보

Overleaf에서는 다양한 기능을 제공하지만, 이 책에서는 이에 대해 다루지 않습니다. 이 환경에 대한 기술적 정보는 Overleaf의 안내 페이지를 참조하시기 바랍니다. 첫 화면이나 파일 목록 화면에서 ‘Help → Documentation’을 클릭하면 안내 페이지로 이동할 수 있습니다.

제 3 장

문법과 코드의 구조

이번 장에서는 \LaTeX 소스 파일의 기본 문법과 소스 코드의 구조를 알아볼 것입니다. 명령은 어떻게 내리는지, 소스 파일은 어떤 구조로 이루어져 있는지, 수식과 주석은 어떻게 입력하는지를 알고 있는 독자는 이 부분을 넘어가도 좋습니다.

3.1 명령어

\LaTeX 은 소스 코드에 입력된 문자 대부분을 그대로 PDF 문서에 조판하므로, 명령어를 사용해야 \LaTeX 에게 명령을 내릴 수 있습니다. 명령어는 ‘제어 문자열’과 ‘환경’으로 구분됩니다. 제어 문자열은 역슬래시 `backslash` ‘`\`’로 시작하는 명령어로, 입력된 위치에서 특정 명령을 실행합니다. 환경은 `\begin`과 `\end`가 쌍으로 이루어진 명령어로, 두 명령어 사이에 입력된 내용에 대해 특정 명령을 실행합니다.

3.1.1 제어 문자열

\LaTeX 에서 대부분의 명령은 제어 문자열 `control sequence`을 사용해 내립니다. 제어 문자열은 문서에 무언가를 삽입하거나, 인자로 지정된 내용에 특정 효과를 적용하거나, 현재 위치에서 문서의 특정 설정을 바꾸도록 선언하는 등의 역할을 합니다. 문서의 레이아웃을 결정하는 내부 변수로 사용되는 제어 문자열도 있습니다.

예를 들어, 다음은 모두 제어 문자열입니다.

```
\LaTeX \documentclass \usepackage \P \@ \\\
```

¹ 한국어 Windows 환경이나 일부 한글 글꼴에서는 원화 기호 ₩로 표시되기도 합니다. 일본어 Windows 환경이나 일부 일본어 글꼴에서는 엔화 기호 ¥로 표시되기도 합니다.

제어 문자열은 다음의 두 가지 형태로 구분됩니다.

- **제어 단어**control word 역슬래시로 시작해 로마자가 이어지는 것.
- **제어 기호**control symbol 역슬래시 뒤에 숫자 또는 기호(정확히는, 로마자 52자가 아닌 모든 문자) 한 개가 있는 것.

제어 단어와 제어 기호의 구분은 중요하지 않지만, 제어 문자열 뒤에 공백 문자가 올 때에는 주의해야 합니다. 제어 단어는 어디까지가 제어 문자열이고 어디부터가 문서 내용인지 구분하기 위해, 뒤에 오는 공백 문자를 무시합니다. 제어 단어 뒤에 공백을 의도적으로 삽입하려면 `_`를 입력하면 됩니다. 제어 기호는 `_`를 제외하고는 뒤에 오는 공백 문자를 무시하지 않습니다.

예를 들어, 'TeX'을 조판하는 제어 문자열 `\TeX`은 다음과 같이 사용할 수 있습니다.

```
\TeX_nicians\_use\_\\TeX\\_well.
```

TeXnicians use TeX well.

제어 문자열의 인자

많은 제어 문자열은 인자argument를 입력받습니다. 중괄호 `{...}` 안에 인자로 지정할 내용을 입력하면 됩니다. 예를 들어, 분수를 입력하는 제어 문자열 `\frac`은 분자와 분모를 각각 인자로 입력받습니다.

```
\[ \pi \approx \frac{355}{113} \]
```

$$\pi \approx \frac{355}{113}$$

인자로 지정할 내용이 문자 하나 또는 제어 문자열 하나일 때에는 중괄호를 생략해도 됩니다. 또, 제어 문자열과 첫 번째 인자 사이나 각 인자 사이에 입력된 공백 문자는 무시됩니다. 즉,

`\frac a b` `\frac ab` `\frac{a}b` `\frac a{b}` `\frac{a} {b}`

모두 `\frac{a}{b}`와 같은 결과를 만듭니다.

인자 중에는 필수로 입력할 필요가 없는 **선택 인자**optional argument도 있습니다. 이는 대괄호 `[...]`를 사용해 지정합니다. 선택 인자를 특별히 지정하지 않으려면 대괄호를 포함해 해당 인자를 생략하면 됩니다. 예를 들어, 근호를 입력하는 제어 문자열 `\sqrt`는 선택 인자인 근지수와 필수 인자인 피제곱근수를 입력받습니다.

```
\[ 2 = \sqrt{4} = \sqrt[3]{8} \]
```

$$2 = \sqrt{4} = \sqrt[3]{8}$$

선택 인자를 지정하는 경우에도 대괄호 앞뒤의 공백 문자는 무시됩니다. 따라서

`\sqrt [a] b` `\sqrt[a] b` `\sqrt [a]{b}` `\sqrt[a]{b}`

모두 `\sqrt[a]{b}`와 같은 결과를 만듭니다. 다만 대괄호는 생략할 수 없습니다.

마지막 인자가 선택 인자인 경우

제어 문자열(과 환경) 중에는 마지막 인자가 선택 인자인 것도 있습니다. 이러한 제어 문자열을 입력한 뒤 대괄호로 시작하는 내용을 입력하면, 대괄호가 선택 인자를 지정하는 것으로 해석되어 의도하지 않은 결과를 얻게 됩니다. 이때에는 제어 문자열 뒤에 `\relax`나 `{}` 등을 붙여 잘못된 해석을 피할 수 있습니다.

예를 들어, 현재 위치에서 줄을 강제로 바꾸도록 하는 `\`는 선택 인자 하나만을 입력받기 때문에, 특별한 조치 없이 대괄호로 시작하는 내용을 입력하면 컴파일 과정에서 오류가 발생합니다. 따라서 다음과 같이 코드를 작성해야 의도한 결과를 얻습니다.

1 `\(x\)`의 값을 구하여라. `\` `\relax`
2 [힌트: 앞 연습문제를 사용하여.]

x 의 값을 구하여라.
[힌트: 앞 연습문제를 사용하여.]

제어 문자열의 유효 범위

제어 문자열 중에는 선언`declaration`을 하는 것도 있습니다. 글꼴을 바꾸는 `\bfseries`나 `\itshape`, 문단을 가운데에 맞춰 정렬시키는 `\centering`, 새로운 명령어를 정의하는 `\newcommand` 등이 선언하는 제어 문자열의 대표적인 예시입니다.

선언형 제어 문자열의 유효 범위는 그 제어 문자열이 입력된 곳부터 제어 문자열이 속한 영역`group`이 끝나는 곳까지입니다. 일반적으로 영역은 중괄호를 사용해 만들 수 있습니다. 예를 들어, 글꼴을 볼드체로 조판하도록 선언하는 `\bfseries`는 중괄호의 유무에 따라 다음의 결과를 만듭니다.

1 글꼴을 `{\bfseries 볼드체로}` 조판하기 `\`
2 글꼴을 `\bfseries 볼드체로` 조판하기

글꼴을 **볼드체로** 조판하기
글꼴을 **볼드체로** 조판하기

환경의 내부도 하나의 영역으로 취급됩니다. 환경 안에 입력된 글자를 크게 조판하도록 하는 `large` 환경을 시작할 때 `\bfseries`를 입력하면 환경 안의 글자가 큰 볼드체로 조판됩니다. 환경이 끝나면 다시 본래 크기와 본래 굵기로 돌아옵니다.

1 글꼴을 `\begin{Large}\bfseries` 큰 볼드체로
`\end{Large}` 조판하기

글꼴을 **큰 볼드체로** 조판하기

전처리부 특히, 선언형 제어 문자열이 전처리부나 document 환경이 시작된 직후에 사용되면
 → 23쪽 문서 전체에 영향을 미칩니다. 필요에 따라 영역을 지정하여 제어 문자열의 유효 범위를 적절하게 지정하십시오.

3.1.2 환경

복잡한 명령이나 넓은 부분에 영향을 미치는 명령을 내릴 때에는 환경environment을 사용합니다. 이름이 $\langle env \rangle$ 인 환경은 $\begin{\langle env \rangle}$ 로 시작해 $\end{\langle env \rangle}$ 로 끝납니다. 환경은 그 안에 입력된 내용에 특정 효과를 적용하거나(제어 문자열보다 더 많은 내용에 효과를 적용할 때 주로 사용합니다), 그 내용을 특수한 형태로 조판하도록 할 때 사용합니다.

예를 들어, 목록을 작성하는 itemize 환경은 아래와 같이 사용합니다.

```
1 \begin{itemize}
2   \item 첫 번째 항목
3   \item 두 번째 항목
4 \end{itemize}
```

- 첫 번째 항목
- 두 번째 항목

환경 중에도 인자를 입력받는 것이 있습니다. 이때에는 각 인자를 $\begin{\langle env \rangle}$ 뒤에 중괄호나 대괄호를 사용해 지정합니다. 예를 들어, 표를 작성하는 tabular 환경은 표의 열 개수와 각 열의 정렬 방법을 지정하는 인자를 입력받습니다.

```
1 \begin{tabular}{c|cc}
2   \hline
3   & 중간고사 & 기말고사 \\
4   해석학 I & 73 & 84 \\
5   해석학 II & 76 & 91 \\
6 \end{tabular}
```

	중간고사	기말고사
해석학 I	73	84
해석학 II	76	91

3.1.3 주석

문서를 작성하다 보면 주석comment을 사용할 일이 생깁니다. 주석을 사용하면 특정 코드의 목적이 무엇인지 설명을 적어 둘 수도 있고, 컴파일 오류의 원인을 분석하기 위해 코드 일부분을 임시로 작동하지 않도록 할 수도 있습니다. \LaTeX 에서 주석은 퍼센트 기호를 사용해 입력합니다. 퍼센트 기호부터 그 줄 끝까지의 내용이 주석으로 처리되어 컴파일 과정에서 무시됩니다.

1 여기에 입력한 문자열은 % 무시됩니다.
2 무시되지 않습니다.

여기에 입력한 문자열은 무시되지 않습니다.

3.2 소스 파일의 구조

소스 파일은 document 환경을 기준으로 세 부분으로 나눌 수 있습니다. 특히, 각종 설정을 담당하는 첫 번째 부분을 전처리부(preamble)라고 부릅니다.

- 1 | document 환경 앞: 문서의 레이아웃이나 디자인과 관련된 부분을 설정하고, 각종 매크로를 정의합니다.
- 2 | document 환경 안: 문서의 내용을 작성합니다. 전처리부에서 정의했던 매크로가 이 환경에서 본격적으로 사용됩니다.
- 3 | document 환경 뒤: 여기에 입력된 모든 내용은 컴파일 시에 무시됩니다.

3.2.1 전처리부

전처리부에서는 각종 사전 설정을 해 줍니다. 먼저 문서를 작성할 때의 토대로 사용할 클래스 파일을 불러오고, 문서를 작성할 때 필요한 패키지를 불러옵니다. 필요하다면 자신이 자주 쓰는 명령어를 직접 정의해 둘 수도 있습니다.

문서의 클래스 지정하기

클래스(class) 파일은 문서의 종류를 지정함으로써, 그에 맞는 명령어를 정의하고 기본적인 레이아웃과 서식을 설정해 주는 파일입니다. 모든 \LaTeX 문서는 전처리부의 첫머리에 사용할 클래스가 지정되어 있어야 합니다.

클래스 중에는 일반적인 문서를 작성할 때 사용되는 것도 있고, 특수한 목적으로 쓰이는 것도 있습니다. 문서를 작성할 때 자주 사용되는 클래스를 표 3.1에 정리하였습니다. 이중 처음 세 가지 클래스는 \LaTeX 개발 팀에서 제공하는 표준 클래스로, 다른 문서 작성 클래스는 이 클래스의 기능을 모두 지원합니다. 이 책에서는 표준 클래스만을 다루겠습니다. 이외에 \LaTeX 에서 사용할 수 있는 다른 문서 클래스는 [12]에서 확인할 수 있습니다.

문서의 클래스를 지정할 때에는 `\documentclass` 제어 문자열을 사용합니다. 제어 문자열의 $\langle class \rangle$ 인자에 사용하려는 클래스의 이름을 입력하면 됩니다.

| 표 3.1 | 자주 사용되는 문서 클래스

클래스 이름	설명
article	일반적인 문서를 작성할 때 사용하는 클래스입니다.
report	보고서를 작성할 때 사용하는 클래스입니다. article 클래스에 비해 더 큰 규모의 문서를 작성할 때 사용합니다.
book	책을 작성할 때 사용하는 클래스입니다. 양면 편집 레이아웃이 적용됩니다.
memoir	다양한 문서를 작성할 때 사용할 수 있는 클래스로, 전문적인 편집을 할 때 문서의 레이아웃을 쉽게 바꿀 수 있는 명령어를 제공합니다.
oblvioir	memoir 클래스를 바탕으로 하여, 한국어 문서를 위한 다양한 기능이 추가되어 있습니다.
beamer	슬라이드 쇼를 작성하는 데 사용됩니다. 비슷한 목적으로 쓰이는 \LaTeX 의 표준 slides 클래스를 대체합니다.

```
\documentclass[<options>]{<class>}
```

<options>: 문서의 클래스에 지정할 옵션

<class>: 사용할 문서 클래스

클래스는 몇 가지 옵션을 제공하며, 이를 통해 레이아웃에 약간의 변화를 줄 수 있습니다. 예를 들어, 앞서 설명한 세 가지 표준 클래스에서는 종이 크기, 기본 글꼴 크기, 2단 편집 여부 등을 옵션으로 제공합니다. 세 클래스에서 제공하는 주요 옵션을 표 3.2에 정리하였습니다.

클래스를 불러올 때 지정하려는 옵션은 `\documentclass`의 *<options>* 인자로 입력하면 됩니다. 여러 옵션을 동시에 지정할 수도 있으며, 이때에는 각 옵션을 쉼표 ‘,’로 구분해 주면 됩니다.

예를 들어, 다음 코드를 \TeX 파일의 첫머리에 입력하여 종이 크기는 A4 용지이고, 본문 글자 크기는 12pt인 article 클래스 문서를 만들 수 있습니다.

```
\documentclass[a4paper, 12pt]{article}
```

패키지 불러오기

문서의 클래스를 지정했다면 패키지(package)를 불러올 차례입니다. 패키지는 추가 기능과 명령어를 제공하는 역할을 합니다. 자주 사용되는 몇 가지 패키지를 표 3.3에 정리하였습니다. [13]에서는 \LaTeX 에서 사용할 수 있는 모든 패키지를 확인할 수 있습니다.

패키지는 `\usepackage` 제어 문자열을 사용해 불러오며, 클래스와 마찬가지로 패키

표 3.2 | 표준 문서 클래스의 주요 옵션

구분	옵션	설명
종이 크기	a4paper	종이 크기를 A4(210 mm × 297 mm)로 설정합니다.
	b5paper	종이 크기를 B5(176 mm × 250 mm)로 설정합니다. ¹
	letterpaper	종이 크기를 레터(8.5 in × 11 in)로 설정합니다. 이 종이 크기가 기본값으로 사용됩니다.
	landscape	종이를 가로로 긴 방향으로 사용하여 문서를 편집합니다.
글자 크기	10pt	본문의 기본 글자 크기를 10 pt로 설정합니다. 이 글자 크기가 기본값으로 사용됩니다.
	11pt	본문의 기본 글자 크기를 11 pt로 설정합니다.
	12pt	본문의 기본 글자 크기를 12 pt로 설정합니다.
다단 편집	onecolumn	본문을 한 단으로 조판합니다. 이 옵션이 기본값입니다.
	twocolumn	본문을 2단으로 조판합니다.
수식 조판	leqno	수식에 번호를 붙인 경우, 이 번호가 왼쪽 끝에 표시되도록 합니다. 이 옵션을 지정하지 않으면 오른쪽 끝에 표시됩니다.
	fleqn	본문과 분리된 글줄에 조판하는 별행수식을 글줄의 가운데가 아니라 왼쪽에 정렬시켜 조판합니다.

¹ 여기의 B5 크기는 국제 표준(ISO) 규격으로, 우리나라와 일본에서 사용하는 일본 공업 표준(JIS) 규격인 182 mm × 257 mm와는 약간의 차이가 있습니다.

지를 불러올 때 그 패키지가 제공하는 옵션을 지정할 있습니다.

```
\usepackage[options]{packages}
```

options: 패키지를 불러올 때 지정할 옵션

packages: 불러올 패키지의 이름

예를 들어, 다음 코드는 `graphicx` 패키지와 `tikz` 패키지를 불러온 후 `kotex` 패키지에 `hangul` 옵션을 지정하여 불러옵니다. 이처럼 `\usepackage`를 한 번 사용하여 여러 패키지를 한꺼번에 불러올 수도 있습니다.

```
1 \usepackage{graphicx, tikz}
2 \usepackage[hangul]{kotex}
```

패키지를 불러올 때 사용할 수 있는 옵션, 정의되는 명령어, 주의 사항 등의 정보는 패키지 안내서 `package documentation`를 통해 확인할 수 있습니다. CTAN에서 패키지의

| 표 3.3 | 자주 사용되는 패키지

패키지 이름	설명
amsmath	미국수학회(AMS)에서 작성한 패키지로, 수식을 작성할 때 유용하게 쓰이는 제어 문자열과 환경을 정의합니다. 아래의 두 패키지도 미국수학회에서 작성한 것입니다.
amssymb	수식에서 사용되는 다양한 글꼴과 기호를 제공합니다.
amsthm	정의나 정리, 증명 등 수학적 문단을 편리하게 작성할 수 있는 명령어를 제공합니다.
geometry	문서의 페이지 크기와 여백 설정을 자유롭게 변경할 수 있게 해 줍니다.
fancyhdr	문서의 머리글과 바닥글의 내용을 자유롭게 변경할 수 있게 해 줍니다.
kotex	한국텍학회(KTS)에서 작성한 패키지로, 한국어 문서를 작성하기 위한 다양한 설정을 갖추어 줍니다.
array	L ^A T _E X에서 표를 작성하는 기능을 확장해 줍니다.
hyperref	문서에 하이퍼링크와 책갈피를 삽입하고, PDF 문서의 각종 속성을 지정하는 기능을 제공합니다.

이름을 검색하거나 터미널에서 `texdoc <manual name>`을 입력해 읽을 수 있습니다.² 많은 경우에 *<manual name>*은 패키지의 이름과 동일합니다. 예를 들어, `mathtools` 패키지의 안내서는 터미널에 `texdoc mathtools`를 입력해 읽을 수 있습니다. 만약 이 명령어를 사용했을 때 다른 패키지의 매뉴얼이 나오거나 현재 패키지의 매뉴얼을 찾을 수 없다는 메시지가 표시되면 CTAN에 패키지의 이름을 검색하시기 바랍니다.

매크로 정의하기

매크로
→ 259쪽

문서를 작성할 때 유용한 매크로를 직접 정의해 둘 수도 있습니다. 긴 코드를 짧게 줄여 주는 역할을 할 수도 있고, 자신이 작성하는 문서에 맞는 논리적 마크업을 정의해 줄 수도 있습니다. 전처리부에서 작성해 둔 매크로를 사용해 문서를 작성하면 본문의 코드를 알아보기도 쉬워지고, 문서의 서식을 적용하고 관리하는 것도 간단해집니다.

예를 들어, 이 책에서 패키지 이름을 입력할 때 사용하는 `\Lpkg` 제어 문자열은 전처리부에 다음을 입력하여 정의합니다.

```
\newcommand*{\Lpkg}[1]{\textsf{#1}}
```

이 제어 문자열은 패키지의 이름을 산세리프체로 조판하도록 합니다. 이렇게 정의한 `\Lpkg` 제어 문자열은 document 환경 안에서 다음과 같이 사용할 수 있습니다.

² `texdoc` 명령어는 컴퓨터에 T_EX 배포판이 설치된 경우에 사용할 수 있습니다.

1 `\Lpkg{kotex}` 패키지는 한국어 문서를 작성하는 데 필수로 사용된다.

`kotex` 패키지는 한국어 문서를 작성하는 데 필수로 사용된다.

매크로를 정의하는 방법은 여기에서 설명하지 않겠습니다. 이 내용이 궁금하신 분은 18.1절의 자세한 설명을 참조하시기 바랍니다.

한국어 문서 작성

영어가 아닌 한국어 문서를 작성하려면 적절한 언어 설정을 갖추어 주어야 합니다. 일반적인 한국어 문서를 작성하려면 예시 문서 2.2에서 했던 것처럼 전처리부에

```
1 \usepackage[hangul]{kotex}
2 \setmainhangulfont[AutoFakeSlant]{나눔명조}
3 \setsanshangulfont[AutoFakeSlant]{나눔고딕}
4 \setmonohangulfont[AutoFakeSlant]{D2coding}
```

을 입력하면 됩니다. 2~4행은 글꼴 설정과 관련된 부분이므로, 이 기능을 사용할 수 없는 레거시 엔진으로 문서를 컴파일 하고자 한다면 이 부분을 전처리부에 입력하지 않아도 됩니다.

문서를 작성할 때 사용할 수 있는 다양한 명령어는 제II편에서 설명합니다. 이 부분에서는 다양한 문자와 기호를 입력하고, 문서의 형식을 갖추며, 여러 서식을 지정하는 방법을 설명합니다. 이를 참조하여 다채로운 문서를 작성해 나가시기 바랍니다.

3.2.2 document 환경

전처리부에서 필요한 설정을 모두 마쳤다면, 비로소 `document` 환경에 문서를 작성할 수 있게 됩니다. 이 환경 안에 문서의 내용을 자유롭게 입력하면 되며, 정의되어 있는 제어 문자열과 환경도 사용할 수 있습니다.

띄어쓰기와 공백 문자

이 책에서는 `[space]` 키를 눌렀을 때 입력되는 문자를 공백 문자,³ `[return]` 또는 `[enter]` 키를 눌렀을 때 입력되는 문자를 개행 문자라고 부르겠습니다. 앞 절에서 몇 번 사용했듯, 공백 문자를 소스 코드에서 명시적으로 나타내야 할 때에는 기호 ‘`␣`’로 나타내겠습니다.

³ `[tab]` 키를 눌렀을 때 입력되는 탭 문자는 공백 문자와 동일하게 처리됩니다.

L^AT_EX은 공백 문자나 한 개의 개행 문자를 단어 사이의 띄어쓰기로 조판합니다. 공백 문자는 여러 개가 입력되더라도 한 개가 입력된 것처럼 처리되므로, 공백 문자를 실수로 여러 개 입력하여 단어 사이의 간격이 불균일하게 조판되는 일을 피할 수 있습니다.

- 1 공백 문자를 여러 번 입력하거나
- 2 줄을 한 번 바꾸어도
- 3 모두 띄어쓰기 한 번으로 처리됩니다.

공백 문자를 여러 번 입력하거나 줄을 한 번 바꾸어도 모두 띄어쓰기 한 번으로 처리됩니다.

필요한 경우에는 앞서 언급했던 `_` 또는 `\space` 제어 문자열을 입력해 단어 사이를 띄어 쓸 수도 있습니다. 이들은 공백 문자와는 달리, 여러 번 입력하면 입력한 만큼의 공백이 만들어집니다.

주석
→ 22쪽 한편, 퍼센트 기호 ‘%’는 해당 줄의 개행 문자까지 주석으로 처리합니다. 따라서, 긴 단어를 여러 줄에 걸쳐 적되 줄을 바꾼 곳에 띄어쓰기를 넣지 않고 싶을 때에는 이 문자를 요긴하게 사용할 수 있습니다. 코드 상에서 줄을 바꿀 때 원치 않는 공백이 문서에 만들어지는 일도 막을 수 있습니다.

- 1 아주 긴 단어를 주석기능을 사용하여%
- 2 띄어쓰기 없이 여러 줄에 걸쳐%
- 3 입력할 수도 있습니다.

아주 긴 단어를 주석기능을 사용하여 띄어쓰기 없이 여러 줄에 걸쳐 입력할 수도 있습니다.

또, 코드에서 각 줄의 첫머리에 입력된 공백 문자는 모두 무시됩니다. 따라서, 코드를 보기 좋도록 적절히 들여쓰기 처리를 해 줄 수도 있습니다. 환경을 사용할 때 환경 안의 코드를 들여쓰는 경우가 많습니다.

틸데 악센트 ‘~’도 띄어쓰기를 조판하지만, 이때에는 앞뒤의 단어 사이에서 줄이 바뀌지 않도록 합니다. 다음과 같은 경우에 유용합니다.

표~1.1 부록~C 변수~\ (x\) Mr.~John Chapter~5

문단 나눔과 개행 문자

개행 문자를 두 번 이상 입력하여 소스 코드에 빈 줄을 만들면 문단이 나뉩니다. 공백 문자와 마찬가지로, 개행 문자를 세 번 이상 입력하더라도 문단 사이에 추가 공백이 만들어지지는 않습니다. 문단 나눔은 2개 이상의 개행 문자 대신 `\par`로 대신할 수도 있습니다. 문단이 시작되면 T_EX은 첫째 줄을 1.5글자 정도 들여쓰므로써 문단이 시작됨을 표시해 줍니다.

1 첫 번째 문단입니다. 문단을 나누려면
 2 리턴 키를 두 번 눌러 주면 됩니다.
 3
 4 두 번째 문단입니다. 리턴 키를 두 번 누르는
 5 대신 `\verb|\par|`를 사용해도 됩니다.`\par`
 6 세 번째 문단입니다.

첫 번째 문단입니다. 문단을 나누려면
 리턴 키를 두 번 눌러 주면 됩니다.

두 번째 문단입니다. 리턴 키를 두 번
 누르는 대신 `\par`를 사용해도 됩니다.

세 번째 문단입니다.

수식 작성

수식 작성
 → 95쪽

수식은 수식 모드`math mode`에서 작성합니다. 문단 속에 섞여 있는 행중수식`inline math`은 `\(와 \)` 사이에 코드를 입력하여 조판하고, 중요하거나 긴 수식은 `\[와 \]`의 사이에 코드를 입력함으로써 별도의 글줄에 표시되는 별행수식`displayed math`으로 조판할 수 있습니다. `\(나 \[`를 입력하면 수식 모드가 아닌 텍스트 모드`text mode`에서 수식 모드로 전환되며, `\TeX`이 수식 코드를 이해할 수 있게 됩니다. 수식 모드에서는 공백 문자나 개행 문자가 무시되며, 개행 문자를 2개 이상 입력하여 문단을 나눈 경우에는 오류가 발생합니다. 반대로, `\)나 \]`를 입력하면 수식 모드에서 텍스트 모드로 전환됩니다.

1 자연로그의 밑 `\(e\)`, 허수 단위 `\(i\)`, 원주율 `\(\pi\)`, 곱셈의 항등원 `\(1\)`과 덧셈의 항등원 `\(0\)` 사이의 관계는 오일러 등식
 2 `\[e^{i\pi} + 1 = 0 \]`
 3 이 설명한다.

자연로그의 밑 e , 허수 단위 i , 원주율 π , 곱셈의 항등원 1과 덧셈의 항등원 0 사이의 관계는 오일러 등식

$$e^{i\pi} + 1 = 0$$

이 설명한다.

위 예시에서 행중수식은 처음 두 줄에 있는 ‘ $e, i, \pi, 1, 0$ ’이며, 별행수식은 셋째 줄에 있는 ‘ $e^{i\pi} + 1 = 0$ ’입니다.

나중에 언급하고자 하는 별행수식에는 번호를 붙여 둘 수도 있습니다. 이러한 수식은 `equation` 환경 안에 작성하면 됩니다. 이 환경을 사용하여 수식을 작성하면 수식의 오른쪽 끝(설정에 따라 왼쪽 끝에 표시될 수도 있음)에 번호가 출력됩니다.

1 이차방정식 `\(ax^2+bx+c=0\)` (단, `\(a \neq 0\)`)의 근은
 2 `\begin{equation} \label{eq:sqroot}`
 3 `x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}`
 4 `\end{equation}`이다.

이차방정식 $ax^2 + bx + c = 0$ (단, $a \neq 0$)의 근은

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (1)$$

이다.

표 3.4 | TeX의 특수 문자

특수 문자	입력 방법	역할
역슬래시 \	<code>\textbackslash</code> , <code>\backslash</code>	제어 문자열을 시작할 때 사용합니다.
여는 중괄호 {	<code>\{</code>	영역을 시작합니다.
닫는 중괄호 }	<code>\}</code>	영역을 끝냅니다.
달러 기호 \$	<code>\\$</code>	수식 모드와 텍스트 모드를 전환합니다.
앰퍼샌드 &	<code>\&</code>	내용을 표 형태로 배치할 때 열을 구분하는 역할을 합니다.
해시 #	<code>\#</code>	명령어를 정의할 때 매개 변수를 나타내는 데 쓰입니다.
서컴플렉스 악센트 ^	<code>\textasciicircum</code> , <code>\^{}</code> , <code>\hat{}</code>	수식 모드에서 위 첨자를 작성하는 데 쓰입니다. 텍스트 모드에서 사용하면 오류가 발생합니다.
언더스코어 _	<code>_</code>	수식 모드에서 아래 첨자를 작성하는 데 쓰입니다. 텍스트 모드에서 사용하면 오류가 발생합니다.
틸데 악센트 ~	<code>\textasciitilde</code> , <code>\~{}</code> , <code>\tilde{}</code>	줄 바꿈을 허용하지 않는 공백을 넣습니다.
퍼센트 기호 %	<code>\%</code>	주석을 작성합니다.

참조 기능

상호 참조
→ 207쪽

LaTeX은 수식 등의 번호를 자동으로 계산하고 참조하는 기능을 제공합니다. 번호 붙은 대상은 컴파일 전까지 번호를 알 수 없고, 번호를 알고 있더라도 앞뒤 내용이 바뀌면 그 번호가 달라질 수 있습니다. 따라서, 번호 붙은 대상을 참조할 때에는 번호를 직접 입력하는 대신, 대상에 가상의 이름을 붙여 두고 이 이름을 사용해 참조해야 합니다.

앞서 수식을 작성한 예시에서는 `\label` 제어 문자열을 사용해 수식에 ‘eq:sqrt’라는 가상의 이름을 붙여 두었습니다. 이 수식을 참조할 때에는 `\ref`를 사용합니다. 올바른 수식의 번호가 다음과 같이 `\ref` 제어 문자열이 있던 곳에 출력됩니다.

1. 앞의 식 (`\ref{eq:sqrt}`)를 사용하면 답을 구할 수 있다.

앞의 식 (1)을 사용하면 답을 구할 수 있다.

특수 문자

제2장에서 설명했듯, 다음 10가지 문자

`\` `{` `}` `$` `&` `#` `^` `_` `~` `%`

| 예시 문서 3.1 | Minimal Template for English Document

```

1 \documentclass[a4paper]{article} % 문서의 클래스를 지정합니다
2
3 \usepackage[T1]{fontenc} % 확장 로마자를 올바르게 조판합니다
4 % 패키지를 불러오고 문서의 설정을 조정합니다
5
6 \title{Title} % 문서의 제목을 중괄호 안에 입력합니다
7 \author{Author} % 저자의 이름을 중괄호 안에 입력합니다
8 \date{\today} % 문서가 작성된 날짜를 중괄호 안에 입력합니다
9
10 \begin{document}
11 \maketitle % 문서의 표지를 생성합니다
12
13 % 문서의 내용을 입력합니다
14
15 \end{document}

```

는 특수하게 처리되므로 문서에 바로 입력할 수 없습니다. 각각의 역할과 문자를 문서에 입력하는 명령어를 표 3.4에 정리하였습니다. 여러 가지 제어 문자열이 제시된 경우, 마지막 것은 수식 모드에서, 나머지는 텍스트 모드에서 사용할 수 있습니다.

과제 \#4는 전체 성적의 40\%를 차지하며, 우수 학생에게는 상금 \\$50가 주어집니다.

과제 #4는 전체 성적의 40%를 차지하며, 우수 학생에게는 상금 \$50가 주어집니다.

3.3 기본적인 템플릿

이상의 내용을 종합하여, 문서를 작성할 때 사용할 수 있는 기본적인 템플릿을 제공합니다. 모든 L^AT_EX 문서는 이 템플릿을 바탕으로 하여 작성할 수 있습니다.

영어 문서 템플릿

영어 문서를 작성할 때에는 예시 문서 3.1을 사용할 수 있습니다.

제1행에서는 클래스와 그 옵션을 지정합니다. 여기에서는 예시로 자주 사용되는 a4paper 옵션을 지정한 article 클래스를 사용하였습니다. 다음으로, 제3~4행에서는 패키지를 불러옵니다. 유니코드 엔진으로 문서를 컴파일할 때에는 제3행을 입력하지 마십시오. 이후 필요한 다른 패키지를 불러오고, 각종 매크로를 정의해 주면 됩니다.

전처리부에 모든 내용을 입력하였다면, 제6~8행에 문서의 제목·저자·날짜를 입력한 후, \maketitle 제어 문자열 아래에 문서의 내용을 작성하면 됩니다.

| 예시 문서 3.2 | Minimal Template for Korean Document

```

1 \documentclass[a4paper]{article} % 문서의 클래스를 지정합니다
2
3 % 패키지를 불러옵니다
4
5 \usepackage[hangul]{kotex} % 한국어 문서를 작성하도록 설정합니다.
6 \setmainhangulfont[AutoFakeSlant]{나눔명조}
7 \setsanshangulfont[AutoFakeSlant]{나눔고딕}
8 \setmonohangulfont[AutoFakeSlant]{D2Coding}
9
10 % 문서의 설정을 조정합니다
11
12 \title{제목} % 문서의 제목을 중괄호 안에 입력합니다
13 \author{저자} % 저자의 이름을 중괄호 안에 입력합니다
14 \date{\today} % 문서가 작성된 날짜를 중괄호 안에 입력합니다
15
16 \begin{document}
17 \maketitle % 문서의 표지를 생성합니다
18
19 % 문서의 내용을 입력합니다
20
21 \end{document}

```

한국어 문서 템플릿

한국어 문서를 작성한다면 예시 문서 3.2를 템플릿으로 사용할 수 있습니다. 이 템플릿의 다른 부분은 예시 문서 3.1과 동일하지만, 한국어 문서를 작성하기 위해 `kotex` 패키지를 불러오고 한글 글꼴 설정을 해 주었습니다. 다른 패키지를 불러오려는 경우, 이를 `kotex` 패키지보다 먼저 불러오는 것이 좋습니다.

한국어 문서는 자유로운 문자 처리를 위해 유니코드 엔진으로 컴파일 하는 것이 좋습니다. 레거시 엔진으로 문서를 컴파일 해야 하는 경우, 한글 글꼴을 설정하는

```
\setmainhangulfont \setsanshangulfont \setmonohangulfont
```

제어 문자열이 작동하지 않으므로 제6~8행을 삭제하고 컴파일 하시기 바랍니다.

참고 | 유니코드 엔진에서는 나눔명조, 나눔고딕, D2Coding 글꼴이 컴퓨터에 설치되어 있어야 합니다. 이 글꼴은 [57]에서 무료로 내려받을 수 있습니다. Overleaf에서는 위 글꼴이 준비되어 있습니다. 대신, `xetexko` 패키지 안내서[23]의 방법을 사용해 다른 한글 글꼴을 사용할 수도 있습니다.

연습문제

3.1. 다음 코드에서 사용된 제어 문자열을 모두 찾아라.

```
1 \documentclass{article}
2 \usepackage[hangul]{kotex}
3 \setmainhangulfont{나눔명조}
4 \begin{document}
5 1년은 52주 하고 하루, 즉
6 \[ 52 \times 7 + 1 = 365 \]
7 일로 이루어져 있다.
8 \end{document}
```

1년은 52주 하고 하루, 즉

$$52 \times 7 + 1 = 365$$

일로 이루어져 있다.

3.2. 다음 코드에서 사용된 환경을 모두 찾아라.

```
1 \documentclass{article}
2 \usepackage[hangul]{kotex}
3 \setmainhangulfont{나눔명조}
4 \begin{document}
5 \begin{center}
6 \begin{tabular}{c|ccc}
7 \hline
8 품목 & 개수 & 단가 & 가격 \\
9 사과 & 3 & 600 & 1,800 \\
10 배 & 5 & 800 & 4,000 \\
11 \end{tabular}
12 \end{center}
13 \end{document}
```

품목	개수	단가	가격
사과	3	600	1,800
배	5	800	4,000

3.3. 문단이 올바르게 나뉘도록 다음 글을 \LaTeX 으로 작성하여라.

해석학이란 연속성을 가지는 대상을 탐구하는 분야이다. 실수에 대한 엄밀한 정의를 바탕으로 극한, 연속성, 미분, 적분과 같은 개념을 다루며, 이를 가지고 측도론, 작용소 이론, 편미분방정식론, 최적화 이론을 비롯한 순수수학과 응용수학의 폭넓은 주제를 논의한다.

대수학이란 대수적 구조의 다양한 성질을 탐구하는 분야이다. 집합의 원소 사이의 연산을 바탕으로 군, 환, 체, 벡터 공간 등의 대수적 구조를 탐구하며, 더 나아가 가환대수학, 표현론, 대수적 정수론, 대수기하학 등의 분야를 연구한다.

기하학이란 위치와 거리 등을 이용하여 공간을 탐구하는 분야이다. 과거의 기하학은 공리를 통한 논증을 중심으로 직관적인 도형의 성질을 탐구하였다. 현대의

기하학은 해석학과 대수학 등의 기법을 이용하며 리만기하학, 사교기하학, 복소기하학, 대수기하학과 같은 갈래가 있다.

- 3.4. 다음 문장을 \LaTeX 에서 조판하기 위한 코드를 작성하여라. (전처리부의 코드 없이 document 환경 안의 코드만 작성하여라.)

새로운 모델의 미국 출시 가격은 이전 모델에 비해 10%가량 상승하여 \$1,300로 결정되었다.

- 3.5. 페이지 크기는 B5이고, 본문의 기본 글꼴 크기는 10 pt이며 수식 번호가 왼쪽에 표시되는 book 클래스의 문서를 작성하려고 한다. 이때 클래스를 불러오는 코드를 작성하여라.
- 3.6. 목록 환경의 형태를 변경하는 기능을 제공하는 enumitem 패키지를 옵션 없이 불러오는 코드를 작성하여라.
- 3.7. 유니코드 엔진에서 다양한 글꼴을 사용할 수 있도록 돕는 fontspec 패키지를 불러오려고 한다. no-math 옵션을 지정하여 이 패키지를 불러오는 코드를 작성하여라.
- 3.8. 이 책에서 괄호 안에 부연 설명을 할 때에는 \paren 제어 문자열을 사용하였다. 이 제어 문자열은 인자로 입력받은 내용을 괄호 안에 작은 크기로 조판한다.

1 세종대왕은 훈민정음을 1446년에 반포\paren{창제는 1443년}하였다.

세종대왕은 훈민정음을 1446년에 반포(창제는 1443년)하였다.

글자를 작게 조판하도록 선언하는 \small 제어 문자열을 사용하여, 위와 같은 동작을 하는 \paren 제어 문자열을 정의하여라.

- 3.9. 다음 글에서 행중수식과 별행수식을 구분하여라.

원주율 π 는 다음 등식을 만족한다.

$$\frac{\pi}{2} = \frac{2}{1} \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{4}{5} \cdot \frac{6}{5} \cdot \frac{6}{7} \cdots$$

- 3.10. 다음 글에서 행중수식과 별행수식을 구분하여라.

로그함수는 \log 로 표기하며, 임의의 두 양수 x 와 y 에 대해

$$\log x + \log y = \log xy$$

를 만족한다.



본문 작성하기

제II편은 수식 기능을 제외한 \LaTeX 의 기본 기능을 설명합니다. 다양한 문자를 사용하여 문서의 내용을 작성하고, 표제와 부록 등을 만들어 목차를 만드는 방법을 알아봅니다. 또, 문서의 내용을 효과적으로 전달하기 위해 여러 서식을 적용하는 방법도 다룹니다. 정리나 정의, 증명 등 여러 수학적 문단을 일정한 형식에 맞추어 작성하는 방법도 알아볼 것입니다.

제4장 문자 입력하기 · 37

제5장 목차 구성하기 · 53

제6장 여러 가지 서식 · 65

제7장 수학적 문단 작성하기 · 81

제 4 장

문자 입력하기

이번 장에서는 \LaTeX 으로 작성하는 문서에 다양한 문자를 입력하는 방법을 알아봅니다. 오늘날의 \LaTeX 시스템에서는 대부분의 로마자나 한글, 몇 가지 기호는 문제 없이 처리되지만, 쉽게 입력할 수 없는 문자를 편리하게 입력하도록 도와 주는 명령어를 소개할 것입니다. 특히, 몇 가지 문장부호는 유의해서 입력해야 합니다.

참고 | 이번 장에서는 영어 또는 한국어 문서를 작성할 때 알아야 할 기본적인 내용만을 설명합니다. 다양한 언어의 문서를 작성하고 문자를 처리하기 위한 방법은 `babel`, `polyglossia`, `kotex`, `ctex`, `luatexja`, `bidi` 패키지[5, 7, 11, 22, 50, 56]를 참조하시기 바랍니다.

레거시 엔진에서의 사전 설정

레거시 엔진은 문자 처리와 글꼴 처리에 약간의 제약이 있어, 한두 가지 사전 설정을 해 주어야 합니다. 첫째로, 2017년 이전의 \TeX 배포판을 사용한다면 다음을 입력해야 합니다(2018년 이후 버전에서는 이 코드가 불필요합니다). 이 코드는 입력 파일에 작성된 문자를 올바르게 해석할 수 있게 해 줍니다.

```
\usepackage[utf8]{inputenc}
```

둘째로, \TeX 배포판의 버전에 관계 없이 전처리부에 다음을 입력합니다. 이는 글꼴을 더 원활하게 처리하고, 생성된 PDF 파일에서 문자 검색을 용이하게 하기 위함입니다.

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

이상의 코드를 입력했다면 레거시 엔진에서의 문자 처리를 위한 사전 설정이 완료됩니다. 예시 문서 2.1과 예시 문서 3.1에서도 이 코드를 확인할 수 있습니다. 유니코드 엔진은 이러한 문제를 해결하기 위해 개발된 엔진이므로, 이러한 설정이 불필요합니다.

| 표 4.1 | 확장 로마자

Üü \{"U\}"{u}	Ôô \^{O}\^{o}	Çç \c{C}\c{c}	Åå \r{A}\r{a}
Éé \{'E\}'{e}	Èè \`{E}\`{e}	Àà \d{A}\d{a}	Ââ \t{A}\t{a}
Žž \.{I}\.{i}	Ññ \~{N}\~{n}	Őő \H{O}\H{o}	Ğğ \u{G}\u{g}
İı \={I}\={i}	Ķķ \b{K}\b{k}	Ĳĳ \k{E}\k{e}	Šš \v{Z}\v{z}
Åå \AA\aa	Hh \Hwithstroke\hwithstroke	Œœ \OE\oe	
Ææ \AE\ae	Ij \IJ\ij	Ŋŋ \NG\ng	Šš \SS\ss ¹
Ðð \DH\dh	ıı \i\j	Øø \O\o	Þþ \TH\th
Đđ \DJ\dj	Łł \L\l		

¹ 엔진이나 글꼴 등에 따라 대문자 에스체트는 다른 형태로 표시될 수 있습니다.

4.1 로마자 입력하기

TeX은 미국에서 개발된 만큼, 기본적인 로마자 52자 A~Z와 a~z를 문제 없이 입출력할 수 있습니다. 또, æ, é, ñ, ø, ß 등의 확장 로마자도 문제 없이 입력받고, 이를 PDF 문서에 조판할 수 있습니다. 그러나 확장 로마자는 영어 키보드에서 입력이 불편하므로, TeX은 이를 편리하게 입력할 수 있는 명령어를 제공합니다. 또, 명령어를 사용하면 대·소문자를 편리하게 전환할 수 있습니다. 이번 절에서는 문서에 로마자와 한글을 입력하는 방법을 알아봅니다.

4.1.1 확장 로마자

표 4.1의 제어 문자열을 사용하면 영어 키보드에 있는 문자만으로도 확장 로마자를 쉽게 입력할 수 있습니다. 인자를 받는 제어 문자열들의 인자를 바꾸면 다른 문자에 악센트를 붙일 수 있습니다.

```
1 Wczoraj zjad\l em troch\k{e}
   ziemniak\'{o}w.
```

Wczoraj zjadłem trochę ziemniaków.

4.1.2 대·소문자 전환

로마자에는 대문자와 소문자의 구분이 있으며, 강조나 디자인 등의 목적으로 이들을 일반적인 상황과 다르게 사용하기도 합니다. 입력된 문자열을 대문자로 전환할 때에는 \MakeUppercase 제어 문자열을, 입력된 문자열을 소문자로 전환할 때에는 \MakeLowercase 제어 문자열을 사용합니다.

`\MakeUppercase{<text>}`

`<text>`: 대문자로 전환할 문자열

`\MakeLowercase{<text>}`

`<text>`: 소문자로 전환할 문자열

You can switch some arbitrary text into `\MakeUppercase{Uppercase}` or `\MakeLowercase{Lowercase}` using commands.

You can switch some arbitrary text into UPPERCASE or lowercase using commands.

상호 참조
→ 207쪽

이 제어 문자열의 인자로 지정된 내용 중 수식 부분이나 상호 참조에 사용되는 제어 문자열인 `\label`이나 `\ref` 등의 인자로 지정된 식별자 부분은 대·소문자 전환이 일어나지 않습니다. 이 부분에서 대·소문자 전환이 일어나면 의도하지 않은 결과를 얻을 수 있기 때문입니다. 이외에도 대·소문자를 전환하지 않을 부분을 직접 지정하려는 경우 `\NoCaseChange` 제어 문자열을 사용할 수 있습니다.

참고 | 2022년 6월 이전 버전의 \LaTeX 은 `\MakeUppercase`나 `\MakeLowercase`의 인자로 지정된 내용 중 수식도 대·소문자를 전환합니다.

플레인 \TeX 의 명령어에 대한 조언

플레인 \TeX 에서는 대·소문자를 전환하는 제어 문자열로 `\uppercase`와 `\lowercase`를 제공합니다. 하지만 이는 기본 로마자 52자 외의 문자는 처리할 수 없으므로, \LaTeX 의 명령어를 사용하는 것이 좋습니다.

4.2 문장 부호 입력하기

문서의 내용을 독자에게 효과적으로 전달하기 위해서는 **문장 부호**를 적절한 곳에 사용해야 합니다. 이번 절에서는 문장 부호를 입력하는 명령어와 주의 사항을 알아봅니다.

4.2.1 유럽 언어의 문장 부호

유럽 언어에서 사용하는 문장 부호는 표 4.2의 명령어를 사용하여 쉽게 입력할 수 있습니다. 불러온 패키지 등으로 인해 이 표의 명령어가 작동하지 않는 경우에는 표 4.5의 제어 문자열을 대신 사용할 수 있습니다.

이제 따옴표, 하이픈·대시, 줄임표를 입력할 때 주의할 점을 알아보겠습니다.

표 4.2 | 문장 부호를 입력하는 문자열

¡ !`	¿ ?`	— --	— ---
‘ ’`	’ ’`	“ ”`	” ”`
« <<	» >>	” ’`	

따옴표

따옴표를 입력할 때는 여는 것과 닫는 것을 구분해 입력해야 합니다. `TeX`에서는 그 레이트 악센트 `‘’`를 사용해 여는 따옴표를 입력하고, 작은따옴표 `’`를 사용해 닫는 따옴표를 입력합니다. 같은 기호를 두 번 입력하면 큰따옴표가 입력되며, `TeX`에서는 큰따옴표 문자 `”`로 큰따옴표를 입력하지 않습니다. `아포스트로피``apostrophe`도 닫는 작은따옴표와 같은 방법으로 입력합니다.

¡ Jessica says, ``Oliver is drawing. He said, `Please help me!`, '`	Jessica says, “Oliver is drawing. He said, ‘Please help me!’”
---	---

위의 예시에서 볼 수 있듯, 작은따옴표와 큰따옴표가 연달아 나온다면 두 따옴표 사이에 `\,`를 입력하여 둘을 구분해 주는 것이 좋습니다. 반대로 큰따옴표와 작은따옴표가 연달아 나오는 경우도 마찬가지입니다.

하이픈과 대시

키보드로 바로 입력할 수 있는 기호 `‘-’`는 문서에 `하이픈``hyphen`을 입력합니다. 이와 비슷한 모양의 기호인 `엔 대시``en dash`는 하이픈 두 개, 즉 `--`로 입력하며, `엠 대시``em dash`는 하이픈 세 개 `---`로 입력합니다.

¡ The boy---who is five-feet tall---is about 2--3 times smarter than his peers.	The boy—who is five-feet tall—is about 2-3 times smarter than his peers.
---	--

한편, 수치를 나타낼 때의 음의 부호는 수식 모드를 사용하거나 `\textminus`를 입력하여 조판하며, 하이픈이나 대시로 입력하는 것이 아닙니다.

¡ You will get \(-2\)~points for each wrong answer.	You will get -2 points for each wrong answer.
---	---

줄임표

줄임표는 마침표를 세 번 찍은 모습을 하고 있지만, 소스 파일에서 단순히 마침표 세 개를 입력하면 점들이 너무 가까이 붙은 형태로 조판됩니다. 대신, `\textellipsis` 제어 문자열, 또는 간단히 `\dots`나 `\ldots`를 사용하면 줄임표를 바르게 조판할 수 있습니다.

The answer for this problem is `\dots` not allowed to be exposed.

The answer for this problem is ... not allowed to be exposed.

참고 | 유니코드 엔진에서는 줄임표가 다른 모양으로 출력되므로, 다음 코드를 전처리부에 입력해야 올바른 형태의 줄임표가 출력됩니다.

```
\UndeclareTextCommand\textellipsis{TU}
```

4.2.2 단어 사이의 간격과 문장 사이의 간격

영어 문서에서는 단어 사이의 간격보다 문장 사이의 간격을 더 넓게 띄우는 관행이 있습니다. 이 때문에, \LaTeX 은 마침표나 물음표, 느낌표 뒤에 공백 문자가 입력되면 문장이 끝났다고 판단하고, 일반적인 띄어쓰기보다 더 넓은 공백을 만듭니다.

Mint chocolate is one of my favorite dessert. It is cool, sweet, and refreshing. I wish everyone would like this dessert.

Mint chocolate is one of my favorite dessert. It is cool, sweet, and refreshing. I wish everyone would like this dessert.

물음표나 느낌표는 항상 문장을 끝낼 때 사용되지만, 마침표는 문장을 끝낼 때뿐 아니라 약어를 표기할 때에도 사용됩니다. 따라서 마침표 바로 앞의 문자가 대문자인 경우 \LaTeX 은 문장이 끝났다고 생각하지 않으며, 일반적인 단어 사이 간격 만큼의 공백이 만들어집니다.

I lived in the U.S. for two months. It had been about a year since I left there.

I lived in the U.S. for two months. It had been about a year since I left there.

따라서, 소문자 뒤에 마침표를 찍었지만 문장이 끝나지 않은 경우, 또 대문자 뒤에 마침표를 찍었지만 문장이 끝난 경우에는 이를 \LaTeX 에게 알려 주어야 합니다. 전자의 경우에는 마침표 뒤의 띄어쓰기를 `_`로 입력하면 되고, 후자의 경우에는 마침표 앞에

| 표 4.3 | 한국어 문장 부호 입력 명령어

문장 부호	명령어	문장 부호	명령어
여는 큰따옴표 “	`` 또는 <code>\textquotedblleft</code>	닫는 큰따옴표 ”	'' 또는 <code>\textquotedblright</code>
여는 작은따옴표 ‘	` 또는 <code>\textquoteleft</code>	닫는 작은따옴표 ’	' 또는 <code>\textquoteright</code>
가운뎃점 ·	<code>\textperiodcentered</code>	3점 줄임표 …	<code>\dots, \ldots, \textellipsis</code>
물결표 ~	<code>\~{}</code> 또는 <code>\textasciitilde</code>	줄표 —	--- 또는 <code>\textemdash</code>

\@를 입력하면 됩니다. 이렇게 함으로써 예외적인 상황에서도 올바른 간격의 공백을 얻을 수 있습니다.

Recently, Thomas is interested in the World War II\@. He reads lots of books, articles, etc.\ during his spare time.

Recently, Thomas is interested in the World War II. He reads lots of books, articles, etc. during his spare time.

반면, 영어가 아닌 많은 언어에서는 문장 뒤의 공백을 단어 사이의 공백과 같은 간격으로 조판합니다. `kotex`이나 `babel` 등 언어 설정 패키지를 불러온 경우, 문장 사이의 간격을 단어 사이의 간격보다 더 넓게 조판할지 동일하게 조판할지를 이 패키지가 자동으로 결정해 줍니다. 단어 사이의 간격과 문장 사이의 간격을 동일하게 조판하는 언어(한국어도 여기에 포함됩니다.)라면 \@나 _ 등을 입력하지 않아도 됩니다.

문장 사이의 간격 옵션을 직접 바꿀 수 있도록 하는 제어 문자열로 `\frenchspacing`과 `\nonfrenchspacing`이 있습니다. 전자는 ‘문장 사이의 간격을 더 넓게 조판하지 않음’을 선언하며, 후자는 반대로 ‘문장 사이의 간격을 더 넓게 조판함’을 선언합니다. 언어 패키지를 통해 원하는 설정을 얻을 수 없는 경우에 이 명령어를 사용하면 됩니다.

4.2.3 한국어 문장 부호

표 4.2에 없는 한국어 문장 부호는 4.3절의 표에 있는 긴 명령어를 통해 입력하거나, 운영 체제의 문자표 삽입 기능을 통해 입력해야 합니다. 먼저, \LaTeX 에서 명령어를 제공하는 한국어 문장 부호를 표 4.3에 정리하였습니다.

이제 줄임표, 붙임표와 줄표, 물결표를 입력할 때의 주의 사항을 알아봅시다.

줄임표

3점 줄임표는 표 4.3에 제시한 세 가지 제어 문자열로 입력합니다. 6점 줄임표는 이 명령어를 두 번 입력하면 됩니다.

이번 주 금요일까지 원고 마감을 할 수 있을까
\dots\dots.

이번 주 금요일까지 원고 마감을 할 수 있
을까…….

참고 | 레거시 엔진에서는 언급한 제어 문자열이 영어의 줄임표 ‘...’를 출력하므로 전처리부에서 다음 선언을 해 주어야 합니다.

```
\DeclareTextCommandDefault{\textellipsis}{\unichar{"2026}}
```

유니코드 엔진에서는 41쪽의 선언을 하지 않아야 한국어의 줄임표가 출력됩니다.

유니코드 엔진에서 kotex 패키지를 불러온 경우, 이 표에 없는 문장 부호 중 6점 줄임표 ‘……’를 \hellipsis로 입력할 수 있습니다.

이번 주 금요일까지 원고 마감을 할 수 있을까
\hellipsis.

이번 주 금요일까지 원고 마감을 할 수 있
을까…….

불임표와 줄표

불임표는 영어 문장 부호의 하이픈과 동일한 방법으로 입력합니다. 또, 줄표는 영어 문장 부호의 엠 대시와 동일한 방법으로 입력합니다. 즉, 불임표는 -로, 줄표는 ---로 입력합니다.

처음-중간-끝 세 부분이 조화롭게 이어지는 --- 주제를 벗어나지 않으며 앞뒤 내용이 매끄럽게 이어지는 --- 글을 쓰는 것은 쉽지 않다.

처음-중간-끝 세 부분이 조화롭게 이어지는 — 주제를 벗어나지 않으며 앞뒤 내용이 매끄럽게 이어지는 — 글을 쓰는 것은 쉽지 않다.

물결표

물결표는 \textasciitilde 제어 문자열, 또는 간단히 ~로 입력합니다.

과제는 8장의 연습문제 1~30번입니다.

과제는 8장의 연습문제 1~30번입니다.

다만, 레거시 엔진에서는 이 코드가 물결표 대신 틸데 악센트 ‘~’를 출력하므로, 이 대신 수식 모드의 닳음 기호 ‘~’를 출력하는 \sim을 사용하는 것이 더 좋을 수 있습니다. 예시 문서 4.1을 사용하면 레거시 엔진에서 더 나은 모양의 물결표를 조판할 수 있습니다.

| 표 4.4 | kopunct 패키지: 한국어 문장 부호를 입력하는 제어 문자열

문장 부호	제어 문자열	문장 부호	제어 문자열
가운뎃점 ·	<code>\cdot</code>	줄임표 …, ……	<code>\cdots, \cdotss</code>
낫표 「, 」	<code>\lnat, \rnat</code>	겹낫표 『, 』	<code>\lNat, \rNat</code>
화살괄호 <, >	<code>\lhwa, \rhwa</code>	겹화살괄호 <<, >>	<code>\lHwa, \rHwa</code>
숨김표 ○, ×	<code>\dongl, \gawi</code>	빠짐표 □	<code>\nemo</code>

한국어 문장 부호를 입력하기 위한 매크로

긴 명령어를 입력해야 하는 가운뎃점이나 줄임표, 그리고 명령어를 통해 입력할 수 없는 문장 부호는 매크로를 정의해 두면 편리하게 입력할 수 있습니다. 예시 문서 4.1의 코드를 `kopunct.sty` 파일로 저장한 후, 전처리부에서 `kopunct` 패키지를 불러 오면 표 4.4의 제어 문자열을 사용하여 한국어 문장 부호를 입력할 수 있습니다.

가운뎃점과 3점 줄임표는 수식 모드에서만 사용할 수 있던 `\cdot`과 `\cdots`를 재정의하여, 텍스트 모드에서도 사용할 수 있도록 만들었습니다. 텍스트 모드에서 이 명령어를 사용하면 문장 부호가 출력됩니다. 이들과 비슷한 이름인 `\cdotss`는 6점 줄임표를 출력하도록 정의하였습니다.

<code>물음표\cdot</code> 느낌표를 남용하면 안 된다.	물음표·느낌표를 남용하면 안 된다.
<code>이게 그\cdots</code> 령게 됐다고 <code>\cdotss</code> ?	이게 그…렇게 됐다고……?

`\dots, \ldots, \textellipsis`는 항상 영어의 줄임표를 조판하도록 정의가 변경되었습니다.

물결표는 기존과 동일한 명령어를 사용하여 입력합니다. 다만, 레거시 엔진에서도 올바른 모양으로 조판되도록 내부 매크로를 재정의하였습니다.

낫표와 화살괄호는 표 4.4의 제어 문자열을 사용하여 입력할 수 있지만, 보통 쌍을 맞추어 사용하므로 `\nat, \Nat, \hwa, \Hwa`를 추가로 정의해 두었습니다. 이들은 인자를 입력받으며, 그 내용을 명령어의 이름에 대응하는 문장 부호로 감싸 조판해 줍니다.

<code>대한민국의 국가는 \hwa{애국가}</code> 이다.	대한민국의 국가는 <애국가>이다.
<code>\Nat{훈민정음}</code> 은 1997년에 유네스코 세계 기록 유산으로 지정되었다.	『훈민정음』은 1997년에 유네스코 세계 기록 유산으로 지정되었다.

| 예시 문서 4.1 | kopunct.sty

```

1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{kopunct}[2024/04/20 Package for providing macros to input
   Korean punctuation marks]
3
4 \newif\ifkopunct@nocdot\kopunct@nocdotfalse
5 \newif\ifkopunct@nocdots\kopunct@nocdotsfalse
6 \newif\ifkopunct@notilde\kopunct@notildefalse
7 \DeclareOption{nocdot}{\kopunct@nocdottrue}
8 \DeclareOption{nocdots}{\kopunct@nocdotstrue}
9 \DeclareOption{notilde}{\kopunct@notildetrue}
10
11 \ProcessOptions\relax
12 \RequirePackage{iftex}
13
14 \NewCommandCopy\textcdot{\textperiodcentered}% text mode cdot
15 \NewCommandCopy\mathcdot{\cdot}% math mode cdot
16 \ifkopunct@nocdot\else
17   \protected\def\cdot{\ifmmode\mathcdot\else\textcdot\fi}% 가운데점
18 \fi
19
20 \iftutex % For Unicode engines
21   \def\lhwa{^^^3008}\def\rhwa{^^^3009}% 화살괄호
22   \def\lHwa{^^^300a}\def\rHwa{^^^300b}% 겹화살괄호
23   \def\lnat{^^^300c}\def\rnat{^^^300d}% 낫표
24   \def\lNat{^^^300e}\def\rNat{^^^300f}% 겹낫표
25
26   \def\textcdots{^^^2026}% text mode cdots
27   \NewCommandCopy\mathcdots{\cdots}% math mode cdots
28   \ifkopunct@nocdots\else
29     \protected\def\cdots{\ifmmode\mathcdots\else\textcdots\fi}% 3점 줄임표
30   \fi
31   \def\cdotss{\textcdots\textcdots}% 6점 줄임표
32   \UndeclareTextCommand\textellipsis{TU}% 영어 줄임표
33
34   \def\dongl{^^^25cb}% 숨김표 (원)
35   \NewCommandCopy\gawi\texttimes % 숨김표 (가위표)
36   \def\nemo{^^^25a1}% 빠짐표
37 \else % For legacy engines
38   \def\lhwa{^e3^80^88}\def\rhwa{^e3^80^89}% 화살괄호
39   \def\lHwa{^e3^80^8a}\def\rHwa{^e3^80^8b}% 겹화살괄호
40   \def\lnat{^e3^80^8c}\def\rnat{^e3^80^8d}% 낫표
41   \def\lNat{^e3^80^8e}\def\rNat{^e3^80^8f}% 겹낫표
42

```

```

43 \DeclareUnicodeCharacter{2026}{\unichar{"2026}}
44 \DeclareRobustCommand\textcdots{^^e2^^80^^a6}% text mode cdots
45 \NewCommandCopy\mathcdots{\cdots}% math mode cdots
46 \ifkopunct@nocdots\else
47   \protected\def\cdots{\ifmmode\mathcdots\else\textcdots\fi}% 3점 줄임표
48 \fi
49 \def\cdotss{\textcdots\textcdots}% 6점 줄임표
50
51 \ifkopunct@notilde\else
52   \newlength\temp@length@a \setlength\temp@length@a{\f@size pt}%
53   \newlength\temp@length@b \setlength\temp@length@b{\f@baselineskip
54   \newcommand\TildeScaleFactor{1.5}% 틸데를 확대할 배율
55   \newlength\TildeRaise \setlength\TildeRaise{-1.35ex}% 틸데를 올릴 길이
56   \newlength\TildeHeight \setlength\TildeHeight{1ex}% 틸데의 높이
57   \newcommand\kopunct@tilde{\raisebox{\TildeRaise}{\TildeHeight}[0pt]
58   {\fontsize{\TildeScaleFactor\temp@length@a}
59   {\baselinestretch\temp@length@b}\selectfont\char"7E}}% 물결표
60   \DeclareTextCompositeCommand{\~}{OT1}{\~}{\kopunct@tilde}
61   \DeclareTextCompositeCommand{\~}{T1}{\~}{\kopunct@tilde}
62 \fi
63
64 \def\dongl{^^e2^^97^^8b}% 숨김표 (원)
65 \NewCommandCopy\gawi\texttimes % 숨김표 (가위표)
66 \def\nemo{^^e2^^96^^a1}% 빠짐표
67 \fi
68
69 \def\hwa#1{{\lhwa #1\rhwa}}% 화살괄호
70 \def\Hwa#1{{\lHwa #1\rHwa}}% 겹화살괄호
71 \def\nat#1{{\lnat #1\rnat}}% 낫표
72 \def\Nat#1{{\lNat #1\rNat}}% 겹낫표
73 \endinput

```

4.3 기호 입력하기

이번 절에서는 \LaTeX 문서에서 제어 문자열을 통해 입력할 수 있는 기호를 알아봅니다.

4.3.1 \LaTeX 의 기본 기호

\LaTeX 은 표 4.5의 제어 문자열을 제공합니다. 기본적인 기호는 이 표의 제어 문자열을 사용하여 입력할 수 있습니다. 이 표의 제어 문자열 중 `\textcircled`는 인자를 하나 입력받아 원 안에 조판합니다.

표 4.5 | L^AT_EX의 기호

« \guillemetleft ¹	\textcompwordmark	¶ \textparagraph ²
» \guillemetright ¹	© \textcopyright ²	· \textperiodcentered
‹ \guilsinglleft	† \textdagger ²	‰ \textpertenthousand
› \guilsinglright	‡ \textdaggerdbl ²	‱ \textperthousand
„ \quotedblbase ¹	\$ \textdollar ²	¿ \textquestiondown ¹
ˆ \textasciicircum	… \textellipsis ²	" \textquotedbl
˜ \textasciitilde	— \textendash ¹	“ \textquotedblleft ¹
* \textasteriskcentered	- \textendash ¹	” \textquotedblright ¹
\ \textbackslash	¡ \textexclamdown ¹	‘ \textquoteleft ¹
\textbar	- \textfiguredash	, \textquoteright ¹
\textbardbl	> \textgreater	® \textregistered ²
○ \textbigcircle	— \texthorizontalbar	§ \textsection ²
{ \textbraceleft ²	< \textless	£ \textsterling ²
} \textbraceright ²	- \textnonbreakinghyphen	™ \texttrademark
• \textbullet	^a \textordfeminine	_ \textunderscore ²
⊗ \textcircled{x}	° \textordmasculine	~ \textvisiblespace

¹ 표 4.2의 명령어를 사용해 입력할 수도 있습니다.

² 표 4.6의 더 짧은 제어 문자열을 사용해 입력할 수도 있습니다.

표 4.6 | 텍스트 모드와 수식 모드에서 모두 사용 가능한 기호

{ \{	_ _	‡ \ddag	£ \pounds
} \}	© \copyright	… \dots ¹	§ \S
\$ \\$	† \dag	¶ \P	

amsfonts 패키지에서 제공하는 기호:

✓ \checkmark	® \circledR	✠ \maltese	¥ \yen
--------------	-------------	------------	--------

¹ \ldots로도 조판할 수 있습니다. 유니코드 엔진에서는 점 사이의 간격이 더 좁은 기호 ‘…’ 또는 가운데점
점을 사용한 기호 ‘…’의 형태로 조판됩니다.

짧은 제어 문자열

앞서 소개한 표 4.5에 제시된 제어 문자열은 매우 길니다. 문서에 기호를 자주 입력한다면 명령어를 입력하는 것이 귀찮을 것입니다. 따라서, L^AT_EX은 자주 사용되는 기호를 편리하게 입력할 수 있도록 짧은 제어 문자열을 제공하며, 이들을 표 4.6에 정리하였습니다. 이 표에 있는 제어 문자열은 수식 모드와 텍스트 모드에서 모두 사용할 수 있습니다.

표 4.7 | textcomp 패키지: 악센트

È \capitalgrave{E}	Å \capitalring{A}	Ǝ \capitalogonek{E}
É \capitalacute{E}	Š \capitalcaron{S}	aa \t{aa}
Ô \capitalcircumflex{O}	Ġ \capitalbreve{G}	AA \capitaltie{AA}
Ñ \capitaltilde{N}	Ī \capitalmacron{I}	â \newtie{a}
Ü \capitaldieresis{U}	Ž \capitaldotaccent{Z}	Â \capitalnewtie{A}
Ő \capitalhungarumlaut{O}	Ç \capitalcedilla{C}	⊗ \textcircled{x}

4.3.2 textcomp 패키지의 기호

textcomp 패키지는 L^AT_EX 문서에서 더 다양한 기호를 사용할 수 있게 해 줍니다. 패키지를 불러 오면 표 4.7과 표 4.8의 기호를 입력할 수 있게 됩니다.

참고 | 2019년 이전 버전의 레거시 엔진이나 2016년 이전 버전의 유니코드 엔진에서는 textcomp 패키지를 불러 와야 두 표에 있는 명령어를 사용할 수 있습니다. 2020년 이후 버전의 레거시 엔진에서는 이 패키지가 기본적으로 불러온 상태로 문서가 조판되며, 2017년 이후 버전의 유니코드 엔진에서는 textcomp 패키지 없이 명령어가 기본 제공됩니다.

표 4.7은 인자를 하나 입력 받는 제어 문자열들을 모아 둔 것입니다. \textcircled를 제외하고는 보통 직접 사용되지 않으며, 표 4.1에 있는 악센트를 입력하는 제어 문자열의 인자로 대문자가 지정되었을 때 L^AT_EX이 내부적으로 사용합니다. 표 4.8에는 인자를 입력받지 않는 제어 문자열을 모아 두었습니다.

L^AT_EX에서 기본 제공하는 제어 문자열과 textcomp 패키지가 제공하는 제어 문자열 중에는 겹치는 것들이 있습니다. 이들은 같은 기호를 출력하지만, 그 모양이 미세하게 다릅니다. textcomp의 것이 더 미려한 기호를 제공하므로, 제어 문자열이 겹치는 기호는 textcomp 패키지에서 제공하는 모양으로 출력됩니다.

\textcompwordmark의 사용

표 4.6에 있는 \textcompwordmark와 표 4.8에 있는

\textcapitalcompwordmark \textascendercompwordmark

는 문자를 출력하거나 공백을 만들지는 않지만, 특수한 목적으로 사용되는 기호를 문서에 입력합니다. 이들은 로마자의 합자ligature 기능이 작동하지 않도록 하거나, 악센트를 두 문자 사이에 조판하려 할 때 사용할 수 있습니다.

표 4.8 | textcomp 패키지: 기호

<code>\textasciigrave</code>	○ <code>\textbigcircle</code>	‰ <code>\textdiscount</code>
<code>\texttildelow</code>	Ω <code>\textohm</code>	€ <code>\textestimated</code>
<code>\textasciibreve</code>	⌈ <code>\textlbrackdbl</code>	◦ <code>\textopenbullet</code>
<code>\textasciicaron</code>	⌋ <code>\textrbrackdbl</code>	SM <code>\textservicemark</code>
” <code>\textacutedbl</code>	↑ <code>\textuparrow</code>	{ <code>\textlquill</code>
“ <code>\textgravedbl</code>	↓ <code>\textdownarrow</code>	} <code>\textrquill</code>
“ <code>\textasciidieresis</code>	★ <code>\textborn</code>	¢ <code>\textcent</code>
˘ <code>\textasciimacron</code>	◊ <code>\textdivorced</code>	£ <code>\textsterling¹</code>
ˆ <code>\textasciiacute</code>	† <code>\textdied</code>	¤ <code>\textcurrency</code>
, <code>\textquotestraightbase</code>	☞ <code>\textleaf</code>	¥ <code>\textyen</code>
„ <code>\textquotestraightdblbase</code>	∞ <code>\textmarried</code>	<code>\textbrokenbar</code>
— <code>\texttwelveldash</code>	♪ <code>\textmusicalnote</code>	§ <code>\textsection¹</code>
— <code>\textthreequartersemdash</code>	= <code>\textdblhyphenchar</code>	© <code>\textcopyright¹</code>
<code>\textcapitalcompwordmark</code>	‡ <code>\textdagger¹</code>	ª <code>\textordfeminine</code>
← <code>\textleftarrow</code>	‡ <code>\textdaggerdbl¹</code>	© <code>\textcopyleft</code>
→ <code>\textrightarrow</code>	<code>\textbardbl</code>	¬ <code>\textlnot</code>
<code>\textascendercompwordmark</code>	‰ <code>\textperthousand</code>	Ⓟ <code>\textcircledP</code>
b <code>\textblank</code>	• <code>\textbullet</code>	® <code>\textregistered</code>
\$ <code>\textdollar¹</code>	°C <code>\textcelsius</code>	° <code>\textdegree</code>
' <code>\textquotesingle</code>	\$ <code>\textdollaroldstyle</code>	± <code>\textpm</code>
* <code>\textasteriskcentered</code>	¢ <code>\textcentoldstyle</code>	² <code>\texttwosuperior</code>
= <code>\textdblhyphen</code>	f <code>\textflorin</code>	³ <code>\textthreesuperior</code>
/ <code>\textfractionsolidus</code>	⌚ <code>\textcolonmonetary</code>	μ <code>\textmu</code>
0 <code>\textzerooldstyle</code>	W <code>\textwon</code>	¶ <code>\textparagraph¹</code>
1 <code>\texttoneoldstyle</code>	N <code>\textnaira</code>	· <code>\textperiodcentered</code>
2 <code>\texttwooldstyle</code>	G <code>\textguarani</code>	※ <code>\textreferencemark</code>
3 <code>\textthreeoldstyle</code>	P <code>\textpeso</code>	¹ <code>\textonesuperior</code>
4 <code>\textfouroldstyle</code>	£ <code>\textlira</code>	º <code>\textordmasculine</code>
5 <code>\textfiveoldstyle</code>	R <code>\textrecipe</code>	√ <code>\textsurd</code>
6 <code>\textsixoldstyle</code>	? <code>\textinterrobang</code>	¹ / ₄ <code>\texttonequarter</code>
7 <code>\textsevenoldstyle</code>	‡ <code>\textinterrobangdown</code>	¹ / ₂ <code>\texttonehalf</code>
8 <code>\texteightoldstyle</code>	d <code>\textdong</code>	³ / ₄ <code>\textthreequarters</code>
9 <code>\textnineoldstyle</code>	™ <code>\texttrademark</code>	€ <code>\texteuro</code>
⟨ <code>\textlangle</code>	‰ <code>\textpertenthousand</code>	× <code>\texttimes</code>
— <code>\textminus</code>	¶ <code>\textpilcrow</code>	÷ <code>\textdiv</code>
⟩ <code>\textrangle</code>	B <code>\textbaht</code>	
℥ <code>\textmho</code>	Nº <code>\textnumero</code>	

¹ 표 4.6의 더 짧은 제어 문자열을 사용해 입력할 수도 있습니다.

<code>1 shelfful 대신 shelf\textcompwordmark ful</code> <code>\\ % suppress `ff' ligature</code> <code>2 b\{g, b\{\textcompwordmark}g,</code> <code>B\{\textcapitalcompwordmark}G</code>	shelfful 대신 shelfful b̂g, b̃g, B̂G
---	---------------------------------------

이 세 가지 제어 문자열이 출력하는 기호의 높이는 각각 1ex, 대문자의 세로 길이 정도, 어센더 정도입니다. 악센트가 조판되는 높이를 조정해야 할 때 이 제어 문자열을 사용할 수 있습니다.

참고 | 유니코드 엔진에서는 악센트 기호의 구현 방식 때문에, `\textcompwordmark` 등을 두 문자 사이에 악센트를 표시하는 데에는 사용할 수 없습니다. 두 문자가 합자로 표시되지 않도록 하는 데에는 사용할 수 있습니다.

유니코드 엔진에서의 기호

유니코드 엔진에서는 표 4.7와 표 4.8의 기호 중 일부가 다른 기호와 같은 것으로 취급되며, 일부는 출력되지 않습니다.

다음 제어 문자열에 대응하는 기호는 유니코드 엔진의 기본 글꼴에서 출력되지 않습니다. 문서의 기본 글꼴을 바꾸는 경우 출력되지 않는 기호는 달라질 수 있습니다.

`\texttildelow` `\textgravedbl` `\textbigcircle` `\textguarani`
`\textinterrobangdown`

또, 다음 제어 문자열은 화살표 오른쪽에 표시된 제어 문자열과 동일하게 작동합니다. 이들은 레거시 엔진에서 작성된 문서와의 코드 호환을 위해 유니코드 엔진에서도 제공되는 것입니다.

<code>\capitalgrave</code>	→	<code>\`</code>
<code>\capitalacute</code>	→	<code>\'</code>
<code>\capitalcircumflex</code>	→	<code>\^</code>
<code>\capitaltilde</code>	→	<code>\~</code>
<code>\capitaldieresis</code>	→	<code>\"</code>
<code>\capitalhungarumlaut</code>	→	<code>\H</code>
<code>\capitalring</code>	→	<code>\r</code>
<code>\capitalcaron</code>	→	<code>\v</code>
<code>\capitalbreve</code>	→	<code>\u</code>
<code>\capitalmacron</code>	→	<code>\=</code>
<code>\capitaldotaccent</code>	→	<code>\.</code>
<code>\capitalcedilla</code>	→	<code>\c</code>
<code>\capitalogonek</code>	→	<code>\k</code>

```

\capitaltie, \newtie, \capitalnewtie
      → \t
\textcapitalcompwordmark, \textascendercompwordmark
      → \textcompwordmark
\textpilcrow      → \textparagraph

```

이상의 제어 문자열이 레거시 엔진에서와 동일한 기호를 출력하도록 하려면 다음 선언을 전처리부에 해 줍니다. $\langle csname \rangle$ 에는 문제가 되는 제어 문자열을 입력합니다.

```

1 \UndeclareTextCommand{<csname>}{TU}
2 \DeclareTextSymbolDefault{<csname>}{TS1}

```

4.3.3 문자 코드로 기호 직접 입력하기

TeX에서 사용할 수 있는 모든 문자에는 문자 코드라고 불리는 번호가 붙어 있으며, 이 번호를 사용해서 기호를 직접 입력할 수도 있습니다. 이때에는 `\symbol` 제어 문자열을 사용합니다.

`\symbol{<character code>}`

$\langle character code \rangle$: 입력하려는 기호의 문자 코드

유니코드 엔진에서는 유니코드에 기반하여 문자 코드가 부여되어 있습니다. 출력하려는 문자의 유니코드 코드 포인트를 `\symbol`의 인자로 지정하면 됩니다. 문자 코드를 16진수로 입력할 때에는 문자 코드 앞에 큰따옴표 "를 입력하면 됩니다.

1 별 모양 기호: `\symbol{"2605} \\ % 16진수`
 2 하트 모양 기호: `\symbol{9829} % 10진수`

별 모양 기호: ★
 하트 모양 기호: ♥

레거시 엔진에서는 ‘글꼴 인코딩’에 따라 문자 코드에 대응되는 기호가 달라집니다. 글꼴 인코딩은 여기에서 설명하기에 내용이 복잡하므로, [32]를 참조하시기 바랍니다.

연습문제

4.1. 다음 문장을 조판하여라.

‘*Cu vi loḡas en Aŭstralio aŭ Usono?*’는 에스페란토로 상대가 호주 혹은 미국에 살고 있는지를 묻는 표현입니다.

4.2. 다음 문장을 조판하여라.

오스트리아에는 유명한 물리학자 슈뢰딩거(Schrödinger)가 있습니다. 미국의 저명한 수학자로는 괴델(Gödel)이 있으며, 스웨덴에는 물리학자 옹스트룀(Ångström), 덴마크에는 물리학자 외르스테드(Ørsted)가 있습니다.

4.3. 다음 문장을 조판하여라.

‘Those’의 ‘th’는 ‘fifth’의 ‘th’와 달리 발음기호 ‘ð’로 표기하며, 한국어의 이응 받침, 영어의 ‘ng’의 발음은 ‘ŋ’로 나타냅니다.

4.4. 다음 문장을 조판하여라.

“달의 향기가 나.” 너의 말이 우리의 숨을 뿔었다.

4.5. 다음 문장을 조판하여라.

계절이 지나가는 하늘에는 — 감히 다 마실 수 없을 만큼의 — 파랑이 가득했다.

4.6. 다음 문장을 조판하여라.

“As you know that...” Maria starts to talk with very worried voice.

4.7. 다음 문장을 조판하여라.

““OX 문제는 맞으면 1점, 틀리면 -1점 처리입니다.” 라고 조교님이 그러시던데?” 그 말을 들은 철수는 등골이 서늘해졌다.

4.8. 다음 문장을 조판하여라.

‘¡Hola!’는 가장 유명한 스페인어 인사말입니다.

4.9. 다음 문장을 조판하여라.

$1\text{B}=36.71\text{W} \rightarrow 10000\text{W}=272.41\text{B}$

4.10. 다음 문장을 조판하여라.

초전도체는 특정 온도와 압력 조건 하에 전기 저항이 0Ω 이 되는 물질을 뜻합니다.

제 5 장

목차 구성하기

형식을 갖춘 문서는 표지, 목차, 머리말, 부록, 참고 문헌 목록, 색인, 발문 등 여러 요소를 적절한 순서로 갖추고 있으며, 본문의 내용은 장과 절로 나뉘어 있습니다. 이들은 문서의 구조를 만들고 목차를 이루며, 이번 절에서는 이를 만드는 방법을 알아봅니다.

5.1 문서의 3단 구성

문서는 중심이 되는 내용이 오는 부분을 가운데로 하여 ‘처음-가운데-마지막’의 세 부분으로 나눌 수 있으며, 각 문서 요소는 세 부분 중 한 곳에 적당한 순서로 나타납니다. 또, 본문과 부록은 편·장·절 등으로 나누어 있어 독자가 내용의 흐름을 쉽게 파악할 수 있습니다.

5.1.1 클래스에 따른 문서의 구성 요소

article이나 report 클래스로 작성하는 문서는 처음 부분과 끝 부분이 비교적 간단하게 구성됩니다. 처음 부분에는 표지와 초록, 목차 정도만 있으며, 끝 부분에는 문서의 성격이나 내용에 따라 참고 문헌 목록과 색인이 있습니다.

book 클래스로 작성하는 문서는 하나의 온전한 책을 이루도록 몇 가지 요소를 더 포함합니다. 처음 부분은 표지로 시작하여, 초록 대신 서문이나 머릿말, 책의 소개와 일러두기, 감사의 말 등이 이어지며 목차로 끝납니다. 마지막 부분은 참고 문헌 목록과 색인 외에 저자의 집필 후기 등이 쓰이기도 합니다. 책의 저작권 정보를 적은 판권지도 표지 안쪽이나 마지막 페이지에 작성됩니다.

클래스에 따른 문서의 구성을 그림 5.1에 나타냈습니다. 각 요소를 작성할 때 사용하는 \LaTeX 명령어도 함께 나타냈습니다. 이들은 이번 장에서 모두 다룰 것입니다.

	article 및 report 클래스	book 클래스
처음	표지: \maketitle 또는 titlepage 환경	
	초록: abstract 환경	감사의 말, 서문 등
	목차: \tableofcontents, \listoftables, \listoffigures	
가운데	본문	
	부록: \appendix	
마지막	추가 정보, 발문 등	
	참고 문헌 목록: thebibliography 환경	
	색인: theindex 환경	

| 그림 5.1 | 클래스에 따른 문서 구성

5.1.2 book 클래스의 추가 명령어

book 클래스는 앞의 두 클래스와 다르게 \frontmatter, \mainmatter, \backmatter 제어 문자열을 제공합니다. 이들은 각각 처음, 가운데, 마지막 부분이 시작됨을 선언하므로, 각 부분의 첫머리에 이들을 입력해 주면 됩니다. 각각의 역할은 다음과 같습니다.

- \frontmatter: 페이지 번호를 1로 초기화하며 소문자 로마 숫자로 바꾸고, 장 표제의 번호가 출력되지 않도록 합니다.
- \mainmatter: 페이지 번호를 1로 초기화하며 아라비아 숫자로 바꾸고, 장 표제의 번호가 출력되도록 합니다.
- \backmatter: 페이지 번호를 초기화하지는 않으며, 장 표제의 번호가 출력되지 않도록 합니다.

위의 세 가지 선언을 통해 책의 중심 부분인 가운데 부분이 구조적으로 처음이나 마지막 부분과 구분될 수 있습니다. 그림 5.1의 요소 중 감사의 말, 서문, 추가 정보, 발문 등은 \mainmatter 선언 전이나 \backmatter 선언 후에 \chapter를 사용해 작성합니다.

5.2 처음 부분

5.2.1 표지

모든 문서는 문서의 정보를 담은 표지로 시작합니다. \LaTeX 표준 클래스의 표지는 문서의 제목, 저자, 날짜로 구성되며, 이들을 각각 `\title`, `\author`, `\date` 제어 문자열의 인자로 지정합니다. 세 가지 정보를 지정한 후 document 환경 안에서 `\maketitle`을 입력하면 문서의 표지가 생성됩니다.

다음은 article 클래스에서의 예시입니다. article 클래스에서는 첫 페이지의 윗부분에 표지가 삽입된 뒤 본문의 글이 이어서 조판됩니다. report와 book 클래스에서는 표지가 첫 페이지 전체를 차지하고 본문은 다음 페이지에서부터 조판됩니다.

```
1 \documentclass{article}
2 ...
3 \title{\LaTeX: 문서 작성 도구}
4 \author{강우현\thanks{KAIST 수리과학과 21학번}}
5 \date{\today}
6 \begin{document}
7 \maketitle
8 \LaTeX 은 수학을 비롯해 자연과학과 공학의 많은 분야에서 문서를 작성하는 도구로 ...
9 \end{document}
```

\LaTeX : 문서 작성 도구

강우현*

2024년 6월 30일

\LaTeX 은 수학을 비롯해 자연과학과 공학의 많은 분야에서 문서를 작성하는 도구로 널리 사용되고 있습니다. 이 도구의 특징과 장단점을 논의하고, ...

*KAIST 수리과학과 21학번

`\maketitle`을 사용할 때, `\title`을 지정하지 않았으면 컴파일 과정에서 오류가 발생하고, `\author`를 지정하지 않으면 경고 메시지를 띄웁니다. `\date`는 필수로 지정하지 않아도 되며, 이때에는 소스 파일을 컴파일 하는 시점의 날짜가 입력됩니다.

여러 명의 저자를 입력할 때에는 `\and`를 각 저자 이름 사이에 입력해 각 저자를 구분합니다. 또, 제목이나 저자, 날짜에 관한 추가 정보를 각주를 통해 제공할 때에는 `\thanks` 제어 문자열을 사용합니다. 각주의 내용을 `\thanks`의 인자로 지정하면 됩니다.

직접 표지 만들기

`\maketitle`이 만드는 표지가 마음에 들지 않는다면 `titlepage` 환경을 사용하여 문서의 표지를 직접 만들 수 있습니다. 표지에 입력하고 싶은 내용을 환경 안에 입력하고 컴파일하면 그 내용만으로 구성된 한 페이지가 만들어집니다. 이 환경 바깥의 내용은 표지와 구분된 페이지에 조판됩니다. `\rule`, `\vspace`, `\hspace`와 제6장의 명령어를 사용해 문서의 내용을 잘 설명하는 표지를 만들어 보십시오.

5.2.2 초록

많은 논문은 표지 다음에 본문의 내용을 요약한 초록을 작성해 둡니다. `abstract` 환경 (`book` 클래스에서는 사용할 수 없음)을 사용하면 초록을 작성할 수 있습니다.

```
1 \begin{abstract}
2   이 문서는 \LaTeX 을 사용하여 문서를 만드는 방
   법을 설명합니다. 수식에 관련된 명령어뿐 아니라
   문서의 서식을 지정하는 명령어도 다룹니다.
3 \end{abstract}
4 이 문서를 읽기 전, 온라인 \LaTeX 플랫폼에 가입
   하거나 컴퓨터에 \TeX 배포판을 설치해 두기를 권장
   합니다.
```

요약

이 문서는 I^AT_EX을 사용하여 문서를 만드는 방법을 설명합니다. 수식에 관련된 명령어뿐 아니라 문서의 서식을 지정하는 명령어도 다룹니다.

이 문서를 읽기 전, 온라인 I^AT_EX 플랫폼에 가입하거나 컴퓨터에 T_EX 배포판을 설치해 두기를 권장합니다.

위 예시와 같이, `abstract` 환경 안에 내용을 작성하면 문단 위에 ‘Abstract’ 또는 ‘요약’이라는 제목이 붙습니다. 또, 문단 양쪽에 약간의 여백이 추가되며 기본 글꼴보다 작은 크기로 조판됩니다.

5.2.3 목차

목차는 `\tableofcontents` 제어 문자열을 입력하여 삽입할 수 있습니다. 5.3.1절에서 설명하는 `\part`, `\chapter`, `\section` 등으로 작성된 표제를 모아 목차를 만들어 줍니

다. \LaTeX 의 목차는 컴파일 후 생성되는 보조 파일(.aux 파일)을 바탕으로 만들어집니다. 따라서 표제의 내용이나 그 페이지 번호에 변동 사항이 생기면 컴파일 과정을 두 번 거쳐야 올바른 목차를 얻게 됩니다.

다음 예시는 article 클래스에서의 목차 생성 예시입니다.

```
1 \tableofcontents
2
3 \section{기본 문법} ...
4 \subsection{패키지} ...
5 \subsection{각종 서식} ...
6
7 \section{수식 문법} ...
8 \subsection{수식 기호} ...
9 \subsection{행렬} ...
```

차 례

1 기본 문법	1
1.1 클래스와 패키지	2
1.2 각종 서식	4
2 수식 문법	7
2.1 수식 기호	10
2.2 행렬	15

캡션
→ 193쪽 표 목차와 그림 목차도 삽입할 수 있습니다. 이 목차의 내용은 표와 그림의 캡션들로 구성됩니다. \listoftables 제어 문자열을 입력하면 표 목차가, \listoffigures 제어 문자열을 입력하면 그림 목차가 만들어집니다. \tableofcontents와 마찬가지로, 두 목차도 내용에 변동 사항이 생기면 컴파일을 두 번 해야 올바른 결과를 얻습니다.

5.3 가운데 부분

가운데 부분에는 문서의 주된 내용이 담겨 있으며, 본문과 부록으로 나눌 수 있습니다. 이번 절에서는 이 부분에서 사용할 수 있는 목차 구성 요소를 알아봅니다.

5.3.1 표제

\LaTeX 에서는 문서의 내용을 분류하고, 편(part), 장(chapter), 절(section) 등 총 7개의 수준으로 구분하여 표제 heading를 붙일 수 있습니다. 이 7개 수준의 장절 표제를 생성하기 위한 제어 문자열을 표 5.1에 정리하였습니다.

절 표제

article 클래스에서 가장 기본적인 표제는 절 section입니다. 절 표제를 만들 때 사용하는 \section(*) 제어 문자열은 다음과 같이 사용합니다. 나머지 6개 수준 표제도 동일한 방법으로 사용합니다.

| 표 5.1 | 장절 표제 명령어

표제 수준	제어 문자열	조판 형식 및 특징
편	\part	하위 수준의 번호에 영향을 주지 않습니다. article 클래스에서는 페이지 나눔 없이, 앞뒤 내용과 분리된 줄에 표제가 조판됩니다. report나 book 클래스에서는 편 번호와 표제만을 조판하는 페이지가 만들어집니다.
장	\chapter	article 클래스에서는 사용할 수 없습니다. 장 번호와 표제로 시작하는 새로운 페이지가 만들어지며, 표제 아래에 본문이 조판됩니다.
절	\section \subsection \subsubsection	앞뒤 내용과 분리된 별도의 줄에 표제가 조판됩니다. \part나 \chapter와는 달리, 표제로 시작하는 새로운 페이지를 만들지는 않습니다.
문단	\paragraph \subparagraph	새로운 줄에 표제가 조판되고, 줄 바꿈 없이 그 옆에 본문이 이어져서 조판됩니다. 문단에 제목을 단 듯한 결과를 얻습니다.

```
\section[<short title>]{<title>}
\section*{<title>}
```

<short title>: 목차, 머리글, 바닥글 등에서 사용할 표제
<title>: 표제

이 제어 문자열을 사용하면 표제 앞에 절 번호도 함께 표시됩니다. 인자로 지정한 표제는 목차와 머리글에도 사용됩니다. 이때 표제가 긴 경우 문서의 레이아웃이 흐트러질 수 있는데, 이때에는 목차와 머리글에서만 사용할 짧은 제목을 선택 인자로 지정해 주면 됩니다.

```
1 \section{기본 문법}
2 이번 절에서는 \LaTeX 에서 문서를 작성할 때 알아
  야 할 기본 문법을 설명합니다.
3
4 \subsection{클래스와 패키지}
5 모든 \LaTeX\ 문서는 그 클래스를 지정해 주어야 하
  며, 다양한 기능을 사용하기 위해서는 여러 패키지를
  불러와야 합니다.
```

1 기본 문법

이번 절에서는 L^AT_EX에서 문서를 작성할 때 알아야 할 기본 문법을 설명합니다.

1.1 클래스와 패키지

모든 L^AT_EX 문서는 그 클래스를 지정해 주어야 하며, 다양한 기능을 사용하기 위해서는 여러 패키지를 불러와야 합니다.

별표 붙은 제어 문자열을 사용하면 표제의 번호가 출력되지 않고, 목차나 머리글에도 영향을 주지 않습니다.

문단 표제

문단에 제목을 붙일 수도 있습니다. 이때에는 `\paragraph(*)`와 `\subparagraph(*)` 제어 문자열을 사용합니다. 절 표제와는 달리 문단 표제에는 번호가 표시되지 않습니다.

```
1 \subsection*{연습 문제}
2 \paragraph{연습문제 1} 군~\langle G \rangle, \langle H \rangle 사
   이에 군 준동형 \langle \varphi \colon G \rightarrow H \rangle가
   주어졌다고 가정하자. 다음을 증명하여라.
```

연습 문제

연습문제 1 군 G, H 사이에 군 준동형 $\varphi: G \rightarrow H$ 가 주어졌다고 가정하자. 다음을 증명하여라.

장 표제

book과 report 클래스에서는 절의 상위 수준 표제로 `\chapter`를 사용할 수 있습니다. `\chapter(*)`를 입력하면 새로운 페이지에 장 표제가 만들어지며, 페이지 위쪽에 큰 글씨로 장 번호와 표제가 조판됩니다.

```
1 \chapter{표와 그림}
2 이번 장에서는 \LaTeX 으로 표를 그리고, 외부 그림 파일을 문서에 삽입하는 방법을 알아봅니다.
3 \section{표 작성하기}
4 표를 그리는 환경에는 \verb|tabular|~환경, \verb|tabular*|~환경, \verb|array|~환경이
   있습니다. 처음의 두 환경은 텍스트 모드에서, 세 번째 환경은 수식 모드에서 사용합니다.
```

제 4 장

표와 그림

이번 장에서는 \LaTeX 으로 표를 그리고, 외부 그림 파일을 문서에 삽입하는 방법을 알아봅니다.

4.1 표 작성하기

표를 그리는 환경에는 `tabular` 환경, `tabular*` 환경, `array` 환경이 있습니다. 처음의 두 환경은 텍스트 모드에서, 세 번째 환경은 수식 모드에서 사용합니다.

article 클래스에서는 \chapter를 사용할 수 없으므로, article 클래스로 작성한 문서들을 모으고 각 문서의 제목을 장 표제로 하여 book이나 report 클래스 문서를 만들 수도 있습니다.

편 표제

특수한 형태의 표제로 편part을 사용할 수 있습니다. \part는 최상위 수준의 표제로 \chapter나 \section보다 수준이 높지만, 편 번호는 장·절의 번호와 독립적으로 계산됩니다(이 책의 목차에서 이를 확인할 수 있습니다). 따라서, 새로운 편을 만들더라도 하위 수준인 장이나 절의 번호가 1부터 다시 시작하지 않습니다.

편 표제는 \part(*)로 만듭니다. article 클래스에서는 새로운 페이지가 만들어지지 않지만, book 클래스와 report 클래스에서는 편 표제만 조판되는 별도의 페이지가 만들어집니다.

5.3.2 부록

부가적인 내용은 부록appendix에 작성할 수 있습니다. L^AT_EX으로 부록을 만들려면 부록이 시작되는 부분에서 \appendix 제어 문자열을 사용해 ‘여기에서 부록이 시작됨’을 선언하면 됩니다. 선언 이후의 절(article 클래스) 또는 장(report 또는 book 클래스) 번호는 1로 초기화되며, A, B, C……와 같이 로마자로 표시됩니다.

article 클래스 문서에서 \appendix를 사용하면 다음과 같은 결과를 얻습니다.

```
1 \appendix
2 \section{\TeX\ 배포판 설치 방법}
3 인터넷 연결 없이 \LaTeX 을 사용하려면 컴퓨터에
  \TeX\ 배포판을 설치해야 합니다.
```

A T_EX 배포판 설치 방법

인터넷 연결 없이 L^AT_EX을 사용하려면 컴퓨터에 T_EX 배포판을 설치해야 합니다.

5.3.3 주석

글을 작성하던 중 부연 설명이 필요하면 주석을 덧붙이기도 합니다. 주석은 목차와는 무관하지만, 본문을 작성할 때 요긴하게 쓰입니다. L^AT_EX에서는 각주와 여백글 기능을 제공하며, 패키지를 사용하면 미주도 작성할 수 있습니다.

각주

각주footnote는 글의 내용 중 짧게 첨언할 내용이 있을 때, 첨언할 부분에 숫자 등으로 표시를 한 후 페이지의 아래쪽에 설명을 덧붙이는 형식의 주석을 뜻합니다. L^AT_EX에서

주석은 `\footnote` 제어 문자열을 사용해 조판합니다. 각주를 덧붙이고 싶은 부분에 `\footnote`를 입력하고, 각주의 내용을 인자로 지정하면 됩니다.

`\footnote[⟨number⟩]{⟨text⟩}`

⟨number⟩: 각주의 번호

⟨text⟩: 각주의 내용

¹ 각주는 부연 설명`\footnote{특히, 중요도가 낮은 내용을 적는 것이 좋습니다.}`을 하는 데 사용합니다.

각주는 부연 설명¹을 하는 데 사용합니다.

¹특히, 중요도가 낮은 내용을 적는 것이 좋습니다.

`\footnote`는 선택 인자를 입력받습니다. 선택 인자로는 각주의 번호로 출력하고 싶은 값을 입력하면 됩니다. 각주의 번호는 원래 자동으로 계산되지만, 선택 인자를 지정한 경우에는 자동으로 계산되지 않습니다.

참고 | `tabular` 환경 등 표를 작성하는 환경에서는 각주 기능이 제대로 작동하지 않습니다.

여백글

\TeX 에서는 `\marginpar` 제어 문자열을 사용해 문서의 좌우 여백에 추가 메모를 적을 수 있습니다. 이를 여백글 `marginal note`이라고 부릅니다. 여백글을 사용하면 여백에 주석을 작성하는 ‘측주’ 기능을 만들 수 있습니다.

`\marginpar[⟨left note⟩]{⟨right note⟩}`

⟨left note⟩: 왼쪽 여백에 작성할 내용

⟨right note⟩: 오른쪽 여백에 작성할 내용

양면 편집을 하는 경우에는 `⟨left note⟩`에 작성된 내용은 안쪽 여백에, `⟨right note⟩`에 작성된 내용은 바깥쪽 여백에 조판됩니다. 또, 2단 편집 시에는 왼쪽 단은 왼쪽 여백에, 오른쪽 단은 오른쪽 여백에 내용이 조판됩니다. 여백글 첫째 줄의 세로 위치는 `\marginpar` 제어 문자열이 입력된 곳에 맞춰집니다.

¹ `\TeX\` 배포판은 TUG`\marginpar{\footnotesize TUG: \TeX\ Users Group}`에서 내려받을 수 있습니다. Windows에서는 `\TeX\ Live`를, macOS에서는 `Mac\TeX` 을 내려받으면 됩니다.

TUG: \TeX Users Group

\TeX 배포판은 TUG에서 내려받을 수 있습니다. Windows에서는 \TeX Live를, macOS에서는 `Mac\TeX`을 내려받으면 됩니다.

\marginpar 제어 문자열은 문단을 시작할 때 사용하면 여백글이 원하는 곳보다 위쪽에 조판될 수 있습니다. 또, 여백글의 첫 번째 단어는 분절 처리가 되지 않습니다. 따라서, 측주를 작성하는 데 사용하려는 경우 다음과 같이 문제를 해결한 \marginnote 제어 문자열을 정의해 \marginpar 제어 문자열 대신 사용하는 것이 편리합니다.

```
\newcommand*\marginnote[1]
{\leavevmode\marginpar{\hspace{0pt}\footnotesize #1}}
```

5.4 마지막 부분

문서의 마지막 부분에는 참고 문헌 목록과 색인이 있습니다. 이번 절에서는 이들을 수동으로 작성하는 방법을 알아봅니다. 이 책의 제 V 편에서는 BibTeX이나 biber, MakeIndex 나 xindy를 사용하여 참고 문헌 목록과 색인을 자동으로 생성하는 방법을 설명합니다. 작은 문서에서는 이번 절의 방법을 사용하여 수동으로 작성해도 되지만, 문서의 규모가 커진다면 제 V 편에서 설명하는 방법을 사용하는 것이 편리할 것입니다.

5.4.1 참고 문헌 목록

참고 문헌
→ 229쪽

참고 문헌 목록 bibliography은 thebibliography 환경을 사용하여 작성합니다. 환경 안에서는 문헌마다 \bibitem 제어 문자열을 입력한 후 그 문헌의 정보를 적절한 서식과 함께 입력해 줍니다.

```
\begin{thebibliography}{\langle longest label \rangle}
```

\langle longest labe \rangle: 목록의 레이블 중 가장 긴 것

```
\end{thebibliography}
```

```
\bibitem[\langle label \rangle]{\langle identifier \rangle}
```

\langle label \rangle: 항목 앞에 붙일 레이블

\langle identifier \rangle: 항목을 인용할 때 사용할 식별자

문헌의 각 항목 앞에는 레이블이 표시됩니다. 각 문헌의 정보를 입력할 때 \bibitem의 선택 인자를 지정하지 않으면 아라비아 숫자로 된 번호가 레이블로써 사용됩니다.

thebibliography 환경은 인자를 하나 입력받습니다. L^AT_EX은 이 인자의 길이를 측정하여 레이블이 왼쪽 여백에 맞도록 조판합니다. 다음의 예시처럼, 문헌의 레이블로 번호를 사용할 때에는 가장 큰 번호의 자리 수만큼 ‘9’를 thebibliography 환경의 인자로 지정하면 됩니다.

```

1 \begin{thebibliography}{9}
2   \bibitem{texbook} Donald E. Knuth.
   \textit{The \TeX book}.
   Addison-Wesley Professional, 1986.
3   \bibitem{latexdps} Leslie Lamport.
   \textit{\LaTeX: A Document
   Preparation System}. Addison-Wesley,
   second edition, 1994.
4 \end{thebibliography}

```

참고 문헌

- [1] Donald E. Knuth. *The T_EXbook*. Addison-Wesley Professional, 1986.
- [2] Leslie Lamport. *L^AT_EX: A Document Preparation System*. Addison-Wesley, second edition, 1994.

한편, `\bibitem`의 선택 인자를 사용하여 문헌의 레이블을 직접 지정해 줄 수도 있습니다. 이때에는 각 레이블의 조판 결과를 확인한 뒤, 길이를 비교하여 가장 긴 레이블을 `thebibliography` 환경의 인자로 지정하면 됩니다.

```

1 \begin{thebibliography}{Lam94}
2   \bibitem[Knu86]{texbook} Donald E.
   Knuth. \textit{The \TeX book}.
   Addison-Wesley Professional, 1986.
3   \bibitem[Lam94]{latexdps} Leslie
   Lamport. \textit{\LaTeX: A Document
   Preparation System}. Addison-Wesley,
   second edition, 1994.
4 \end{thebibliography}

```

참고 문헌

- [Knu86] Donald E. Knuth. *The T_EXbook*. Addison-Wesley Professional, 1986.
- [Lam94] Leslie Lamport. *L^AT_EX: A Document Preparation System*. Addison-Wesley, second edition, 1994.

문헌의 인용

목록에 작성한 문헌을 인용할 때에는 `\cite` 제어 문자열을 사용합니다.

`\cite[<additional info.>]{<identifier>}`

<additional info.>: 페이지 번호 등 인용하는 문헌의 추가 정보

<identifier>: 인용하려는 문헌의 식별자

이 제어 문자열의 인자로 참조하려는 문헌의 식별자를 입력하면 되며, 그에 대응하는 식별자가 *<identifier>* 인자로 지정된 `\bibitem`의 레이블이 출력됩니다.

1 자세한 내용은 `\cite{texbook}`을 참조하여라.

자세한 내용은 [Knu86]을 참조하여라.

5.4.2 색인

색인index은 theindex 환경 안에 작성합니다. 이 환경 안에 \item 제어 문자열을 사용하여 색인의 항목을 작성합니다. 하위 수준의 항목을 만들려는 경우에는 \subitem과 \subsubitem을 사용합니다. 항목과 항목 사이에 빈 공간을 삽입하려면 \indexspace를 입력하면 됩니다. 이 환경을 사용하여 작성된 색인은 2단으로 조판됩니다.

```

1 \begin{theindex}
2   \textbf{ㄱ}
3   \item 곱위상, 5
4   \item 기저, 3
5
6   \indexspace
7   \textbf{ㅁ}
8   \item 부분기저, 3
9
10  \indexspace
11  \textbf{ㅅ}
12  \item 순서위상, 4
13
14  \indexspace
15  \textbf{ㅇ}
16  \item 열린집합, 1
17  \item 위상, 1
18    \subitem 여유한위상, 2
19    \subitem 이산위상, 2
20    \subitem 자명위상, 2
21  \item 위상 공간, 1
22 \end{theindex}

```

찾아보기

ㄱ	ㅇ
곱위상, 5	열린집합, 1
기저, 3	위상, 1
	여유한위상, 2
ㅁ	이산위상, 2
부분기저, 3	자명위상, 2
	위상 공간, 1
ㅅ	
순서위상, 4	

제 6 장

여러 가지 서식

제4장과 제5장의 명령어만으로도 형식을 갖춘 문서를 작성할 수 있습니다. 하지만, 문서에 다양한 서식을 사용하면 더 효과적으로 내용을 독자에게 전달할 수 있습니다. 이번 장에서는 \LaTeX 에서 서식을 지정할 때 사용할 수 있는 명령어들을 알아봅니다. 이들은 크게 문단 단위의 서식과 문자 단위의 서식으로 구분됩니다.

6.1 다양한 형태의 문단

이번 장에서는 다양한 형태의 문단을 만드는 방법을 알아봅니다. \LaTeX 은 다양한 항목을 나열할 때 사용하는 목록 환경, 인용한 글을 앞뒤 문단과 구분할 때 사용하는 인용문·운문 환경을 제공합니다. 또, 글줄의 양 끝을 여백에 맞추는 대신 왼쪽, 가운데, 오른쪽에 맞추어 정렬하기 위한 명령어도 사용할 수 있습니다.

6.1.1 목록

목록은 많은 수의 주제가 나열될 때 효과적으로 사용할 수 있는 형태의 문단입니다. 항목 사이에 순서가 있거나 번호를 붙일 때에는 번호 목록을 만드는 것이 효과적이고, 항목 간의 순서가 중요하지 않다면 번호라는 불필요한 정보가 없는 구분점 목록을 만드는 것이 좋습니다. 용어를 제시하고 그에 대한 설명을 적어야 한다면 사전식 목록을 만드는 것이 좋습니다.

\LaTeX 에서는 위 세 가지 형태의 목록을 만들 수 있으며, 각각 `enumerate`, `itemize`, `description` 환경을 사용해 만듭니다. 목록 환경 안에서 각 항목은 `\item` 제어 문자열을 사용하여 구분합니다.

<pre> 1 \begin{enumerate} 2 \item \verb enumerate 환경입니다. 3 \item 이 환경은 번호 목록을 만듭니다. 4 \end{enumerate} </pre>	<ol style="list-style-type: none"> 1. <code>enumerate</code> 환경입니다. 2. 이 환경은 번호 목록을 만듭니다.
<pre> 1 \begin{itemize} 2 \item \verb itemize 환경입니다. 3 \item 이 환경은 구분점 목록을 만듭니다. 4 \end{itemize} </pre>	<ul style="list-style-type: none"> • <code>itemize</code> 환경입니다. • 이 환경은 구분점 목록을 만듭니다.
<pre> 1 \begin{description} 2 \item[소개] \verb description 환경 3 \item[설명] 사전식 목록을 만듭니다. 4 \end{description} </pre>	<p>소개 <code>description</code> 환경</p> <p>설명 사전식 목록을 만듭니다.</p>

예시에서 볼 수 있듯, 사전식 목록에서는 키워드를 `\item`의 선택 인자로 지정합니다. 번호 목록이나 구분점 목록에서도 `\item`에 선택 인자를 지정할 수 있으며, 이때에는 번호나 구분점 대신 해당 내용이 항목의 구분 기호로 조판됩니다. 번호 목록에서 `\item`에 선택 인자를 지정한 경우, 해당 항목은 번호 계산에 영향을 주지 않습니다.

<pre> 1 \begin{enumerate} 2 \item 첫째 항목입니다. 3 \item[(1'\)] 기호를 바꾸어 보았습니다. 4 \item 둘째 항목입니다. 5 \end{enumerate} </pre>	<ol style="list-style-type: none"> 1. 첫째 항목입니다. 1'. 기호를 바꾸어 보았습니다. 2. 둘째 항목입니다.
<pre> 1 \begin{itemize} 2 \item 첫째 항목입니다. 3 \item[(\circ)] 기호를 바꾸어 보았습니다. 4 \item 둘째 항목입니다. 5 \end{itemize} </pre>	<ul style="list-style-type: none"> • 첫째 항목입니다. ○ 기호를 바꾸어 보았습니다. • 둘째 항목입니다.

세 가지 환경을 사용하여 조판한 목록의 형태가 마음에 들지 않는다면 `enumerate` 패키지[9]와 `enumitem` 패키지[4]를 사용해 모습을 변경할 수 있습니다.

목록 환경의 중첩

목록 환경 안에서 목록 환경을 다시 사용하면 하위 단계의 목록을 만들 수 있습니다. 목록 환경의 종류를 교차하여 중첩할 수도 있으며, `enumerate` 및 `itemize` 환경은 4단계 이내에서, `description` 환경은 6단계 이내에서 사용할 수 있습니다.

```

1 \begin{itemize}
2   \item 남은 작업
3   \begin{enumerate}
4     \item 1장, 4장, 10장 재작성
5     \item 16장, 17장, 20장, 21장 재작성
6     \item 참고 문헌 정보 검토
7     \item 색인 항목 검사
8   \end{enumerate}
9   \item 부분별 고민 중인 사항
10  \begin{description}
11    \item[제III편] 수식 번호 기능을 설명할 위
        치: 9장 vs. 10장
12    \item[제VI편] 정수 값과 길이 값 문법을 설
        명할 위치: 본문 vs. 부록
13  \end{description}
14 \end{itemize}

```

• 남은 작업

1. 1장, 4장, 10장 재작성
2. 16장, 17장, 20장, 21장 재작성
3. 참고 문헌 정보 검토
4. 색인 항목 검사

• 부분별 고민 중인 사항

제III편 수식 번호 기능을 설명할 위
치: 9장 vs. 10장

제VI편 정수 값과 길이 값 문법을 설
명할 위치: 본문 vs. 부록

6.1.2 운문과 인용문

LaTeX은 목록을 작성하는 환경 외에도, 운문이나 인용문을 적절한 서식으로 작성하기 위한 환경을 제공합니다. 이 환경을 사용하면 운문이나 인용문을 앞뒤의 글과 구분하기 위해 환경 상하좌우에 약간의 여백이 추가됩니다.

운문

시나 노래 가사 등 운문을 작성할 때에는 `verse` 환경을 사용합니다. 각 행은 줄바꿈 할 때처럼 `\\`로 구분하며, 문단을 나누면 새로운 연이 만들어집니다.

```

1 아래는 \verb|verse| 환경을 사용한 예시입니다.
2 \begin{verse}
3   \verb|\\|로 행을 구분하고 \\
4   문단을 나누어 \\ 연을 구분합니다.
5
6   한 행의 내용이 길면
7   둘째 줄부터는 들여쓰기 처리됩니다.
8 \end{verse}

```

아래는 `verse` 환경을 사용한 예시입니다.

```

\\로 행을 구분하고
문단을 나누어
연을 구분합니다.

한 행의 내용이 길면 둘째 줄부
터는 들여쓰기 처리됩니다.

```

위 예시처럼 글줄의 길이가 한 행의 내용보다 짧아 행 안에서 줄바꿈이 일어나는 경우, 두 번째 줄부터는 왼쪽에 여백이 추가됩니다.

인용문

인용문은 `quotation` 및 `quote` 환경을 사용해 작성합니다. 전자는 각 문단 첫 줄을 들여쓰기 처리하며, 후자는 들여쓰기 처리를 하지 않습니다.

<pre> 1 아래는 \verb quotation 환경을 사용한 예시입니 다. 2 \begin{quotation} 3 여러 문단에 걸친 긴 내용을 인용할 때에 이 환경을 사용합니다. \par 4 각 문단 첫머리는 들여쓰기 처리됩니다. 5 \end{quotation} </pre>	<p>아래는 <code>quotation</code> 환경을 사용한 예시입니다.</p> <p>여러 문단에 걸친 긴 내용을 인용할 때에 이 환경을 사용합니다.</p> <p>각 문단 첫머리는 들여쓰기 처리됩니다.</p>
<pre> 1 아래는 \verb quote 환경을 사용한 예시입니다. 2 \begin{quote} 3 한 문단 이내의 짧은 내용이나, 여러 사람의 인용 문을 열거할 때에 이 환경을 사용합니다. \par 4 각 문단 첫머리를 들여쓰지는 않지만, 문단 사이에 약간의 공백이 있습니다. 5 \end{quote} </pre>	<p>아래는 <code>quote</code> 환경을 사용한 예시입니다.</p> <p>한 문단 이내의 짧은 내용이나, 여러 사람의 인용문을 열거할 때에 이 환경을 사용합니다.</p> <p>각 문단 첫머리를 들여쓰지는 않지만, 문단 사이에 약간의 공백이 있습니다.</p>

6.1.3 문단 정렬 바꾸기

\TeX 은 문서를 이루는 각 글줄의 양 끝을 가지런하게 정렬합니다. 하지만 제목, 표제, 이름, 부가 정보 등을 작성하거나 강조가 필요할 때에는 글줄을 한쪽에만 맞추어 정렬하기도 합니다.

\LaTeX 에서는 `flushleft`, `center`, `flushright` 환경을 사용해 문단의 정렬 방식을 바꿀 수 있습니다. 각 환경은 왼쪽, 가운데, 오른쪽에 맞추어 글줄을 정렬하며, 환경의 위아래에 약간의 여백을 만듭니다. 다음 예시는 \LaTeX 의 `\maketitle` 제어 문자열 없이 정렬 환경을 사용해 문서 첫머리를 만들어 본 것입니다.

```

1 \begin{center} \Large 한국어 문서에서의 \LaTeX\ 사용 \end{center}
2 \begin{flushright} KAIST 수학기초연구회 강우현 \end{flushright} \par
3 문서 작성 도구인 \LaTeX 은 한국어 문서에도 유용하게 사용할 수 있습니다. 영어권 국가에서 만들어진 \LaTeX 으로 한국어 문서에 적합한 서식을 얻는 방법을 알아봅니다.

```

한국어 문서에서의 L^AT_EX 사용

KAIST 수확문제연구회 강우현

문서 작성 도구인 L^AT_EX은 한국어 문서에도 유용하게 사용할 수 있습니다. 영어권 국가에서 만들어진 L^AT_EX으로 한국어 문서에 적합한 서식을 얻는 방법을 알아봅니다.

환경 대신 `\raggedright`, `\centering`, `\raggedleft` 제어 문자열을 사용할 수도 있습니다. 이 제어 문자열은 선언형 제어 문자열로, 본문에서 이를 직접 사용하기보다는 특정 환경 안에서 정렬을 바꾸고자 할 때 사용하는 것이 좋습니다.

```
1 \begin{itemize} \centering
2   \item 이렇게 하면
3   \item 가운데 정렬된 목록을 얻습니다.
4 \end{itemize}
```

- 이렇게 하면
- 가운데 정렬된 목록을 얻습니다.

참고 | 문단 정렬을 바꾼 후 현재 문단이 끝나지 않으면 문단 정렬이 변경되지 않습니다. 따라서

```
{\centering This is a sample sentence.}
```

에서는 `\centering`이 무효하며, `{\centering ... sentence.\par}`와 같이 입력하여 영역 안에서 문단을 끝내 주어야 합니다.

ragged2e 패키지를 사용한 문단 정렬

T_EX은 필요한 경우 글줄 끝에 오는 긴 단어를 중간에서 끊도록 합니다. 영어 등 유럽 언어에서는 이때 분철 처리도 합니다. 하지만 앞서 설명한 명령어를 사용하여 문단의 정렬 방식을 바꾸면 단어 중간에서 줄바꿈이 일어나지 않습니다. 따라서, 경우에 따라 지나치게 짧은 글줄이 만들어져 보기 좋지 않은 결과를 얻을 수 있습니다.

마틴 슈뢰더(Martin Schröder)가 작성한 `ragged2e` 패키지[45]의 명령어를 사용하면 문단의 정렬하지 않은 쪽 글줄이 ‘덜 빼돌빼돌’해집니다. L^AT_EX의 기본 명령어 대신

`\RaggedRight`, `\Centering`, `\RaggedLeft` 제어 문자열과
`FlushLeft`, `Center`, `FlushRight` 환경

을 사용하면 됩니다. 또, 양쪽 정렬로 되돌리는 `\justifying` 제어 문자열도 사용할 수 있습니다. 다음은 두 명령어를 사용했을 때의 왼쪽 정렬 결과를 비교한 것입니다.

```

1 \begin{flushleft}
2   아주긴단어를 ... 비교해보십시오.
3 \end{flushleft}
4 \begin{FlushLeft}
5   아주긴단어를 ... 비교해보십시오.
6 \end{FlushLeft}

```

아주긴단어를 여러개 작성한뒤
왼쪽정렬해 보았습니다. 위아래 문단이
조판된결과를 비교해보십시오.

아주긴단어를 여러개 작성한뒤 왼쪽정렬
해 보았습니다. 위아래 문단이 조판된결
과를 비교해보십시오.

6.2 문자의 서식

이번 절에서는 문자에 서식을 지정하는 방법을 알아봅니다. 중요한 부분에 강조 표시를 하거나 글꼴을 바꾸는 방법, 첨자를 넣는 방법을 알아봅니다. 서식을 지정하는 명령어는 선언형과 인자형 두 가지로 구분할 수 있습니다. 선언형은 ‘여기서부터 서식을 사용함’을 선언하며, 인자형은 ‘인자로 지정된 부분에 서식을 적용’합니다.

6.2.1 특정 부분 강조하기

특정 부분을 강조할 때에는 선언형인 `\em` 또는 인자형인 `\emph` 제어 문자열을 사용합니다. 강조 서식이 지정되면 글자가 이탤릭체로 조판됩니다. 강조 서식이 지정된 상태에서 강조 서식을 한 번 더 적용하면 원래의 글꼴로 조판됩니다.

```

1 The word {\em contento} means 'happy'
  in Spanish.
2 The word \emph{triste} means 'sad' in
  Spanish.

```

The word *contento* means ‘happy’ in Spanish. The word *triste* means ‘sad’ in Spanish.

```

1 \em If you emphasize the text that is
  \emph{already being emphasized}, then
  it will become upright.

```

If you emphasize the text that is already being emphasized, then it will become upright.

`\underline` 제어 문자열을 사용하면 인자로 지정한 내용에 밑줄을 쳐 내용을 강조할 수 있습니다. 이때 밑줄을 그은 내용은 줄바꿈 처리가 되지 않으므로 주의해야 합니다.

ktex 패키지를 불러온 경우 `\dotemph` 제어 문자열을 사용할 수 있으며, 이는 인자로 지정한 문자열을 **드러냄표**를 사용해 강조합니다.

```

1 한국어 글에서는 주로 \underline{밑줄}이나
  \dotemph{드러냄표}를 사용해 내용을 강조한다.

```

한국어 글에서는 주로 밑줄이나 **드러냄표**를 사용해 내용을 강조한다.

6.2.2 글꼴의 모양 바꾸기

\LaTeX 2 ϵ 의 글꼴 모양은 ‘패밀리’, ‘시리즈’, ‘셰이프’ 세 가지 속성에 따라 지정합니다.

- 패밀리 family 글꼴의 전체적인 모양을 가리킵니다. ‘타입페이스(typeface)’라는 용어로도 부릅니다. 세리프(serif), 산세리프(sans-serif), 고정폭(monospaced) 세 가지 패밀리를 사용할 수 있습니다.
- 시리즈 series 획의 두께를 가리킵니다. ‘웨이트(weight)’라는 용어로도 부릅니다. 미디엄(medium)과 볼드(bold) 시리즈를 사용할 수 있습니다.
- 셰이프 shape 글꼴에 변형을 준 형태를 뜻합니다. 정체(upright), 이탤릭체(italic), 기울임꼴(slanted), 작은 대문자(small caps) 네 가지 셰이프를 사용할 수 있습니다.

글꼴을 변경하는 제어 문자열들을 표 6.1에 정리하였습니다. 위쪽의 제어 문자열은 인자형, 아래쪽의 제어 문자열은 선언형입니다. 기본 글꼴을 지정하는 `\textnormal`이 나 `\normalfont`는 세리프 패밀리, 미디엄 시리즈, 정체 셰이프를 사용합니다.

<code>\textbf{글꼴}</code> 을 바꿈으로써 <code>{\sffamily \itshape}</code> 강조 효과를 만들 수 있습니다.	글꼴 을 바꿈으로써 강조 효과 를 만들 수 있습니다.
<code>You can \textbf{emphasize} text by changing {\sffamily\itshape fonts}.</code>	You can emphasize text by changing <i>fonts</i> .

위 예시에서 확인할 수 있듯, 세 가지 속성은 서로 교차하여 지정할 수도 있습니다. 또, 셰이프 중 ‘작은 대문자’ 스타일은 다른 스타일과도 교차하여 사용할 수 있습니다. 다만, 글꼴에 따라 특정 조합을 사용하지 못할 수 있습니다. 예를 들어, \LaTeX 의 기본 산세리프 글꼴은 볼드 이탤릭체를 제공하지 않습니다.

글꼴을 변경할 때에는 표 6.1의 명령어를 바로 사용하는 것보다, 특정 상황에서 적절한 글꼴을 사용하는 매크로를 정의해 두고 document 환경에서는 정의한 매크로를 사용하는 것이 좋습니다.

매크로
→ 259쪽

이전 버전 명령어에 대한 조언

\LaTeX 표준 클래스에서는 `\it`, `\rm`, `\bf`, `\sf`, `\tt`, `\sl`, `\sc` 등의 제어 문자열로도 글꼴을 바꿀 수 있습니다. 이들은 \LaTeX 2.09나 플레인 \TeX 의 선언형 글꼴 명령어입니다.

\LaTeX 2 ϵ 에서는 공식적으로 이 명령어들이 지원되지 않으므로, 패키지나 클래스에 따라 전혀 다른 동작을 할 수도 있습니다. \LaTeX 표준 클래스에서는 하위 호환성을 위해

표 6.1 | 텍스트 모드의 글꼴

구분	종류	제어 문자열	예시
기본 글꼴	—	<code>\textnormal</code>	기본 글꼴입니다.
		<code>\normalfont</code>	This is the default font.
패밀리	세리프	<code>\textrm</code>	세리프 패밀리 글꼴입니다.
		<code>\rmfamily</code>	Current Font Is in Roman Family.
	산세리프	<code>\textsf</code>	산세리프 패밀리 글꼴입니다.
		<code>\sffamily</code>	Current Font Is in Sans-serif Family.
	고정폭	<code>\texttt</code>	고정폭 패밀리 글꼴입니다.
		<code>\ttfamily</code>	Current Font Is in Monospaced Family.
시리즈	미디엄	<code>\textmd</code>	미디엄 시리즈 글꼴입니다.
		<code>\mdseries</code>	Current Font Belongs to the Medium Series.
	볼드	<code>\textbf</code>	볼드 시리즈 글꼴입니다.
		<code>\bfseries</code>	Current Font Belongs to the Bold Series.
셰이프	정체	<code>\textup</code>	정체 셰이프 글꼴입니다.
		<code>\upshape</code>	Current Font Is in Upright Shape.
	이탤릭체	<code>\textit</code>	<i>이탤릭체 셰이프 글꼴입니다.</i> ¹
		<code>\itshape</code>	<i>Current Font Is in Italic Shape.</i>
	기울임꼴	<code>\textsl</code>	<i>기울임꼴 셰이프 글꼴입니다.</i>
		<code>\slshape</code>	<i>Current Font Is in Slanted Shape.</i>
	작은 대문자	<code>\textsc</code>	작은 대문자 셰이프 글꼴입니다. ²
		<code>\scshape</code>	CURRENT FONT IS IN SMALL-CAPS SHAPE.

¹ 많은 한글 글꼴은 이탤릭체를 지원하지 않으므로, 정체나 기울임꼴이 대신 사용됩니다.² 한글에는 대·소문자의 구분이 없으므로, 정체와 특별한 차이 없이 조판됩니다.

이 제어 문자열을 제공할 뿐입니다. 또, 이 명령어를 사용하면 글꼴의 세 가지 속성을 교차해 지정할 수 없습니다. 따라서 표 6.1의 제어 문자열을 사용해 글꼴을 변경하는 것이 바람직합니다.

6.2.3 글꼴의 크기 바꾸기

LaTeX에서는 총 10단계로 글꼴 크기를 지정할 수 있습니다. 글꼴 크기는 문서의 클래스 옵션으로 지정한 기본 글꼴 크기에 대해 상대적으로 정해집니다. 글꼴 크기를 지정하는 제어 문자열과 환경을 표 6.2에 정리하였습니다. 제어 문자열에는 선언형만 있고 인자형은 없으며, 대신 같은 이름의 환경을 사용할 수 있습니다.

표 6.2 | 글꼴 크기를 바꾸는 명령어

크기	선언형 명령어	환경	예시
4단계 작게	<code>\tiny</code>	tiny 환경	4단계 작게 tiny
3단계 작게	<code>\scriptsize</code>	scriptsize 환경	3단계 작게 scriptsize
2단계 작게	<code>\footnotesize</code>	footnotesize 환경	2단계 작게 footnotesize
1단계 작게	<code>\small</code>	small 환경	1단계 작게 small
기본 크기	<code>\normalsize</code>	normalsize 환경	기본 크기 normalsize
1단계 크게	<code>\large</code>	large 환경	1단계 크게 large
2단계 크게	<code>\Large</code>	Large 환경	2단계 크게 Large
3단계 크게	<code>\LARGE</code>	LARGE 환경	3단계 크게 LARGE
4단계 크게	<code>\huge</code>	huge 환경	4단계 크게 huge
5단계 크게	<code>\Huge</code>	Huge 환경	5단계 크게 Huge

글꼴을 `\begin{huge}크게\end{huge}` 바꿀 수도, `{\scriptsize 작게}` 바꿀 수도 있습니다.

글꼴을 크게 바꿀 수도, 작게 바꿀 수도 있습니다.

문서의 기본 글꼴 크기에 따라 각 명령어가 설정하는 실제 글꼴 크기는 표 6.3에서 확인할 수 있습니다. 다만, 문서를 작성할 때에는 글꼴의 실제 크기가 아닌 상대적인 크기 차이를 기준으로 글꼴 크기를 결정하는 것이 좋습니다.

6.2.4 기타 서식

LaTeX에서 기본 제공하는 서식 기능 중 나머지 기능으로는 첨자 기능과 숫자 모양 기능이 있습니다.

위 첨자와 아래 첨자

흔한 경우는 아니지만, 수식 바깥에서도 첨자를 사용합니다. 대표적인 예시는 영어의 서수 표기입니다(항상 첨자를 사용하는 것은 아닙니다). 위 첨자는 `\textsuperscript`를, 아래 첨자는 `\textsubscript`를 사용해 입력합니다.

`<superscript>`

`<superscript>`: 위 첨자로 입력할 내용

`\textsubscript{<subscript>}`

`<subscript>`: 아래 첨자로 입력할 내용

표 6.3 | 기본 글꼴 크기에 따라 명령어가 설정하는 실제 글꼴 크기

글꼴 크기	10 pt 문서	11 pt 문서	12 pt 문서
4단계 작게	5 pt	6 pt	6 pt
3단계 작게	7 pt	8 pt	8 pt
2단계 작게	8 pt	9 pt	10 pt
1단계 작게	9 pt	10 pt	10.95 pt
기본 크기	10 pt	10.95 pt	12 pt
1단계 크게	12 pt	12 pt	14.4 pt
2단계 크게	14.4 pt	14.4 pt	17.28 pt
3단계 크게	17.28 pt	17.28 pt	20.74 pt
4단계 크게	20.74 pt	20.74 pt	24.88 pt
5단계 크게	24.88 pt	24.88 pt	24.88 pt

1 Tony's team ranked
32nd among 287 teams
in the competition.

Tony's team ranked 32nd among 287
teams in the competition.

숫자의 모양

한국어 타이포그래피에서는 자주 사용되는 개념이 아니지만, 로마자처럼 대·소문자가 구분되는 문자를 사용하는 언어에서는 ‘대문자 숫자(lining figures)’와 ‘소문자 숫자(old-style figures)’라는 개념이 흔히 사용됩니다. 이들은 아라비아 숫자의 형태를 가리키는 것으로, 전자는 영어의 대문자처럼 0부터 9까지의 숫자의 높이가 모두 동일하지만 후자는 영어의 소문자처럼 숫자에 따라 그 높이가 달라집니다. 다음은 \LaTeX 기본 글꼴을 사용하여 두 가지 모양으로 숫자를 조판해 본 것입니다.

대문자 숫자 0 1 2 3 4 5 6 7 8 9
소문자 숫자 0 1 2 3 4 5 6 7 8 9

\LaTeX 에서는 `\oldstylenums` 제어 문자열을 제공하며, 이를 사용하면 대문자 숫자를 기본으로 사용하는 글꼴에서 소문자 숫자를 조판할 수 있게 됩니다. 인자로는 소문자 형태로 표기할 아라비아 숫자를 입력합니다(아라비아 숫자와 마침표, 쉼표 이외의 문자는 이상한 문자로 출력될 수 있습니다).

```
1 Thomas is \oldstylenums{183}
centimeters tall.
```

Thomas is 183 centimeters tall.

다만, 한 문서에서 대문자 숫자와 소문자 숫자를 섞어서 사용하는 것(수식 부분에만 대문자 숫자를 사용하는 것은 예외)은 일반적이지 않습니다. 따라서 문서에서 소문자 숫자를 조판할 때에는 `\oldstylenums`를 사용하는 것보다는 소문자 숫자를 기본으로 사용하는 글꼴을 문서의 기본 글꼴로 변경하는 것이 좋습니다.

참고 | 글꼴에 따라서는 소문자 숫자를 제공하지 않으며, 이 경우에는 `\oldstylenums`를 사용하더라도 대문자 숫자나 다른 기호가 조판됩니다.

6.3 프로그램 코드 조판하기

프로그램의 소스 코드를 \LaTeX 으로 조판하려면 역슬래시나 앰퍼샌드, 중괄호 등을 별도의 제어 문자열을 사용하여 입력해야 합니다. 이는 매우 귀찮은 일이므로, 입력한 문자열을 \LaTeX 문법으로 해석하지 않고 문자 그대로 조판하도록 하는 방법을 사용하는 것이 편리합니다.

verbatim 환경

여러 줄에 걸쳐 작성하는 코드는 `verbatim` 환경을 사용해 조판할 수 있습니다. 환경 안에 조판하려는 코드를 그대로 입력하면 됩니다.

```
1 \begin{verbatim}
2 #include <stdio.h>
3 void main() {
4     printf("Hello, world!");
5     return;
6 }
7 \end{verbatim}
```

```
#include <stdio.h>
void main() {
    printf("Hello, world!");
    return;
}
```

위 예시에서 확인할 수 있듯, `verbatim` 환경에 입력한 코드는 관행에 따라 고정폭 글꼴로 조판됩니다.

입력한 공백 문자를 명시적으로 표시해야 한다면 별표 붙은 `verbatim*` 환경을 사용할 수 있습니다. 이 환경에서는 공백 문자가 기호 ‘`␣`’로 표시됩니다.

```

1 You have to print precisely the
  following:
2 \begin{verbatim*}
3 ID    Font name
4 4     Times New Roman
5 35    Helvetica
6 78    Palatino
7 103   Courier
8 \end{verbatim*}

```

You have to print precisely the following:

```

IDFontname
4TimesNewRoman
35Helvetica
78Palatino
103Courier

```

\verb 제어 문자열

그대로 조판하려는 내용이 짧은 경우에는 `\verb` 또는 `\verb*` 제어 문자열을 사용할 수 있습니다. 이 제어 문자열을 사용하면 행중수식처럼 코드가 글줄 안에 섞여서 표시 됩니다.

\verb(*)⟨delim.⟩⟨string⟩⟨delim.⟩

⟨delim.⟩: 영역을 지정하기 위한 경계 문자

⟨string⟩: 그대로 조판할 문자열

⟨delim.⟩ 인자에는 어디부터 어디까지를 프로그램 코드로 취급할 것인지를 경계를 나타내는 문자로, 로마자 52자와 별표(*), 공백 문자를 제외한 임의의 문자 1개를 사용할 수 있습니다. ⟨string⟩에서 사용하지 않는 문자를 하나 고른 뒤 ⟨string⟩의 앞뒤 — 즉 ⟨delim.⟩ 인자의 자리 — 에 한 개씩 입력하고, 그 앞에 `\verb`나 `\verb*`를 입력하면 됩니다. 일반적으로는 세로줄 ‘|’이나 덧셈 기호 ‘+’를 경계 문자로 사용합니다.

```

1 `\'TeX'을 조판하려면 \verb|\'TeX|이라고 입력하
  면 되고, `\'LaTeX'을 조판하려면 \verb+\'LaTeX+
  이라고 입력하면 된다.

```

‘T_EX’을 조판하려면 `\TeX`이라고 입력하면 되고, ‘L_AT_EX’을 조판하려면 `\LaTeX`이라고 입력하면 된다.

`\verb` 대신 `\verb*`를 사용하면 `verbatim*` 환경과 마찬가지로 공백 문자가 기호 ‘_’로 바뀌어 명시적으로 표시됩니다.

참고 | T_EX이 제어 문자열의 인자를 처리하는 방식과 `\verb` 및 `\verb*`의 작동 방식 때문에, `\verb`와 `\verb*`는 다른 제어 문자열이나 환경의 인자로 사용할 수 없습니다. 환경 안에서는 사용할 수 있습니다.

다른 패키지 사용하기

L^AT_EX에서 프로그램 코드를 조판하는 데에는 다른 패키지의 도움을 받을 수 있습니다. 레슬리 램포트의 `alltt` 패키지, 라이너 쇼프(Rainer Schöpf)의 `verbatim` 패키지, 티모시 반 잔트(Timothy Van Zandt)의 `fancyvrb` 패키지, 제프리 푸어(Geoffrey M. Poore)의 `fvextra` 패키지는 다양한 기능이 추가된 코드 작성 환경을 정의해 줍니다.

`fancyvrb` 패키지와 프랭크 미텔바흐의 `shortvrb` 패키지는 `\verb` 제어 문자열 대신 문자 하나로 문단 속에 짧은 코드를 조판할 수 있도록 하는 기능을 제공합니다. 또, 프로그래밍 언어에 맞는 구문 강조(syntax highlighting) 기능을 제공하는 패키지도 있습니다. 카르스텐 하인즈(Carsten Heinz)가 작성한 `listings` 패키지나 제프리 푸어의 `minted` 패키지가 이러한 기능을 제공합니다.

이들 패키지에 대한 사용 방법은 각각의 패키지 안내서[6, 20, 29, 38, 39, 44, 55]를 참조하시기 바랍니다. 각 패키지의 특징과 차이점은 [33]의 4.2절에서 읽어볼 수 있습니다.

연습문제

- 6.1. 다음을 조판하여라. 목록에서 사용된 화학식 세 개는 각각 `\(\mathrm{NaBH_4}\)`, `\(\mathrm{LiAlH_4}\)`, `\(\mathrm{H_2CrO_4}\)`로 입력한다.

1. How to produce alcohol
 - (a) Via Reduction
 - i. Use metal catalyst (rarely used)
 - ii. Use $\mathrm{NaBH_4}$
 - iii. Use $\mathrm{LiAlH_4}$
 - (b) Via Grignard Reagent
 - i. Use Grignard reagent
2. How to oxidize alcohol
 - (a) Complete oxidization
 - i. Use $\mathrm{H_2CrO_4}$
 - (b) When isolation of aldehyde is needed
 - i. Use PCC
 - ii. Swern oxidation
 - iii. DMP oxidation

6.2. 다음을 조판하여라.

식물의 분류

1. 비관다발 식물
2. 관다발 식물
 - (a) 비종자 관다발 식물
 - (b) 종자 식물
 - i. 겉씨식물
 - ii. 속씨식물
 - A. 외떡잎식물
 - B. 쌍떡잎식물

6.3. 다음을 조판하여라.

KAIST 수리과학과에서 개설하는 기초선택 과목의 목록은 다음과 같다.

- MAS109 (선형대수학개론)
- MAS110 (데이터과학을 위한 선형대수학)
- MAS201 (응용미분방정식)
- MAS202 (응용해석학)
- MAS250 (확률및통계)

6.4. 다음을 조판하여라.

페르마는 자신의 저서에 이런 글귀를 남겼다.

나는 이 문제에 대한 놀라운 증명을 알고 있다. 여백이 부족하여 여기에 적지 않는다.

여기에서 ‘이 문제’는 페르마의 마지막 정리를 가리킨다.

6.5. 문단 정렬을 변경하는 환경을 사용하여 다음을 조판하여라.

우리가 이번 장에서 중요하게 생각해야 할 것은 오직
함수를 미분할 수 있는가?

이다. 함수의 미분 가능성은 여러 가지 정리를 통해 증명할 수 있다.

수학과 박 모 교수
2023년 봄학기 해석학 I 강의에서

6.6. 다음 문장을 \textit이나 \itshape 없이 조판하여라.

한국어 문장을 기울임꼴을 사용하여 강조하는 것은 적절하지 않다. 기울임꼴 대신 밑줄이나 드래프트를 사용하는 것이 적절하며, 글꼴을 바꾸어 강조하는 것도 좋은 방법이다.

6.7. 다음 문장을 조판하여라.

KAIST 수학문제연구회에는 **학술부**, **홍보부**, **ML부**, **수학동아부** 등의 부서가 있습니다.

6.8. 다음 문장을 조판하여라.

오늘은 **2023년 11월 21일**입니다.
It is 21st, March 2024 today.

6.9. 다음 문장을 조판하여라.

I hate Mondays, but I love Fridays!

6.10. verbatim 환경을 사용하여 다음을 조판하여라.

```
from cs1robots import *

load_world('worlds/harvest1.wld')
hubo=Robot()

def turn_right():
    for i in range(3):
        hubo.turn_left()
def pick():
    hubo.move()
    hubo.pick_beeper()
```


제 7 장

수학적 문단 작성하기

이번 장에서는 \LaTeX 에서 정의, 정리, 증명 등 수학적 문단을 일정한 형식에 맞춰 조판하는 방법을 알아보겠습니다. \LaTeX 은 수학적 문단을 조판하기 위한 명령어를 제공하며, 많은 문서에서 사용하는 `amsthm` 패키지를 사용하면 이 기능을 확장할 수 있습니다.

7.1 기본적인 수학적 문단 만들기

수학적 문단을 작성하는 기본적인 방법은 정리형 환경을 정의하고, 그 환경을 사용하여 문단을 작성하는 것입니다. 정의된 정리형 환경을 사용하면 제목과 번호가 붙어 있는 수학적 문단을 올바르게 조판할 수 있습니다. 이번 절에서는 정리형 환경을 정의하고 사용하는 기본적인 방법을 알아봅니다.

7.1.1 정리형 환경 정의하기

정리형 환경을 정의하는 `\newtheorem*` 제어 문자열은 아래와 같이 사용합니다. (별표 붙은 `\newtheorem*`는 `amsthm` 패키지를 불러와야만 사용할 수 있습니다.)

```
\newtheorem{<name>}[<ref. counter>]{<heading>}[<parent counter>]
\newtheorem*{<name>}{<heading>}
```

`<name>`: 정의하려는 환경의 이름

`<ref. counter>`: 번호를 공유할 환경의 이름

`<heading>`: 이 환경을 사용할 때 문단 첫머리에 출력할 내용

`<parent counter>`: 번호를 중속시킬 상위 카운터

카운터
→ 269쪽

이 제어 문자열을 사용하면 이름이 `<name>`인 환경과 카운터가 정의되며, 이 환경을 사용해 문단을 조판할 때에는 첫머리에 `<heading>`에 입력된 내용이 번호와 함께 출

력됩니다. `\newtheorem*` 제어 문자열을 사용해 환경을 정의하면 해당 환경을 사용해 문단을 조판할 때 문단의 번호가 조판되지 않습니다. 선택 인자인 $\langle \text{ref. counter} \rangle$ 와 $\langle \text{parent counter} \rangle$ 에 대해서는 다음 절에서 설명합니다.

예를 들어, 보조정리를 작성하기 위한 lemma 환경과 번호 없이 공리를 작성하기 위한 axiom 환경을 정의하려면 아래와 같이 입력합니다.

```
1 \newtheorem{lemma}{Lemma}
2 \newtheorem*{axiom}{Axiom}
```

7.1.2 정의한 환경 사용하기

`\newtheorem`으로 정의한 환경은 아래와 같이 사용합니다.

```
\begin{<name>}[<subtitle>]
```

$\langle \text{name} \rangle$: 정의한 환경의 이름

$\langle \text{subtitle} \rangle$: 문단의 번호 뒤에 붙일 부제

```
\end{<name>}
```

선택 인자를 지정하면 문단의 이름과 번호 뒤에 부제가 작성됩니다.

위에서 정의한 lemma 환경을 사용해 문단을 작성하면 그 문단의 첫머리에 ‘Lemma 1’, ‘Lemma 2’……가 출력되며, axiom 환경을 사용해 문단을 작성하면 문단의 첫머리에 번호 없이 ‘Axiom’이 출력됩니다. 또, 아래와 같이 선택 인자를 지정하여 부제를 붙일 수도 있습니다.

```
1 \begin{lemma}[Zorn]
2   모든 연쇄가 상계를 가지도록 하는 순서집합은 극대
   원을 가진다.
3 \end{lemma}
4 \begin{axiom}[Choice]
5    $\set{\emptyset} \notin X$ 인 임의의 집합
    $X$ 의 선택 함수가 존재한다.
6 \end{axiom}
```

Lemma 1 (Zorn). 모든 연쇄가 상계를 가지도록 하는 순서집합은 극대원을 가진다.

Axiom (Choice). $\emptyset \notin X$ 인 임의의 집합 X 의 선택 함수가 존재한다.

참고 | amsmath 패키지를 불러오지 않은 경우에는 $\langle \text{subtitle} \rangle$ 인자로 지정한 내용도 볼드체로 조판되지만, amsmath 패키지를 불러왔다면 이 인자의 내용은 위의 예시처럼 보통 굵기의 글꼴로 조판됩니다.

7.1.3 증명 작성하기

amsthm 패키지는 증명을 작성하기 위한 proof 환경을 제공합니다. 이 환경을 사용하면 환경 안의 첫 번째 문단 첫머리에 ‘Proof’(한국어 문서에서는 ‘증명’)가 이탤릭체로 출력되고, 마지막 문단 오른쪽 끝에 증명 완료 기호 ‘□’가 출력됩니다.

```
\begin{proof}[<alt. title>]
```

<alt. name>: ‘Proof’나 ‘증명’ 대신 사용할 문단 제목

```
\end{proof}
```

<pre>1 \begin{proof} 2 귀납법을 사용해 증명한다. ... 따라서 주어진 정 리가 성립한다. 3 \end{proof}</pre>	증명. 귀납법을 사용해 증명한다. ... 따라서 주어진 정리가 성립한다. □
---	--

특정 증명의 제목을 변경하고자 한다면 환경에 선택 인자 *<alt. title>*을 지정하면 됩니다. 인자로 입력된 내용이 이탤릭체로 문단 첫 부분에 출력됩니다.

<pre>1 \begin{proof}[Proof of Lemma 1] 2 ... Thus we proved the lemma. 3 \end{proof}</pre>	<i>Proof of Lemma 1.</i> ... Thus we proved the lemma. □
--	--

proof 환경 밖에서 증명 완료 기호를 출력하려면 문단의 마지막에 \qed 제어 문자열을 사용하면 됩니다.

<pre>1 ... 따라서 주어진 정리가 성립한다. \qed</pre>	... 따라서 주어진 정리가 성립한다. □
---	-------------------------

증명 완료 기호의 위치 수정하기

proof 환경 안의 내용이 특정 환경(목록 작성 환경이나 quote 등)이나 별행수식으로 끝나는 경우, 아래와 같이 증명 완료 기호가 환경 또는 수식 아래에 출력됩니다.

<pre>1 \begin{proof} 2 ... Hence we showed that 3 \[G:H = \frac{ G }{ H }. \] 4 \end{proof}</pre>	<div style="display: flex; align-items: center;"> <div style="flex: 1;"> <i>Proof.</i> ... Hence we showed that </div> <div style="flex: 1; text-align: center;"> $G:H = \frac{ G }{ H }.$ </div> <div style="flex: 0 0 auto; text-align: right;">□</div> </div>
--	---

이러한 경우, 환경이나 수식의 끝에 `\qedhere` 제어 문자열을 입력하여 이 기호가 출력되는 위치를 수정할 수 있습니다.

```
1 \begin{proof}
2   ... Hence we showed that
3   \[ |G:H| = \frac{|G|}{|H|}. \qedhere
4   \]
```

Proof. ... Hence we showed that

$$|G:H| = \frac{|G|}{|H|}. \quad \square$$

7.2 다양한 형태의 수학적 문단

이번 절에서는 번호나 글꼴 등이 기본 설정과 다른 형태로 표시되는 수학적 문단을 만드는 방법을 알아봅니다.

7.2.1 환경의 번호 다루기

문단의 번호가 다른 형태로 계산되도록 하거나, 다른 모습으로 표시되도록 할 수도 있습니다. 문단 번호를 계산할 때에는 \LaTeX 의 카운터 기능을 사용하므로, 필요한 경우 19.1절을 참조하면 좋습니다.

다른 번호의 하위 번호로 만들기

수학적 문단의 번호를 문서의 장 또는 절의 하위 번호로 만들어 장 또는 절 번호가 바뀔 때마다 문단 번호가 다시 1부터 시작되도록 하려면, `\newtheorem`의 $\langle parent\ counter \rangle$ 인자에 해당 번호를 출력하는 카운터의 이름을 입력하면 됩니다. 예를 들어, `theorem` 환경의 문단 번호를 장 번호의 하위 번호로 만들고자 한다면 전처리부에

```
\newtheorem{theorem}{Theorem}[chapter]
```

를 입력하면 됩니다. 이후 `theorem` 환경을 사용하면 아래 예시와 같이 문단의 번호가 장 번호와 함께 표시됩니다.

```
1 \begin{theorem}[Zermelo]
2   임의의 집합은 정렬 가능하다.
3 \end{theorem}
```

Theorem 7.1 (Zermelo). 임의의 집합은 정렬 가능하다.

이후 새로운 장이 시작되는 경우, `theorem` 환경의 번호는 1부터 다시 시작되어 문단의 제목이 ‘Theorem 9.1’, ‘Theorem 9.2’……로 표시됩니다.

다른 환경과 번호 연동시키기

여러 수학적 문단이 번호를 공유하기도 합니다. 예를 들어, [15]에서는 정리, 보조정리, 따름정리, 명제가 모두 번호를 공유합니다. 따라서 제3장에서는 Proposition 2 다음에 Theorem 3이 등장하고, 그 다음에는 Proposition 3 대신 Proposition 4가 있습니다. 이처럼, 한 정리형 환경의 번호를 (미리 정의된) 다른 정리형 환경의 번호와 연동시키려는 경우에는 `\newtheorem`의 `\langle ref. counter \rangle` 인자에 해당 번호를 기록하는 카운터의 이름을 입력합니다.

예를 들어, 앞에서 정의한 theorem 환경과 번호가 연동되는 proposition 환경은 아래와 같이 정의합니다.

```
\newtheorem{proposition}[theorem]{Proposition}
```

이렇게 정의한 proposition 환경을 사용하면 아래의 결과를 얻습니다.

```
1 \begin{proposition}[Hausdorff]
2   임의의 순서집합은 극대 연쇄를 가진다.
3 \end{proposition}
```

Proposition 7.2 (Hausdorff). 임의의 순서집합은 극대 연쇄를 가진다.

반대로, theorem 환경도 proposition 환경과 번호를 공유하므로, 이후 theorem 환경을 사용하면 ‘7.3’이 문단 번호로 출력됩니다.

```
1 \begin{theorem}
2   집합족  $\mathcal{X}$ 의 원소들의 곱집합은  $\mathcal{X}$ 의 선택
   함수들의 집합으로 간주할 수 있다.
3 \end{theorem}
```

Theorem 7.3. 집합족 X 의 원소들의 곱집합은 X 의 선택 함수들의 집합으로 간주할 수 있다.

번호가 제목 앞에 표시되도록 하기

amsthm 패키지에서 제공하는 `\swapnumbers` 제어 문자열을 입력하면 이 이후에 `\newtheorem`으로 정의되는 정리형 환경의 번호와 제목의 순서를 바꿉니다. [43]에서 수학적 문단이 이러한 형식으로 조판되어 있습니다.

예를 들어, corollary 환경을 정의할 때

```
1 \swapnumbers
2 \newtheorem{corollary}[theorem]{Corollary}
```

라고 입력하면 이 환경에서는 문단 번호가 ‘Corollary’보다 먼저 출력됩니다.

```

1 \begin{corollary}
2   공집합이 아닌 집합들의 곱집합은 공집합이 아니다.
3 \end{corollary}

```

7.4 Corollary. 공집합이 아닌 집합들의 곱집합은 공집합이 아니다.

\swapnumbers를 한 번 더 입력한 뒤에 정의한 환경에서는 원래대로 문단의 이름 뒤에 문단 번호가 출력됩니다.

7.2.2 문단 스타일 바꾸기

참고 | 수학적 문단의 스타일 기능은 amsthm 패키지를 불러와야 사용할 수 있습니다.

책에 따라 정의 문단은 정체로, 정리 문단은 이탤릭체나 기울임꼴로 조판하는 등 문단의 스타일에 구분을 두기도 합니다. amsthm 패키지는 다음 세 가지 기본 스타일을 제공하며, plain 스타일을 기본으로 사용합니다.

- plain 스타일 상하 여백 있음, 제목 볼드체, 내용 이탤릭체
- definition 스타일 상하 여백 있음, 제목 볼드체, 내용 정체
- remark 스타일 상하 여백 없음, 제목 이탤릭체, 내용 정체

plain 스타일은 주로 정리, 보조정리, 따름정리, 명제 등 뒤에 증명이 따르는 문단을 작성하는 문단에 사용하고, definition 스타일은 이름대로 정의나 표기법 등을 작성하는 문단에 사용합니다. remark는 참고 사항 등에 사용합니다.

\theoremstyle 제어 문자열을 사용하면 \newtheorem 제어 문자열로 정의되는 환경의 스타일을 바꿀 수 있습니다. 인자로 사용하려는 스타일 이름을 입력하면 됩니다.

\theoremstyle{<style>}

<style>: 사용할 스타일의 이름

\theoremstyle은 이후 정의되는 정리형 환경이 <style> 스타일을 사용할 것을 선언합니다. 예를 들어, 다음과 같이 환경을 정의하면 definition 환경은 definition 스타일로, remark 환경은 remark 스타일로 조판됩니다.

```

1 \theoremstyle{definition}
2 \newtheorem*{definition}{Definition}
3 \theoremstyle{remark}
4 \newtheorem*{remark}{Remark}

```

```

1 \begin{definition}[선택 함수]
2   집합  $(X)$ 의 선택 함수는  $(\forall x \in X \ ; \ f(x) \in x)$ 를 만족시키는 함수
    $(f \colon X \rightarrow \bigcup X)$ 이다.
3 \end{definition}
4 선택 함수는 선택 공리에서 중요한 개념이다.
5 \begin{remark}
6   만약  $(\emptyset \in X)$ 이면  $(X)$ 의 선택
   함수가 존재할 수 없다.
7 \end{remark}

```

Definition (선택 함수). 집합 X 의 선택 함수는 $\forall x \in X \ f(x) \in x$ 를 만족시키는 함수 $f: X \rightarrow \bigcup X$ 이다.

선택 함수는 선택 공리에서 중요한 개념이다.

Remark. 만약 $\emptyset \in X$ 이면 X 의 선택 함수가 존재할 수 없다.

새로운 스타일 만들기

`\newtheoremstyle` 제어 문자열을 사용하면 새로운 스타일을 정의하거나 기존의 스타일을 변경할 수 있습니다. 이 제어 문자열은 아홉 개의 인자를 입력받습니다. 이중 2, 3, 5, 9번째 인자는 빈칸으로 두면 plain 또는 definition 스타일의 기본값이 사용되며, 4번째와 6번째 인자는 빈칸으로 두면 기본 글꼴이 사용됩니다.

```

\newtheoremstyle{<name>}{<top margin>}{<bottom margin>}{<body font>}
{<indent>}{<title font>}{<title separator>}{<title separation>}{<title format>}

```

`<name>`: 정의할 스타일의 이름
`<top margin>`: 문단의 위쪽 여백 길이
`<bottom margin>`: 문단의 아래쪽 여백 길이
`<body font>`: 본문 글꼴 서식
`<indent>`: 첫 줄을 들여쓸 간격
`<title font>`: 제목 글꼴 서식
`<title separator>`: 제목 뒤에 붙일 구분 기호(또는 문장부호)
`<title separation>`: 제목과 본문 사이의 간격
`<title format>`: 제목의 형식

`<title separation>` 인자에는 길이 값을 입력해야 하지만, 예외로 공백 문자를 입력해 일반적인 띄어쓰기와 같은 간격이 삽입되도록 하거나, `\newline`을 입력해 본문을 제목과 별도의 줄에 조판하도록 할 수 있습니다. `<title format>` 인자는 `\thmname`, `\thmnumber`, `\thmnote` 제어 문자열을 사용해 지정해야 합니다. 사용 방법이 복잡하므로 패키지의 안내서[53, p. 9]를 참조하시기 바랍니다.

다음은 스타일을 새로 정의하는 예시 코드입니다. `slplain` 스타일의 다른 부분은 모두 plain 스타일과 동일하지만, 환경 안의 내용을 이탤릭체가 아닌 기울임꼴로 —

[34]의 본문처럼 — 조판하도록 합니다. 이탤릭체 대신 기울임꼴을 사용하면 강조 효과를 얻을 수 있으면서 동시에 수식과도 서식이 구분됩니다.

```
1 \newtheoremstyle{slplain}
2   {}{}{\normalfont\bfseries}{}{\normalfont\slshape}{}{1em}{}{}
```

이 스타일을 사용하여 theorem 환경을 정의했다면 다음의 결과를 얻습니다.

```
1 \begin{theorem}
2   Let  $(Y)$  be a ... is also regular.
3 \end{theorem}
```

Theorem 8.5. *Let Y be a subspace of X , and suppose that X is regular. Then Y is also regular.*

7.3 예시 코드

많은 문서에서는 정형화된 종류의 수학적 문단을 사용합니다. 자주 쓰이는 것에는

정의(definition), 공리(axiom), 정리(theorem), 명제(proposition), 보조정리(lemma), 따름정리(corollary), 예시(example), 참고(remark)

정도가 있습니다. 이들을 정의하는 예시 코드를 아래에 제시합니다. 이 코드를 바탕으로, 자신이 필요한 정리형 환경만을 취해 정의하거나 원하는 대로 수정하여 사용하기 바랍니다.

```
1 \usepackage{amsthm}
2
3 \newtheorem*{axiom}{Axiom}
4 \newtheorem{theorem}{Theorem}
5 \newtheorem{proposition}[theorem]{Proposition}
6 \newtheorem{lemma}[theorem]{Lemma}
7 \newtheorem{corollary}[theorem]{Corollary}
8
9 \theoremstyle{definition}
10 \newtheorem*{definition}{Definition}
11 \newtheorem*{definitions}{Definitions}
12 \newtheorem*{example}{Example}
13 \newtheorem*{examples}{Examples}
14
15 \theoremstyle{remark}
16 \newtheorem*{remark}{Remark}
17 \newtheorem*{remarks}{Remarks}
```


한국어 문서에서는 스타일 정의 기능을 사용하여 다음과 같이 정리형 환경을 정의할 수 있습니다.

```

1 \usepackage{amsthm}
2
3 \newtheoremstyle{korean}
4   {}{}{\normalfont\sffamily\bfseries}{}{\normalfont}{}{1em}{}
5 \theoremstyle{korean}
6 \newtheorem*{axiom}{공리}
7 \newtheorem{theorem}{정리}
8 \newtheorem{proposition}[theorem]{명제}
9 \newtheorem{lemma}[theorem]{보조정리}
10 \newtheorem{corollary}[theorem]{따름정리}
11
12 \newtheorem*{definition}{정의}
13 \newtheorem*{definitions}{정의}
14 \newtheorem*{example}{예시}
15 \newtheorem*{examples}{예시}
16
17 \newtheoremstyle{koreanremark}
18   {0pt}{0pt}{\normalfont\sffamily}{}{\normalfont}{}{1em}{}
19 \theoremstyle{koreanremark}
20 \newtheorem*{remark}{참고}
21 \newtheorem*{remarks}{참고}

```

theorem 환경을 정의하는 코드의 맨 끝에 ‘[section]’을 덧붙이면 정리·명제·보조정리·따름정리를 작성할 때 문단 번호의 앞에 절 번호를 표시할 수 있습니다. \chapter를 사용할 수 있는 report 및 book 클래스 문서에서는 ‘[chapter]’를 덧붙여 절 번호 대신 장 번호를 표시할 수도 있습니다.

연습문제

이번 장의 연습문제는 그림 7.1~7.2의 문서를 작성하는 것입니다. 하나의 article 클래스 문서에 답안을 작성하시기 바랍니다. 수식을 작성하는 방법은 제 III 편을 참조하시기 바랍니다.

- 7.1. \newtheorem과 \theoremstyle 제어 문자열을 사용해 정의(definition), 정리(theorem), 보조정리(lemma), 따름정리(corollary), 참고(remark)를 작성하기 위한 환경을 정의하라. (힌트: 문단의 스타일과 번호에 유의하여야.)

벡터 공간의 기저와 차원

권도형

2024년 1월 18일

Definition. S 를 벡터 공간 V 의 공집합이 아닌 부분집합이라고 하자. S 의 생성은 S 를 포함하는 모든 V 의 부분공간의 교집합이며, $\text{span}(S)$ 라고 표기한다. 즉, 다음과 같이 정의한다.

$$\text{span}(S) = \bigcap_{S \subset W \subset V} W$$

Definition. V 의 부분집합 S 가 선형 독립이고 $\text{span}(S) = V$ 일 때, S 를 V 의 기저라고 한다. 또, $|S|$ 를 V 의 차원이라고 하고, 이를 $\dim V$ 라고 표기한다.

Lemma 1. $V = \text{span}\{v_1, v_2, \dots, v_n\}$ 라 두자. 그러면 V 의 선형 독립인 부분집합 S 는 $|S| \leq n$ 을 만족시킨다.

증명. $|S| = m$ 이라고 하고 $S = \{u_1, \dots, u_m\}$ 으로 두자. 그러면 다음을 만족하는 V 의 스칼라 체의 원소 a_1, \dots, a_n 이 존재한다.

$$u_1 = a_1 v_1 + a_2 v_2 + \dots + a_n v_n$$

일반성을 잃지 않고 a_1 이 0이 아니라고 둘 수 있으므로, 다음과 같이 쓸 수 있다.

$$V = \text{span}\{u_1, v_2, \dots, v_n\}, \quad S_1 = S - \{s_1\}$$

S_1 은 선형 독립이므로, 아래와 같이 쓰면 V 의 스칼라 체의 원소 b_2, \dots, b_n 중 적어도 하나는 0이 아니다.

$$u_2 = b_1 u_1 + b_2 v_2 + \dots + b_n v_n$$

다시 일반성을 잃지 않고 V 를 생성하는 집합에서 v_2 대신 u_2 를 넣고 S_1 에서 s_2 를 뺀 집합 S_2 를 잡을 수 있으며, 이를 v_n 이 u_n 으로 바뀔 때까지 반복한다. 그 전에 S 에서 공집합 S_i 를 얻으면 원하는 결과가 나온다. 이 과정이 끝난 후에도 S_i 에 원소가 남아 있다면 S 가 선형 독립임에 모순이므로 증명은 끝이 난다. \square

Remark. 벡터 공간의 차원은 유일하다.

Theorem 2. 유한차원 벡터 공간 V 의 선형 독립인 부분집합 S 에 V 의 원소를 추가하여 V 를 생성할 수 있다.

증명. 선형 독립인 집합 $S = \{v_1, \dots, v_m\}$ 를 생각하자. $\text{span}(S) = V$ 라면 원소를 추가할 필요가 없이 S 가 기저가 된다. 만약 아니라면, $V \setminus \text{span}(S)$ 의 원소 v_{m+1} 를 뽑아서 S 에 추가하는 과정을 반복한다. Lemma 1에 의해, 이 과정은 $\dim V - m$ 번 안에 끝난다. \square

Corollary 3. V 를 유한차원 벡터 공간이라고 하고, U 와 W 를 V 의 부분공간이라 하자. 그러면 다음이 성립한다.

$$\dim U + \dim W = \dim(U + W) + \dim(U \cap W)$$

증명. $U \cap W$ 의 차원을 n 이라 하고, $B = \{x_1, \dots, x_n\}$ 을 $U \cap W$ 의 기저라 하자. 그러면 Theorem 2에 의해 B 를 U 의 기저 $B_1 = \{x_1, \dots, x_n, y_1, \dots, y_m\}$ 과 W 의 기저 $B_2 = \{x_1, \dots, x_n, z_1, \dots, z_l\}$ 로 확장할 수 있다. 그러면 다음은 자명하다.

$$U + W = \text{span}\{x_1, \dots, x_n, y_1, \dots, y_m, z_1, \dots, z_l\}$$

따라서, 주어진 따름정리의 증명은 $B_1 \cup B_2 = \{x_1, \dots, x_n, y_1, \dots, y_m, z_1, \dots, z_l\}$ 이 선형 독립임을 보이는 것으로 충분하다.

V 의 스칼라 체의 원소 $a_1, \dots, a_n, b_1, \dots, b_m, c_1, \dots, c_l$ 에 대해

$$\sum_{i=1}^n a_i x_i + \sum_{j=1}^m b_j y_j + \sum_{k=1}^l c_k z_k = 0 \quad (*)$$

으로 두자. 그러면

$$\sum_{i=1}^n a_i x_i + \sum_{j=1}^m b_j y_j \in U$$

이므로 $\sum_{k=1}^l c_k z_k \in U$, 즉 $\sum_{k=1}^l c_k z_k \in U \cap W$ 이다. B 는 $U \cap W$ 의 기저이므로

$$\sum_{k=1}^l c_k z_k = \sum_{i=1}^n d_i x_i$$

이고, 따라서

$$\sum_{k=1}^l c_k z_k - \sum_{i=1}^n d_i x_i = 0$$

이다. B_2 는 선형 독립이므로 각각의 k 와 i 에 대해 $c_k = 0$, $d_i = 0$ 이다. 그러므로,

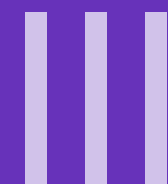
(*)에서

$$\sum_{i=1}^n a_i x_i + \sum_{j=1}^m b_j y_j = 0$$

이므로, 각각의 i, j, k 에 대해 다음을 얻는다.

$$a_i = 0, \quad b_j = 0, \quad c_k = 0 \quad \square$$

- 7.2. 연습문제 7.1에서 정의한 정의 작성 환경을 사용해 첫 번째 Definition을 작성하여라.
- 7.3. 연습문제 7.1에서 정의한 정의 작성 환경을 사용해 두 번째 Definition을 작성하여라.
- 7.4. 연습문제 7.1에서 정의한 보조정리 작성 환경을 사용해 Lemma 1을 작성하여라.
- 7.5. 연습문제 7.1에서 정의한 참고 작성 환경을 사용해 Remark를 작성하여라.
- 7.6. 연습문제 7.1에서 정의한 정리 작성 환경을 사용해 Theorem 2를 작성하여라.
- 7.7. 연습문제 7.1에서 정의한 따름정리 작성 환경을 사용해 Corollary 3을 작성하여라.
- 7.8. proof 환경을 사용해 Lemma 1의 증명을 작성하여라.
- 7.9. proof 환경을 사용해 Theorem 2의 증명을 작성하여라.
- 7.10. proof 환경을 사용해 Corollary 3의 증명을 작성하여라. (힌트: `\qedhere`를 사용하여라.)



수식 작성하기

제Ⅲ편에서는 수식과 관련된 내용을 알아봅니다. 수식을 작성하는 기본적인 명령어에서 비롯하여, 다양한 형태의 수식을 표현하는 방법과 여러 줄에 걸친 수식을 작성하는 방법을 알아봅니다. 또, 수학적 대상 사이의 관계를 효과적으로 나타낼 수 있는 가환 도표를 그리는 방법도 알아봅니다.

제8장 기본적인 수식 작성하기 95

제9장 다양한 수식 작성하기 123

제10장 별행수식 작성하기 141

제11장 가환 도표 그리기 151

제 8 장

기본적인 수식 작성하기

대부분의 사람들이 \LaTeX 을 사용하는 이유는 수식이 많은 문서를 작성하기 위함입니다. 이번 장에서 드디어 수식을 작성하는 방법을 소개합니다. \LaTeX 으로 수식을 작성하는 방법과 함께, 분수나 근호, 각종 기호 등 기본적인 수식 구성 요소를 어떻게 입력하는지 알아봅시다.

8.1 \LaTeX 에서의 수식 입력

이번 절에서는 제3장의 내용에 더하여, \LaTeX 으로 수식을 작성할 때 참고할 점들을 설명합니다. 수식을 입력할 때 사용하는 명령어는 다음 절에서부터 설명합니다.

8.1.1 수식 모드

수식은 일반적인 본문의 글꼴과는 다른 글꼴로 조판되고, 기호와 기호 사이의 간격, 첨자의 위치, 구분선의 두께와 길이 등이 세심하게 조절됩니다. 예를 들어, 중학교 수학 교과서는 이차방정식을 ' $ax^2 + bx + c = 0$ '이라고 조판하지, ' $ax^2+bx+c=0$ '이라고 조판하지 않습니다.

\LaTeX 에서 수식을 조판하기 위해서는 수식 코드를 \backslash (와 \backslash)로 감싸야 합니다(`math` 환경을 사용해도 됩니다). 이렇게 하면 입력된 코드가 수식을 올바르게 조판하도록, 일반 텍스트를 조판할 때와는 다르게 해석됩니다. 문자에 따라 올바른 글꼴을 자동으로 선택하고, 각 기호 사이의 간격을 알아서 조절해 줍니다. 즉, 앞 문단에서 예로 든 이차 방정식을 조판하려면

```
 $\backslash(ax^2+bx+c=0\backslash)$ 
```

이라고 입력하면 됩니다. 다음 예시도 생각할 수 있습니다.

입력

```
\(2 + 2 \times 2 = 8\)
```

```
\((x+1)(x+2) = x^2+3x+2\)
```

```
\(\cos^2 x + \sin^2 x = 1\)
```

출력

$$2 + 2 \times 2 = 8$$

$$(x + 1)(x + 2) = x^2 + 3x + 2$$

$$\cos^2 x + \sin^2 x = 1$$

\TeX 이 수식 코드를 해석할 수 있는 상태를 수식 모드`math mode`라고 부르며, 그렇지 않은 상태를 텍스트 모드`text mode`라고 부릅니다. `\(`를 입력하면 텍스트 모드에서 수식 모드로 전환되고, `\)`를 입력하면 수식 모드에서 텍스트 모드로 전환됩니다.

앞 예시들에서 이미 눈치채신 분도 있겠지만, 수식 모드에서는 소스 파일에 입력된 공백 문자가 기호 사이의 간격에 영향을 주지 않습니다. 따라서, 덧셈 기호나 등호 앞뒤에 공백 문자가 있든 없든 \TeX 은 공백을 알아서 조절하여 동일한 결과물을 만들어 냅니다.

입력

```
\(12^2+16=160\)
```

```
\(12^2 + 16 = 160\)
```

```
\( 1 2 ^ 2 + 1 6 = 1 6 0 \)
```

출력

$$12^2 + 16 = 160$$

$$12^2 + 16 = 160$$

$$12^2 + 16 = 160$$

하지만 \TeX 이 수식의 의미 자체를 이해할 수 있는 프로그램은 아니라서, 필요한 경우에는 간격을 사용자가 직접 조절해 주어야 합니다. 일반적으로는 8.2.2절의 내용만을 지켜도 충분히 보기 좋은 문서를 얻을 수 있습니다.

이외에도 수식 모드는 코드를 해석할 때 텍스트 모드와 몇 가지 차이를 보입니다. 정리하자면 아래와 같지만, 보통은 방금 설명한 첫 번째 항목만 알고 있으면 됩니다.

- 제어 단어 뒤의 공백 문자를 제외하고, 모든 공백 문자가 무시됩니다. 기호와 기호 사이의 간격은 \TeX 의 알고리즘을 따라 자동으로 조정됩니다.
- 텍스트 모드와는 다르게, 로마자가 수식 이탤릭체로 조판됩니다. 수식 이탤릭체는 본문을 조판하기 위한 이탤릭체와 다르므로, 둘을 섞어서 사용하면 안 됩니다.
- 수식 요소나 수식 기호를 입력하기 위한 명령어를 사용할 수 있게 됩니다. 텍스트 모드에서 이들을 사용하면 컴파일 과정에서 오류가 발생합니다.
- 중괄호는 하위 수식을 구분하는 데 사용합니다. 선언형 명령어의 유효 범위는 `\begingroup`과 `\endgroup`으로 지정합니다.¹

¹ 중괄호를 사용해도 선언형 명령어의 유효 범위를 지정할 수 있지만, 중괄호가 앞뒤 수식과의 간격에 영향을 미칠 수 있습니다.

8.1.2 행중수식과 별행수식

앞 절에서 설명한 `\(...\)` 명령어를 사용하면 수식이 글줄과 섞여 나타납니다. 하지만

$$e^{i\theta} = \cos \theta + i \sin \theta$$

처럼 중요하거나 긴 수식은 앞뒤 글과 구분하여 작성해 줍니다. 더 중요한 식이라면

$$e^{i\pi} + 1 = 0 \tag{1}$$

처럼 번호를 붙여 두기도 합니다. 이러한 수식을 **별행수식** `displayed math`이라고 부르며, `\[...\]`로 수식 코드를 감싸면(또는 `displaymath` 환경 안에 수식 코드를 작성하면) 별행수식을 입력할 수 있습니다. 또, `equation` 환경 안에 수식 코드를 작성하면 별행수식이 조판될 뿐 아니라, 수식 옆에 번호까지 출력됩니다. 따라서 위의 첫 번째 수식은

```
\[ e^{i\theta} = \cos \theta + i \sin \theta \]
```

를 입력하면, 또 두 번째 수식은

```
1 \begin{equation}
2   e^{i\pi} + 1 = 0
3 \end{equation}
```

이라고 입력하면 얻을 수 있습니다. 반대로, `\(...\)` 안에 작성한 — 즉, 본문에 섞여서 나타나는 — 수식은 **행중수식** `inline math`이라고 부릅니다.

행중수식은 좁은 공간에 수식을 조판해야 하고, 별행수식은 더 넓은 공간에 자유롭게 수식을 조판할 수 있기 때문에, 같은 코드로 작성한 수식도 행중수식이나 별행수식이냐에 따라 다르게 나타납니다.

| 입력 | 출력 | 입력 | 출력 |
|---|--------------------------|--------------------------------|-----------------|
| <code>\(\frac{1}{3}\)</code> | $\frac{1}{3}$ | <code>\(\binom{10}{5}\)</code> | $\binom{10}{5}$ |
| <code>\[\frac{1}{3}\]</code> | $\frac{1}{3}$ | <code>\[\binom{10}{5}\]</code> | $\binom{10}{5}$ |
| <code>\(\lim_{n \rightarrow 0}\)</code> | $\lim_{n \rightarrow 0}$ | <code>\(\sum_{k=1}^n\)</code> | $\sum_{k=1}^n$ |
| <code>\[\lim_{n \rightarrow 0}\]</code> | $\lim_{n \rightarrow 0}$ | <code>\[\sum_{k=1}^n\]</code> | $\sum_{k=1}^n$ |

이처럼 수식이 조판되는 모습을 수식의 **스타일**이라고 부릅니다. 행중수식을 조판할 때에는 ‘행중수식 스타일’이 사용되고, 별행수식을 조판할 때에는 ‘별행수식 스타일’이 사용됩니다.

8.1.3 수식 관련 패키지

미국 수학회(American Mathematical Society, AMS)는 \TeX 개발 초기부터 \TeX 의 강력한 수식 기능을 바탕으로 하여 $\mathcal{AMS-}\TeX$ 이라는 매크로 집합을 개발하였습니다. 이 기능이 현재의 \LaTeX 까지 이어져 $\mathcal{AMS-}\LaTeX$ 이라는 클래스 및 패키지 모음으로 제공됩니다. 이 패키지는 수식을 작성할 때 유용한 명령어를 제공하고 다양한 수식 기호와 글꼴을 사용할 수 있게 해 주기 때문에, \LaTeX 으로 작성하는 많은 문서에서는 이 패키지를 불러옵니다. $\mathcal{AMS-}\LaTeX$ 의 기능은 `amsmath` 패키지[52]를 불러와 사용할 수 있으며, 미국 수학회는 이외에도 다음 패키지와 클래스를 제공합니다.

- 정의와 정리, 명제, 증명 등을 작성하기 위한 `amsthm` 패키지
- 다양한 기호를 입력할 수 있는 보조 글꼴과 독일 활자체, 칠판 볼드체, 필기체 글꼴을 제공하는 `amsfonts` 패키지
- `amsfonts` 패키지의 보조 글꼴에 있는 기호를 입력하기 위한 명령어를 정의하는 `amssymb` 패키지
- 미국 수학회 양식의 문서를 작성하기 위한 `amsart`, `amsproc`, `amsbook` 클래스
- 가환 도표를 작성하기 위한 `amscd` 패키지

이외에도 여러 유용한 패키지가 있습니다. 대표적으로 모르텐 호그홀름(Morten Høgholm)이 작성한 `mathtools` 패키지[21]는 수식 작성 기능을 더 강화해 줍니다. 패키지 내부에서 `amsmath` 패키지를 불러온 다음 이 패키지의 버그를 수정하고, 몇 가지 기능을 확장해 주는 명령어를 추가로 제공합니다. 이 책에서는 \LaTeX 의 기본 수식 작성 기능과 함께 `amsmath`, `amssymb`, `mathtools` 패키지의 기능을 함께 소개하겠습니다.

8.2 수식 기호

수식을 입력할 때에는 다양한 기호를 사용합니다. 이번 절에서는 \LaTeX 과 미국 수학회의 패키지가 제공하는 기호와 그에 대응하는 제어 문자열을 소개합니다. 기호의 의미와 형태에 따라 이항 연산자, 대형 연산자·적분 기호, 로그형 함수, 관계 기호, 화살표, 괄호·여닫는 기호, 문장 부호, 일반 기호, 그리스 문자로 구분하였습니다.

기호를 입력할 때 지키면 좋은 규칙이 몇 가지 있습니다. 이들은 반드시 지킬 필요는 없지만, 더 보기 좋은 문서를 만들고 싶다면 참고하는 것이 좋습니다. 이는 8.2.1절에서 확인할 수 있습니다.

표 8.1 | 이항 연산자

| $+$ | $+$ | $-$ | $-$ | $*$ | \star^1 | | |
|--------------------|--|------------|--------------------------------|-------------|--------------------------------|------------------|----------------------------------|
| \amalg | <code>\amalg</code> | \cup | <code>\cup</code> | \oplus | <code>\oplus</code> | \times | <code>\times</code> |
| \ast | <code>\ast</code> ¹ | \dagger | <code>\dagger</code> | \oslash | <code>\oslash</code> | \triangleleft | <code>\triangleleft</code> |
| \bigcirc | <code>\bigcirc</code> | \ddagger | <code>\ddagger</code> | \otimes | <code>\otimes</code> | \triangleright | <code>\triangleright</code> |
| \bigtriangledown | <code>\bigtriangledown</code> ³ | \diamond | <code>\diamond</code> | \pm | <code>\pm</code> | \unlhd | <code>\unlhd</code> ² |
| \bigtriangleup | <code>\bigtriangleup</code> ³ | \div | <code>\div</code> | \rhd | <code>\rhd</code> ² | \unrhd | <code>\unrhd</code> ² |
| \bullet | <code>\bullet</code> | \lhd | <code>\lhd</code> ² | \setminus | <code>\setminus</code> | \uplus | <code>\uplus</code> |
| \cap | <code>\cap</code> | \mp | <code>\mp</code> | \sqcap | <code>\sqcap</code> | \vee | <code>\vee</code> ³ |
| \cdot | <code>\cdot</code> | \odot | <code>\odot</code> | \sqcup | <code>\sqcup</code> | \wedge | <code>\wedge</code> ³ |
| \circ | <code>\circ</code> | \ominus | <code>\ominus</code> | \star | <code>\star</code> | \wr | <code>\wr</code> |

amssymb 패키지에서 제공하는 기호:

| | | | | | | | |
|---|--------------------------------|---|--------------------------------|---|------------------------------|---|-------------------------------|
| ⌋ | <code>\barwedge</code> | ⊛ | <code>\circledast</code> | ⋈ | <code>\dotplus</code> | ⋈ | <code>\rightthreetimes</code> |
| ⊠ | <code>\boxdot</code> | ⊙ | <code>\circledcirc</code> | ⌋ | <code>\doublebarwedge</code> | ⋈ | <code>\rtimes</code> |
| ⊞ | <code>\boxminus</code> | ⊖ | <code>\circleddash</code> | ⋈ | <code>\gtrdot</code> | ⋈ | <code>\smallsetminus</code> |
| ⊕ | <code>\boxplus</code> | ⊔ | <code>\Cup</code> ⁴ | ⋈ | <code>\intercal</code> | ⋈ | <code>\veebar</code> |
| ⊗ | <code>\boxtimes</code> | ∪ | <code>\curlyvee</code> | ⋈ | <code>\leftthreetimes</code> | | |
| ⌋ | <code>\Cap</code> ⁴ | ∩ | <code>\curlywedge</code> | ⋈ | <code>\lessdot</code> | | |
| • | <code>\centerdot</code> | ⋈ | <code>\divideontimes</code> | ⋈ | <code>\ltimes</code> | | |

¹ *와 \ast는 같은 기호를 출력합니다.

² latexsym 또는 amsfonts 패키지를 불러와야 사용할 수 있습니다.

³ ▽와 △은 각각 \varbigtriangledown과 \varbigtriangleup으로도, ∨와 ∧는 각각 \lor와 \land로도 입력할 수 있습니다.

⁴ ⌋과 ⊔은 각각 \doublecap과 \doublecup으로도 입력할 수 있습니다.

이항 연산자

두 대상 사이의 연산을 나타내는 이항 연산자는 표 8.1의 명령어를 사용해 입력합니다. 자주 쓰이는 것들을 모으면 아래와 같습니다.

입력

`\(a + b - c\)`
`\(a \times b \div c\)`
`\(a \pm b \mp c\)`
`\(a \cdot b \circ c\)`
`\(a \oplus b \otimes c\)`
`\(a \wedge b \vee c\)`
`\(a \cap b \cup c \setminus d\)`

출력

$a + b - c$
 $a \times b \div c$
 $a \pm b \mp c$
 $a \cdot b \circ c$
 $a \oplus b \otimes c$
 $a \wedge b \vee c$
 $a \cap b \cup c \setminus d$

표 8.2 | 대형 연산자와 적분 기호

| | | | |
|---------------------------------|--------------------------------------|------------------------------------|-------------------------------|
| \bigcap <code>\bigcap</code> | \bigotimes <code>\bigotimes</code> | \bigwedge <code>\bigwedge</code> | \prod <code>\prod</code> |
| \bigcup <code>\bigcup</code> | \bigsqcup <code>\bigsqcup</code> | \coprod <code>\coprod</code> | \int <code>\smallint</code> |
| \odot <code>\bigodot</code> | \biguplus <code>\biguplus</code> | \int <code>\int</code> | \sum <code>\sum</code> |
| \oplus <code>\bigoplus</code> | \bigvee <code>\bigvee</code> | \oint <code>\oint</code> | |

amsmath 패키지에서 제공하는 기호:

| | | | |
|----------------------------|------------------------------|------------------------------|--|
| \iint <code>\iint</code> | \iiint <code>\iiint</code> | \iiint <code>\iiint</code> | $\int \dots \int$ <code>\int \dots \int</code> |
|----------------------------|------------------------------|------------------------------|--|

대형 연산자와 적분 기호

합의 기호 ‘ \sum ’나 곱의 기호 ‘ \prod ’ 등의 대형 연산자나 ‘ \int ’ 등의 적분 기호는 표 8.2에서 확인할 수 있습니다. 왼쪽의 기호는 행중수식 스타일에서의 모습, 오른쪽의 기호는 별행수식 스타일에서의 모습입니다.

입력

`\(\sum_{n=1}^{\infty} a_n\)`

`\[\sum_{n=1}^{\infty} a_n\]`

`\(\prod_{i \neq j} k_i\)`

`\[\prod_{i \neq j} k_i\]`

`\(\int_a^x f(t) \, dt\)`

`\[\int_a^x f(t) \, dt\]`

출력

$\sum_{n=1}^{\infty} a_n$

$\sum_{n=1}^{\infty} a_n$

$\prod_{i \neq j} k_i$

$\prod_{i \neq j} k_i$

$\int_a^x f(t) dt$

$\int_a^x f(t) dx$

참자 문법
→ 116쪽

위의 예시에서 볼 수 있듯, 대형 연산자와 적분 기호의 계산 범위는 참자 문법을 사용해 입력합니다. 입력한 참자는 수식의 스타일이나 앞의 기호에 따라 올바른 위치에 배치됩니다. 또, 적분식을 작성할 때에는 피적분함수와 적분 변수 사이에 `\,`를 입력해 약간의 공백을 넣는 것이 좋습니다.

로그형 함수

삼각함수나 로그함수, 극한 연산자 등 로그형 함수 log-like function는 로마자 정체로 적습니다. 이들은 각각의 이름 앞에 역슬래시를 붙인 `\sin`, `\log`, `\lim` 등의 제어 문자열을 사용해 입력합니다. 특히, 극한이나 최대·최소 등을 나타내는 연산자는 앞과 같이 참자

표 8.3 | 로그형 함수

| | | | | | | | | | |
|---------------------|----------------------|-------------------|-------------------------------|----------------------|----------------------------------|------------------|-------------------------------|-------------------|-------------------------------|
| <code>arccos</code> | <code>\arccos</code> | <code>coth</code> | <code>\coth</code> | <code>hom</code> | <code>\hom</code> | <code>ln</code> | <code>\ln</code> | <code>sinh</code> | <code>\sinh</code> |
| <code>arcsin</code> | <code>\arcsin</code> | <code>csc</code> | <code>\csc</code> | <code>inf</code> | <code>\inf¹</code> | <code>log</code> | <code>\log</code> | <code>sup</code> | <code>\sup¹</code> |
| <code>arctan</code> | <code>\arctan</code> | <code>deg</code> | <code>\deg</code> | <code>ker</code> | <code>\ker</code> | <code>max</code> | <code>\max¹</code> | <code>tan</code> | <code>\tan</code> |
| <code>arg</code> | <code>\arg</code> | <code>det</code> | <code>\det¹</code> | <code>lg</code> | <code>\lg</code> | <code>min</code> | <code>\min¹</code> | <code>tanh</code> | <code>\tanh</code> |
| <code>cos</code> | <code>\cos</code> | <code>dim</code> | <code>\dim¹</code> | <code>lim</code> | <code>\lim¹</code> | <code>Pr</code> | <code>\Pr¹</code> | | |
| <code>cosh</code> | <code>\cosh</code> | <code>exp</code> | <code>\exp¹</code> | <code>lim inf</code> | <code>\liminf¹</code> | <code>sec</code> | <code>\sec</code> | | |
| <code>cot</code> | <code>\cot</code> | <code>gcd</code> | <code>\gcd¹</code> | <code>lim sup</code> | <code>\limsup¹</code> | <code>sin</code> | <code>\sin</code> | | |

amsmath 패키지에서 제공하는 기호:

| | | | | | |
|----------------------|-----------------------------------|---------------|--------------------------------------|---------------|--------------------------------------|
| <code>injlim</code> | <code>\injlim¹</code> | \varinjlim | <code>\varinjlim¹</code> | \varinjlim | <code>\varinjlim¹</code> |
| <code>projlim</code> | <code>\projlim¹</code> | \varprojlim | <code>\varprojlim¹</code> | \varprojlim | <code>\varprojlim¹</code> |

¹ 별행수식 스타일에서 이 제어 문자열 뒤에 입력된 첨자는 대형 연산자처럼 기호의 위아래에 조판됩니다.

문법으로 계산 범위를 지정해 줄 수 있습니다. \LaTeX 에서 사용할 수 있는 로그형 함수는 표 8.3에서 확인할 수 있습니다.

| 입력 | 출력 | 입력 | 출력 |
|---------------------------|------------|--|-------------------------------|
| <code>\(\sin x\)</code> | $\sin x$ | <code>\(\lim f(x)\)</code> | $\lim f(x)$ |
| <code>\(\sin^2 x\)</code> | $\sin^2 x$ | <code>\(\lim_{x \rightarrow 0} f(x)\)</code> | $\lim_{x \rightarrow 0} f(x)$ |
| <code>\[\sin^2 x\]</code> | $\sin^2 x$ | <code>\[\lim_{x \rightarrow 0} f(x)\]</code> | $\lim_{x \rightarrow 0} f(x)$ |

이 표에 없는 로그형 함수를 입력할 때에는 그 이름을 `\operatorname{(*)}` 제어 문자열의 인자로 지정하면 됩니다(이 제어 문자열은 amsmath 패키지에서 제공합니다). 극한 `lim` 이나 상한 `sup`처럼 계산 범위를 적어 주어야 하는 경우에는 별표 있는 것을, 그렇지 않은 경우에는 별표 없는 것을 사용하면 됩니다.

| 입력 | 출력 |
|---|------------------------------------|
| <code>\[\operatorname{arsinh}^{-1}x\]</code> | $\operatorname{arsinh}^{-1} x$ |
| <code>\[\operatorname{rank}AB\]</code> | $\operatorname{rank} AB$ |
| <code>\[\operatorname{lcm}_{n \in A} a_n\]</code> | $\operatorname{lcm}_{n \in A} a_n$ |

자주 사용하는 함수가 있다면 이를 입력하는 짧은 제어 문자열을 정의해 두는 것이 편리합니다. 이때에는 전처리부에서 amsmath 패키지의 `\DeclareMathOperator{(*)}` 제어 문자열을 사용합니다. 앞과 마찬가지로, 별표 붙은 것은 계산 범위를 적어 주어야 하는 경우에 사용합니다.

`\DeclareMathOperator(*){<csname>}{<name>}`

`<csname>`: 연산자를 입력하기 위해 정의할 제어 문자열

`<name>`: 연산자의 이름

예를 들어, 전처리부에

```
1 \DeclareMathOperator{\sech}{sech}
2 \DeclareMathOperator{\csch}{csch}
```

를 입력해 두면 쌍곡선 시컨트함수 `sech`와 쌍곡선 코시컨트함수 `csch`를 `\sech`, `\csch`로 입력할 수 있습니다(이 두 제어 문자열은 \LaTeX 이 제공하지 않습니다).

법 연산자

법 연산자 ‘`mod`’는 로마자 정체로 적긴 하지만, 다른 로그형 함수와 다르게 입력합니다. \LaTeX 은 이항 연산자로서 입력하는 `\bmod`(binary ‘`mod`’)와 괄호 안에 법을 작성하는 `\pmod`(parenthesized ‘`mod`’) 제어 문자열을 제공합니다. `\pmod` 제어 문자열의 인자로 법으로써 사용할 값(또는 식)을 입력해 주면 됩니다.

입력

```
\(5 \bmod 3 = 2\)
\((5 \equiv 2 \pmod{3})\)
```

출력

```
5 mod 3 = 2
5 ≡ 2 (mod 3)
```

`amsmath` 패키지는 `\pmod`를 변형한 `\mod`와 `\pod`를 제공합니다. `\pmod`와 달리 `\mod`는 괄호를 조판하지 않고, `\pod`는 ‘`mod`’를 조판하지 않습니다.

입력

```
\(5 \equiv 2 \mod{3}\)
\((5 \equiv 2 \pod{3})\)
```

출력

```
5 ≡ 2 mod 3
5 ≡ 2 (3)
```

관계 기호

두 대상 사이의 관계를 나타내는 관계 기호는 표 8.4의 제어 문자열로 입력합니다. 자주 쓰이는 것들을 모으면 아래와 같습니다.

입력

```
\(a = b \equiv c\)
\((a \sim b \approx c)\)
\((a \simeq b \cong c)\)
\((a \vdash b \vdash c)\)
```

출력

```
 $a = b \equiv c$ 
 $a \sim b \approx c$ 
 $a \simeq b \cong c$ 
 $a \vdash b \vdash c$ 
```

표 8.4 | 관계 기호

| : : | < < | = = | > > |
|-------------------|----------------------------|--------------------------|--|
| \approx \approx | \geq \geq ¹ | \neq \neq ¹ | \sim \sim |
| \asymp \asymp | \gg \gg | \ni \ni ¹ | \simeq \simeq |
| \bowtie \bowtie | \in \in | \notin \notin | \smile \smile |
| \cong \cong | \Join \Join ² | \parallel \parallel | \sqsubset \sqsubset ² |
| \dashv \dashv | \leq \leq ¹ | \perp \perp | \sqsubseteq \sqsubseteq ² |
| \doteq \doteq | \ll \ll | \prec \prec | \sqsupset \sqsupset ² |
| \equiv \equiv | \mid \mid | \preceq \preceq | \sqsupseteq \sqsupseteq |
| \frown \frown | \models \models | \propto \propto | \subset \subset |

amssymb 패키지에서 제공하는 기호:

| | | | | |
|---|-----------------------------|---------------------------------------|---------------------------------|---------------------------------------|
| \approx \approxeq | \gtrsim \gnsim | \ngeq \ngeq | \nVdash \nVdash | \succapprox \succapprox |
| \backsimeq \backepsilon | \gtrapprox \gtrapprox | \ngeqslant \ngeqslant | \nvDash \nvDash | \succneq \succneq |
| \backsim \backsim | \gtreqless \gtreqless | \ngtr \ngtr | \nvdash \nvdash | \succnsim \succnsim |
| \backsimeq \backsimeq | \gtreqqlless \gtreqqlless | \nleq \nleq | \pitchfork \pitchfork | \succsim \succsim |
| \because \because | \gtrless \gtrless | \nleqq \nleqq | \precapprox \precapprox | \supset \supset |
| \between \between | \gtrsim \gtrsim | \nleqslant \nleqslant | $\prec curlyeq$ \prec curlyeq | \supseteq \supseteq |
| \blacktriangleleft \blacktriangleleft | | \nless \nless | \precnapprox \precnapprox | \supsetneq \supsetneq |
| \blacktriangleright \blacktriangleright | | \nmid \nmid | \precneq \precneq | \supsetneqq \supsetneqq |
| \bumpeq \bumpeq | \gvertneqq \gvertneqq | \nparallel \nparallel | \precnsim \precnsim | \therefore \therefore |
| \bumpeq \bumpeq | \leqq \leqq | \nprec \nprec | $\prec sim$ \prec sim | \thickapprox \thickapprox |
| \circeq \circeq | \leqslant \leqslant | \npreceq \npreceq | \risingdotseq \risingdotseq | \thicksim \thicksim |
| \curlyeqprec \curlyeqprec | \lessapprox \lessapprox | \nshortmid \nshortmid | \shortmid \shortmid | \trianglelefteq \trianglelefteq |
| \curlyeqsucc \curlyeqsucc | \lesseqgtr \lesseqgtr | \nshortparallel \nshortparallel | | \trianglelefteq \trianglelefteq |
| \doteqdot \doteqdot ³ | \lesseqqgtr \lesseqqgtr | \nsim \nsim | \shortparallel \shortparallel | \trianglerighteq \trianglerighteq |
| \eqcirc \eqcirc | \lessgtr \lessgtr | \nsubseteq \nsubseteq | \smallfrown \smallfrown | \varpropto \varpropto |
| \eqsim \eqsim | \lessssim \lessssim | \nsubseteqeq \nsubseteqeq | \smallsmile \smallsmile | \varsubsetneq \varsubsetneq |
| \eqslantgtr \eqslantgtr | \lll \lll ³ | \nsucc \nsucc | \Subset \Subset | \varsubsetneqq \varsubsetneqq |
| \eqslantless \eqslantless | \lnapprox \lnapprox | \nsucceq \nsucceq | \subseteq \subseteq | \varsupsetneq \varsupsetneq |
| \fallingdotseq \fallingdotseq | \lneq \lneq | \nsupseteq \nsupseteq | \subseteq \subseteq | \varsupsetneqq \varsupsetneqq |
| \geqq \geqq | \lneqq \lneqq | \nsupseteqeq \nsupseteqeq | \subseteq \subseteq | \vartriangle \vartriangle |
| \geqslant \geqslant | \lnsim \lnsim | \ntriangleleft \ntriangleleft | \succapprox \succapprox | \vartriangleleft \vartriangleleft |
| \ggg \ggg ³ | \lvertneqq \lvertneqq | \ntrianglelefteq \ntrianglelefteq | | \vartriangleright \vartriangleright |
| \gnapprox \gnapprox | \multimap \multimap | \ntriangleright \ntriangleright | | \Vdash \Vdash |
| \gneq \gneq | \ncong \ncong | \ntrianglerighteq \ntrianglerighteq | | \VDash \VDash |
| \gneqq \gneqq | \ngeq \ngeq | \nVDash \nVDash | \succcurlyeq \succcurlyeq | \Vdash \Vdash |

¹ \geq, \leq, \neq 는 각각 \ge, \le, \ne로도 입력할 수 있습니다. \ni 는 \owns로도 입력할 수 있습니다.

² latexsym 또는 amsfonts 패키지를 불러와야 사용할 수 있습니다.

³ \doteq 는 \Doteq로도 입력할 수 있습니다. \ggg 와 \lll 는 각각 \gggtr, \lllless로도 입력할 수 있습니다.

입력

$\backslash(a < b \backslash \leq c \backslash)$
 $\backslash(a \backslash \geq b > c \backslash)$
 $\backslash(a \backslash \prec b \backslash \preceq c \backslash)$
 $\backslash(a \backslash \succeq b \backslash \succ c \backslash)$
 $\backslash(a \backslash \vartriangleleft b \backslash \trianglelefteq c \backslash)$
 $\backslash(a \backslash \triangleright b \backslash \trianglerighteq c \backslash)$
 $\backslash(a \backslash \subset b \backslash \subseteq c \backslash)$
 $\backslash(a \backslash \supseteq b \backslash \supset c \backslash)$
 $\backslash(a \backslash \in b \backslash \ni c \backslash)$
 $\backslash(a \backslash \mid b \backslash \parallel c \backslash)$
 $\backslash(a \backslash \perp b \backslash)$
 $\backslash(a : b \backslash)$

출력

$a < b \leq c$
 $a \geq b > c$
 $a \prec b \preceq c$
 $a \succeq b \succ c$
 $a \triangleleft b \trianglelefteq c$
 $a \triangleright b \trianglerighteq c$
 $a \subset b \subseteq c$
 $a \supseteq b \supset c$
 $a \in b \ni c$
 $a \mid b \parallel c$
 $a \perp b$
 $a : b$

기호를 입력하는 제어 문자열 앞에 $\backslash not$ 을 붙이면 기호에 빗금이 그려져 부정 기호를 출력할 수 있습니다. `amssymb` 패키지를 불러오면 많은 부정 기호를 사용할 수 있으므로, 이 패키지의 기호를 사용하는 것도 좋습니다.

입력

$\backslash(a \backslash \neq b \backslash \not\equiv c \backslash)$
 $\backslash(a \backslash \not\sim b \backslash \ncong c \backslash)$
 $\backslash(a \backslash \subsetneq b \backslash \subsetneqq c \backslash)$
 $\backslash(a \backslash \nsupseteq b \backslash \supsetneqq c \backslash)$
 $\backslash(a \backslash \notin b \backslash)$
 $\backslash(a \backslash \nmid b \backslash \nparallel c \backslash)$

출력

$a \neq b \not\equiv c$
 $a \not\sim b \ncong c$
 $a \subsetneq b \subsetneqq c$
 $a \nsupseteq b \supsetneqq c$
 $a \notin b$
 $a \nmid b \nparallel c$

mathtools 패키지의 관계 기호

쌍점 기호 ‘:’를 자세히 보면 세로로 가운데에 있지 않아, 기호 ‘:=’ 처럼 다른 관계 기호와 어울려 사용했을 때 어색함을 느낄 수도 있습니다. `mathtools` 패키지는 이를 보정한 쌍점을 조판하는 $\backslash vcentcolon$ 을 정의합니다.

입력

$\backslash(a : b \backslash)$
 $\backslash(f := g \backslash)$

출력

$a : b$
 $f := g$

입력

$\backslash(a \backslash vcentcolon b \backslash)$
 $\backslash(f \backslash vcentcolon= g \backslash)$

출력

$a : b$
 $f := g$

`mathtools` 패키지는 높이가 조정된 쌍점을 포함한 관계 기호를 추가로 정의하며, 이를 표 8.5에서 확인할 수 있습니다.

표 8.5 | mathtools 패키지: 관계 기호

| | | | | | | | |
|-------------|---------------------------|-------------|-------------------------|--------------|-------------------------|--------------|-----------------------------|
| $\approx::$ | <code>\Approxcolon</code> | \coloneq | <code>\colondash</code> | \dashcolon | <code>\Dashcolon</code> | $:$ | <code>\ordinarycolon</code> |
| \approx | <code>\approxcolon</code> | \coloneq | <code>\Coloneq</code> | \dashcolon | <code>\dashcolon</code> | \sim | <code>\Simcolon</code> |
| \approx | <code>\Colonapprox</code> | \coloneq | <code>\coloneq</code> | \dcolon | <code>\dblcolon</code> | \sim | <code>\simcolon</code> |
| \approx | <code>\colonapprox</code> | \colonsim | <code>\Colonsim</code> | \eqcolon | <code>\Eqcolon</code> | \centcolon | <code>\vcentcolon</code> |
| \coloneq | <code>\Colondash</code> | \colonsim | <code>\colonsim</code> | \eqcolon | <code>\eqcolon</code> | | |

화살표

각종 화살표는 표 8.6의 제어 문자열로 입력합니다. 명제 사이의 동치 관계를 나타낼 때에는 `\Longlefttrightarrow` 대신 화살표 좌우에 넓은 공간을 두는 `\iff`를 사용합니다. 함의 관계를 나타낼 때에도 같은 이유로 `\implies`와 `\impliedby`를 사용합니다.

`\[x^2 = a, x>0 \iff x = \sqrt{a} \]`

$$x^2 = a, x > 0 \iff x = \sqrt{a}$$

괄호

표 8.7에서는 \LaTeX 에서 사용할 수 있는 괄호를 확인할 수 있습니다. 느낌표와 물음표를 제외하고, 홀수 열에 있는 것은 여는 것, 짝수 열에 있는 것은 닫는 것으로 취급됩니다.

입력

`\((n-r)!\,r!\)`

`\(\{x \mid f(x) = 0\}\)`

`\(\langle g_1, g_2 \mid r \rangle\)`

`\(\lfloor x \rfloor + \lceil y \rceil\)`

출력

$(n-r)!r!$

$\{x \mid f(x) = 0\}$

$\langle g_1, g_2 \mid r \rangle$

$\lfloor x \rfloor + \lceil y \rceil$

특이 사항으로, 느낌표와 물음표는 문장 부호가 아니라 닫는 괄호처럼 취급됩니다. 팩토리얼 뒤에 숫자나 문자 등이 이어진다면 위 예시와 같이 `\,`를 입력해 약간의 공백을 만드는 것이 좋습니다.

또, 이 표에는 없지만 괄호로써 자주 사용되는 기호로 세로선을 조판하는 `|`와 겹세로선을 조판하는 `\|`도 있습니다. 이들은 표 8.9에서 확인할 수 있습니다.

입력

`\(|a||x|)\)`

`\(\lvert \sin x \rvert\)`

출력

$|a||x|$

$|\sin x|$

괄호 크기 조절
→ 118쪽

괄호 사이에 분수와 같이 크기가 큰 식이 올 때에는 8.4절의 명령어를 사용하여 괄호 사이의 식을 감싸도록 괄호의 높이를 조절할 수 있습니다.

| 표 8.6 | 화살표

| | | | | | | | |
|------------------------|--|--------------------------|-------------------------------------|---------------------|--------------------------------------|----------------------|--|
| \Downarrow | <code>\Downarrow</code> | \leftarrow | <code>\leftharpoonup</code> | \longrightarrow | <code>\longrightarrow</code> | \rightarrow | <code>\rightharpoonup</code> |
| \downarrow | <code>\downarrow</code> | \Leftrightarrow | <code>\Leftrightarrow</code> | \mapsto | <code>\mapsto</code> | \Rrightarrow | <code>\Rrightarrow</code> |
| \hookleftarrow | <code>\hookleftarrow</code> | \leftrightharpoons | <code>\leftrightharpoons</code> | \nearrow | <code>\nearrow</code> | \searrow | <code>\searrow</code> |
| \hookrightarrow | <code>\hookrightarrow</code> | \Longleftarrow | <code>\Longleftarrow</code> | \nwarrow | <code>\nwarrow</code> | \swarrow | <code>\swarrow</code> |
| \Leftrightarrow | <code>\iff¹</code> | \longleftarrow | <code>\longleftarrow</code> | \Rightarrow | <code>\Rightarrow</code> | \Uparrow | <code>\Uparrow</code> |
| \leadsto | <code>\leadsto²</code> | \Longleftrightharpoons | <code>\Longleftrightharpoons</code> | | | \Uparrow | <code>\uparrow</code> |
| \Leftarrow | <code>\Leftarrow</code> | \longleftrightharpoons | <code>\longleftrightharpoons</code> | | | \Updownarrow | <code>\Updownarrow</code> |
| \leftarrow | <code>\leftarrow³</code> | \longmapsto | <code>\longmapsto</code> | \rightarrow | <code>\rightarrow³</code> | \downarrow | <code>\downarrow</code> |
| \leftharpoondown | <code>\leftharpoondown</code> | \Longrightarrow | <code>\Longrightarrow</code> | \rightharpoondown | <code>\rightharpoondown</code> | | |
| amsmath 패키지에서 제공하는 기호: | | | | \Rightarrow | <code>\implies¹</code> | \Leftarrow | <code>\impliedby¹</code> |
| amssymb 패키지에서 제공하는 기호: | | | | | | | |
| \circlearrowleft | <code>\circlearrowleft</code> | \leftleftarrows | <code>\leftleftarrows</code> | \nLeftrightarrow | <code>\nLeftrightarrow</code> | \Rightarrow | <code>\Rightarrow</code> |
| \circlearrowright | <code>\circlearrowright</code> | \leftrightarrows | <code>\leftrightarrows</code> | \nleq | <code>\nleq</code> | \Rsh | <code>\Rsh</code> |
| \curvearrowleft | <code>\curvearrowleft</code> | \leftrightharpoons | <code>\leftrightharpoons</code> | | | \twoheadleftarrow | <code>\twoheadleftarrow</code> |
| \curvearrowright | <code>\curvearrowright</code> | \leftrightsquigarrow | <code>\leftrightsquigarrow</code> | | | \twoheadrightarrow | <code>\twoheadrightarrow</code> |
| \dashleftarrow | <code>\dashleftarrow⁴</code> | \Lleftarrow | <code>\Lleftarrow</code> | \nRightarrow | <code>\nRightarrow</code> | \upharpoonleft | <code>\upharpoonleft</code> |
| \dashrightarrow | <code>\dashrightarrow⁴</code> | \looparrowleft | <code>\looparrowleft</code> | \nrightarrow | <code>\nrightarrow</code> | \upharpoonright | <code>\upharpoonright⁵</code> |
| \downdownarrows | <code>\downdownarrows</code> | \looparrowright | <code>\looparrowright</code> | \rightarrowtail | <code>\rightarrowtail</code> | \upuparrows | <code>\upuparrows</code> |
| \downharpoonleft | <code>\downharpoonleft</code> | \Lsh | <code>\Lsh</code> | \rightleftarrows | <code>\rightleftarrows</code> | | |
| \downharpoonright | <code>\downharpoonright</code> | \nLeftarrow | <code>\nLeftarrow</code> | \rightrightarrows | <code>\rightrightarrows</code> | | |
| \leftarrowtail | <code>\leftarrowtail</code> | \nleq | <code>\nleq</code> | \rightsquigarrow | <code>\rightsquigarrow</code> | | |

¹ 기호 좌우의 간격이 같은 모양의 다른 기호보다 넓습니다.

² latexsym 패키지 또는 amsfonts 패키지를 불러와야 사용할 수 있습니다.

³ \leftarrow 는 `\gets`로도, \rightarrow 는 `\to`로도 입력할 수 있습니다.

⁴ amsfonts 패키지를 불러와도 사용할 수 있습니다.

⁵ `\restriction`으로도 입력할 수 있습니다.

문장 부호

수식의 문장 부호는 표 8.8의 명령어로 입력합니다. 일상 언어의 문장 부호는 문장의 휴지나 종지를 나타내지만, 수식 문장 부호는 나열된 대상이나 표현을 구분 짓습니다.

입력

`\(\lambda x \ldotp f(x)\)`
`\(f \colon X \to Y\)`
`\[x_1 + x_2 + \cdots + x_n]`
`\[a_1, a_2, \ldots, a_n]`

출력

$\lambda x. f(x)$
 $f: X \rightarrow Y$
 $x_1 + x_2 + \cdots + x_n$
 a_1, a_2, \dots, a_n

표 8.7 | 괄호

| | | | | | | | |
|-------------------------|-------------|-------------|-------------|-----------|-----------|-----------|-----------|
| $($ | $(^1$ | $)$ | $)^1$ | $[$ | $[^2$ | $]$ | $]^2$ |
| $!$ | $!$ | $?$ | $?$ | | | | |
| $\{$ | $\{^3$ | $\}$ | $\}^3$ | \langle | \langle | \rangle | \rangle |
| \lceil | \lceil | \rceil | \rceil | \lfloor | \lfloor | \rfloor | \rfloor |
| amsmath 패키지에서 제공하는 기호: | | | | | | | |
| \lvert | \lvert | \rvert | \rvert | $\! $ | $\! $ | $\! $ | $\! $ |
| amsfonts 패키지에서 제공하는 기호: | | | | | | | |
| \ulcorner | \urcorner | \llcorner | \lrcorner | | | | |

¹ mathtools 패키지를 불러온 경우, `\lparen`과 `\rparen`으로도 입력할 수 있습니다.

² 대괄호는 `\lbrack`과 `\rbrack`으로 입력할 수 있습니다.

³ {와 }는 `\lbrace`와 `\rbrace`로도 입력할 수 있습니다.

표 8.8 | 문장 부호

| | | | | | |
|------------------------|----------|----------|----------|----------|----------|
| $,$ | $,$ | $;$ | $;$ | | |
| \cdot | \cdot | \cdot | \cdot | $:$ | $:$ |
| \ldots | \ldots | \cdots | \cdots | \vdots | \vdots |
| \ldots^1 | \ldots | \cdots | \cdots | \ddots | \ddots |
| amsmath 패키지에서 제공하는 기호: | | | | | |
| \dotsb | \dotsc | \dotsi | \dotsm | \dotso | |

¹ ...는 `\mathellipsis`로도 입력할 수 있습니다. amsmath 패키지를 불러오지 않았다면 `\dots`를, 불러왔다면 `\hdots`를 사용할 수도 있습니다.

줄임표 줄임표를 입력하는 명령어에는 `\dots`도 있습니다. amsmath 패키지를 불러오지 않았다면 `\ldots`와 동일한 기호를 조판하고, amsmath 패키지를 불러온 경우에는 상황에 따라 `\ldots`와 `\cdots` 중 적절한 것을 골라 줍니다.

→ 110쪽

일반 기호

앞의 어떤 분류에도 속하지 않는 일반 기호는 표 8.9에 정리하였습니다. 프라임 기호를 입력할 때에는 기호 ‘ $'$ ’을 조판하는 제어 문자열 `\prime`을 위 첨자로 입력하거나, 이와 같은 역할을 하는 ‘ $'$ ’를 입력합니다.

입력

`\(f' = g^{\prime}\)`
`\(f'' = g^{\prime\prime}\)`

출력

$f' = g'$
 $f'' = g''$

| 표 8.9 | 일반 기호

| . | . | ' | ' | / | / | | ¹ |
|-------------------------|-----------------------------------|--------------|---------------------------|-------------------------------|---|---------------------|-----------------------|
| \parallel | $\ $ ² | \diamond | <code>\diamondsuit</code> | \Im | <code>\Im</code> | \mho ³ | \spadesuit |
| \aleph | <code>\aleph</code> | ℓ | <code>\ell</code> | \imath | <code>\imath</code> | ∇ | <code>\nabla</code> |
| \angle | <code>\angle</code> | \emptyset | <code>\emptyset</code> | ∞ | <code>\infty</code> | \natural | <code>\natural</code> |
| \backslash | <code>\backslash</code> | \exists | <code>\exists</code> | \jmath | <code>\jmath</code> | \neg ² | \triangle |
| \bot | <code>\bot</code> | \flat | <code>\flat</code> | \mathdollar ⁴ | <code>\mathdollar⁴</code> | ∂ | <code>\partial</code> |
| \Box ³ | <code>\Box³</code> | \forall | <code>\forall</code> | \mathparagraph ⁴ | <code>\mathparagraph⁴</code> | \prime | <code>\prime</code> |
| \clubsuit | <code>\clubsuit</code> | \hbar | <code>\hbar</code> | \mathsection ⁴ | <code>\mathsection⁴</code> | \Re | <code>\Re</code> |
| \Diamond ³ | <code>\Diamond³</code> | \heartsuit | <code>\heartsuit</code> | \mathsterling ⁴ | <code>\mathsterling⁴</code> | \sharp | <code>\sharp</code> |

amssymb 패키지에서 제공하는 기호:

| | | | |
|--|--|--|--|
| \backprime <code>\backprime</code> | \blacktriangledown <code>\blacktriangledown</code> | \Finv <code>\Finv</code> | \sphericalangle <code>\sphericalangle</code> |
| \Bbbk <code>\Bbbk</code> | \textcircled{S} <code>\circledS</code> | \Game <code>\Game</code> | \square <code>\square</code> |
| \beth <code>\beth</code> | \complement <code>\complement</code> | \gimel <code>\gimel</code> | \triangledown <code>\triangledown</code> |
| \bigstar <code>\bigstar</code> | \daleth <code>\daleth</code> | \hslash <code>\hslash</code> | \varnothing <code>\varnothing</code> |
| \blacklozenge <code>\blacklozenge</code> | \diagdown <code>\diagdown</code> | \lozenge <code>\lozenge</code> | |
| \blacksquare <code>\blacksquare</code> | \diagup <code>\diagup</code> | \measuredangle <code>\measuredangle</code> | |
| \blacktriangle <code>\blacktriangle</code> | \eth <code>\eth</code> | \nexists <code>\nexists</code> | |

¹ |는 `\vert`로도 입력할 수 있습니다.

² ||는 `\Vert`로도, \neg 은 `\not`으로도 입력할 수 있습니다.

³ `latexsym` 또는 `amsfonts` 패키지를 불러와야 사용할 수 있습니다.

⁴ 표 4.6의 더 짧은 제어 문자열을 사용해 입력할 수도 있습니다.

그리스 문자

그리스 문자는 표 8.10의 명령어로 입력합니다. 간단히 역슬래시 뒤에 입력하려는 문자의 이름을 입력하면 되며, 첫 글자를 대문자로 입력하면 대문자를 얻습니다. 몇 가지 소문자는 역슬래시와 그리스 문자의 이름 사이에 ‘var’를 입력해 변형된 꼴의 문자를 얻을 수 있습니다. 또, `amsmath` 패키지를 불러오면 대문자를 입력할 때 ‘var’를 입력해 이탤릭체 대문자를 조판할 수 있습니다.

입력

`\(\Pi, \varPi\quad \Phi, \varPhi\)`
`\(\pi, \varpi\quad \phi, \varphi\)`

출력

Π, Π Φ, Φ
 π, π ϕ, ϕ

소문자 오미크론이나 대문자 알파, 베타처럼 같은 모양의 로마자가 있는 경우에는 `\omicron`이나 `\Alpha`, `\Beta` 등의 제어 문자열이 정의되어 있지 않으며, 대응하는 로마자를 입력해야 합니다.

| 표 8.10 | 그리스 문자

| | | | | |
|---------------------|-------------------|---------------------------|-----------------------|---------------------|
| α \alpha | ι \iota | σ \sigma | ϑ \vartheta | Λ \Lambda |
| β \beta | κ \kappa | τ \tau | ϖ \varpi | Ξ \Xi |
| γ \gamma | λ \lambda | υ \upsilon | ϱ \varrho | Π \Pi |
| δ \delta | μ \mu | ϕ \phi | ς \varsigma | Σ \Sigma |
| ϵ \epsilon | ν \nu | χ \chi | φ \varphi | Υ \Upsilon |
| ζ \zeta | ξ \xi | ψ \psi | Γ \Gamma | Φ \Phi |
| η \eta | π \pi | ω \omega | Δ \Delta | Ψ \Psi |
| θ \theta | ρ \rho | ε \varepsilon | Θ \Theta | Ω \Omega |

amsmath 패키지에서 제공하는 기호:

| | | | | |
|--------------------|----------------------|--------------------|------------------------|--------------------|
| Γ \varGamma | Λ \varLambda | Π \varPi | Υ \varUpsilon | Ψ \varPsi |
| Δ \varDelta | Ξ \varXi | Σ \varSigma | Φ \varPhi | Ω \varOmega |
| Θ \varTheta | | | | |

amssymb 패키지에서 제공하는 기호:

| | |
|--------------|-----------------------|
| F \digamma | \varkappa \varkappa |
|--------------|-----------------------|

| 표 8.11 | 수식 악센트

| | | | | |
|----------------------|----------------------|----------------------------|---|---|
| \acute{a} \acute a | \check{a} \check a | \grave{a} \grave a | \tilde{a} \tilde a | \widetilde{a}^1 \widetilde a ¹ |
| \bar{a} \bar a | \ddot{a} \ddot a | \hat{a} \hat a | \vec{a} \vec a | |
| \breve{a} \breve a | \dot{a} \dot a | \mathring{a} \mathring a | \widehat{a}^1 \widehat a ¹ | |

amsmath 패키지에서 제공하는 기호:

| | |
|----------------------|---------------------|
| \ddot{a} \ddddot a | \ddot{a} \dddot a |
|----------------------|---------------------|

¹ 이 악센트는 인자로 지정된 내용에 따라 길이가 늘어납니다.

악센트

표 8.11의 기호는 다른 수식 문자 위에 표시하는 악센트들입니다. 제어 문자열의 인자로 지정한 문자 위에 악센트를 표시해 줍니다. 로마자 i 와 j 에 악센트를 표시하려면 \imath 와 \jmath 를 대신 사용하여 문자 위의 점을 없애 줍니다.

이 표의 악센트 중 $\widehat{}$ 과 $\widetilde{}$ 를 제외하면 그 길이가 자동으로 늘어나지 않으므로, 인자로 는 문자 하나만을 지정해야 합니다. $\widehat{}$ 과 $\widetilde{}$ 는 문자 세 개 정도의 너비까지 늘어날 수 있습니다.

입력

\widehat{a} , \widehat{AB} , \widehat{ABC}

출력

\hat{a} , \widehat{AB} , \widehat{ABC}

8.2.1 비슷한 기호의 구분

비슷한 모양의 기호를 입력할 때에는 수식의 의미에 맞는 명령어를 사용하는 것이 좋습니다. 여기에서 설명하는 규칙은 반드시 지켜야 하는 것은 아니지만, 수식을 더 보기 좋게 조판하려는 경우 도움이 될 것입니다.

이항 연산자와 대형 연산자

이항 연산자와 대형 연산자는 서로 대응되는 같은 모양의 기호가 있습니다. 예를 들어, 합집합 연산을 나타내는 기호 `\cup` 'U'와 `\bigcup` 'U', 직접합 연산을 나타내는 기호 `\oplus` '⊕'와 `\bigoplus` '⊕', 논리곱 연산을 나타내는 기호 `\wedge` '∧'와 `\bigwedge` '∧'가 있습니다.

크기가 더 작은 것은 이항 연산자로, 두 개의 대상 사이에 쓰여 이 둘 사이의 연산을 나타냅니다. 크기가 더 큰 것은 대형 연산자로, 연산을 수행할 대상을 일반적인 형태로 나타내고(인덱스를 붙이거나 집합의 대표 원소로 표시하는 등) 그 왼쪽에 대형 연산자를 씌우으로써 조건을 만족하는 대상 모두에 대한 연산을 나타냅니다.

입력

`\(A_1 \cup A_2\)`

`\(\bigcup_{i=1}^{\infty} A_i\)`

`\[V_1 \oplus V_2\]`

`\[\bigoplus_{i=1}^n V_i\]`

`\[a \wedge b\]`

`\[\bigwedge_{a \in S} a\]`

출력

$A_1 \cup A_2$

$\bigcup_{i=1}^{\infty} A_i$

$V_1 \oplus V_2$

$\bigoplus_{i=1}^n V_i$

$a \wedge b$

$\bigwedge_{a \in S} a$

이들을 반대로 사용하면 기호의 의미에도 맞지 않고 보기에 좋지 않은 결과를 얻으므로, 두 기호를 올바른 곳에 사용하기 바랍니다. 특히, 대형 연산자를 사용해야 할 자리에 이항 연산자를 사용하면 별행수식에서 계산 범위가 기호의 위아래에 표시되지 않습니다.

줄임표

줄임표에는 점을 아래쪽에 찍는 `\ldots`(low dots)와 가운데에 찍는 `\cdots`(centered dots)가 있습니다. 연산자나 관계 기호 사이, 곱할 대상 사이에는 점을 가운데에 찍어야

하고, 그 외의 경우에는 점을 아래에 찍어야 합니다.²

입력

```
\[x_1 + x_2 + \cdots + x_n\]
\[x_1 < x_2 < \cdots < x_n\]
\[X = x_1 x_2 \cdots x_n\]
\[x_1, x_2, \ldots, x_n\]
\[x = 0.x_1x_2\ldots x_n\]
```

출력

```
x_1 + x_2 + \cdots + x_n
x_1 < x_2 < \cdots < x_n
X = x_1x_2 \cdots x_n
x_1, x_2, \ldots, x_n
x = 0.x_1x_2 \ldots x_n
```

amsmath 패키지를 불러온 후 `\dots`를 입력하면, 뒤에 오는 기호에 따라 `\ldots`와 `\cdots` 중 적절한 것이 사용됩니다. 수식의 끝에 사용되거나 아래의 (*)처럼 곱할 대상 사이에 있다면 `\dots`가 둘 중 어느 것이 적절한지 판단할 수 없어, `\ldots`가 사용됩니다.

입력

```
\[x_1 + x_2 + \dots + x_n\]
\[x_1 < x_2 < \dots < x_n\]
\[X = x_1 x_2 \dots x_n\]
\[x_1, x_2, \dots, x_n\]
\[x = 0.x_1x_2\dots x_n\]
```

출력

```
x_1 + x_2 + \cdots + x_n
x_1 < x_2 < \cdots < x_n
X = x_1x_2 \ldots x_n \quad \cdots(*)
x_1, x_2, \ldots, x_n
x = 0.x_1x_2 \ldots x_n
```

올바른 기호를 판단할 수 없을 때에는 표 8.8의 다섯 가지 제어 문자열을 사용해 어떤 상황인지를 알려 주어야 합니다. 각 기호는 다음과 같은 상황에 사용됩니다.

- 이항 연산자나 관계 기호 사이에는 `\dotsb`를, (dots between binary)
- 쉼표 사이에 올 때에는 `\dotsc`를, (dots between commas)
- 적분 기호(또는 대형 연산자) 사이에는 `\dotsi`를, (dots between integrals)
- 곱할 항목들 사이에 올 때에는 `\dotsm`을, (multiplication dots)
- 다른 경우에는 `\dotso`를 입력합니다. (dots between others)

상황에 알맞은 명령어를 입력하면 적절한 줄임표가 출력됩니다.

입력

```
\[x_1+x_2+x_3+\dotsb\]
\[x_1, x_2, x_3, \dotsc\]
\[X = x_1x_2\dotsm x_n\]
\[x = 0.x_1x_2x_3\dotso\]
```

출력

```
x_1 + x_2 + x_3 + \cdots
x_1, x_2, x_3, \cdots
X = x_1x_2 \cdots x_n
x = 0.x_1x_2x_3 \ldots
```

² 동아시아에서는 일상 언어에서의 문장부호를 가운데에 찍기 때문에, 그 영향을 받아 수식에서도 줄임표를 항상 가운데에 찍기도 합니다.

8.2.2 모양은 같지만 앞뒤의 간격이 다른 기호

이번 절에서는 기호 앞뒤의 간격을 올바르게 표시하기 위한 방법을 알아봅니다. 대부분의 경우 \TeX 이 자동으로 간격을 조절하지만, 같은 모양의 기호가 여러 의미로 쓰일 때에는 \TeX 이 이를 올바르게 처리하도록 적절한 명령어를 사용해야 합니다.

세로줄과 겹세로줄

세로줄은 `\mid`, `|`, `\lvert`, `\rvert`로 입력할 수 있습니다. 마찬가지로, 겹세로줄은 `\parallel`, `\|`, `\lVert`, `\rVert`로 입력할 수 있습니다. 기호를 입력할 때 사용한 명령어에 따라 그 의미가 다르게 해석되므로, 기호 앞뒤의 공간도 다르게 표시됩니다.

먼저, `\mid`와 `\parallel`은 관계를 나타내는 기호로, 기호 앞뒤의 간격이 넓습니다. 약수와 배수 관계, 평행 관계를 나타낼 때뿐 아니라 집합의 정의, 조건부 확률 등을 나타낼 때에도 이들을 사용합니다. 반면, `|`와 `\|`는 괄호로 사용됩니다. 절댓값, 유한 집합의 크기, 행렬식, 노름 등에 사용합니다.

입력

```
\(p \mid mn\)
\(\{x \mid x > y\}\)
\(\Pr(B \mid A)\)
\(\l \parallel l'\)
\((y = |x+a|)\)
\(\|x\| + \|y\| = 0\)
```

출력

```
 $p \mid mn$ 
 $\{x \mid x > y\}$ 
 $\Pr(B \mid A)$ 
 $l \parallel l'$ 
 $y = |x + a|$ 
 $\|x\| + \|y\| = 0$ 
```

경우에 따라서는 괄호의 의미로 사용한 `|`나 `\|`가 불필요한 공백을 만들 때가 있습니다. 이때에는 이들 대신 `\lvert`, `\rvert`, `\lVert`, `\rVert`를 사용합니다. ‘`\l`’로 시작하는 것은 여는 괄호, ‘`\r`’로 시작하는 것은 닫는 괄호로 사용하면 됩니다.

입력

```
\(|\sin x|\)
\(|-x|\)
\(|A*|)
```

출력

```
 $|\sin x|$ 
 $|-x|$ 
 $|A^*|$ 
```

입력

```
\(\lvert\sin x\rvert\)
 $\lvert -x \rvert$ 
 $\lvert A^* \rvert$ 
```

출력

```
 $|\sin x|$ 
 $|-x|$ 
 $|A^*|$ 
```

이 대신, 8.4절의 괄호 높이를 조절하는 명령어를 사용해도 문제가 해결됩니다.

입력

```
\(\left|\sin x\right|\)
\(\left|-x\right|\)
```

출력

```
 $|\sin x|$ 
 $|-x|$ 
```


역슬래시

역슬래시는 `\setminus`와 `\backslash` 제어 문자열로 입력합니다. 전자는 이항 연산자로 취급되어 기호 앞뒤에 약간의 간격을 두고 조판되며, 후자는 그렇지 않습니다.

입력

`\(A \setminus B\)`
`\(H\backslash G/K\)`

출력

$A \setminus B$ (집합의 차)
 $H \setminus G/K$ (이중 잉여류)

참고 | 역슬래시와 달리, 빗금은 분수를 나타내든 나눗셈 연산을 나타내든 잉여류나 다른 식을 나타내든 항상 /로 입력하며, 기호 앞뒤의 간격을 넓히지 않습니다.

쉼표, 마침표, 쌍점

쉼표, 마침표와 쌍점은 문장부호로 쓰일 때와 그렇지 않을 때를 구분해 주는 것이 좋습니다. 먼저, 쉼표는 대부분의 경우에 간단히 ,로 입력하면 문제가 없습니다. 다만, 큰 수를 나타낼 때 천 단위를 구분하는 기호로 사용할 때에는 중괄호로 감싸 주어 기호 뒤쪽의 공간을 없애 주어야 합니다.

입력

`\(100, 200, 300\)`
`\(100\{, \}200\{, \}300\)`

출력

100,200,300 (나열)
 100,200,300 (큰 수)

마침표나 쌍점도 대부분의 경우에는 간단히 .와 :로 입력하면 문제가 없습니다. 다만, 람다 대수 등에서 변수를 뒤의 식과 구분할 때, 함수의 정의역과 공역 또는 대응 관계를 표기할 때에는 `\ldotp`나 `\colon`을 입력하여 문장 부호로써 사용하는 것이 좋습니다. 람다 대수 식에서 변수의 유형(type)을 나타내는 쌍점은 `\{:\}`로 입력합니다.

입력

`\(0.35 = 35\%\)`
`\(\lambda x\{:\}\tau\ldotp f(x)\)`
`\(4:x = y:5\)`
`\(\{x : x > y\}\)`
`\(f\colon X \to Y\)`

출력

$0.35 = 35\%$
 $\lambda x:\tau. f(x)$
 $4 : x = y : 5$
 $\{x : x > y\}$
 $f : X \rightarrow Y$

이외에, 가운데뎡점을 출력하는 제어 문자열에는 `\cdot`과 `\cdotp`가 있습니다. `\cdot`은 다른 이항 연산자처럼 기호 양쪽에 약간의 간격을 두며, `\cdotp`는 문장 부호처럼 기호의 뒤쪽에만 약간의 간격을 둡니다. 기호의 의미에 따라 두 명령어를 적절히 구분하여 사용하시기 바랍니다.

8.3 기본적인 수식 요소

이번 절에서는 분수, 근호, 첨자 등 기본적인 수식 요소를 입력하는 문법을 설명합니다.

8.3.1 분수와 이항 계수

분수를 입력할 때에는 `\frac` 제어 문자열을 사용합니다. 이항 계수를 입력할 때에는 `amsmath` 패키지의 `\binom` 제어 문자열을 사용합니다.

| <code>\frac{⟨over⟩}{⟨under⟩}</code> | <code>\binom{⟨over⟩}{⟨under⟩}</code> |
|---|--|
| <code>⟨over⟩</code> : 분수의 분자
<code>⟨under⟩</code> : 분수의 분모 | <code>⟨over⟩</code> : 위쪽에 조판할 식
<code>⟨under⟩</code> : 아래쪽에 조판할 식 |

분수와 이항 계수는 별행수식에서 조판할 때보다 행중수식에서 조판할 때 작은 크기로 표시됩니다. 분자나 분모 자리에서도 작게 표시됩니다.

| 입력 | 출력 | 입력 | 출력 |
|------------------------------|-------------------|------------------------------|-----------------|
| <code>\(1+ \frac ab\)</code> | $1 + \frac{a}{b}$ | <code>\(2 \binom ab\)</code> | $2\binom{a}{b}$ |
| <code>\[1+ \frac ab\]</code> | $1 + \frac{a}{b}$ | <code>\[2 \binom ab\]</code> | $2\binom{a}{b}$ |
| <code>\(\frac{1+a}b\)</code> | $\frac{1+a}{b}$ | <code>\(\binom{2a}b\)</code> | $\binom{2a}{b}$ |
| <code>\[\frac{1+a}b\]</code> | $\frac{1+a}{b}$ | <code>\[\binom{2a}b\]</code> | $\binom{2a}{b}$ |

연분수

연분수를 조판할 때에는 더 보기 좋은 결과를 위해 `\frac` 대신 `\cfrac` 제어 문자열을 사용하는 것이 좋습니다.

| <code>\cfrac[⟨align⟩]{⟨over⟩}{⟨under⟩}</code> |
|---|
| <code>⟨align⟩</code> : 분자를 정렬시킬 방향
<code>⟨over⟩</code> : 분수의 분자
<code>⟨under⟩</code> : 분수의 분모 |

```

1 \[x = 1 +
2   \cfrac{1}{2 +
3     \cfrac{1}{3+\cfrac{14}}}\]
```

$$x = 1 + \frac{1}{2 + \frac{1}{3 + \frac{1}{4}}}$$

`<align>` 인자는 분자의 정렬 방향을 결정하며, `l`, `c`, `r` 중 하나를 입력할 수 있습니다. 기본값은 분자를 가운데에 배치하는 `c`이며, `l`을 입력하면 왼쪽에, `r`를 입력하면 오른쪽에 분자가 배치됩니다.

입력

$$\backslash[x = 1 + \backslashcfrac[l]{1}{\dots}\backslash]$$

$$\backslash[x = 1 + \backslashcfrac[r]{1}{\dots}\backslash]$$

출력

$$x = 1 + \frac{1}{2 + \frac{1}{3 + \frac{1}{a}}}$$

$$x = 1 + \frac{1}{2 + \frac{1}{3 + \frac{1}{a}}}$$

8.3.2 근호

`근호`를 입력할 때에는 `\sqrt` 제어 문자열을 사용합니다. 선택 인자를 지정하지 않으면 제곱근을, 선택 인자를 지정하면 거듭제곱근을 입력할 수 있습니다.

$$\backslash\sqrt[\langle rad. index \rangle]{\langle radicand \rangle}$$

`<rad. index>`: 근지수

`<radicand>`: 피제곱근수

입력

$$\backslash(\backslash\sqrt{3x}\backslash)$$

$$\backslash(\backslash\sqrt{3}\backslash, x\backslash)$$

$$\backslash(\backslash\sqrt{3 + x}\backslash)$$

$$\backslash(\backslash\sqrt[3]{x}\backslash, y\backslash)$$

출력

$$\sqrt{3x}$$

$$\sqrt{3}x$$

$$\sqrt{3 + x}$$

$$\sqrt[3]{x}y$$

예시에서 확인할 수 있듯, 근호 뒤에 문자 등이 이어지면 `\,`를 입력해 약간의 간격을 만들어 주는 것이 좋습니다.

윗줄과 밑줄

근호를 그릴 때에는 인자로 지정된 내용의 길이에 맞춰 가로선이 그려집니다. 이와 비슷한 수식 요소로 윗줄과 밑줄이 있습니다. 윗줄을 그을 때에는 `\overline` 제어 문자열을, 밑줄을 그을 때에는 `\underline` 제어 문자열을 사용합니다.

$$\backslash\overline{\langle subformula \rangle}$$

`<subformula>`: 윗줄을 그을 수식

$$\backslash\underline{\langle subformula \rangle}$$

`<subformula>`: 밑줄을 그을 수식

| 입력 | 출력 |
|--|-------------------------------|
| <code>\(\overline{a+b}\)</code> | $\overline{a+b}$ |
| <code>\(\overline{a}+\overline{b}\)</code> | $\overline{a}+\overline{b}$ |
| <code>\(\underline{x-y}\)</code> | $\underline{x-y}$ |
| <code>\(\underline{x}-\underline{y}\)</code> | $\underline{x}-\underline{y}$ |

8.3.3 첨자

위 첨자를 입력할 때에는 서컴플렉스 ‘^’를, 아래 첨자를 입력할 때에는 밑줄 문자 ‘_’를 사용합니다. 같은 역할을 하는 `\sp`와 `\sb` 제어 문자열을 사용할 수도 있습니다. 명령어 직전 기호에 첨자를 붙여 줍니다.

| <code>^{\langle superscript \rangle}</code> | <code>_{\langle subscript \rangle}</code> |
|--|---|
| $\langle superscript \rangle$: 위 첨자로 조판할 내용 | $\langle subscript \rangle$: 아래 첨자로 조판할 내용 |

| 입력 | 출력 | 입력 | 출력 |
|------------------------|---------|------------------------------|---------|
| <code>\(x^a\)</code> | x^a | <code>\(x\sp a\)</code> | x^a |
| <code>\(x_b\)</code> | x_b | <code>\(x\sb b\)</code> | x_b |
| <code>\(x_1^2\)</code> | x_1^2 | <code>\(x\sb 1\sp 2\)</code> | x_1^2 |
| <code>\(x^2_1\)</code> | x_1^2 | <code>\(x\sp 2\sb 1\)</code> | x_1^2 |

중괄호를 사용해 수식을 묶어 준다면 해당 부분 전체가 하나의 ‘덩어리’로 인식되므로, 첨자가 중괄호가 없을 때와 다른 위치에 나타날 수 있습니다.

| 입력 | 출력 | 입력 | 출력 |
|----------------------|-----------|--------------------------|---------------|
| <code>x_a^b</code> | x_a^b | <code>\{x_a\}^b</code> | $\{x_a\}^b$ |
| <code>x^a_b</code> | x_b^a | <code>\{x^a\}_b</code> | $\{x^a\}_b$ |
| <code>(x^a)^b</code> | $(x^a)^b$ | <code>\{(x^a)\}^b</code> | $\{(x^a)\}^b$ |
| <code>(x_a)_b</code> | $(x_a)_b$ | <code>\{(x_a)\}_b</code> | $\{(x_a)\}_b$ |

같은 방향의 첨자를 두 번 연속으로 사용하면 컴파일 과정에서 오류가 발생하므로, 중괄호를 적절하게 사용해야 합니다. 예를 들어, `\(a^b^c\)`는 오류를 일으키며, ‘ a^{b^c} ’를 조판하려면 `\(a^{\{b^c\}}\)`를, ‘ a^{b^c} ’를 조판하려면 `\(\{a^b\}^c\)`를 입력해야 합니다. (전자의 ‘c’가 후자의 ‘c’보다 더 작습니다.) 수식의 의미를 고려했을 때 전자의 방식으로 첨자를 사용하는 것이 일반적이며, 후자와 같이 ‘ a^b ’에 위 첨자 ‘c’를 붙일 때에는 ‘ $(a^b)^c$ ’처럼 괄호를 씌우는 것이 좋습니다.

우리나라 고등학교 교과서의 조합 기호나 화학식에서는 기호 왼쪽에 첨자를 입력하기도 합니다. 이때에는 빈 수식 `\{ }`을 입력하고 첨자를 입력합니다.

입력

```
\({}_5\mathrm{C}_3\)
```

```
\({}_1^3\mathrm{H}^{1+}\)
```

출력

$${}_5C_3$$

$${}_1^3H^{1+}$$

빈 수식을 입력하지 않으면, 다음과 같이 왼쪽 첨자가 그 앞의 기호에 붙습니다.

$${}_{n+1}C_{r+1} = {}_nC_r + {}_nC_{r+1}$$

이러한 일을 막기 위해, 왼쪽 첨자 앞에 빈 수식을 꼭 입력하기 바랍니다.

8.3.4 기타 요소

일반 텍스트

수식을 작성하더라도, 조건을 설명할 때에는 종종 텍스트 모드를 사용할 필요가 있습니다. 집합의 정의를 설명할 때나 대상을 구체적인 설명으로 지칭할 때가 그 예입니다. 이 때에는 `amsmath` 패키지의 `\text` 제어 문자열을 사용합니다. 인자로 텍스트 모드에서 조판하려는 내용을 입력하면 됩니다. 필요하다면 이 안에서 다시 수식 모드를 사용할 수도 있습니다.

```
\[(\text{원의 넓이}) = \pi \times
```

```
(\text{반지름의 길이})^2\]
```

$$(\text{원의 넓이}) = \pi \times (\text{반지름의 길이})^2$$

```
\[A = \{x \mid \text{\textit{\textbackslash}(x)}\text{는 짝수}\}\]
```

$$A = \{x \mid x \text{는 짝수}\}$$

수식과 수식 사이의 간격

한 줄에 별행수식을 여러 개 작성하거나, 별행수식을 작성하고 그 옆에 조건을 달아 둘 때에는 식과 식 사이를 넓게 띄워 주는 것이 좋습니다. 이때에는 각각 1em과 2em의 공백을 삽입하는 `\quad`와 `\quad\quad`를 사용합니다. 일반적으로는 `\quad\quad`를 사용하며, 수식을 배치할 공간이 좁거나 특별히 가까이 조판하고 싶은 수식이 있는 경우에는 `\quad`를 사용하면 됩니다.

```
1 두 함수
```

```
2 \[f(x) = \sin x, \quad\quad
```

```
3 g(x) = \cos x\]
```

```
4 는 주기가 \((2\pi)\)인 주기함수이다.
```

두 함수

$$f(x) = \sin x, \quad g(x) = \cos x$$

는 주기가 2π 인 주기함수이다.

표 8.12 | 높이를 조절할 수 있는 기호

| | | | | | | | |
|---|--------------|---|-----------------|---|---------------------|---|----------------------|
| (| (|) |) | [| [|] |] |
| { | \{ | } | \} | < | \angle ¹ | > | \rangle ¹ |
| | ² | | \ ² | / | / | \ | \backslash |
| ⌈ | \lceil | ⌋ | \rceil | ⌊ | \lfloor | ⌋ | \rfloor |
| ↑ | \uparrow | ↗ | \Uparrow | ↓ | \downarrow | ↘ | \Downarrow |
| ↕ | \updownarrow | ↕ | \Updownarrow | | | | |

반드시 크기를 조절하는 명령어 뒤에 입력해야 하는 기호:

| | | | | | | | |
|---|------------|---|------------|---|-------------|---|-------------|
| (| \lgroup |) | \rgroup | ⎵ | \lmoustache | ⎶ | \rmoustache |
| | \arrowvert | | \arrowVert | | \bracevert | | |

amsmath 패키지의 기호:

| | | | | | | | |
|--|--------|--|--------|--|--------|--|--------|
| | \lvert | | \rvert | | \lVert | | \rVert |
|--|--------|--|--------|--|--------|--|--------|

¹ 크기를 조절할 때에는 <와 >로 간단히 입력할 수 있습니다.

² 같은 모양의 기호를 조판하는 \mid나 \parallel은 사용할 수 없습니다.

8.4 괄호의 높이 조절

괄호로 감싼 식에 분수, 대형 연산자, 적분식 등 큰 식이 포함되어 있다면 괄호의 높이를 조정해 주어야 합니다. 괄호의 높이는 수동으로 조절할 수도 있고, 괄호 사이에 오는 식에 맞추어 자동으로 조절되도록 할 수도 있습니다. 높이를 조절할 수 있는 기호는 표 8.12에서 확인할 수 있습니다.

8.4.1 자동 조절

\left와 \right 제어 문자열을 사용하면 괄호 사이에 작성되는 식의 크기에 맞추어 괄호의 크기가 자동으로 조절됩니다. 표 8.7의 기호 중 왼쪽 기호로 사용할 것을 \left 뒤에, 오른쪽 기호로 사용할 것을 \right 뒤에 입력하면 됩니다.

입력

\(\left(x \right)\)

\(\left(x_n^k \right)\)

\[\left(\frac{ab}{b} \right)\]

\[\left(\int f(x) \, dx \right)\]

출력

(x)

(x_n^k)

($\frac{a}{b}$)

($\int f(x) dx$)

어디부터 어디까지의 식에 괄호의 높이를 맞추어야 하는지 \TeX 이 알아야 하므로, \left와 \right는 짝을 맞추어 사용해야 합니다(괄호의 모양은 짝이 맞지 않아도 됩니다).

표 8.7의 기호 외에 마침표 ‘.’를 `\left`나 `\right` 뒤에 입력하면 한쪽 괄호를 생략할 수 있습니다.

```
\[ \left. \frac{df}{dx} \right|_{x=0}
\]
```

$$\left. \frac{df}{dx} \right|_{x=0}$$

분수나 집합 등을 작성할 때처럼, 가운데에 쓰이는 구분선의 높이를 양쪽 괄호와 맞추려면 `\left`와 `\right`의 사이에서 `\middle`을 사용하면 됩니다. 조건부 확률이나 집합 표기 등에서는 구분선 좌우에 `\;`를 입력하여 `\mid`를 사용했을 때처럼 공백을 넣어 주는 것이 좋습니다.

```
\[ \left. \frac{dy}{dt} \middle/ \frac{dx}{dt} \right. \]
```

$$\frac{dy}{dt} \middle/ \frac{dx}{dt}$$

```
\[ \left\{ \frac{a+bi}{2} \;\middle|\; a, b \in \mathbb{R} \right\} \]
```

$$\left\{ \frac{a+bi}{2} \;\middle|\; a, b \in \mathbb{R} \right\}$$

한편, `\left...\right`로 감싼 식의 앞이나 뒤에는 약간의 공백이 삽입됩니다. 이 공백을 없애려는 경우에는 해당 위치에 `\!`를 입력하면 됩니다.

입력

`\(f(x) = x\)`

`\(f\left(x\right) = x\)`

`\(f!\left(x\right) = x\)`

출력

$f(x) = x$

$f(x) = x$

$f(x) = x$

8.4.2 수동 조절

괄호의 높이를 수동으로 조절할 때에는 네 단계의 높이를 사용할 수 있습니다. 기호 앞에 `\big`을 붙이면 1단계 높게, `\Big`을 붙이면 2단계 높게, `\bigg`을 붙이면 3단계 높게, `\Bigg`을 붙이면 4단계 높게 조판됩니다.

기본 높이: $() [] \{ \} \langle \rangle / \backslash ||| \uparrow \uparrow \downarrow \downarrow \updownarrow$

1단계 높게: $() [] \{ \} \langle \rangle / \backslash ||| \uparrow \uparrow \downarrow \downarrow \updownarrow$

2단계 높게: $() [] \{ \} \langle \rangle / \backslash ||| \uparrow \uparrow \downarrow \downarrow \updownarrow$

3단계 높게: $() [] \{ \} \langle \rangle / \backslash ||| \uparrow \uparrow \downarrow \downarrow \updownarrow$

4단계 높게: $() [] \{ \} \langle \rangle / \backslash ||| \uparrow \uparrow \downarrow \downarrow \updownarrow$

`\big`, `\Big`, `\bigg`, `\Bigg`을 사용하면 기호의 앞뒤에 공간이 없이 조판됩니다. 여는 괄호 앞에는 이들 대신 `\bigl……`, `\Biggl`을, 닫는 괄호 앞에는 `\bigr……`, `\Biggr`를 사용하여 적절한 결과를 얻을 수 있도록 하는 것이 좋습니다.

| | |
|---|--|
| <code>\[\bigl x + y \bigr \]</code> | $ x + y $ |
| <code>\[\log \biggl(\prod_{i=1}^n x_i \biggr) = \sum_{i=1}^n \log x_n \]</code> | $\log\left(\prod_{i=1}^n x_i\right) = \sum_{i=1}^n \log x_n$ |

`\bigm……`, `\Biggm`을 사용하면 기호가 관계 기호처럼 취급되어, 기호 양쪽에 약간의 공백이 생깁니다. 앞서 설명했듯 조건부 확률이나 집합 등의 표기에서 사용합니다.

| | |
|--|---|
| <code>\[\biggl\{ \frac{a+bi}{2} \biggm a, b \in R \biggr\} \]</code> | $\left\{ \frac{a+bi}{2} \mid a, b \in R \right\}$ |
| <code>\[\Pr \bigl(Y > y_0^2 \bigr) \bigm X > x_0^2 \bigr] \]</code> | $\Pr(Y > y_0^2 \mid X > x_0^2)$ |

기호의 앞뒤에 간격을 넣지 않을 때에는 `\big……`, `\Bigg`를 사용합니다.

| | |
|---|--------------------------------------|
| <code>\[\frac{dy}{dt} \bigg/ \frac{dx}{dt} \]</code> | $\frac{dy}{dt} \bigg/ \frac{dx}{dt}$ |
|---|--------------------------------------|

대부분의 경우에는 자동 조절 기능만으로도 충분히 좋은 결과를 얻지만, 이 기능을 사용할 수 없을 때나 적절하지 않은 결과를 만들 때도 있습니다. 이러한 경우에 수동 조절 기능이 유용하게 사용됩니다. 대표적으로 다음과 같은 상황이 있습니다.

- 수식 중간에 줄을 바꾸어야 할 때: `\left...\right`로 감싼 식은 중간에 줄이 바뀌지 않습니다. 큰 괄호로 감싼 식의 가운데에서 줄을 바꾸어야 한다면 수동 조절 기능을 사용해야 합니다.
- 기호가 크게 조판될 때: 괄호가 감싼 식에 대형 연산자가 포함된 경우, 괄호가 지나치게 높게 조판됩니다. 괄호가 계산 범위를 살피만 감싸는 높이(대부분의 경우 `\bigg`의 높이)를 사용하는 것이 적당합니다.
- 기호가 작게 조판될 때: 괄호로 감싼 식을 괄호로 또 감싼 경우, 특별한 요소가 없다면 안팎의 괄호가 같은 크기로 조판됩니다. 두 쌍의 괄호가 잘 구분되도록 바깥쪽의 괄호를 더 큰 것으로 바꾸어 주는 것이 좋습니다.

연습문제

8.1. 텍스트 모드와 수식 모드에 유의하여 다음을 조판하여라.

(a) 임의의 삼각형을 생각하자. 이 삼각형의 세 변의 길이가 각각 a, b, c 일 때, 부등식 $a + b > c$ 가 성립한다.

(b) 질량이 m 인 물체에 힘 F 를 가했을 때, 그 물체의 가속도 a 는 $F = ma$ 라는 관계를 만족한다.

(c) 기압이 P , 부피가 V , 온도가 T 인 공간에 이상 기체가 n 몰 있을 때,

$$PV = nRT$$

라는 관계가 성립한다. 이때, R 는 이상 기체 상수이다.

8.2. 다음 수식을 조판하여라.

$$(a) \bigsqcup_{i \in I} A_i = \bigcup_{i \in I} \{(x, i) \mid x \in A_i\} \quad (\text{분리합집합의 정의})$$

$$(b) \int_{\partial D} P dx + Q dy = \iint_D \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dx dy \quad (\text{그린 정리})$$

$$(c) \limsup A_n = \bigcap_{n=1}^{\infty} \bigcup_{m=n}^{\infty} A_m \quad (\text{집합의 상한의 정의})$$

8.3. 다음 수식을 조판하여라.

$$(a) \cos^2 x + \sin^2 x = 1, \quad \cosh^2 x - \sinh^2 x = 1$$

$$(b) \lim_{x \rightarrow a} f(x) = f(a)$$

$$(c) (p-1)! \bmod [1 + 2 + \cdots + (p-1)] = p-1$$

8.4. 다음 수식을 조판하여라.

$$(a) gH = \{gh \mid h \in H\} \quad (\text{잉여류의 정의})$$

$$(b) (a \sim b) \wedge (b \sim c) \implies a \sim c \quad (\text{추이적 관계})$$

$$(c) x \in U \cup V \iff (x \in U) \vee (x \in V) \quad (\text{두 집합의 합집합})$$

8.5. 다음 수식을 조판하여라.

$$(a) g \circ f: X \rightarrow Z, \quad g \circ f: x \mapsto g(f(x)) \quad (\text{합성함수의 정의})$$

$$(b) (\forall \epsilon > 0)(\exists \delta > 0)[f(B(x, \delta)) \subset B(f(x), \epsilon)] \quad (\text{함수의 연속의 정의})$$

$$(c) \hat{i} \times \hat{j} = \hat{k} \quad (\text{표준 단위벡터의 외적})$$

8.6. 다음 수식을 조판하여라.

$$(a) \frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \quad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} \quad (\text{코시-리만 방정식})$$

$$(b) \varphi = 1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \dots}}} \quad (\text{황금비의 연분수 표현})$$

$$(c) \binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k} \quad (\text{파스칼의 법칙})$$

8.7. 다음 수식을 조판하여라.

$$(a) \sqrt{x \pm y + 2\sqrt{xy}} = |\sqrt{x} \pm \sqrt{y}|$$

$$(b) \sqrt[m]{\sqrt[n]{x}y} = \sqrt[mn]{x} \sqrt[n]{y}$$

$$(c) \overline{a + bi} = a - bi$$

8.8. 다음 수식을 조판하여라.

$$(a) D_{2n} = \langle r, s \mid r^n = s^2 = (rs)^2 = 1 \rangle \quad (\text{정이면체군의 정의})$$

$$(b) e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots \quad (\text{지수함수의 테일러 급수})$$

$$(c) g_{ij}v^j = v_i$$

8.9. 다음을 조판하여라.

(a) 이자를 1년에 한 번씩 받는 경우, 연 금리가 $r\%$ 일 때의 원리금은 다음과 같이 계산한다.

$$(\text{원리금}) = (\text{원금}) \times \left(1 + \frac{r}{100}\right)^{(\text{예금 기간})}$$

(b) 환 $(R, +, \times)$ 의 두 원소 a 와 b 에 대하여, 두 등식

$$a + b = b + a \quad \text{와} \quad a \times b = b \times a$$

가 성립한다. 이를 ‘교환법칙’이라고 부른다.

8.10. 괄호의 크기에 유의하여 다음 수식을 조판하여라.

$$(a) \left[-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V(x) \right] \psi(x) = E\psi(x) \quad (\text{슈뢰딩거 방정식})$$

$$(b) \left(P + a \frac{n^2}{V^2} \right) (V_b n) = nRT \quad (\text{판데르발스 상태 방정식})$$

$$(c) G(f) = \left\{ (x, f(x)) \mid x \in D \right\} \quad (\text{함수의 그래프의 정의})$$

제 9 장

다양한 수식 작성하기

이번 장에서는 더 다양한 수식을 작성하기 위해 사용할 수 있는 요소를 알아봅니다. 표 형태의 구조, 분수 형태의 구조, 기호 위아래에 기호를 입력하는 구조 등 다양한 형태의 식과 여러 글꼴을 사용하는 방법을 다룹니다.

9.1 표 형태의 구조

이번 절에서는 행렬 형태와 경우 형태의 요소를 조판하는 방법을 알아봅니다. 일반적인 표 형태의 구조는 제12장에서 설명하는 array 환경을 사용해 만들 수 있습니다.

9.1.1 행렬

행렬을 작성할 때에는 amsmath 패키지에서 정의하는 행렬 작성 환경을 사용합니다. 행렬 환경은 행렬을 감싸는 괄호에 따라 표 9.1의 여섯 가지가 정의되어 있습니다. 이 표에 없는 괄호를 사용하고 싶다면 matrix 환경을 사용한 후 \left와 \right 제어 문자열과 원하는 괄호를 사용해 행렬을 감싸 주면 됩니다.

표 9.1 | amsmath 패키지: 행렬 작성 환경

| 환경 이름 | 예시 | 환경 이름 | 예시 | 환경 이름 | 예시 |
|---------|--|---------|--|---------|--|
| matrix | $\begin{matrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{matrix}$ | bmatrix | $\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$ | vmatrix | $\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$ |
| pmatrix | $\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$ | Bmatrix | $\begin{Bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{Bmatrix}$ | Vmatrix | $\begin{Vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{Vmatrix}$ |

행렬을 작성할 때에는 한 행의 내용을 작성한 후 다음 행으로 넘어가며, 한 행 안에서 열의 구분은 &로, 행과 행의 구분은 \\로 합니다.

```
1 \begin{displaymath}
2   \begin{bmatrix}
3     3 & 1 & -4 & 0 \\
4     2 & -3 & 11 & -7 \\
5     -5 & 0 & 0 & 9
6   \end{bmatrix}
7 \end{displaymath}
```

$$\begin{bmatrix} 3 & 1 & -4 & 0 \\ 2 & -3 & 11 & -7 \\ -5 & 0 & 0 & 9 \end{bmatrix}$$

행렬의 모든 성분은 가운데에 맞추어 정렬됩니다. 모든 성분을 왼쪽이나 오른쪽에 맞추어 정렬하려면 mathtools 패키지의 별표 붙은 행렬 환경을 사용하면 됩니다.¹ 선택 인자로 l, c, r를 입력하면 성분이 각각 왼쪽, 가운데, 오른쪽에 맞추어 정렬됩니다. 선택 인자를 생략했을 때에는 기본값인 c가 사용됩니다.

```
1 \begin{displaymath}
2   \begin{vmatrix*}[l]
3     0.98 & 0.7 \\
4     0.4 & 1.03
5   \end{vmatrix*}
6 \end{displaymath}
```

$$\begin{vmatrix} 0.98 & 0.7 \\ 0.4 & 1.03 \end{vmatrix}$$

```
1 \begin{displaymath}
2   \begin{pmatrix*}[r]
3     -3 & 7 \\
4     2 & -11
5   \end{pmatrix*}
6 \end{displaymath}
```

$$\begin{pmatrix} -3 & 7 \\ 2 & -11 \end{pmatrix}$$

카운터
→ 269쪽

참고 | 하나의 행렬 환경은 최대 10 × 10 행렬까지만 그릴 수 있습니다. 행렬 환경이 그릴 수 있는 행렬의 최대 크기는 MaxMatrixCols 카운터에 저장되어 있으며, 이 값을 바꾸면 더 큰 행렬을 그릴 수 있습니다.

큰 행렬의 생략 표시

일반적인 상황을 나타내는 큰 행렬을 그릴 때에는 줄임표를 사용해 생략 표시를 하기도 합니다. 이때에는 표 8.8에 있는 \hdots(\cdots를 사용하기도 합니다.), \vdots, \ddots를 사용합니다.

¹ 열마다 정렬 방식을 다르게 지정하려면 array 환경을 사용할 수 있습니다.

```

1 \begin{displaymath}
2   \begin{bmatrix}
3     a_{11} & a_{12} & \hdots & a_{1n} \\
4     a_{21} & a_{22} & \hdots & a_{2n} \\
5     \vdots & \vdots & \ddots & \vdots \\
6     a_{n1} & a_{n2} & \hdots & a_{nn}
7   \end{bmatrix}
8 \end{displaymath}

```

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

여러 열에 걸쳐 줄임표를 그리는 방식을 사용하려면 `amsmath` 패키지의 `\hdotsfor` 제어 문자열을 사용할 수 있습니다. 선택 인자를 입력하면 `\cdots`로 입력한 줄임표의 점 사이 간격에 해당 값을 곱한 만큼 점과 점 사이의 간격이 넓어집니다.

`\hdotsfor[⟨spacing⟩]{⟨columns⟩}`

⟨spacing⟩: 점과 점 사이의 간격을 넓힐 배수
 ⟨columns⟩: 줄임표를 그릴 열의 수

```

1 \begin{displaymath}
2   \begin{bmatrix}
3     a_{11} & a_{12} & \hdots & a_{1n} \\
4     a_{21} & a_{22} & \hdots & a_{2n} \\
5     \hdotsfor[1.5]{4} \\
6     a_{n1} & a_{n2} & \hdots & a_{nn}
7   \end{bmatrix}
8 \end{displaymath}

```

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

작은 행렬

`amsmath` 패키지의 `smallmatrix` 환경을 사용하면 행렬을 작게 그릴 수 있습니다. 이 환경에는 그릴 수 있는 행렬의 크기에 제한이 없습니다.

```

1 \[ I = \left[ \begin{smallmatrix}
2 1&0&0 \\ 0&1&0 \\ 0&0&1
3 \end{smallmatrix} \right] \]

```

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

작은 행렬에 괄호를 함께 조판하는 `psmallmatrix`, `bsmallmatrix`, `Bsmallmatrix`, `vsmallmatrix`, `Vsmallmatrix` 환경은 `mathtools` 패키지에서 제공합니다.

```

1 \[ I = \begin{bsmallmatrix}
2 1&0&0 \\ 0&1&0 \\ 0&0&1
3 \end{bsmallmatrix} \]

```

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

또, 본래 크기의 행렬과 마찬가지로, `mathtools` 패키지는 별표 붙은 작은 행렬 환경을 제공합니다. `l`, `c`, `r` 중 하나를 선택 인자로 지정함으로써 모든 성분의 정렬 방향을 지정해 줍니다.

작은 행렬은 행렬을 행중수식으로 작성해야 할 때 유용합니다. 하지만 행이 3개보다 많은 행렬은 공간을 많이 차지하므로 별행수식으로 작성하는 것이 좋습니다.

9.1.2 경우

여러 식을 세로로 나열해 중괄호로 묶고, 각 식의 옆에 그 조건을 적는 형태의 구조를 작성할 때에는 `cases` 환경을 사용합니다. 이러한 `cases` 환경은 조각적으로 정의된 (piecewisely defined) 함수를 나타낼 때나 연립방정식 등을 묶어서 나타낼 때 사용할 수 있습니다.

```

1 \begin{displaymath}
2 |x| =
3 \begin{cases}
4 x & x \geq 0 \text{일 때} \\
5 -x & x < 0 \text{일 때}
6 \end{cases}
7 \end{displaymath}

```

$$|x| = \begin{cases} x & x \geq 0 \text{일 때} \\ -x & x < 0 \text{일 때} \end{cases}$$

예시에서 확인할 수 있듯, 환경 안에서 식과 조건을 입력하는 방법은 `matrix` 환경과 동일합니다. 각 식은 `\\`로 구분하며, 식과 조건을 구분할 때에는 `&`를 입력하면 됩니다.

`mathtools` 패키지는 별표 붙은 `cases*` 환경을 제공합니다. 이 환경은 조건을 적는 2열을 텍스트 모드에서 조판합니다.

```

1 \begin{displaymath}
2   a_n =
3   \begin{cases*}
4     \frac{1}{n} & \text{\(n\)이 홀수일 때} \\
5     -\frac{1}{n} & \text{\(n\)이 짝수일 때}
6   \end{cases*}
7 \end{displaymath}

```

$$a_n = \begin{cases} \frac{1}{n} & n\text{이 홀수일 때} \\ -\frac{1}{n} & n\text{이 짝수일 때} \end{cases}$$

추가 경우 환경

mathtools 패키지는 cases* 환경 외에 여섯 가지 경우 환경을 추가로 제공합니다. 위 예시의 분수에서 알 수 있듯, cases(*) 환경은 각 식을 행중수식 스타일로 조판합니다. 하지만, dcases(*) 환경을 사용하면 각 수식을 별행수식 스타일로 조판할 수 있습니다.

rcases(*) 환경은 중괄호를 왼쪽이 아니라 오른쪽에 표시합니다. 또, drcases(*) 환경은 각 수식을 별행수식 스타일로 조판하면서 중괄호를 오른쪽에 표시합니다. 각 환경의 별표 붙은 환경은 cases* 환경과 마찬가지로, 조건을 작성하는 2열을 텍스트 모드에서 조판합니다.

```

1 \begin{displaymath}
2   \begin{drcases}
3     \sinh x = \frac{e^x - e^{-x}}{2} \\
4     \cosh x = \frac{e^x + e^{-x}}{2} \\
5     \tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}
6   \end{drcases}
7   \text{쌍곡선 함수}
8 \end{displaymath}

```

$$\left. \begin{aligned} \sinh x &= \frac{e^x - e^{-x}}{2} \\ \cosh x &= \frac{e^x + e^{-x}}{2} \\ \tanh x &= \frac{e^x - e^{-x}}{e^x + e^{-x}} \end{aligned} \right\} \text{쌍곡선 함수}$$

9.2 분수 형태의 구조

이번 절에서는 분수나 이항 계수와 같이, 두 개의 식을 위아래로 쌓은 형태의 구조와 관련된 명령어를 다룹니다.

9.2.1 스타일이 지정된 분수와 이항 계수

8.3.1절에서는 분수를 조판하는 `\frac`과 이항 계수를 조판하는 `\binom` 제어 문자열을 소개하였으며, 이들은 현재의 수식 스타일에 따라 다른 크기로 조판됩니다. `amsmath` 패키지는 현재 수식 스타일과 무관하게, 일정한 스타일로 분수나 이항 계수를 조판하는 제어 문자열을 제공합니다. `\tfrac`과 `\tbinom`은 행중수식 스타일로, `\dfrac`과 `\dbinom`은 별행수식 스타일로 이들이 조판됩니다.

| 입력 | 출력 | 입력 | 출력 |
|----------------------------|---------------|-----------------------------|----------------|
| <code>\(\tfrac ab\)</code> | $\frac{a}{b}$ | <code>\[\tbinom ab\]</code> | $\binom{a}{b}$ |
| <code>\(\dfrac ab\)</code> | $\frac{a}{b}$ | <code>\[\dbinom ab\]</code> | $\binom{a}{b}$ |

9.2.2 일반화된 분수

`amsmath` 패키지의 `\genfrac` 제어 문자열은 두 개의 식을 위아래로 쌓는 형태의 식을 입력할 때 사용합니다. 이러한 구조를 일반화된 분수 *generalized fraction*라고 부릅니다. 이 제어 문자열은 여섯 개의 인자를 입력받습니다.

```
\genfrac{<left delim.>}{<right delim.>}{<thickness>}{<style>}
{<numerator>}{<denominator>}
```

`<left delim.>`: 왼쪽 구분 기호

`<right delim.>`: 오른쪽 구분 기호

`<thickness>`: 구분선의 두께

`<style>`: 이 식을 조판할 스타일

`<numerator>`: 분자 자리(위쪽)에 조판할 식

`<denominator>`: 분모 자리(아래쪽)에 조판할 식

첫 번째와 두 번째 인자는 생략하면 구분 기호가 출력되지 않으며, 세 번째 인자를 생략하면 `\frac`과 같은 두께가 사용됩니다. 네 번째 인자는 비워 두면 현재 스타일을 사용하고, 0~3을 입력하면 각각 별행수식, 행중수식, 첨자, 이중 첨자 스타일로 위쪽 식과 아래쪽 식이 조판됩니다.

이 제어 문자열을 사용하면 두 식을 위아래로 쌓는 형태의 식을 조판하는 제어 문자열을 정의할 수 있습니다. 예를 들어, `\frac`과 `\binom`은 다음과 같이 정의되었다고 생각할 수 있습니다.

```
1 \newcommand\frac{\genfrac{}{}{}{}{}}
2 \newcommand\binom{\genfrac(){}{}{}}
```


또, 식을 조판하는 스타일이 행중수식 스타일로 고정된 `\tfrac`이나 별행수식 스타일로 고정된 `\dbinom` 등은 다음과 동치입니다.

```
1 \newcommand\tfrac{\genfrac{}{}{}1}
2 \newcommand\dbinom{\genfrac(){}{0}}
```

다른 예시로, 정수론에서 사용하는 르장드르 기호는 다음과 같이 정의할 수 있습니다.

```
\newcommand\legendre{\genfrac(){}{}}
```

이렇게 정의한 `\legendre` 제어 문자열은 다음과 같이 사용합니다.

```
1 \[ \legendre{ab}p = \legendre ap
   \legendre bp \]
```

$$\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$$

9.2.3 분자와 분모를 두 줄로 적기

분자나 분모의 내용이 길어 이를 두 줄로 조판하고자 한다면 `\mathtools` 패키지의 `\splitfrac`과 `\splitdfac` 제어 문자열을 사용합니다. 윗줄에 조판할 내용을 첫 번째 인자로, 아랫줄에 조판할 내용을 두 번째 인자로 지정하면 됩니다. 전자는 후자에 비해 두 줄 사이의 간격을 좁게 조판합니다.

```
1 \[\frac{\splitfrac{a+b+c}{+d+e+f+g}}
2      {\splitdfac{a+b+c}{+d+e+f+g}}\]
```

$$\frac{a+b+c}{a+b+c} \\ +d+e+f+g$$

9.3 복잡한 식의 첨자

이번 절에서는 첨자를 여러 줄로 입력하거나, 각종 기호에서 계산 범위로써 사용되는 첨자를 조판하는 위치를 변경하는 방법을 알아봅니다. 첨자를 기호의 왼쪽에 입력할 때 유용한 명령어도 알아봅니다.

9.3.1 첨자 여러 줄로 입력하기

대형 연산자에 복잡한 계산 범위를 입력하거나, 다변수 함수의 미분계수를 나타낼 때에는 첨자를 여러 줄로 입력해야 하기도 합니다. 이때에는 `amsmath` 패키지의 `\substack` 제어 문자열을 사용합니다. 첨자로 입력할 내용을 인자로 지정하면 되며, 줄을 바꿀 때에는 `\\`를 입력합니다.

```

1 \begin{displaymath}
2 \int_{-\infty}^{\infty} f(x) \, dx =
3 \lim_{\substack{a \rightarrow -\infty \\ b \rightarrow \infty}} \int_a^b f(x) \, dx
4 \end{displaymath}

```

$$\int_{-\infty}^{\infty} f(x) dx = \lim_{\substack{a \rightarrow -\infty \\ b \rightarrow \infty}} \int_a^b f(x) dx$$

amsmath 패키지는 subarray 환경도 제공합니다. 이 환경은 인자로 l, c, r 중 하나를 입력받아 각 줄을 왼쪽, 가운데, 오른쪽에 맞추어 정렬합니다.

```

1 \begin{displaymath}
2 \left. \frac{\partial^2 f}{\partial x \partial y} \right|_{\substack{x=1 \\ y=-1}} = 1
3 \end{displaymath}

```

$$\left. \frac{\partial^2 f}{\partial x \partial y} \right|_{\substack{x=1 \\ y=-1}} = 1$$

9.3.2 대형 연산자의 계산 범위

별행수식에서 대형 연산자와 몇 가지 로그형 함수는 계산 범위가 기호의 위아래에 조판되며, 적분 기호의 계산 범위는 기호 오른쪽 위아래에 조판됩니다. 특정 분야에서는 계산 범위를 다른 위치에 조판하기도 하며, 그 외에도 필요에 따라 계산 범위를 일반적인 상황과 다르게 조판할 때도 있습니다.

기호(대형 연산자, 적분 기호, 로그형 함수) 뒤에 \limits를 입력하고 첨자 문법을 사용하면 해당 첨자는 수식의 현재 스타일에 관계없이 기호 위아래에 조판되며, \nolimits를 입력하면 기호의 오른쪽 위아래에 조판됩니다. \displaylimits는 별행수식 스타일에서는 \limits처럼 동작하고, 그 외의 스타일에서는 \nolimits처럼 동작합니다.

```

1 \[\sum\nolimits' a_n, \quad
2 \int\limits_a^b f(x) \, dx\]

```

$$\sum' a_n, \quad \int_a^b f(x) dx$$

계산 범위 위치의 기본값 변경하기

amsmath 패키지는 패키지를 불러올 때 옵션을 지정하여 대형 연산자, 적분 기호, 로그형 함수의 계산 범위를 조판하는 위치의 기본값을 변경할 수 있습니다.

- `sumlimits`와 `nosumlimits` 옵션은 대형 연산자의 계산 범위의 위치를 설정합니다. 전자가 기본으로 사용되며, 후자를 옵션으로 지정하면 계산 범위는 기호의 오른쪽 위아래에 조판됩니다.
- `intlimits`와 `nointlimits` 옵션은 적분 기호의 계산 범위의 위치를 설정합니다. 후자가 기본으로 사용되며, 전자를 옵션으로 지정하면 계산 범위는 기호의 위아래에 조판됩니다.
- `namelimits`와 `nonamelimits` 옵션은 특정 로그형 함수의 계산 범위의 위치를 설정합니다. 전자가 기본으로 사용되며, 후자를 옵션으로 지정하면 계산 범위는 기호의 오른쪽 위아래에 조판됩니다.

다만, 행중수식 스타일로 수식을 조판할 때에는 패키지를 불러올 때 지정한 옵션에 관계없이 항상 기호의 우상단·우하단에 첨자가 조판됩니다. 이때에 첨자를 기호 위아래에 조판하려면 앞서 설명한 `\limits`를 사용해야 합니다.

첨자와 계산 범위 함께 조판하기

대형 연산자에 첨자를 붙이면서 계산 범위까지 함께 조판하려는 경우에는 `amsmath` 패키지의 `\sideset` 제어 문자열을 사용합니다.

`\sideset{<left>}{<right>}{<operator>}`

`<left>`: 기호 왼쪽에 붙일 첨자
`<right>`: 기호 오른쪽에 붙일 첨자
`<operator>`: 첨자를 붙일 연산자

```
\[ \sideset{a^b}{c^d}\prod_{i\in I} A_i ]
```

$$a^b \prod_{i \in I}^d A_i$$

`<operator>` 자리에는 대형 연산자, 적분 기호, 로그형 함수만 입력해야 하며, 이외의 기호를 입력하면 오류가 발생합니다. 따라서 다른 기호의 왼쪽에 첨자를 붙일 때에는 이 제어 문자열을 사용할 수 없습니다.

9.3.3 기호 왼쪽에 첨자 입력하기

8.3.3절에서는 왼쪽 첨자를 입력하는 방법을 언급하였습니다. 하지만, \LaTeX 의 기본 첨자 문법만으로는 만족스럽지 않은 결과를 얻을 때도 있습니다.

예를 들어, 다음 예시에서는 문자 ‘B’ 왼쪽의 첨자 ‘5’와 ‘10’이 오른쪽이 아니라 왼쪽으로 정렬되어 조판됩니다.

```
1 \[ {}_5^{10}\mathrm{B}^{3+} \]
```

$${}^{10}_5\mathrm{B}^{3+}$$

이때에는 mathtools 패키지의 `\prescript` 제어 문자열을 사용하여 문제를 해결할 수 있습니다. 이 제어 문자열을 사용하면 기호 왼쪽의 첨자가 오른쪽으로 정렬되어 조판됩니다.

`\prescript{<superscript>}{<subscript>}{<arg.>}`

`<superscript>`: 왼쪽에 적을 위 첨자

`<subscript>`: 왼쪽에 적을 아래 첨자

`<arg.>`: 첨자를 붙일 문자

```
1 \[ \prescript{10}{5}{\mathrm{B}}^{3+} \]
```

$${}^{10}_5\mathrm{B}^{3+}$$

`\prescript` 제어 문자열을 사용하였을 때, 기호 오른쪽에 위 첨자와 아래 첨자를 모두 사용하지 않았다면 양쪽 첨자의 위치가 일치하지 않을 수 있습니다. 이때에는 빈 자리에 `\mathstrut` 제어 문자열을 입력해 주면 위치를 교정할 수 있습니다.

아래 예시에서는 왼쪽 아래 첨자 ‘b’와 세로 위치가 맞지 않는 오른쪽 아래 첨자 ‘c’를 교정하였습니다.

```
1 \[ \prescript{a}{b}{X}_c, \quad \quad
```

```
2 \prescript{a}{b}{X}_c^{\mathstrut} \]
```

$${}_b^aX_c, \quad {}_b^aX_c$$

9.4 기호 위아래에 기호 입력하기

이번 절에서는 기호 위아래에 다른 기호를 입력하는 방법을 알아봅니다. 이 기능을 사용하면 기존의 이항 연산자나 관계 기호에 새로운 의미를 부여하거나 부연 설명을 덧붙일 수 있습니다.

9.4.1 화살표 위아래에 기호 입력하기

연산을 통해 수학적 대상의 변화 과정을 나타낼 때에는 화살표 위아래에 부가 설명을 붙여 줌으로써 독자의 이해를 도울 수 있습니다. `amsmath` 패키지와 `mathtools` 패키지는 이때 사용할 수 있는 제어 문자열을 제공하며, 이를 표 9.2에 정리하였습니다.

표 9.2 | 화살표 위아래에 기호를 입력하는 제어 문자열

amsmath 패키지에서 제공하는 기호:

| | | | |
|--------------------|------------------------------------|---------------------|-------------------------------------|
| \xleftarrow{abc} | <code>\xleftarrow[def]{abc}</code> | \xrightarrow{abc} | <code>\xrightarrow[def]{abc}</code> |
|--------------------|------------------------------------|---------------------|-------------------------------------|

mathtools 패키지에서 제공하는 기호:

| | | | |
|----------------------------|--|-------------------------|--|
| \hookleftarrow{abc} | <code>\hookleftarrow[def]{abc}</code> | \xhookrightarrow{abc} | <code>\xhookrightarrow[def]{abc}</code> ¹ |
| \hookrightarrow{abc} | <code>\hookrightarrow[def]{abc}</code> | \xhookleftarrow{abc} | <code>\xhookleftarrow[def]{abc}</code> ¹ |
| \Leftrightarrow{abc} | <code>\Leftrightarrow[def]{abc}</code> | \xmapsto{abc} | <code>\xmapsto[def]{abc}</code> |
| \leftharpoonup{abc} | <code>\leftharpoonup[def]{abc}</code> | \xrightarrow{abc} | <code>\xrightarrow[def]{abc}</code> |
| \leftharpoonup{abc} | <code>\leftharpoonup[def]{abc}</code> | \xrightarrow{abc} | <code>\xrightarrow[def]{abc}</code> |
| \leftrightharpoonup{abc} | <code>\leftrightharpoonup[def]{abc}</code> | \xrightarrow{abc} | <code>\xrightarrow[def]{abc}</code> |
| \leftrightharpoonup{abc} | <code>\leftrightharpoonup[def]{abc}</code> | \xrightarrow{abc} | <code>\xrightarrow[def]{abc}</code> |
| \leftrightharpoonup{abc} | <code>\leftrightharpoonup[def]{abc}</code> | \xrightarrow{abc} | <code>\xrightarrow[def]{abc}</code> |
| \leftrightharpoonup{abc} | <code>\leftrightharpoonup[def]{abc}</code> | \xrightarrow{abc} | <code>\xrightarrow[def]{abc}</code> |
| \leftrightharpoonup{abc} | <code>\leftrightharpoonup[def]{abc}</code> | \xrightarrow{abc} | <code>\xrightarrow[def]{abc}</code> |

¹ `\xleftarrow`나 `\xrightarrow`와 달리, 인자로 짧은 내용이 지정되더라도 `\long...arrow`의 길이가 유지됩니다.

```

1 \begin{displaymath}
2   \begin{bmatrix} 1&0&0 \\ 0&0&1 \\ 0&1&0 \end{bmatrix} \xrightarrow{\text{2행} \rightarrow \text{3행}} \begin{bmatrix} 1&0&0 \\ 0&1&0 \\ 0&0&1 \end{bmatrix}
3   \begin{bmatrix} 1&0&0 \\ 0&1&0 \\ 0&0&1 \end{bmatrix} \xrightarrow{\text{2행} \leftrightarrow \text{3행}} \begin{bmatrix} 1&0&0 \\ 0&1&0 \\ 0&0&1 \end{bmatrix}
4   \begin{bmatrix} 1&0&0 \\ 0&1&0 \\ 0&0&1 \end{bmatrix} \xrightarrow{\text{2행} \leftrightarrow \text{3행}} \begin{bmatrix} 1&0&0 \\ 0&1&0 \\ 0&0&1 \end{bmatrix}
5 \end{displaymath}

```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \xrightarrow[\text{2행} \leftrightarrow \text{3행}]{\text{E.R.O. (1)}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

9.4.2 기호 위아래에 기호 덧붙이기

특정 분야에서는 새로운 표기법을 정의할 때 등호 ‘=’ 위에 ‘def’라는 문자열을 적은 기호 ‘ $\stackrel{\text{def}}{=}$ ’를 사용하기도 합니다. 이처럼, 관계 기호 위에 다른 기호를 덧붙일 때에는 `\stackrel` 제어 문자열을 사용합니다.

`\stackrel{\langle above \rangle}{\langle symbol \rangle}`

`\langle above \rangle`: 관계 기호 위에 덧붙일 기호

`\langle symbol \rangle`: 기호를 덧붙일 관계 기호

입력

$$\backslash[A \ \stackrel{\mathrm{def}}{=} B \backslash]$$

출력

$$A \stackrel{\mathrm{def}}{=} B$$

`stackrel` 패키지를 불러오면 `\stackrel`의 이항 연산자 버전인 `\stackbin` 제어 문자열을 사용할 수 있게 됩니다. 또, 두 제어 문자열이 선택 인자를 입력받아 기호 아래에도 다른 기호를 덧붙일 수 있게 됩니다.

`\stackrel[⟨below⟩]{⟨above⟩}{⟨symbol⟩}`
`\stackbin[⟨below⟩]{⟨above⟩}{⟨symbol⟩}`

⟨below⟩: 관계 기호·이항 연산자 아래에 덧붙일 기호

⟨above⟩: 관계 기호·이항 연산자 위에 덧붙일 기호

⟨symbol⟩: 기호를 덧붙일 관계 기호·이항 연산자

입력

$$\backslash[A \ \stackrel{x}{\sim} B \backslash]$$

출력

$$A \stackrel{x}{\sim} B$$

$$\backslash[A \ \stackbin{x}{\cup} B \backslash]$$

$$A \stackbin{x}{\cup} B$$

관계 기호가 아닌 임의의 기호 위나 아래에 다른 기호를 덧붙일 때에는 `amsmath` 패키지의 `\overset`과 `\underset` 제어 문자열을 사용합니다. `\overunderset` 제어 문자열을 사용하면 기호를 위아래에 덧붙입니다.

입력

$$\backslash[\overset{*}{X} \backslash]$$

출력

$$X^*$$

$$\backslash[\underset{*}{X} \backslash]$$

$$X_*$$

$$\backslash[\overunderset{\wedge}{\vee}{X} \backslash]$$

$$\hat{X}_\vee$$

9.4.3 식 위아래에 화살표·괄호 그리기

수식의 위나 아래에 해당 수식의 길이에 맞는 화살표나 괄호를 그릴 수도 있습니다. 이때 사용할 수 있는 제어 문자열을 표 9.3에 정리하였습니다. 특히, 괄호를 조판하는 제어 문자열 네 개는 첨자 문법을 사용하여 괄호 위나 아래에 다른 내용을 입력할 수 있습니다.

$$\backslash[\ \vec{p} = \overrightarrow{\mathrm{AB}} \backslash]$$

$$\vec{p} = \overrightarrow{AB}$$

$$\backslash[\ a^n = \underbrace{a \cdot a \cdots a}_{\text{\text{(n) times}}} \backslash]$$

$$a^n = \underbrace{a \cdot a \cdots a}_{n \text{ times}}$$

표 9.3 | 식 위아래에 그리는 화살표와 괄호

| | |
|--|--|
| \overleftarrow{abc} <code>\overleftarrow{abc}</code> | \overrightarrow{abc} <code>\overrightarrow{abc}</code> |
| \overbrace{abc} <code>\overbrace{abc}</code> | \underbrace{abc} <code>\underbrace{abc}</code> |
| amsmath 패키지에서 제공하는 기호: | |
| \overleftrightarrow{abc} <code>\overleftrightarrow{abc}</code> | $\underleftrightarrow{abc}$ <code>\underleftrightarrow{abc}</code> |
| \underlineleftarrow{abc} <code>\underlineleftarrow{abc}</code> | \underrightarrow{abc} <code>\underrightarrow{abc}</code> |
| mathtools 패키지에서 제공하는 기호: | |
| \overbracket{abc} <code>\overbracket{abc}</code> | \underbracket{abc} <code>\underbracket{abc}</code> |

특히, `\overbracket`과 `\underbracket`은 선택 인자를 입력받아 괄호의 모양을 조정할 수 있습니다.

`\overbracket[⟨thickness⟩][⟨height⟩]{⟨subformula⟩}`
`\underbracket[⟨thickness⟩][⟨height⟩]{⟨subformula⟩}`

⟨thickness⟩: 획의 두께

⟨height⟩: 괄호의 높이

⟨subformula⟩: 위 또는 아래에 괄호를 그릴 수식

입력

`\(\underbracket{wxyz}\)`

`\(\underbracket[.4pt]{wxyz}\)`

`\(\underbracket[4pt][4pt]{wxyz}\)`

출력

$wxyz$

$wxyz$

$wxyz$

9.5 수식 글꼴

수식의 문자는 사용된 글꼴에 따라 의미가 달라지기도 합니다. 예를 들어, **u**, **v**처럼 볼드체는 벡터를 나타내고, *T*처럼 흘림체는 집합의 모임을 나타내고, \mathbb{R} 처럼 칠판 볼드체는 특정 수 집합을 나타냅니다. 또, **p**와 같은 독일 활자체(프락투르체)는 환의 아이디얼을 나타내기도 합니다.

LaTeX에서는 표 9.4의 제어 문자열을 사용해 다른 글꼴을 사용할 수 있습니다. 여기의 제어 문자열은 모두 인자형이며, 선언형 제어 문자열은 없습니다. 수식 글꼴은 로마자, 숫자, 그리스 문자 대문자에만 영향을 미치며, 글꼴에 따라 이중 일부에만 유효할 수도 있습니다.

표 9.4 | 수식 모드의 글꼴

| 글꼴 | 제어 문자열 | 예시 |
|---------|--------------------------|--|
| 기본 | — | $abcd\ ABCD\ 1234\ \Gamma\Delta\Theta\Lambda$ |
| 이탤릭체 | <code>\mathit</code> | $abcd\ ABCD\ 1234\ \Gamma\Delta\Theta\Lambda$ |
| 수식 이탤릭체 | <code>\mathnormal</code> | $abcd\ ABCD\ 1234\ \Gamma\Delta\Theta\Lambda$ |
| 정체 | <code>\mathrm</code> | $abcd\ ABCD\ 1234\ \Gamma\Delta\Theta\Lambda$ |
| 볼드체 | <code>\mathbf</code> | $\mathbf{abcd\ ABCD\ 1234\ \Gamma\Delta\Theta\Lambda}$ |
| 산세리프체 | <code>\mathsf</code> | $abcd\ ABCD\ 1234\ \Gamma\Delta\Theta\Lambda$ |
| 고정폭 서체 | <code>\mathtt</code> | $\mathtt{abcd\ ABCD\ 1234\ \Gamma\Delta\Theta\Lambda}$ |
| 흘림체 | <code>\mathcal</code> | $\mathcal{ABCDEFGHIJ}^1$ |
| 독일 활자체 | <code>\mathfrak</code> | $\mathfrak{abcd\ ABCD\ 1234}^{2,3}$ |
| 칠판 볼드체 | <code>\mathbb</code> | $\mathbb{ABCDEFGHIJ}^{1,3}$ |

¹ 로마자 대문자에만 적용됩니다.

² 로마자와 숫자에만 적용됩니다.

³ `amssymb` 패키지나 `amsfonts` 패키지를 불러와야 사용할 수 있습니다.

입력

선분~`\(\mathrm{AB}\)`

벡터~`\(\mathbf{u}\)`

위상~`\(\mathcal{T}\)`

아이디얼~`\(\mathfrak{p}\)`

실수 집합~`\(\mathbb{R}\)`

출력

선분 AB

벡터 **u**

위상 \mathcal{T}

아이디얼 \mathfrak{p}

실수 집합 \mathbb{R}

수식에서 사용되는 이탤릭체는 ‘수식 이탤릭체’로, 수식을 더 보기 좋게 조판할 수 있도록 텍스트 모드에서의 이탤릭체와는 다르게 디자인 되어 있습니다. `\mathit`로 사용할 수 있는 (텍스트 모드의) 이탤릭체는 여러 글자로 이루어진 변수 이름 등에 사용합니다. 주로 전산학 분야에서 사용합니다.

입력

`\(\mathit{head} :: \mathit{tail}\)`

`\(\mathit{num}_1 + \mathit{num}_2\)`

출력

$head :: tail$

$num_1 + num_2$

필기체 글꼴

\LaTeX 과 `amsfonts` 패키지에서 기본으로 제공하지 않는, ‘ \mathcal{A} , \mathcal{B} , \mathcal{C} ’와 같은 필기체 글꼴을 수식에 사용하기도 합니다. 이 글꼴은 `mathrsfs` 패키지를 불러온 후 `\mathscr`

제어 문자열을 통해 사용합니다. 한편, `mathrsfs` 패키지 대신 `calrsfs` 패키지를 불러오면 `\mathcal`로 흘림체 대신 필기체 글꼴을 사용할 수 있습니다.

```
1 \[ H^0(\mathcal U, \mathscr F) =
   \mathscr F(X) \]
```

$$H^0(\mathcal{U}, \mathscr{F}) = \mathscr{F}(X)$$

볼드체 글꼴

`amsfonts` 패키지는 문자를 볼드체로 조판하는 `\boldsymbol` 제어 문자열과 `\pmb` 제어 문자열을 제공합니다. `\mathbf`와 달리 `\boldsymbol`은 (대형 연산자나 높이가 조절된 괄호 등을 제외하고) \LaTeX 기본 글꼴에서 제공하는 모든 기호를 볼드체로 조판합니다. 로마자와 숫자, 그리스 대문자의 경우, 표 9.4의 제어 문자열(단, `\mathbb`는 제외)을 사용해 글꼴을 바꾸었다면 해당 글꼴의 볼드체 버전으로 조판됩니다.

```
1 \[U \in \mathcal T, \quad
2 \boldsymbol{U \in \mathcal T}\]
```

$$U \in \mathcal{T}, \quad \boldsymbol{U \in \mathcal{T}}$$

`\pmb` 제어 문자열은 볼드체를 제공하지 않는 기호도 볼드체처럼 보이도록 기호를 여러 번 겹쳐 조판해 줍니다. 조판 결과가 보기 좋지 않으므로, 다른 방법으로는 볼드체를 사용할 수 없을 때에 한해 이 제어 문자열을 사용하는 것이 좋습니다.

```
1 \[\sum_{n=1}^{\infty} a_n, \quad
2 \boldsymbol{\mathop{\pmb{\sum}}_{n=1}
   ^{\infty} a_n}, \quad
3 \pmb{\sum_{n=1}^{\infty} a_n}\]
```

$$\sum_{n=1}^{\infty} a_n, \quad \boldsymbol{\sum_{n=1}^{\infty} a_n}, \quad \pmb{\sum_{n=1}^{\infty} a_n}$$

`\boldsymbol`과 `\pmb` 제어 문자열은 인자로 지정된 수식과 나머지 부분 사이의 간격에 영향을 미칠 수 있으므로, 조판 결과가 이상하다면 별도의 명령어를 사용하여 기호 사이의 간격을 고쳐야 합니다. 특히, 위 예시와 같이 대형 연산자를 볼드체로 조판한 경우에는 `\mathop`를 사용해야 계산 범위가 올바른 위치에 배치됩니다.

기호를 정체로 입력하는 명령어에 관한 조언

수식을 작성하다 보면 내용을 정체로 작성해야 하는 경우가 생깁니다. 이때 사용할 수 있는 명령어로 `\operatorname{(*)}`, `\mathrm`, `\text`가 있습니다. 이 명령어는 모두 다른 의미를 가지고 있고, 경우에 따라 다른 결과를 만들 수 있으므로 구분하여 사용하는 것이 좋습니다.

- `\operatorname{(*)}` 명령어의 이름에서 알 수 있듯 연산자(로그형 함수)의 이름을 입력할 때 사용합니다. 최소공배수, 역쌍곡선함수 등 \LaTeX 에서 제공하지 않는 연산자를 입력할 때 이 제어 문자열을 사용하면 됩니다.
- `\mathrm` 연산자의 이름이 아닌, 다른 수학적 의미를 가지는 로마자를 정체로 적을 때 사용합니다. 적분식에서의 'd'를 정체로 적거나 변수의 이름에 첨자로써 설명을 붙일 때 사용합니다.
- `\text` 수식 식이 아닌 일반 텍스트를 작성할 때 사용합니다. 특히, 수식 앞뒤에서 기본 글꼴이 아닌 다른 글꼴을 사용하고 있었다면 `\text`로 입력한 문자열에도 그 글꼴이 사용되므로, 내용이 항상 정체로 조판된다고 할 수도 없습니다.

다음은 세 가지 명령어를 올바르게 구분해서 사용한 예시입니다. 여기에서 `\max` 대신 `\operatorname*{max}`를 사용한 것은 예시를 들기 위함입니다.

```
1 \[ x_{\mathrm{max}} =
   \operatorname*{max}_{i \in I} x_i =
   (\text{\(x_i\)}들 중 최대인 것) \]
```

$$x_{\max} = \max_{i \in I} x_i = (x_i \text{들 중 최대인 것})$$

이전 버전 명령어에 대한 조언

글꼴을 전환하는 플레인 \TeX 및 \LaTeX 2.09 의 선언형 명령어 `\rm`, `\it`, `\mit`, `\bf`, `\cal`, `\tt`는 이전 문서와의 호환을 위해 제공됩니다. 71쪽에서 설명한 것과 같은 이유로, 이 선언형 명령어 대신 \LaTeX 2\epsilon 의 인자형 명령어를 사용하는 것이 바람직합니다. `\rm` 대신 `\mathrm`을, `\it` 대신 `\mathit`을, `\mit` 대신 `\mathnormal`을, `\bf` 대신 `\mathbf`을, `\cal` 대신 `\mathcal`을, `\tt` 대신 `\mathtt`를 사용하십시오.

9.5.1 볼드체 수식

수식 앞뒤의 내용이 볼드체 글꼴로 조판되고 있었다면 수식 전체를 더 진한 글꼴로 조판하여 앞뒤의 글꼴과 어울리도록 할 수 있습니다.² 이때에는 각 기호에 `\boldsymbol` 등을 사용하지 않고, 텍스트 모드에서 `\boldmath`를 입력하여 수식 자체를 볼드체로 조판하도록 선언해 줍니다. `\unboldmath` 제어 문자열은 수식을 원래의 글꼴로 조판하도록 선언합니다. 표 9.4의 글꼴 명령어와는 달리, 이 두 제어 문자열은 텍스트 모드에서 사용해야 합니다.

² 글꼴에 따라 수식의 의미가 달라질 수도 있으므로 이 기능은 주의해서 사용해야 합니다.

집합 \mathcal{X} 가 x 를 원소로 가지면 이를 $\{x \in \mathcal{X}\}$ 로 나타내고, \mathcal{X} 에 속한다}고 말한다.

집합 X 가 x 를 원소로 가지면 이를 $x \in X$ 로 나타내고, x 가 X 에 속한다고 말한다.

연습문제

9.1. 다음 수식을 조판하여라.

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21} \quad (\text{이차 정사각행렬의 행렬식})$$

9.2. 다음 문장을 조판하여라.

$$\text{If } A = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \text{ then } A^{-1} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}.$$

9.3. 다음 수식을 조판하여라.

$$\begin{cases} u_{tt} - c^2 \Delta u = 0, & t > 0, \mathbf{x} \in \mathbb{R}^3 \\ u(\mathbf{x}, 0) = \phi(\mathbf{x}), \quad u_t(\mathbf{x}, 0) = \psi(\mathbf{x}), & \mathbf{x} \in \mathbb{R}^3 \\ u(0, t) = 0, & t > 0 \end{cases} \quad (\text{파동 방정식})$$

9.4. 다음 수식을 조판하여라. (힌트: mathtools 패키지의 환경을 사용하면 편리하다.)

$$(-1)^n = \begin{cases} 1, & n \text{이 짝수일 때} \\ -1, & n \text{이 홀수일 때} \end{cases}$$

9.5. 첨자의 위치에 주의하여 다음 수식을 조판하여라.

$$(a) \quad X_1 \cdots X_l = \sum_{\gamma} \prod_{\{v, v'\}} \mathbf{E}[\xi_v \xi_{v'}] \odot_{v''} \xi_{v''}$$

$$(b) \quad \bigcup_{i=1}^{\infty} A_i \cap \left(\bigcap_{j \in J} B_j \right) = \bigcap_{j \in J} A_i \cap \left(\bigcup_{i=1}^{\infty} B_j \right)$$

9.6. 다음 수식을 조판하여라.

$$\int_{-\infty}^{\infty} f(x) dx = \lim_{\substack{a \rightarrow -\infty \\ b \rightarrow \infty}} \int_a^b f(x) dx \quad (\text{이상 적분})$$

9.7. 다음 수식을 조판하여라.

$$\mathbf{B}(P) = \frac{\mu_0}{4\pi} \int \frac{\mathbf{I} \times \hat{\mathbf{r}}'}{r'^2} dl \quad (\text{비오-사바르 법칙})$$

제 10 장

별행수식 작성하기

이번 장에서는 별행수식과 관련된 `amsmath` 패키지와 `mathtools` 패키지의 기능을 알아봅니다. 여러 줄로 별행수식을 조판하는 기능과 수식의 번호를 계산하고 조판하는 기능을 다룰 것입니다.

10.1 기본 환경

\LaTeX 에는 `displaymath` 환경과 `equation` 환경이 정의되어 있고, `amsmath` 패키지를 불러오면 `equation*` 환경이 추가로 정의됩니다. 앞서 설명했듯 `equation` 환경은 번호가 붙은 별행수식을 작성하는 환경이며, `displaymath` 환경은 번호 없는 별행수식을 작성하는 환경입니다. `equation*` 환경은 `displaymath` 환경, 그리고 `\[...\]` 명령어와 같은 역할을 합니다.

1 다음을 가우스-보네 정리라고 한다.

2 `\begin{equation}`

3 `\int_M K \, dA + \int_{\partial M} k_g \, ds = 2\pi\chi(M)`

4 `\end{equation}`

5 이때 `\(M\)`이 경계가 없는 곡면이라면 다음 식이 성립한다.

6 `\begin{equation*}`

7 `\int_M K \, dA = 2\pi\chi(M)`

8 `\end{equation*}`

다음을 가우스-보네 정리라고 한다.

$$\int_M K \, dA + \int_{\partial M} k_g \, ds = 2\pi\chi(M) \quad (10.1)$$

이때 M 이 경계가 없는 곡면이라면 다음 식이 성립한다.

$$\int_M K \, dA = 2\pi\chi(M)$$

이 세 가지 환경과 `\[...\]` 명령어는 여러 줄에 걸쳐 수식을 작성하는 기능을 지원하지 않습니다. 수식이 글줄보다 길어지더라도 자동으로 줄이 바뀌지 않으며, `\\`를

입력하여 줄을 수동으로 나눌 수도 없습니다. 따라서 긴 수식을 조판해야 하거나 여러 수식을 함께 조판하려면 다음 절에서 설명하는 환경을 사용해야 합니다.

10.2 여러 줄의 수식

amsmath 패키지는 여러 줄에 걸쳐 수식을 작성하기 위한 환경으로 다음을 제공합니다.

| | | | | |
|-----------|---------|--------|----------|----------|
| split | | | | |
| multline | gather | align | alignat | flalign |
| multline* | gather* | align* | alignat* | flalign* |

첫째 줄에 있는 split 환경은 다른 별행수식 작성 환경 안에서 사용해야 하며, 둘째 줄과 셋째 줄에 있는 환경은 텍스트 모드에서 사용해야 합니다. equation과 equation* 환경의 관계처럼, 둘째 줄에 있는 환경을 사용하면 수식의 번호가 조판되며, 셋째 줄에 있는 환경을 사용하면 수식 번호가 조판되지 않습니다. 이번 절에서는 적절한 환경을 사용해 여러 줄로 수식을 조판하는 방법을 알아보겠습니다.

위의 환경 안에서는 공통적으로 \\로 줄을 바꾸고, &로 정렬 지점을 설정합니다. 다만 gather처럼 정렬 지점 없이 수식을 조판하는 환경에서는 &를 사용할 수 없습니다.

10.2.1 하나의 수식을 여러 줄로 조판하기

한 개의 수식을 여러 줄에 걸쳐 조판할 때에는 multline 환경이나 split 환경을 사용합니다.

정렬 지점 없이 조판: multline 환경

multline 환경은 하나의 수식을 특별한 정렬 지점 없이 여러 줄에 걸쳐 조판할 때 사용합니다. 첫 줄은 왼쪽 끝에, 마지막 줄은 오른쪽 끝에 조판되며, 나머지 줄들은 가운데에 정렬되어 조판됩니다.

```

1 \begin{multline}
2 X = a+b+c-d+e \\ -f+g-h+i+j \\
3 -k-l+m+n-o
4 \end{multline}

```

$$\begin{aligned}
 X &= a + b + c - d + e \\
 &\quad - f + g - h + i + j \\
 &\quad - k - l + m + n - o \quad (10.2)
 \end{aligned}$$

\shoveleft 제어 문자열은 현재 행을 왼쪽 끝에 조판합니다. 반대로, \shoveright는 현재 행을 오른쪽 끝에 조판합니다. 정렬 방향을 바꿀 행 전체를 이들의 인자로 지정하면 됩니다. 단, 첫째 행과 마지막 행에서는 이 제어 문자열을 사용할 수 없습니다.

```

1 \begin{multline*}
2   X = a+b+c-d+e \\
3   \shoveright{-f+g-h+i+j} \\
4   \shoveleft{-k-l+m+n-o} \\
5   +p-q-r-s+t
6 \end{multline*}

```

$$\begin{aligned}
 X = a + b + c - d + e \\
 \qquad \qquad \qquad - f + g - h + i + j \\
 - k - l + m + n - o \\
 \qquad \qquad \qquad + p - q - r - s + t
 \end{aligned}$$

정렬 지점을 기준으로 조판: `split` 환경

수식의 각 줄을 특정 지점을 기준으로 정렬하여 조판하려면 별행수식 작성 환경 안에서 `split` 환경을 사용합니다. 일반적으로 정렬 지점은 관계 기호의 왼쪽에 설정합니다.

```

1 \begin{equation}
2   \begin{split}
3     X &= a+b+c-d+e \\
4   \end{split}
5 \end{equation}

```

$$\begin{aligned}
 X = a + b + c - d + e \\
 \qquad \qquad \qquad - f + g - h + i + j
 \end{aligned} \tag{10.3}$$

이 환경은 `gather` 환경 안의 수식 하나를 여러 줄에 걸쳐 조판할 때에도 사용할 수 있습니다. 단, `multline` 환경 안에서는 사용할 수 없습니다.

10.2.2 여러 수식을 세로로 나열하기

여러 수식을 세로로 나열하여 조판할 때에는 `gather` 환경과 `align` 환경을 사용합니다.

정렬 지점 없이 조판: `gather` 환경

여러 수식을 정렬하지 않고 단순히 나열하고자 한다면 `gather` 환경을 사용합니다.

```

1 \begin{gather}
2   X = a_1+b_1-c \\
3   Y = a_2-b_2
4 \end{gather}

```

$$X = a_1 + b_1 - c \tag{10.4}$$

$$Y = a_2 - b_2 \tag{10.5}$$

`gather` 환경을 사용하면 `equation` 환경을 여러 번 사용했을 때보다 각 수식 사이의 공간이 줄어들므로, 여러 수식을 세로로 나열해야 한다면 `gather` 환경을 사용하는 것이 좋습니다.

정렬 지점을 기준으로 조판: align 환경

align 환경을 사용하면 특정 지점을 기준으로 수식을 정렬할 수 있습니다. split 환경을 사용할 때와 마찬가지로, 주로 관계 기호의 왼쪽에 정렬 지점을 설정합니다.

```
1 \begin{align}
2   X_1 &= a_1 + b_1 - c \quad \backslash\backslash X_2 &= a_2 - b_2
3 \end{align}
```

$$X_1 = a_1 + b_1 - c \quad (10.6)$$

$$X_2 = a_2 - b_2 \quad (10.7)$$

여러 줄에 걸쳐 식의 변화 과정을 작성할 때에도 align 환경을 사용합니다. 이때, 식의 오른쪽에 &&를 입력하고 \text를 사용하면 변화 과정에 대한 부연 설명을 작성할 수도 있습니다.

```
1 \begin{align*}
2   X &= a+b-a-b \quad \backslash\backslash
3   &= a+b-b-a \quad \&\& \text{by commutativity} \quad \backslash\backslash
4   &= a+(b-b)-a \quad \&\& \text{by associativity} \quad \backslash\backslash
5   &= a-a = 0
6 \end{align*}
```

$$\begin{aligned} X &= a + b - a - b \\ &= a + b - b - a && \text{by commutativity} \\ &= a + (b - b) - a && \text{by associativity} \\ &= a - a = 0 \end{aligned}$$

eqnarray 환경에 대한 조언

L^AT_EX은 align(*) 환경과 비슷한 환경으로 eqnarray 환경과 eqnarray* 환경을 제공합니다. 이 환경에서는 &를 관계 기호의 앞뒤에 하나씩 입력하여 수식을 정렬합니다.

```
1 \begin{eqnarray}
2   X_1 &=& a_1 + b_1 - c \quad \backslash\backslash X_2 &=& a_2 - b_2
3 \end{eqnarray}
```

$$X_1 = a_1 + b_1 - c \quad (10.8)$$

$$X_2 = a_2 - b_2 \quad (10.9)$$

하지만, 이 환경은 일반적인 수식에 비해 &가 입력된 부분의 간격을 넓게 조판하므로 조판 결과가 보기 좋지 않습니다. 따라서, eqnarray(*) 환경 대신 align(*) 환경을 사용하는 것을 권장합니다.

10.2.3 여러 수식을 여러 열과 행으로 조판하기

align 환경은 단순히 수식을 세로로 나열하는 것뿐 아니라, 가로 방향으로도 나열할 수 있습니다. 같은 행에 수식을 여러 열로 나열하는 경우 정렬 지점 외에 수식과 수식 사이에도 &를 입력해 주면 됩니다.

```
1 \begin{align*}
2   A_{11} &= P_{11}+Q_{11} & A_{12} &= P_{12}-Q_{12} \\
3   A_{21} &= P_{21}-Q_{21} & A_{22} &= P_{22}+Q_{22} \\
4 \end{align*}
```

$$\begin{array}{ll} A_{11} = P_{11} + Q_{11} & A_{12} = P_{12} - Q_{12} \\ A_{21} = P_{21} - Q_{21} & A_{22} = P_{22} + Q_{22} \end{array}$$

앞서 설명했던, &&를 입력한 뒤 \text를 사용하여 수식에 대한 설명을 적는 것도 이 기능을 사용한 것입니다.

align 환경을 사용해 수식을 여러 열로 배치한 경우, 첫 열 왼쪽의 여백, 마지막 열 오른쪽의 여백, 그리고 열과 열 사이의 공간이 균일하도록 조판됩니다.

열과 열 사이의 간격을 직접 지정: alignat 환경

수식과 수식 사이의 간격을 원하는 대로 조정하려는 경우에는 alignat 환경을 사용합니다. 열과 열 사이에 공백이 만들어지지 않으므로, \quad나 \qquad 등을 사용해 수식 사이의 공간을 원하는 대로 지정할 수 있습니다.

```
1 \begin{alignat*}{2}
2   A_{11} &= P_{11}+Q_{11} &\qquad A_{12} &= P_{12}-Q_{12} \\
3   A_{21} &= P_{21}-Q_{21} &\qquad A_{22} &= P_{22}+Q_{22} \\
4 \end{alignat*}
```

$$\begin{array}{ll} A_{11} = P_{11} + Q_{11} & A_{12} = P_{12} - Q_{12} \\ A_{21} = P_{21} - Q_{21} & A_{22} = P_{22} + Q_{22} \end{array}$$

한편, 이 환경은 한 줄에 배치할 수식의 수를 인자로 지정해 주어야 합니다. 따라서 한 줄에는 &가 $2 \times (\text{인자로 지정한 값}) - 1$ 개 있어야 합니다. 위 예시에서는 한 줄에 식이 두 개씩 배치되므로 인자로 2를 지정하였고, 한 줄에는 &가 세 개 입력되었음을 확인할 수 있습니다.

좌우 여백 없이 조판: flalign 환경

align 환경과 alignat 환경 외에, flalign 환경을 사용할 수도 있습니다. 이 환경은 align 환경과 동일한 역할을 하지만, 첫 수식의 왼쪽과 마지막 수식의 오른쪽에 여백을 만들지 않습니다.

```

1 \begin{flalign*}
2   A_{11} &= P_{11}+Q_{11} & A_{12} &= P_{12}-Q_{12} & A_{13} &= \\
   &P_{13}+Q_{13} \\
3   A_{21} &= P_{21}-Q_{21} & A_{22} &= P_{22}+Q_{22} & A_{23} &= \\
   &P_{23}-Q_{23} \\
4 \end{flalign*}

```

$$A_{11} = P_{11} + Q_{11}$$

$$A_{12} = P_{12} - Q_{12}$$

$$A_{13} = P_{13} + Q_{13}$$

$$A_{21} = P_{21} - Q_{21}$$

$$A_{22} = P_{22} + Q_{22}$$

$$A_{23} = P_{23} - Q_{23}$$

10.2.4 수식과 수식 사이의 일반 텍스트

align 등의 환경으로 수식을 정렬해 조판한 경우, 수식의 정렬을 흐뜨러뜨리지 않고 수식과 수식 사이에 일반 텍스트를 입력해야 하는 경우도 있습니다. 이때에는 amsmath 패키지의 \intertext 제어 문자열을 사용합니다. 8.3.4절에서 설명한 \text와 마찬가지로, 입력하고자 하는 일반 텍스트를 인자로 지정하면 됩니다.

mathtools 패키지는 \shortintertext 제어 문자열을 제공합니다. 이 제어 문자열은 \intertext와 같은 동작을 하지만, 일반 텍스트가 조판되는 글줄 위아래에 여백을 더 적게 만듭니다. 짧은 내용을 일반 텍스트로 조판하는 경우에 이 제어 문자열을 사용하면 지면을 효율적으로 사용할 수 있습니다.

```

1 By Taylor's series, we have that
2 \begin{align*}
3   \cos x &= 1 - \frac{1}{2!}x^2 + \frac{1}{4!}x^4 - \frac{1}{6!}x^6 + \\
   &\cdots \\
4   \shortintertext{and} \\
5   \sin x &= x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 - \frac{1}{7!}x^7 + \\
   &\cdots. \\
6   \intertext{Therefore, we obtain} \\
7   \cos x + i \sin x &= 1 + ix - \frac{1}{2!}x^2 - \frac{i}{3!}x^3 + \cdots \\
   &\backslash

```

```

8   &= 1 + ix + \frac{1}{2!}(ix)^2 + \frac{1}{3!}(ix)^3 + \cdots \\
9   &= e^{ix}.
10 \end{align*}

```

By Taylor's series, we have that

$$\cos x = 1 - \frac{1}{2!}x^2 + \frac{1}{4!}x^4 - \frac{1}{6!}x^6 + \cdots$$

and

$$\sin x = x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 - \frac{1}{7!}x^7 + \cdots$$

Therefore, we obtain

$$\begin{aligned} \cos x + i \sin x &= 1 + ix - \frac{1}{2!}x^2 - \frac{i}{3!}x^3 + \cdots \\ &= 1 + ix + \frac{1}{2!}(ix)^2 + \frac{1}{3!}(ix)^3 + \cdots \\ &= e^{ix}. \end{aligned}$$

참고 | `\intertext`나 `\shortintertext`는 `split` 환경 안에서는 사용할 수 없습니다.

10.3 수식의 번호

이번 절에서는 수식 번호와 관련된 여러 작업을 소개합니다. `gather`나 `align` 등의 환경에서 특정 줄은 수식 번호를 표시하지 않도록 할 수도 있고, 번호 대신 별표 등의 원하는 문구를 표시할 수도 있습니다. 여러 수식이 번호를 공유하고, 그 하위 번호로써 구분되도록 하는 방법도 알아봅니다.

10.3.1 번호 지우거나 임의의 기호 붙이기

번호 지우기

`gather`, `align`, `alignat`, `flalign` 환경은 각 줄에 수식 번호를 조판합니다. 이때, 특정 줄에 `\notag` 제어 문자열을 입력하면 그 줄에서는 수식 번호가 출력되지 않습니다.

```

1 \begin{align}
2   f'(x) &= (g \circ h)'(x) \\
3   &= g'(h(x)) \cdot h'(x) \notag \\
4   &= (g' \circ h)(x) \cdot h'(x) \\
5 \end{align}

```

$$f'(x) = (g \circ h)'(x) \quad (10.10)$$

$$\begin{aligned} &= g'(h(x)) \cdot h'(x) \\ &= (g' \circ h)(x) \cdot h'(x) \end{aligned} \quad (10.11)$$

임의의 기호 붙이기

`\tag`와 `\tag*`를 사용하면 수식에 임의의 기호를 붙일 수 있습니다. 기호로 붙이고 싶은 내용을 인자로 지정해 주면 됩니다. 전자는 기호 앞뒤에 소괄호를 함께 조판하지만, 후자는 소괄호를 조판하지 않습니다.

```
1 You might have seen this equation:
2 \[ 2 + 2 \times 2 = 8 \tag{\(*) \]
3 Obviously, this equation is wrong.
```

You might have seen this equation:

$$2 + 2 \times 2 = 8 \quad (*)$$

Obviously, this equation is wrong.

번호 붙는 수식 환경 안에서도 `\tag(*)`를 사용할 수 있습니다. 이 경우 해당 수식의 번호는 계산에 포함되지 않습니다.

```
1 \begin{align}
2 f(x) &= e^{ix} \label{eq:cpxexp} \\
3 f(x) &= \cos x + i \sin x \\
4 \tag{\ref{eq:cpxexp}\(')\} \\
5 g(x) &= e^x \\
6 \end{align}
```

$$f(x) = e^{ix} \quad (10.12)$$

$$f(x) = \cos x + i \sin x \quad (10.12')$$

$$g(x) = e^x \quad (10.13)$$

10.3.2 하위 번호 만들기

관련된 수식이 여러 개 있다면 이들에 번호를 각각 부여하는 대신, 수식들 전체에 번호를 하나만 부여하고 그 하위의 번호를 각 수식에 붙여 줄 수도 있습니다. `amsmath` 패키지의 `subequations` 환경을 사용하면 이러한 방식으로 번호를 붙일 수 있습니다. 하나의 `subequations` 환경이 수식 번호 하나를 부여받으며, 이 환경 안에 있는 수식은 그 번호의 하위 번호를 부여받습니다.

```
1 \begin{subequations}
2 We define hyperbolic cosine and sine functions as follows:
3 \begin{align}
4 \cosh x &= \frac{1}{2} \left( e^x + e^{-x} \right) \\
5 \sinh x &= \frac{1}{2} \left( e^x - e^{-x} \right) \\
6 \end{align}
7 \end{subequations}
8 Hyperbolic tangent function is defined as
9 \begin{equation}
```

```

10 \tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x + e^{-x}}{e^x - e^{-x}},
11 \end{equation}
12 which is similar to trigonometric tangent function.

```

We define hyperbolic cosine and sine functions as follows:

$$\cosh x = \frac{1}{2} (e^x + e^{-x}) \quad (10.14a)$$

$$\sinh x = \frac{1}{2} (e^x - e^{-x}) \quad (10.14b)$$

Hyperbolic tangent function is defined as

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x + e^{-x}}{e^x - e^{-x}}, \quad (10.15)$$

which is similar to trigonometric tangent function.

10.3.3 번호를 조판할 위치

amsmath 패키지는 수식 번호를 조판할 위치와 관련된 옵션으로 leqno, reqno 옵션과 centertags, tbtags 옵션을 제공합니다.

leqno 옵션과 reqno 옵션

leqno 옵션은 수식의 번호를 왼쪽 끝에, reqno 옵션은 수식의 번호를 오른쪽 끝에 조판하도록 설정합니다. 기본값은 reqno 옵션이지만, AMS 클래스에서는 leqno 옵션이 기본값입니다.

```

1 % leqno 옵션 사용
2 \begin{equation}
3 f(x) = x \ln x + x
4 \end{equation}

```

$$(9) \quad f(x) = x \ln x + x$$

```

1 % reqno 옵션 사용
2 \begin{equation}
3 f(x) = x \ln x + x
4 \end{equation}

```

$$f(x) = x \ln x + x \quad (9)$$

centertags 옵션과 tbtags 옵션

이 두 옵션은 equation이나 gather 환경 안에 사용된 split 환경에 번호를 붙일 때, 환경의 가운데나 위·아래 중 어느 쪽에 붙일지 결정합니다. 기본값은 번호를 가운데에

붙이는 centertags이지만, tbtags 옵션을 사용하면 번호가 마지막 줄(reqno 옵션 사용 시)이나 첫째 줄(leqno 옵션 사용 시)에 맞춰 조판됩니다.

1 % centertags 옵션 사용

2 \begin{equation}

3 \begin{split}

4 T(x) &= 1+x+2x^2+3x^3 \\\

5 &= 4x^4+5x^5+6x^6+\dotsb

6 \end{split}

7 \end{equation}

$$\begin{aligned} T(x) &= 1 + x + 2x^2 + 3x^3 \\ &= 4x^4 + 5x^5 + 6x^6 + \dots \end{aligned} \quad (14)$$

1 % tbtags 옵션 사용

2 \begin{equation}

3 \begin{split}

4 T(x) &= 1+x+2x^2+3x^3 \\\

5 &= 4x^4+5x^5+6x^6+\dotsb

6 \end{split}

7 \end{equation}

$$\begin{aligned} T(x) &= 1 + x + 2x^2 + 3x^3 \\ &= 4x^4 + 5x^5 + 6x^6 + \dots \end{aligned} \quad (14)$$

제 11 장

가환 도표 그리기

가환 도표(commutative diagram)는 여러 대상과 그 사이의 여러 사상들의 호환 관계를 나타내는 도표로, 주로 범주론과 대수학 및 그 응용 분야에서 많이 등장합니다. 이번 장에서는 tikz-cd 패키지를 사용하여 가환 도표를 그리는 방법을 알아보겠습니다.

11.1 기본적인 가환 도표 그리기

tikz-cd 패키지를 불러오면 tikzcd 환경을 사용할 수 있게 됩니다. 수식 모드 안에서 이 환경을 사용해 가환 도표를 그려 주면 됩니다.¹ 이번 절에서는 다음 예시를 사용해 가환 도표를 그리는 과정을 알아보겠습니다. 이번 장의 나머지 세 개의 절에서는 tikz-cd 패키지가 제공하는 옵션을 설명합니다.

$$\begin{array}{ccccc}
 & & \pi_1(U_1) & & \\
 & \nearrow i_1 & \downarrow & \searrow j_1 & \\
 \pi_1(U_1 \cap U_2) & & \pi_1(U_1 \cup U_2) & \dashrightarrow k & \pi_1(V) \\
 & \searrow i_2 & \uparrow & \nearrow j_2 & \\
 & & \pi_1(U_2) & &
 \end{array}$$

1단계: 대상 배치하기

가환 도표에 있는 각 대상은 표의 형태로 배치되어 있다고 생각할 수 있습니다. 예시로 주어진 도표에서 화살표를 지우고 가상의 선을 그리면 아래의 그림과 같은 형태로 각

¹ tikzcd 환경은 텍스트 모드와 수식 모드에서 모두 사용할 수 있지만, 가환 도표는 일반적으로 수식으로 취급되므로 수식 작성 환경에서 사용하는 것이 바람직합니다.

대상이 배치되어 있다고 생각할 수 있습니다.

| | | |
|-----------------------|-----------------------|------------|
| | $\pi_1(U_1)$ | |
| $\pi_1(U_1 \cap U_2)$ | $\pi_1(U_1 \cup U_2)$ | $\pi_1(V)$ |
| | $\pi_1(U_2)$ | |

tikzcd 환경 안에 대상을 배치할 때에는 matrix나 array 환경과 같은 문법을 사용합니다. 즉, 같은 행 안에 있는 대상은 앰퍼샌드 &로 구분하고, 행과 행은 \\로 구분합니다. 예시에 있던 각 대상을 tikzcd 환경 안에 배치하면 다음과 같습니다.

```

1 \begin{displaymath}
2   \begin{tikzcd}
3     & \pi_1(U_1) & \\
4     \pi_1(U_1 \cap U_2) & \pi_1(U_1 \cup U_2) & \pi_1(V) \\
5     & \pi_1(U_2) & \\
6   \end{tikzcd}
7 \end{displaymath}

```

$$\pi_1(U_1)$$

$$\pi_1(U_1 \cap U_2)$$

$$\pi_1(U_1 \cup U_2)$$

$$\pi_1(V)$$

$$\pi_1(U_2)$$

2단계: 화살표 그리기

가환 도표에서 각 대상 사이의 관계는 화살표를 사용하여 나타냅니다. 화살표를 그릴 때에는 \arrow 제어 문자열을 사용합니다.

\arrow[*<options>*]{*<direction>*}

<options>: 화살표에 지정할 옵션

<direction>: 화살표를 그릴 방향

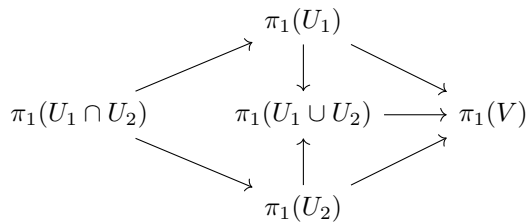
화살표가 출발하는 대상이 있는 칸에 명령어를 입력하며, *<direction>* 인자를 통해 화살표의 목적지를 지정해 줍니다. 이 인자에는 위쪽을 나타내는 u, 왼쪽을 나타내는 l, 오른쪽을 나타내는 r, 아래쪽을 나타내는 d를 조합하여 입력합니다. r를 입력하면 바로 오른쪽 칸을 가리키는 화살표가, ul을 입력하면 왼쪽 위의 칸을 가리키는 화살표가, ddd를 입력하면 3칸 아래쪽을 가리키는 화살표가 그려집니다.

주어진 예시에서, $\pi_1(U_1)$ 이 있는 칸에서 출발하는 화살표는 바로 아래 칸인 $\pi_1(U_1 \cup U_2)$ 로 가는 것과 오른쪽 아래 칸인 $\pi_1(V)$ 로 가는 것, 이렇게 두 개가 있습니다. 따라서 $\pi_1(U_1)$ 이 있는 칸에 화살표를 그리는 코드를 추가하면

```
\pi_1(U_1) \arrow{d} \arrow{dr}
```

이 됩니다. 이런 식으로, 예시에 있는 화살표를 모두 그리면 아래와 같습니다.

```
1 \begin{displaymath}
2   \begin{tikzcd}
3     & \pi_1(U_1) \arrow{d} \arrow{dr} \\
4     \pi_1(U_1 \cap U_2) \arrow{ur} \arrow{dr} & \pi_1(U_1 \cup U_2) \arrow{r} & \pi_1(V) \\
5     & \pi_1(U_2) \arrow{u} \arrow{ur} \\
6   \end{tikzcd}
7 \end{displaymath}
```



`\arrow`의 선택 인자인 *<options>*에는 화살표에 지정할 옵션을 입력합니다. 이를 통해 다양한 모양의 화살표를 그리고, 위치나 방향을 조정할 수 있습니다. 여러 옵션을 동시에 지정하려면 각 옵션을 쉼표로 구분해 주면 됩니다.

주어진 예시에서는 점선으로 그려진 화살표가 세 개 있으며, 이는 `dashed` 옵션을 사용하여 구현할 수 있습니다. 따라서, $\pi_1(U_1 \cup U_2)$ 가 있는 칸의 코드는 다음과 같습니다.

```
\pi_1(U_1 \cup U_2) \arrow[dashed]{r}
```

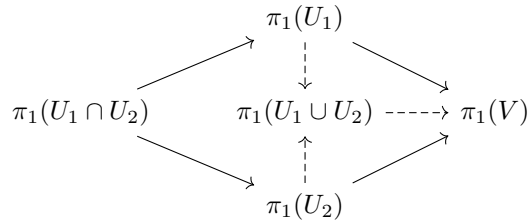
이러한 식으로 세 개의 화살표에 `dashed` 옵션을 지정하면 다음 결과를 얻습니다.

```
1 \begin{displaymath}
2   \begin{tikzcd}
3     & \pi_1(U_1) \arrow{d} \arrow{dr} \\
4     \pi_1(U_1 \cap U_2) \arrow{ur} \arrow{dr} & \pi_1(U_1 \cup U_2) \arrow{r} & \pi_1(V) \\
5     & \pi_1(U_2) \arrow{u} \arrow{ur} \\
6   \end{tikzcd}
7 \end{displaymath}
```

```

6 \end{tikzcd}
7 \end{displaymath}

```



화살표에 지정할 수 있는 옵션은 11.2.2절에서 알아봅니다.

3단계: 화살표에 레이블 붙이기

화살표에는 대상 사이에 정의되는 함수나 사상 등을 레이블로 붙이기도 합니다. 레이블은 다음의 문법으로 입력하며, 이를 레이블을 붙일 화살표 명령어 뒤에 입력하면 됩니다.

레이블 \longrightarrow [*options*]{*label*}

options: 레이블에 지정할 옵션

label: 레이블로 표시할 내용

하나의 화살표에 여러 개의 레이블이 달려 있다면 이 명령어를 계속 이어붙이면 됩니다.

예시에서 $\pi_1(U_2)$ 에서 $\pi_1(V)$ 로 가는 화살표에는 ‘ j_2 ’가 레이블로 붙어 있습니다. 이를 화살표에 붙이기 위해서는 $\pi_1(U_2)$ 가 있는 칸의 코드를 아래와 같이 작성합니다.

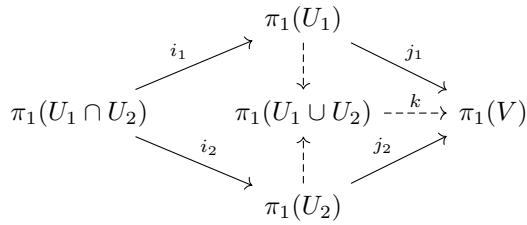
```
\pi_1(U_2) \arrow[dashed]{u} \arrow{ur}{j_2}
```

이렇게 5개의 레이블을 모두 붙이면 아래의 결과를 얻습니다.

```

1 \begin{displaymath}
2 \begin{tikzcd}
3 & \pi_1(U_1) \arrow[dashed]{d} \arrow{dr}{j_1} \\
4 \pi_1(U_1 \cap U_2) \arrow{ur}{i_1} \arrow{dr}{i_2} & \pi_1(U_1 \cup U_2) \arrow[dashed]{r}{k} & \pi_1(V) \\
5 & \pi_1(U_2) \arrow[dashed]{u} \arrow{ur}{j_2}
6 \end{tikzcd}
7 \end{displaymath}

```



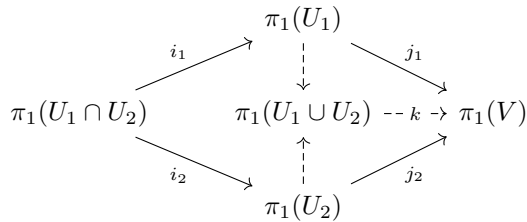
레이블의 선택 인자인 $\langle options \rangle$ 는 레이블이 조판될 위치를 변경해 줍니다. 대표적으로 swap 옵션은 화살표를 기준으로 레이블이 현재 위치의 반대편에 표시되도록 합니다. 또, description 옵션은 레이블이 화살표 위에 표시되도록 합니다.

앞의 결과에서 i_2 과 j_2 에 swap 옵션을, k 에 description 옵션을 지정하면 다음과 같이 목표하던 결과를 얻게 됩니다.

```

1 \begin{displaymath}
2   \begin{tikzcd}
3     & \pi_1(U_1) \arrow[dashed]{d} \arrow{dr}{j_1} \\
4     \pi_1(U_1 \cap U_2) \arrow{ur}{i_1} \arrow{dr}[swap]{i_2} & \pi_1(U_1 \cup U_2) \arrow[dashed]{r}[description]{k} & \pi_1(V) \\
5     & \pi_1(U_2) \arrow[dashed]{u} \arrow{ur}[swap]{j_2}
6   \end{tikzcd}
7 \end{displaymath}

```



레이블에 지정할 수 있는 옵션은 11.2.3절에서 알아봅니다.

11.1.1 화살표의 대체 문법

tikz-cd 패키지는 화살표를 그리는 다른 문법을 제공합니다. 이는 화살표의 옵션과 방향, 레이블의 옵션과 레이블의 내용을 대괄호 안에 모두 작성하는 방식입니다. 이 방식을 사용할 때에도 앞서 설명한 $\backslash arrow$ 제어 문자열을 사용합니다.

`\arrow[⟨direction, options, and labels⟩]`

⟨direction, options, and labels⟩: 화살표의 방향, 화살표의 옵션, 레이블의 옵션, 라벨

이 방식을 사용할 때에는 화살표의 방향이나 레이블도 하나의 옵션처럼 입력합니다. 레이블을 입력할 때에는 "*⟨label⟩*" *⟨label options⟩*'의 형식으로 입력합니다. 만약 레이블의 내용에 쉼표가 포함되어 있으면 *⟨label⟩*을 중괄호로 감싸야 하며, 레이블에 여러 옵션을 지정하기 위해 쉼표를 사용할 때에도 *⟨label options⟩*를 중괄호로 감싸야 합니다.

예를 들어, 앞서 예시로 들었던 도표의 화살표 중 'k'를 포함한 것은 다음의 두 가지 방법으로 입력할 수 있습니다.

```
1 \arrow[dashed]{r}[description]{k}
2 \arrow[r, dashed, "k" description]
```

축약된 명령어

화살표를 그리는 `\arrow` 제어 문자열은 `\ar`를 입력하여 같은 결과를 얻을 수 있습니다. 또, 화살표가 출발하는 칸을 기준으로 인접한 8칸을 가리키는 화살표는 방향을 명령어의 이름에 포함한 다음 명령어를 사용하여 그릴 수도 있습니다.

`\rar` `\lar` `\dar` `\uar` `\drar` `\urar` `\dlar` `\ular`

따라서, 위에서 예시로 든 화살표는 다음의 네 가지 방법으로도 그릴 수 있습니다.

```
1 \ar[dashed]{r}[description]{k}
2 \ar[r, dashed, "k" description]
3 \rar[dashed][description]{k}
4 \rar[dashed, "k" description]
```

이처럼, 축약된 명령어를 사용하면 예시 가환 도표는 다음과 같이 더 간단한 코드를 사용하여 그릴 수도 있습니다.

```
1 \begin{displaymath}
2   \begin{tikzcd}
3     & \pi_1(U_1) \dar[dashed] \drar["j_1"] \ll
4     \pi_1(U_1 \cap U_2) \urar["i_1"] \drar["i_2" swap] &
       \pi_1(U_1 \cup U_2) \rar[dashed, "k" description] & \pi_1(V) \\
5     & \pi_1(U_2) \uar[dashed] \urar["j_2" swap]
6   \end{tikzcd}
7 \end{displaymath}
```

11.2 각종 옵션

이번 절에서는 tikzcd 환경을 사용해 가환 도표를 그릴 때 사용할 수 있는 옵션을 설명합니다. 이 패키지는 tikz 패키지의 기능을 사용하여 가환 도표를 그리므로, tikz 패키지의 기능도 함께 사용할 수 있습니다. 이 책에서는 tikz-cd 패키지가 제공하는 옵션만 설명하겠습니다.

옵션을 지정하는 방법

가환 도표와 관련된 옵션은 다음의 세 가지 위치에 지정할 수 있습니다. 주어진 코드의 `<options>` 부분에 원하는 옵션들을 입력하면 됩니다. 많은 옵션은 ‘키-값’ 형식으로 지정합니다.

- 전처리부 `\tikzcdset` 제어 문자열을 사용하면 문서에 그려지는 모든 가환 도표에 옵션을 지정할 수 있습니다.

```
\tikzcdset{<options>}
```

이 제어 문자열을 사용하는 방법은 패키지의 안내서[47]를 참조하시기 바랍니다.

- tikzcd 환경 tikzcd 환경을 시작할 때 선택 인자로 옵션을 지정하면 해당 환경으로 그려지는 가환 도표의 모든 요소(화살표와 레이블 등)에 입력한 옵션이 적용됩니다.

```
\begin{tikzcd}[<options>] ... \end{tikzcd}
```

- 옵션을 지정할 요소 앞 절에서 설명했듯, `\arrow`의 선택 인자로 지정된 옵션은 그 화살표에만 영향을 미치며, 레이블의 선택 인자로 지정된 옵션은 그 레이블에만 영향을 미칩니다.

단, 레이블에 효과를 미치는 옵션을 화살표의 옵션으로 지정하는 경우에는 그 화살표에 달린 모든 레이블에 해당 옵션이 적용됩니다.

11.2.1 tikzcd 환경의 옵션

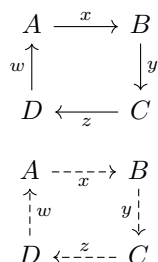
앞서 설명했듯, tikzcd 환경에 입력한 옵션은 환경 안의 모든 화살표나 레이블에 영향을 미칩니다. 예를 들어, 앞 절에서 알아본 `dashed` 옵션과 `swap` 옵션을 환경의

옵션으로 지정하면 다음의 결과를 얻습니다. 위쪽의 도표와 달리 아래쪽의 도표에서는 모든 화살표가 점선으로 그려지고, 모든 레이블이 반대쪽에 배치되었습니다.

```

1 \begin{displaymath}
2   \begin{tikzcd}
3     A \arrow{r}{x} & B \arrow{d}{y} \\
4     D \arrow{u}{w} & C \arrow{l}{z}
5   \end{tikzcd}
6 \end{displaymath}
7 \begin{displaymath}
8   \begin{tikzcd}[dashed, swap]
9     A \arrow{r}{x} & B \arrow{d}{y} \\
10    D \arrow{u}{w} & C \arrow{l}{z}
11  \end{tikzcd}
12 \end{displaymath}

```



다만, 화살표나 레이블에 지정할 수 없고 tikzcd 환경에 지정해야 하는 옵션도 있습니다. 이들은 가환 도표를 그릴 때의 크기를 조정합니다.

대상과 대상 사이의 간격

대상과 대상 사이의 간격은 아래의 세 가지 옵션을 사용해 지정합니다. 옵션 뒤에 등호를 입력하고 원하는 간격 값이나 사전 정의된 6가지 키워드 tiny, small, scriptsize, normal, large, huge 중 하나를 입력하면 됩니다. 기본값은 normal입니다.

- row sep 옵션 행과 행 사이의 간격을 입력된 값으로 설정합니다.
- column sep 옵션 열과 열 사이의 간격을 입력된 값으로 설정합니다.
- sep 옵션 행과 행, 열과 열 사이의 간격을 모두 입력된 값으로 설정합니다.

입력

```
\begin{tikzcd} ... \end{tikzcd}
```

```
\begin{tikzcd}[sep = small] ... \end{tikzcd}
```

```
\begin{tikzcd}[sep = large] ... \end{tikzcd}
```

```
\begin{tikzcd}[sep = 2em] ... \end{tikzcd}
```

출력

 $A \longrightarrow B$
 $A \rightarrow B$
 $A \longrightarrow B$
 $A \longrightarrow B$

압축된 스타일의 가환 도표

cramped 옵션은 각 대상 사이의 넓은 여백을 없애고 압축된 형태의 가환 도표를 그릴 수 있도록 합니다. 공간을 효율적으로 사용해야 할 때 유용합니다.

표 11.1 | tikz-cd 패키지: 화살표의 모양

| 옵션 | 모양 | 옵션 | 모양 | 옵션 | 모양 |
|-------------------------------|---|--------------------------------|---|----------------------------------|---|
| <code>leftarrow</code> |  | <code>rightarrow</code> |  | <code>leftrightharrow</code> |  |
| <code>Leftarrow</code> |  | <code>Rightarrow</code> |  | <code>Leftrightarrow</code> |  |
| <code>mapsfrom</code> |  | <code>mapsto</code> |  | <code>dash</code> |  |
| <code>Mapsfrom</code> |  | <code>Mapsto</code> |  | <code>equal</code> |  |
| <code>leftsquigarrow</code> |  | <code>rightsquigarrow</code> |  | <code>leftrightsquigarrow</code> |  |
| <code>hookleftarrow</code> |  | <code>hookrightarrow</code> |  | | |
| <code>leftarrowtail</code> |  | <code>rightarrowtail</code> |  | | |
| <code>twoheadleftarrow</code> |  | <code>twoheadrightarrow</code> |  | | |
| <code>dashleftarrow</code> |  | <code>dashrightarrow</code> |  | | |
| <code>leftharpoonup</code> |  | <code>rightharpoonup</code> |  | | |
| <code>leftharpoondown</code> |  | <code>rightharpoondown</code> |  | | |

입력

```
\begin{tikzcd} ... \end{tikzcd}
```

```
\begin{tikzcd}[cramped] ... \end{tikzcd}
```

출력

$$A \longrightarrow B$$

$$A \longrightarrow B$$

11.2.2 화살표의 옵션

이번 절에서는 화살표와 관련된 옵션을 알아봅니다. 여기의 옵션은 화살표의 모양을 변경하는 것, 화살표의 방향을 변경하는 것, 화살표의 위치를 변경하는 것으로 구분할 수 있습니다.

화살표의 모양

화살표에 지정할 수 있는 옵션들 중에는 화살표의 모양을 변경하는 것이 자주 사용됩니다. 표 11.1은 화살표의 모양을 바꿀 수 있는 옵션을 정리한 것입니다. 이 표에 있는 옵션을 입력하면 화살표의 머리, 꼬리, 선의 모양이 모두 변경됩니다.

한편, 머리, 꼬리, 선의 모양 중 일부만을 변경하는 명령어도 있으며, 이를 표 11.2에 정리하였습니다. 이 표의 명령어를 조합하면 표 11.1에 없는 더 다양한 화살표를 만들 수도 있습니다. 예를 들어, `\rar[tail, dashed, two heads]`를 입력해 그릴 수 있습니다.

| 표 11.2 | tikz-cd 패키지: 화살표의 부분별 모양

| 꼬리 모양 | | 선의 모양 | | 머리 모양 | |
|---------|---|----------|---|-----------|---|
| 명령어 | 모양 | 옵션 | 모양 | 옵션 | 모양 |
| maps to |  | dashed |  | to head |  |
| hook |  | squiggly |  | two heads |  |
| hook' |  | | | harpoon |  |
| tail |  | | | harpoon' |  |
| no tail |  | | | no head |  |

tikz-cd 패키지가 제공하는 화살표의 머리·꼬리·선의 모양 옵션 외에, `bend left`나 `bend right`, `controls` 등 TikZ에서 곡선을 그리는 옵션을 사용하면 굽은 화살표나 자기 자신을 가리키는 화살표를 만들 수 있습니다. 자세한 내용은 tikz 패키지 매뉴얼 [48, pp. 838–842]의 관련 부분을 참조하시기 바랍니다.

화살표의 방향

화살표의 방향을 지정하는 기본적인 방법은 11.1절에서 설명했듯 `l`, `r`, `u`, `d`를 조합해 출발지를 기준으로 이동할 방향을 정해 주는 것입니다. 하지만 화살표의 방향을 절대 위치를 사용하여 정해줄 수도 있고, `\arrow`가 입력된 곳을 목적지로 하는 `—` 즉, 일반적인 화살표와 방향이 반대인 `—` 화살표를 그릴 수도 있습니다.

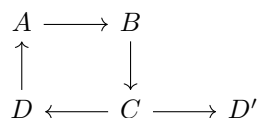
화살표의 출발지를 지정할 때에는 `from` 옵션을, 출발지를 지정할 때에는 `to` 옵션을 지정합니다. 옵션 뒤에 등호를 입력하고 원하는 위치를 입력해 주면 됩니다. 이때 위치를 지정하는 방법은 세 가지가 있습니다.

- 방향을 사용한 상대 위치 앞서 설명했듯 `l`, `r`, `u`, `d`를 조합해 현재 위치를 기준으로 한 상대 위치를 지정합니다. 예를 들어 `uu1`은 현재 칸을 기준으로 2칸 위, 1칸 왼쪽에 위치한 곳을 가리킵니다.
- 좌표를 사용한 절대 위치 ‘`(row number)-(column number)`’의 형식으로, 번호를 사용해 대상의 절대 위치를 지정합니다. 예를 들어 3-2는 제3행 제2열을 가리킵니다.
- 이름을 사용한 위치 출발지나 목적지로 삼을 위치에 붙여 둔 가상의 이름을 사용합니다. 가상의 이름은 TikZ의 문법을 따라 붙일 수 있습니다.


```

1 \begin{tikzcd}
2   A \arrow[from=d] \arrow{1-2} & B \\
3   D & C \arrow[from=1-2] \arrow[to=D'] \\
4   & & & D'
\end{tikzcd}

```



예시에 from과 to 옵션을 사용해 화살표를 그리는 방법을 제시하였습니다. D' 이 있는 칸에서는 대상에 이름을 붙이고, 이 이름을 사용해 화살표를 그리는 방법도 확인할 수 있습니다. 한편, A 가 있는 칸의 $\arrow{1-2}$ 에서 확인할 수 있듯, $\arrow[#1]{#2}$ 는 $\arrow[to=#2, #1]$ 과 동일합니다.

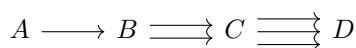
화살표의 위치

두 대상 사이에 여러 개의 화살표를 그리는 경우, 별도의 조정을 해 주지 않으면 화살표들이 겹쳐진 채로 그려집니다. 이때에는 shift left 옵션과 shift right 옵션을 사용합니다.

```

1 \begin{tikzcd}
2   A \rar
3   & B \rar[shift left] \rar[shift
4   & C \rar[shift left=2] \rar
5   & D
6 \end{tikzcd}

```



예시에서 볼 수 있듯, 옵션 뒤에 등호를 입력하고 화살표를 이동시킬 거리를 기본 거리의 배수로써 지정해 줄 수도 있습니다. 또, 이동시킬 거리를 단위를 사용해 직접 지정해 줄 수도 있습니다.

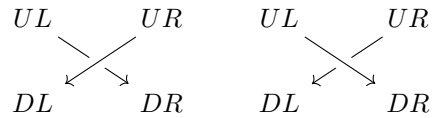
3차원 가환 도표

3차원 가환 도표를 그리려는 경우, 한 화살표가 다른 화살표의 위에 있는 듯한 효과를 만들 수 있습니다. 이때에는 crossing over 옵션을 사용합니다. 서로 교차되는 화살표 중 위쪽에 있는 화살표에 옵션을 지정하면 되며, 아래쪽에 있는 화살표는 끊어진 형태로 그려집니다. 이때, crossing over 옵션이 지정된 화살표를 더 나중에 그려야 합니다.

```

1 \begin{tikzcd}
2   UL \drar & UR \dlar[crossing over] \\
3   DL & DR
4 \end{tikzcd}
5 \quad
6 \begin{tikzcd}
7   UL & UR \drar \\
8   DL & DR \arrow[from=ul, crossing
9     over]
\end{tikzcd}

```



아래쪽에 있는 화살표가 끊어지는 정도는 `crossing over clearance` 옵션으로 변경할 수 있습니다. 기본값은 `1.5ex`이며, '`crossing over clearance = 2ex`'와 같이 원하는 값을 옵션 뒤에 등호를 입력한 후 지정하면 됩니다.

참고 | `crossing over` 옵션에 의해 끊어진 화살표의 경우, 레이블이 화살표의 가운데 지점에 표시되지 않습니다. 11.2.3절에서 언급하는 `near start` 옵션 등을 사용해 레이블을 조판할 위치를 옮겨 주어야 레이블이 정상적으로 조판됩니다.

11.2.3 레이블의 옵션

레이블에 지정하는 옵션 중 대부분은 레이블의 위치를 변경하는 역할을 합니다.

레이블을 반대쪽·화살표 위에 배치하기

`tikz-cd` 패키지에서 제공하는 옵션 중 레이블에 지정하는 옵션은 레이블의 위치를 변경합니다. 가장 먼저, 앞서 설명했듯 `swap` 옵션은 화살표 반대쪽에 레이블이 배치되도록 합니다.

입력

```

... \arrow[r]{f} ...
... \arrow[r][swap]{f} ...
... \arrow[r, "f"] ...

```

출력

```

A \xrightarrow{f} B
A \xrightarrow[swap]{f} B
A \xrightarrow["f"]{} B

```

세 번째 예시와 같이, `swap` 대신 따옴표 '를 옵션으로 지정하여 간단하게 레이블을 반대쪽에 배치할 수도 있습니다.

`marking` 옵션과 `description` 옵션을 사용하면 레이블을 화살표 위에 배치할 수 있습니다. 후자를 사용하면 레이블이 있는 부분에 화살표의 선이 그려지지 않습니다.

입력

```
... \arrow{r}[marking]{\circ} ...
... \arrow{r}[description]{(1)} ...
```

출력

$A \xrightarrow{\circ} B$
 $A \xrightarrow{(1)} B$

레이블을 표시할 지점 변경하기

다음 7개의 옵션이나 pos 옵션을 사용하면 레이블을 화살표의 중간 지점이 아니라 꼬리 근처나 머리 근처에 배치할 수도 있습니다.

at start very near start near start midway
near end very near end at end

pos 옵션은 레이블을 배치하고 싶은 위치를 0(화살표의 꼬리 쪽)과 1(화살표의 머리 쪽) 사이의 실수로 지정할 수 있습니다. 예를 들어, near start 옵션은 pos=0.25를 입력한 것과 같습니다.

```
1 \begin{tikzcd}[sep = large]
2   X \rar[] [near start]{f}[swap]{g}[near
   end]{h} & Y
3 \end{tikzcd}
```

$$X \xrightarrow[f]{h} Y$$

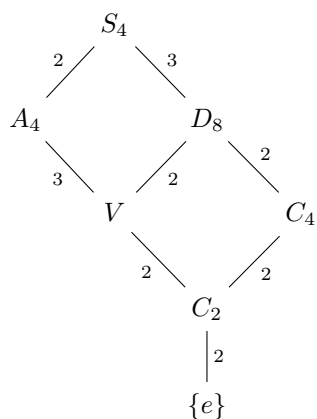
이외에도, draw나 circle 등 TikZ의 노드에 사용할 수 있는 옵션을 레이블에 사용할 수도 있습니다. tikz 패키지의 매뉴얼[48, pp. 223-261]을 참조하십시오.

연습문제

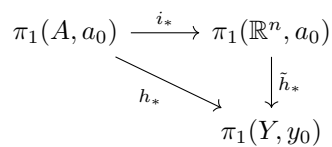
11.1. 다음 가환 도표를 작성하여라.

$$\begin{array}{ccc} L & \longrightarrow & M & \longrightarrow & N \\ \downarrow & & & & \downarrow \\ I_L^0 & & & & I_N^0 \\ \downarrow & & & & \downarrow \\ I_L^1 & & & & I_N^1 \\ \downarrow & & & & \downarrow \\ \vdots & & & & \vdots \end{array}$$

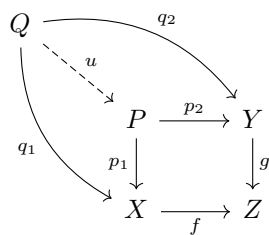
11.2. 다음 가환 도표를 작성하여라.



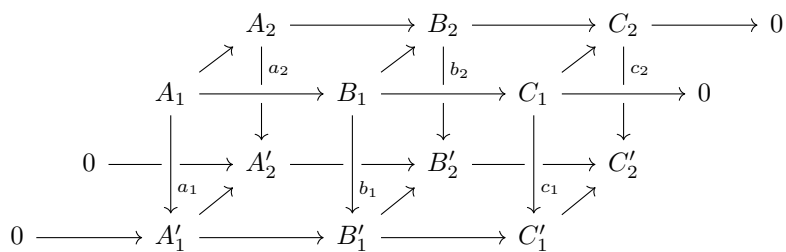
11.3. 다음 가환 도표를 작성하여라.



11.4. 다음 가환 도표를 작성하여라.



11.5. 다음 가환 도표를 작성하여라.



IV

표와 그림 다루기

제 IV 편에서는 \LaTeX 문서에 표와 그림을 넣는 방법을 알아봅니다. 표를 작성할 때에는 `tabular` 환경과 `array` 환경을 사용하며, 그림을 삽입할 때에는 `graphicx` 패키지의 `\includegraphics` 제어 문자열을 사용합니다. 마지막 장에서는 표와 그림을 배치하는 방법을 알아봅니다.

제12장 표 작성하기 · 167

제13장 그림 넣기 · 185

제14장 표와 그림의 배치 · 193

제 12 장

표 작성하기

이번 장에서는 tabular 환경과 array 환경을 사용해 표를 작성하는 방법을 알아봅니다. 또, 다양한 인자를 조정하여 표의 모습을 변경하고, 전체 폭이 고정된 표를 그리는 방법도 알아봅니다.

두 환경은 환경 안에 입력된 내용이 조판되는 모드만 제외하면 같은 동작을 하고, 두 환경을 사용하는 문법도 같습니다. 따라서 두 환경을 사용하는 방법을 이번 장에서 함께 설명하겠습니다.

12.1 기본적인 표 그리기

다음은 tabular 환경을 사용한 간단한 예시입니다. center 환경 안에 표를 작성하면 표가 글줄 가운데에 배치됩니다.

```

1 \begin{center}
2   \begin{tabular}{|l|c|r|}
3     \hline
4     왼쪽 & 가운데 & 오른쪽 \\ \hline
5     정렬된 & 정렬된 & 정렬된 \\ \hline
6     1열 & 2열 & 3열 \\ \hline
7   \end{tabular}
8 \end{center}

```

| | | |
|-----|-----|-----|
| 왼쪽 | 가운데 | 오른쪽 |
| 정렬된 | 정렬된 | 정렬된 |
| 1열 | 2열 | 3열 |

다음은 array 환경을 사용한 가장 간단한 예시입니다. 예시에서 볼 수 있듯 array 환경은 수식 모드에서 사용해야 하며, 각 칸의 내용도 수식 모드에서 조판됩니다.

```

1 \begin{displaymath}
2   \begin{array}{|l|c|r|}
3     \hline
4     left & center & left \\ \hline
5     aligned & aligned & aligned \\ \hline
6     column & column & column \\ \hline
7   \end{array}
8 \end{displaymath}

```

| | | |
|----------------|----------------|----------------|
| <i>left</i> | <i>center</i> | <i>left</i> |
| <i>aligned</i> | <i>aligned</i> | <i>aligned</i> |
| <i>column</i> | <i>column</i> | <i>column</i> |

예시에서 볼 수 있듯, 두 환경은 인자를 하나 입력받습니다. l, c, r를 입력한 개수만큼 열이 생성됩니다. 이때 l로 생성된 열은 왼쪽에, c로 생성된 열은 가운데에, r로 생성된 열은 오른쪽에 맞추어 내용이 정렬됩니다. 각 열을 생성하는 문자의 좌우에 세로선 |을 입력하면 표에 세로 구분선이 그려집니다.

환경 안에는 표에 작성할 내용을 입력합니다. 한 행의 내용을 모두 작성한 뒤 다음 행으로 넘어가는 순서로 입력하며, 행과 행의 구분은 \\로, 한 행 안에서의 열 구분은 &로 합니다. 행과 행 사이의 구분선은 \\ 뒤에 \hline을 입력해 그릴 수 있습니다.

위의 예시를 작성하기 위한 코드를 시각적으로 나타내면 그림 12.1과 같습니다.

```

\begin{tabular}
{ | l | c | r | }
\hline
왼쪽 & 가운데 & 오른쪽 \\ \hline
정렬된 & 정렬된 & 정렬된 \\ \hline
1열 & 2열 & 3열 \\ \hline
\end{tabular}

```

■ 그림 12.1 | 표를 작성하는 코드의 시각화

l, c, r로 생성된 열은 입력된 내용을 줄바꿈 없이 한 줄로 조판합니다. 즉, 조판할 내용이 길면 페이지의 여백을 침범한 채로 내용이 조판됩니다. 만약 칸에 긴 내용을 입력하고 줄바꿈이 일어나도록 하려면 열을 생성하는 인자로 p{<width>}를 사용하면 됩니다. <width>에는 해당 열의 폭을 지정하면 되며, 이 폭에 맞추어 글줄의 길이가 결정되고 줄바꿈 처리가 진행됩니다. 이때, array 환경을 사용하더라도 p 인자로 생성한 열의 내용은 텍스트 모드에서 조판되므로 주의하시기 바랍니다.

길이
→ 280쪽


```

1 \begin{center}
2   \begin{tabular}{c|p{6cm}}
3     \hline
4     1행 & \texttt{p} 인자로 생성한 열에서는 긴 내용이 여러 줄로 조판됩니다. \\ \hline
5     2행 & 표 안에는 지나치게 긴 내용을 입력하지 않는 것이 좋습니다. \\ \hline
6   \end{tabular}
7 \end{center}

```

| | |
|----|-----------------------------------|
| 1행 | p 인자로 생성한 열에서는 긴 내용이 여러 줄로 조판됩니다. |
| 2행 | 표 안에는 지나치게 긴 내용을 입력하지 않는 것이 좋습니다. |

12.2 환경의 세부 문법

앞 절의 내용만으로도 기본적인 표를 작성하는 데에는 충분할 것입니다. 하지만 더 복잡한 표를 작성하려면 환경을 사용하는 자세한 방법을 알아야 합니다. 이번 절에서는 `tabular` 및 `array` 환경의 세부 문법을 설명하겠습니다. 한편, `array` 패키지[31]를 불러오면 \LaTeX 의 기본 표 작성 기능을 확장할 수 있습니다. 이 패키지의 기능도 이번 절에서 함께 알아보겠습니다.

`tabular` 환경과 `array` 환경은 다음과 같이 두 가지 인자를 입력받습니다. 필수 인자인 $\langle spec. \rangle$ 에 대해서는 12.2.1절에서, 선택 인자인 $\langle pos. \rangle$ 에 대해서는 12.2.2절에서 알아봅니다.

`\begin{tabular}[\langle pos. \rangle]{\langle spec. \rangle}`

$\langle pos. \rangle$: 앞뒤 텍스트와의 세로 정렬

$\langle spec. \rangle$: 열의 개수와 각 열의 속성

`\end{tabular}`

`\begin{array}[\langle pos. \rangle]{\langle spec. \rangle}`

$\langle pos. \rangle$: 앞뒤 수식과의 세로 정렬

$\langle spec. \rangle$: 열의 개수와 각 열의 속성

`\end{array}`

12.2.1 열 속성 인자

필수 인자인 $\langle spec. \rangle$ 에는 표 12.1의 인자를 조합하여 입력합니다. 여기에는 열을 생성하는 인자, 열과 열 사이를 구분할 방식을 지정하는 인자, 열의 속성을 지정하는 인자가 있습니다. 많은 열을 한꺼번에 생성할 수 있는 반복 인자도 있습니다.

| 표 12.1 | table 및 array 환경의 열 속성 인자

| 구분 | 인자 | 설명 |
|-------------|--------------------------|--------------------------------------|
| 열
생성 인자 | l | 왼쪽 정렬된 한 줄짜리 열을 생성합니다. |
| | c | 가운데 정렬된 한 줄짜리 열을 생성합니다. |
| | r | 오른쪽 정렬된 한 줄짜리 열을 생성합니다. |
| | p{<width>} | 폭이 <width>인 위쪽 정렬된 문단형 열을 생성합니다. |
| | m{<width>} ¹ | 폭이 <width>인 가운데 정렬된 문단형 열을 생성합니다. |
| | b{<width>} ¹ | 폭이 <width>인 아래쪽 정렬된 문단형 열을 생성합니다. |
| 열
구분 인자 | | 좌우 열 사이에 구분선을 삽입합니다. |
| | @{<delim.>} | 좌우 열 사이에 여백 없이 <delim.>의 내용을 삽입합니다. |
| | !{<delim.>} ¹ | 좌우 열 사이에 여백을 두고 <delim.>의 내용을 삽입합니다. |
| 서식
지정 인자 | >{<format>} ¹ | 이 인자 오른쪽 열의 첫머리에 <format>을 삽입합니다. |
| | <{<format>} ¹ | 이 인자 왼쪽 열의 끝부분에 <format>을 삽입합니다. |
| 반복 인자 | *{<num.>}{<spec.>} | <spec.>을 <num.>번 반복합니다. |

¹ array 패키지를 불러와야 사용할 수 있습니다.

열 생성 인자

열을 생성할 때에는 l, c, r, p{<width>}, m{<width>}, b{<width>}를 사용할 수 있습니다 (m과 b는 array 패키지를 불러와야 사용할 수 있습니다). l, c, r 인자는 앞 절에서 설명하였으므로 여기에서는 설명을 생략하겠습니다. p, m, b는 표의 내용을 지정한 폭 <width>에 맞추어 내용을 조판하는 열을 생성합니다. 입력된 내용이 <width>보다 길다면 줄바꿈 처리가 일어나고, 글줄의 양 끝이 가지런해지도록 양쪽 정렬 처리됩니다. 각 인자는 같은 행 다른 열의 내용이 이 열의 위쪽, 가운데, 아래쪽에 맞춰 조판되도록 합니다. 앞 페이지의 예시와 다음 두 예시를 비교해 보십시오.

```

1 \begin{center}
2   \begin{tabular}{c|m{4.5cm}}
3     \hline
4     1행 & \texttt{m} 인자로 생성한 열 ... 행의 가운데에 맞춰 조판됩니다. \\ \hline
5   \end{tabular}\quad
6   \begin{tabular}{c|b{4.5cm}}
7     \hline

```

```

8   1행 & \texttt{b} 인자로 생성한 열 ... 행의 아래쪽에 맞춰 조판됩니다. \\ \hline
9   \end{tabular}
10  \end{center}

```

| | | | |
|----|--|----|--|
| 1행 | m 인자로 생성한 열 옆 칸의 내용은 행의 가운데에 맞춰 조판됩니다. | 1행 | b 인자로 생성한 열 옆 칸의 내용은 행의 아래쪽에 맞춰 조판됩니다. |
|----|--|----|--|

열 구분 인자

표를 이루는 각 열의 좌우에는 약간의 여백이 삽입되어 인접한 열과 시각적으로 구분됩니다. $\langle spec. \rangle$ 인자의 열 지정자 사이에 세로줄 ‘|’을 입력하면 좌우 두 열 사이에 세로 구분선을 그릴 수 있습니다. 이 문자를 여러 번 입력해 다중 세로선을 그릴 수도 있습니다.

$\langle \{ \langle delim. \rangle \} \rangle$ 를 인자로 사용하면 열과 열 사이의 여백 없이, 열 사이에 $\langle delim. \rangle$ 에 입력된 내용을 넣습니다. 예를 들어, $\langle \{ \} \rangle$ 를 입력하면 열과 열 사이의 여백을 없앨 수 있고, $\langle \{ \rangle \} \rangle$ 를 입력하면 열과 열 사이에 기본 여백 대신 5mu의 작은 공백이 삽입되도록 할 수 있습니다.

다음 예시는 열 사이의 간격을 조정하여 연립방정식의 각 항을 열 맞춰 조판한 것입니다. $\langle \{ \} \rangle$ 인자로 열 사이의 간격을 조정하지 않으면 열과 열 사이의 간격이 넓어져 보기 좋지 않은 결과를 얻습니다.

```

1 \begin{displaymath}
2   \left\{ \begin{array}{@{}r@{\;}c@{\;}r@{\>}c@{\>}l@{}}
3     2x_1 & + & 3x_2 & = & 4 \\
4     -3x_1 & - & x_2 & = & -1
5   \end{array} \right.
6 \end{displaymath}

```

$$\begin{cases} 2x_1 + 3x_2 = 4 \\ -3x_1 - x_2 = -1 \end{cases}$$

이 기능을 응용하면 열의 내용을 소숫점을 기준으로 정렬할 수도 있습니다. 소숫점을 기준으로 해당 열을 두 개의 열로 나누어 조판하고, 이 두 열 사이에 소숫점이 입력되도록 하는 것입니다. 표의 첫 행은 뒤에서 설명하는 $\langle multicolumn \rangle$ 제어 문자열을 사용해 두 열을 합쳐 작성하면 됩니다.

```

1 \begin{center}
2   \begin{tabular}{c|r@{.}l}
3     정보 & \multicolumn{2}{c}{측정값} \\
4     \hline
5     신장 & 180&3\,cm \\
6     체중 & 76&23\,kg \\
7   \end{tabular}
8 \end{center}

```

| 정보 | 측정값 |
|----|----------|
| 신장 | 180.3 cm |
| 체중 | 76.23 kg |

array 패키지를 불러오면 @ 대신 !를 사용할 수도 있습니다. 이 인자를 사용하면 @ 인자와 같은 동작을 하되, 열과 열 사이의 기본 여백이 삭제되지 않습니다. 예를 들어, 열과 열을 세로선 대신 쌍점을 사용해 구분하려 할 때 이 인자를 사용할 수 있습니다.

```

1 \begin{center}
2   \begin{tabular}{c|c!{:}c}
3     \hline
4     분야 & A국 & B국 \\
5     \hline
6     득점 & 3 & 1 \\
7     유효슈팅 & 10 & 8 \\
8     파울 & 2 & 4 \\
9   \end{tabular}
10 \end{center}

```

| 분야 | A국 | B국 |
|------|----|----|
| 득점 | 3 | 1 |
| 유효슈팅 | 10 | 8 |
| 파울 | 2 | 4 |

서식 지정 인자

한 열의 모든 칸의 첫 부분이나 끝 부분에 같은 코드를 넣을 일이 있다면 array 패키지의 $\langle format \rangle$ 과 $\langle format \rangle$ 인자를 사용할 수 있습니다. $\langle format \rangle$ 를 입력하면 이 인자 오른쪽에 있는 열의 첫머리에 $\langle format \rangle$ 을 삽입한 채로 내용이 조판되고, 반대로 $\langle format \rangle$ 를 입력하면 이 인자 왼쪽에 있는 열의 끝부분에 $\langle format \rangle$ 을 삽입한 채로 내용이 조판됩니다.

예를 들어,

```
>\sffamily c | l r<{원} >\{({}r<{\%})\}
```

를 $\langle spec. \rangle$ 인자로 입력한 경우 제1열은 산세리프체로 내용이 조판되며, 제3열은 각 칸의 끝에 '원'이 추가된 채로 조판되고, 제4열은 수식 모드에서 조판되며 끝에 '%'가 붙습니다.

```

1 \begin{center}
2   \begin{tabular}
3     >{\sffamily}c|lr<{원}>
4     {\{}r<{\%}\}}
5     \hline
6     A1 & 사과 & 3000 & 3.1 \\
7     A2 & 배 & 1500 & -2.8 \\
8     B1 & 볼펜 & 1800 & 1.3 \\
9   \end{tabular}
10 \end{center}

```

| | | | |
|----|----|-------|-------|
| A1 | 사과 | 3000원 | 3.1% |
| A2 | 배 | 1500원 | -2.8% |
| B1 | 볼펜 | 1800원 | 1.3% |

서식이 지정된 열의 특정 칸에는 이 서식이 적용되지 않도록 하려면 다음 절에서 설명하는 `\multicolumn` 제어 문자열을 사용할 수 있습니다. 예를 들어, 위 예시에서 제1행을 다음과 같이 추가할 수 있습니다.

```

1 ...
2 \hline
3 코드 & \sffamily 이름
4 & \multicolumn{1}{c}{\sffamily 가격}
5 & \multicolumn{1}{c}{\sffamily 증가율}
6 ...

```

| 코드 | 이름 | 가격 | 증가율 |
|----|----|-------|-------|
| A1 | 사과 | 3000원 | 3.1% |
| A2 | 배 | 1500원 | -2.8% |
| B1 | 볼펜 | 1800원 | 1.3% |

반복 인자

같은 속성의 열을 반복해서 생성해야 한다면 `*{<num.>}{<spec.>}`을 인자로 지정할 수 있습니다. 이 인자는 `<spec.>`을 `<num.>`번 반복 입력한 것과 같은 효과를 냅니다. 예를 들어, 열과 열 사이에 구분선이 있는 6열짜리 표를 그린다면 인자로

```
|c|c|c|c|c|c|
```

를 지정할 수도 있지만, 간단히 `|*{6}{c}|`를 인자로 지정해도 같은 결과를 얻습니다.

```

1 KAIST 수리과학과에서 개설하는 주요 과목은 다음과 같다.
2 \begin{center}
3   \begin{tabular}{|*{6}{c}|}
4     \hline
5     과목명 & 과목코드 & 과목 구분 & 개설 학기 & 핵심과목 여부 & 학점 \\
6     미적분학 I & MAS101 & 기초필수 & 봄, 가을 & 아니오 & 3

```

```

7      ...
8      르베그적분론 & MAS441 & 전공선택 & 가을 & 아니오 & 3 \\ \hline
9      \end{tabular}
10 \end{center}

```

KAIST 수리과학과에서 개설하는 주요 과목은 다음과 같다.

| 과목명 | 과목코드 | 과목 구분 | 개설 학기 | 핵심과목 여부 | 학점 |
|---------|--------|-------|-------|---------|----|
| 미적분학 I | MAS101 | 기초필수 | 봄, 가을 | 아니오 | 3 |
| 선형대수학개론 | MAS109 | 기초선택 | 봄, 가을 | 아니오 | 3 |
| 선형대수학 | MAS212 | 전공선택 | 봄, 가을 | 예 | 3 |
| 해석학 I | MAS241 | 전공선택 | 봄 | 예 | 4 |
| 현대대수학 I | MAS311 | 전공선택 | 봄 | 예 | 4 |
| 미분기하학개론 | MAS321 | 전공선택 | 가을 | 예 | 4 |
| 위상수학 | MAS331 | 전공선택 | 봄 | 예 | 4 |
| 르베그적분론 | MAS441 | 전공선택 | 가을 | 아니오 | 3 |

* 인자의 $\langle spec. \rangle$ 부분에는 표 12.1의 모든 인자를 사용할 수 있으므로, 다음과 같이 사용할 수도 있습니다.

```

1 \begin{displaymath}
2   \begin{array}{*{3}{r@{\;}c@{\;}}r@{\>}c@{\>}l}
3     x_1 & - & 2x_2 & + & x_3 & 2x_4 & = & 1 \\
4     x_1 & + & x_2 & - & x_3 & + & x_4 & = & 2 \\
5     x_1 & + & 7x_2 & - & 5x_3 & - & x_4 & = & 3
6   \end{array}
7 \end{displaymath}

```

$$\begin{array}{rcl}
 x_1 - 2x_2 + x_3 - 2x_4 & = & 1 \\
 x_1 + x_2 - x_3 + x_4 & = & 2 \\
 x_1 + 7x_2 - 5x_3 - x_4 & = & 3
 \end{array}$$

12.2.2 세로 정렬 인자

tabular 및 array를 사용해 작성한 표는 기본적으로 문자와 똑같이 취급됩니다. 따라서 별도의 문단에 표를 작성한 것이 아니라면 표의 앞뒤에 텍스트나 수식이 올 수도 있습니다. 이때 표의 위쪽, 가운데, 아래쪽 중 어디에 맞춰 글줄이 오도록 할지를 환경의 $\langle pos. \rangle$ 인자가 결정합니다. 기본값인 c는 표의 가운데에 글줄이 오도록 하며, t와 b는 각각 표의 위쪽과 아래쪽에 맞춰 글줄이 오도록 합니다.

예를 들어, 여러 표를 한 줄에 나란히 배치할 때 이 인자를 유용하게 사용할 수 있습니다. 높이가 다른 표를 배치할 때 첫 행, 가운데, 마지막 행 중 어디에 맞춰서 배치할지 결정할 수 있습니다.

```
1 \begin{center}
2   \begin{tabular}[t]{c|cc} ... \end{tabular}\quad
3   \begin{tabular}[t]{c|cc} ... \end{tabular}
4 \end{center}
```

| 퀴즈 | 점수 | 평균 | 퀴즈 | 점수 | 평균 |
|----|------|------|----|------|------|
| 1 | 17.5 | 16.3 | 1 | 13.5 | 16.6 |
| 2 | 13.5 | 15.1 | 2 | 16.0 | 15.1 |
| 3 | 20.0 | 12.7 | 3 | 19.5 | 12.8 |
| | | | 4 | 11.0 | 14.4 |

위 예시에서 선택 인자를 c나 b로 바꾸었을 때 어떤 결과를 얻는지는 직접 컴파일 하여 확인해 보시기 바랍니다.

12.3 환경 안에서 사용할 수 있는 명령어

이번 절에서는 tabular 및 array 환경 안에서 사용할 수 있는 명령어를 알아봅니다. 행을 구분하는 \\와 한 행 안에서 열을 구분하는 & 외에, 칸을 합치는 명령어와 가로선을 그리는 명령어를 사용할 수 있습니다.

12.3.1 칸 합치기

여러 열 합치기

\multicolumn 제어 문자열을 사용하면 한 행 안의 여러 칸을 합칠 수 있습니다.

```
\multicolumn{<num.>}{<spec.>}{<text>}
```

<num.>: 합칠 열의 개수

<spec.>: 합친 칸의 열 속성

<text>: 합친 칸에 작성할 내용

<spec.>에는 표 12.1의 열 생성 인자 하나를 입력하고, 필요한 경우 열 구분 인자나 열 속성 인자를 입력한 열 생성 인자 앞뒤에 적절히 입력하면 됩니다. 합친 칸 안에 작성할 내용은 <text> 인자로 지정하십시오. 이렇게 합친 칸에 조판할 내용은 \multicolumn의

`<text>` 인자로 지정해야 하며, 이 인자가 아닌 곳에 내용이 있다면 컴파일 과정에서 오류가 발생합니다.

```

1 \begin{center}
2   \begin{tabular}{|c|c|c|c|}
3     \hline
4     A1 & B1 & C1 & D1 \\ \hline
5     A2 & \multicolumn{3}{c}{B2:D2} \\ \hline
6     \multicolumn{2}{|c|}{A3:B3} & C3 & D3 \\ \hline
7     A4 & B4 & C4 & D4 \\ \hline
8   \end{tabular}
9 \end{center}

```

| | | | |
|-------|-------|----|----|
| A1 | B1 | C1 | D1 |
| A2 | B2:D2 | | |
| A3:B3 | | C3 | D3 |
| A4 | B4 | C4 | D4 |

`\multicolumn`을 사용하면 합칠 대상이 되는 열들에 지정된 서식(>, <로 지정한 것)이 무시되고, 각 열의 오른쪽에 지정된 구분 기호(|, @, ! 등)가 무시됩니다. 합칠 칸에 제1 열이 포함되어 있다면 표 가장 왼쪽에 지정된 구분 기호도 무시됩니다. 따라서 `<spec.>` 인자를 지정할 때 구분선이 사라지지 않도록 주의해야 합니다.

예를 들어, 위 예시에서 B2, C2, D2 칸을 합칠 때에는 B2, C2, D2 칸의 오른쪽에 있는 구분선이 사라지고, B2 칸의 왼쪽에 있는 구분선은 사라지지 않습니다. 이때 칸 오른쪽의 구분선이 사라지지 않도록 하려면 `<spec.>` 인자로 `c|`를 입력하여 구분선이 그려지도록 하면 됩니다. 반면, A3, B3 칸을 합칠 때에는 A3 칸의 왼쪽과 오른쪽, B3 칸의 오른쪽에 있는 구분선이 모두 사라집니다. 따라서 `<spec.>` 인자로 `|c|`를 입력하여 양쪽의 구분선을 다시 그려 주었습니다.

구분선을 특별히 지정하지 않은 경우와의 차이를 비교하기 위해, 두 `<spec.>` 인자에 입력한 |를 모두 삭제한 후 조판한 결과를 아래에 제시합니다.

```

1 \begin{center}
2   \begin{tabular}{|c|c|c|c|}
3     \hline
4     A1 & B1 & C1 & D1 \\ \hline
5     A2 & \multicolumn{3}{c}{B2:D2} \\ \hline

```



```

6 \multicolumn{2}{c}{A3:B3} & C3 & D3 \\ \hline
7 A4 & B4 & C4 & D4 \\ \hline
8 \end{tabular}
9 \end{center}

```

| | | | |
|-------|-------|----|----|
| A1 | B1 | C1 | D1 |
| A2 | B2:D2 | | |
| A3:B3 | | C3 | D3 |
| A4 | B4 | C4 | D4 |

여러 행 합치기

여러 행에 걸친 칸을 합치는 기능은 \LaTeX 이 기본으로 제공하지 않습니다. 그러므로, 이 기능을 제공하는 `multirow` 패키지를 먼저 불러와야 합니다.

```
\usepackage{multirow}
```

패키지를 불러왔다면 이 패키지에서 제공하는 `\multirow` 제어 문자열을 사용하여 칸을 합칠 수 있습니다.

```
\multirow[⟨pos.⟩]{⟨num.⟩}{⟨width⟩}{⟨text⟩}
```

⟨pos.⟩: 합친 칸의 내용을 정렬할 위치
 ⟨num.⟩: 합칠 행의 개수
 ⟨width⟩: 합친 칸에서 내용을 조판할 글줄의 폭
 ⟨text⟩: 합친 칸에 작성할 내용

⟨num.⟩ 인자에는 양수를 입력할 수도, 음수를 입력할 수도 있습니다. 양수를 입력하면 제어 문자열이 입력된 칸을 기준으로 아래 방향으로 칸을 합치며, 음수를 입력하면 제어 문자열이 입력된 칸을 기준으로 위 방향으로 칸을 합칩니다.

⟨width⟩ 인자에는 길이 값을 입력할 수도 있고, 이외에 별표 *와 등호 =를 입력할 수도 있습니다. *를 입력하면 ⟨text⟩에 입력한 내용이 한 줄로 조판되도록 합친 칸의 폭을 \LaTeX 이 자동으로 계산합니다. 합치려는 칸들이 p, m 또는 b 인자로 생성된 열이라면, 이 열의 폭을 그대로 사용하고자 할 때 =를 사용할 수 있습니다.

```

1 \begin{center}
2 \begin{tabular}{|c|c|c|c|}
3 \hline

```

```

4   A1 & \multirow{3}{*}{B1:B3} & C1 & D1 \\ \cline{1-1}\cline{3-4}
5   A2 & & C2 & D2 \\ \cline{1-1}\cline{3-4}
6   A3 & & \multirow{2}{*}{C3:C4} & D3 \\ \cline{1-2}\cline{4-4}
7   A4 & B4 & & D4 \\ \hline
8   \end{tabular}
9   \end{center}

```

| | | | |
|----|-------|-------|----|
| A1 | B1:B3 | C1 | D1 |
| A2 | | C2 | D2 |
| A3 | | C3:C4 | D3 |
| A4 | B4 | | D4 |

여러 행에 걸쳐 칸을 합친 경우에는 위 예시의 B2, B3, C4 칸처럼 칸을 빈 상태로 두어야 합니다. 또, `\hline`으로 그린 가로 구분선은 합친 칸을 가로지르므로, 뒤쪽에서 설명하는 `\cline`을 `\hline` 대신 사용해 그려 주어야 합니다.

선택 인자 (*pos.*)은 `tabular` 및 `array` 환경의 (*pos.*) 인자와 같은 역할을 하며, `t`, `c`, `b` 중 하나를 입력합니다. 기본값은 `c`로 합친 칸의 내용을 칸의 가운데에 조판하도록 하며, `t` 또는 `b`를 입력하면 내용이 합친 칸의 위쪽이나 아래쪽에 맞춰 조판됩니다.

```

1 \begin{center}
2   \begin{tabular}{|c|c|c|c|c|}
3     \hline
4     \multirow[t]{3}{*}{위쪽} & B1 & \multirow[c]{3}{*}{가운데} & D1 & \\
5     \multirow[b]{3}{*}{아래쪽} & & & & \\
6     & B2 & & D2 & \\
7     & B3 & & D3 & \\
8   \end{tabular}
9 \end{center}

```

| | | | | |
|----|----|-----|----|-----|
| 위쪽 | B1 | 가운데 | D1 | 아래쪽 |
| | B2 | | D2 | |
| | B3 | | D3 | |

행과 열 동시에 합치기

행과 열을 동시에 합치려면 `\multicolumn` 제어 문자열과 `\multirow` 제어 문자열을 같이 사용합니다. 합치고자 하는 칸의 각 행에서 `\multicolumn`을 사용해서 행별로 먼저 합쳐 준 다음, 이들 중 맨 위 칸이나 맨 아래 칸에서 `\multirow`를 사용하여 행들을

합쳐 주면 됩니다.

다음 예시에서는 B2~B3칸과 C2~C3칸을 `\multicolumn`으로 합친 뒤, 합쳐진 B2:B3 칸에 `\multirow`를 입력하여 4개의 칸을 합쳤습니다.

```

1 \begin{center}
2   \begin{tabular}{|c|c|c|c|}
3     \hline
4     A1 & A2 & A3 & A4 \\ \hline
5     B1 & \multicolumn{2}{c|}{\multirow{2}{*}{B2:C3}} & B4 \\ \cline{1-1}\cline{4-4}
6     C1 & \multicolumn{2}{c|}{} & C4 \\ \hline
7     D1 & D2 & D3 & D4 \\ \hline
8   \end{tabular}
9 \end{center}

```

| | | | |
|----|-------|----|----|
| A1 | A2 | A3 | A4 |
| B1 | B2:C3 | | B4 |
| C1 | | | C4 |
| D1 | D2 | D3 | D4 |

12.3.2 가로선 그리기

첫 번째 절에서 설명했듯, 가로선은 `\hline` 제어 문자열을 사용해 그립니다. 이 제어 문자열을 여러 번 연달아 입력하면 다중 가로선을 그릴 수도 있습니다.

한 행의 일부 열에만 가로선을 그리려면 `\hline` 대신 `\cline` 제어 문자열을 사용합니다.

`\cline{<range>}`

<range>: 가로줄을 그릴 열의 범위

인자 *<range>*에는 가로줄을 그릴 열의 범위를 *<i>-<j>*의 형식으로 입력하면 됩니다. 첫 열의 번호는 1입니다. 중간중간 끊어져 있는 가로선을 그리려면 이 제어 문자열을 여러 번 사용하면 됩니다(따라서 `\cline`으로는 다중 가로선을 그릴 수 없습니다). 예를 들어, 앞 페이지의 예시에서 `\cline{2-2}\cline{4-4}`는 제2열과 제4열에만 가로선을 그리고, 나머지 열에는 가로선을 그리지 않도록 합니다.

12.4 표 전체 폭 조정하기

tabular 환경을 사용해 작성한 표는 전체 폭이 표의 내용에 따라 변합니다. 표 전체의 폭을 고정하고 셀 사이의 간격이나 셀 크기를 조정하면서 표를 그릴 수도 있습니다.

12.4.1 tabular* 환경

L^AT_EX에서는 tabular* 환경을 제공하며, 이 환경을 사용하면 표의 전체 폭을 고정하여 표를 그릴 수 있습니다. $\langle pos. \rangle$ 와 $\langle spec. \rangle$ 인자는 tabular와 같은 방법으로 지정합니다.

```
\begin{tabular*}{\langle width \rangle}[\langle pos. \rangle]{\langle spec. \rangle}
```

$\langle width \rangle$: 표 전체의 폭
 $\langle pos. \rangle$: 앞뒤 텍스트와의 세로 정렬
 $\langle spec. \rangle$: 열의 개수와 각 열의 속성

```
\end{tabular*}
```

예를 들어, $\langle width \rangle$ 인자로 `\linewidth`를 지정하면 글줄의 길이와 일치하는 폭의 표가 만들어집니다. 다음으로, $\langle spec. \rangle$ 인자를 일반적인 tabular 환경처럼 지정한 뒤 첫 열의 왼쪽에

```
!\extracolsep{\fill}}
```

을 입력하면 됩니다. 이 명령어는 각 열 사이의 빈 공간을 균등하게 넓히는 역할을 해주어 표 전체의 폭이 $\langle width \rangle$ 에 지정된 길이에 맞게 해 줍니다.¹ 첫 열의 왼쪽에 세로선을 그리고 싶다면, 위의 명령어 대신 다음 예시처럼 첫 열의 왼쪽에

```
!\vline\extracolsep{\fill}}
```

을 입력하면 됩니다.

```
1 \begin{tabular*}{\linewidth}{!\vline\extracolsep{\fill}}lccr|}
2   \hline
3   왼쪽 & 가운데 & 가운데 & 오른쪽 \\ \hline
4   정렬된 & 정렬된 & 정렬된 & 정렬된 \\ \hline
5   1열 & 2열 & 3열 & 4열 \\ \hline
6 \end{tabular*}
```

¹ 자세히 설명하자면, `\extracolsep`은 현재 위치에서 오른쪽에 있는 모든 열에 인자로 지정된 만큼의 공간을 추가로 삽입하도록 하는 역할을 하며, `\fill`은 빈 공간을 최대한 채우는 ‘무한’ 길이를 뜻합니다.

| | | | |
|-----|-----|-----|-----|
| 왼쪽 | 가운데 | 가운데 | 오른쪽 |
| 정렬된 | 정렬된 | 정렬된 | 정렬된 |
| 1열 | 2열 | 3열 | 4열 |

tabular* 환경의 단점

tabular* 환경에는 두 가지 단점이 있습니다. 첫째는 아래 예시에서 확인할 수 있듯 세로 구분선이 열과 열 사이의 정가운데에 그려지지 않는다는 점입니다.

```

1 \begin{tabular*}{\linewidth}{!{\vline\extracolsep{\fill}}l|c|c|r|}
2   \hline
3   왼쪽 & 가운데 & 가운데 & 오른쪽 \\ \hline
4   정렬된 & 정렬된 & 정렬된 & 정렬된 \\ \hline
5   1열 & 2열 & 3열 & 4열 \\ \hline
6 \end{tabular*}

```

| | | | |
|-----|-----|-----|-----|
| 왼쪽 | 가운데 | 가운데 | 오른쪽 |
| 정렬된 | 정렬된 | 정렬된 | 정렬된 |
| 1열 | 2열 | 3열 | 4열 |

둘째는 각 열에서 내용이 작성되는 공간의 폭이 표 전체의 폭에 맞추어 자동으로 결정되도록 할 수 없다는 점입니다. 각 열의 폭은 표 전체의 폭과 독립적으로 결정되고 남은 공간은 공백으로 채워지므로, 공간을 효율적으로 사용하기 어렵습니다.

12.4.2 tabularx 환경

tabular* 환경의 단점은 tabularx 패키지에서 제공하는 tabularx 환경을 사용하여 해결할 수 있습니다. 이 환경은 tabular* 환경과 같은 인자를 입력받습니다.

```
\begin{tabularx}{\langle width \rangle}[\langle pos. \rangle]{\langle spec. \rangle}
```

$\langle width \rangle$: 표 전체의 폭

$\langle pos. \rangle$: 앞뒤 텍스트와의 세로 정렬

$\langle spec. \rangle$: 열의 수, 정렬 방법 및 구분선

```
\end{tabularx}
```

여기에서 $\langle spec. \rangle$ 인자를 지정할 때에는 l, c, r나 p 대신 X로도 열을 생성할 수 있습니다. X 인자로 생성된 열의 폭은 $\langle width \rangle$ 인자에 따라 자동으로 계산됩니다. 다음 예시를 참고하시기 바랍니다.

```

1 \begin{tabularx}{\linewidth}{|c|X|X|c|}
2   \hline
3   1행 1열 & \texttt{X} 인자로 생성된 & 각 열은 & 1행 4열 \\ \hline
4   2행 1열 & 나머지 열을 조판하고 & 남은 공간을 & 2행 4열 \\ \hline
5   3행 1열 & 완전히 채우도록 & 균등하게 분배됩니다. & 3행 4열 \\ \hline
6 \end{tabularx}

```

| | | | |
|-------|-------------|-------------|-------|
| 1행 1열 | x 인자로 생성된 | 각 열은 | 1행 4열 |
| 2행 1열 | 나머지 열을 조판하고 | 남은 공간을 | 2행 4열 |
| 3행 1열 | 완전히 채우도록 | 균등하게 분배됩니다. | 3행 4열 |

c 인자로 생성된 열은 각 칸의 내용에 맞추어 폭이 결정되었지만, x 인자로 생성된 두 열은 표의 전체 폭에서 다른 열을 조판하고 남은 공간을 완전히 채우도록 폭이 결정되었음을 확인할 수 있습니다. 각 열의 폭은 균등하게 분배됩니다.

연습문제

12.1. tabular 환경을 사용하여 다음 표를 작성하여라.

| 과목 | 중간고사 | 기말고사 |
|-------|------|------|
| 선형대수학 | 79 | 93 |
| 해석학 I | 83 | 88 |

12.2. array 환경을 사용하여 다음 표를 작성하여라.

| e_1 | e_2 | $e_1 \wedge e_2$ | $e_1 \vee e_2$ | $e_1 \rightarrow e_2$ |
|-------|-------|------------------|----------------|-----------------------|
| T | T | T | T | T |
| T | F | F | T | F |
| F | T | F | T | T |
| F | F | F | F | T |

12.3. 다음 표를 작성하여라. 마지막 열의 폭은 7cm로 하여라.

| 이름 | 기호 | 설명 |
|-----|----------|---|
| 논리합 | \vee | 두 명제 중 하나 이상이 참이면 참을 반환하고, 둘이 모두 거짓이면 거짓을 반환한다. |
| 논리곱 | \wedge | 두 명제가 모두 참이면 참을 반환하고, 하나 이상이 거짓이면 거짓을 반환한다. |
| 부정 | \neg | 명제가 참이면 거짓을 반환하고, 거짓이면 참을 반환한다. |

12.4. > 또는 < 인자를 사용하여 다음 표를 작성하여라.

| 문자 | 8진수 | 10진수 | 16진수 |
|----|------|------|------|
| ! | '041 | 33 | "21 |
| 0 | '060 | 48 | "30 |
| = | '075 | 61 | "3D |
| L | '114 | 76 | "4C |

12.5. * 인자를 사용하여 다음 표를 작성하여라.

| | | | | | | | |
|--------|---------|-----|------|---------|---------|------|-------|
| arccos | \arccos | csc | \csc | ker | \ker | min | \min |
| arcsin | \arcsin | deg | \deg | lg | \lg | Pr | \Pr |
| arctan | \arctan | det | \det | lim | \lim | sec | \sec |
| arg | \arg | dim | \dim | lim inf | \liminf | sin | \sin |
| cos | \cos | exp | \exp | lim sup | \limsup | sinh | \sinh |
| cosh | \cosh | gcd | \gcd | ln | \ln | sup | \sup |
| cot | \cot | hom | \hom | log | \log | tan | \tan |
| coth | \coth | inf | \inf | max | \max | tanh | \tanh |

12.6. 다음 표를 작성하여라.

| | 전설 모음 | | 후설 모음 | |
|-----|-------|-------|-------|-------|
| | 평순 모음 | 원순 모음 | 평순 모음 | 원순 모음 |
| 고모음 | ㅣ | ㄱ | ㅡ | ㅓ |
| 중모음 | ㅐ | ㅑ | ㅕ | ㅗ |
| 저모음 | ㅐ | | ㅓ | |

12.7. 다음 표를 작성하여라.

| | | 양순음 | 치조음 | 경구개음 | 연구개음 | 후음 |
|-----|----|-----|-----|------|------|----|
| 파열음 | 평음 | ㅂ | ㄷ | | ㄱ | |
| | 경음 | ㅃ | ㄸ | | ㄲ | |
| | 격음 | ㅍ | ㅌ | | ㅋ | |
| 파찰음 | 평음 | | | ㅈ | | |
| | 경음 | | | ㅊ | | |
| | 격음 | | | ㅉ | | |
| 마찰음 | 평음 | | ㅅ | | | ㅎ |
| | 경음 | | ㅆ | | | |
| 비음 | | ㅁ | ㄴ | | ㅇ | |
| 유음 | | | ㄹ | | | |

12.8. array 환경을 사용하여 다음 행렬을 그려라.

$$A = \begin{pmatrix} \begin{array}{|c|c|c|} \hline \lambda_1 & 1 & 0 \\ \hline 0 & \lambda_1 & 1 \\ \hline 0 & 0 & \lambda_1 \\ \hline \end{array} & & \\ & \begin{array}{|c|c|} \hline \lambda_2 & 1 \\ \hline 0 & \lambda_2 \\ \hline \end{array} & \\ & & \begin{array}{|c|} \hline \lambda_3 \\ \hline \end{array} & \\ & & & \ddots & \\ & & & & \begin{array}{|c|c|} \hline \lambda_n & 1 \\ \hline 0 & \lambda_n \\ \hline \end{array} \end{pmatrix}$$

제 13 장

그림 넣기

\LaTeX 에서는 두 가지 방법으로 문서에 그림을 넣을 수 있습니다. 첫 번째는 \LaTeX 의 명령어를 사용해 직접 그림을 그리는 것이고, 두 번째는 다른 방법으로 준비한 그림 파일을 \LaTeX 문서에 삽입하는 것입니다. 이번 장에서는 \LaTeX 문서에 그림을 넣는 여러 방법을 설명합니다.

13.1 직접 그림 그리기

\LaTeX 에서 그림을 직접 그리는 경우, \LaTeX 을 사용하는 것처럼 코드를 입력함으로써 모든 요소를 그려야 합니다. 따라서 그 과정이 — 일반적인 문서를 작성하는 것보다도 더 — 비직관적이므로 익숙해지기 어렵고, 즉흥적이거나 추상적인 그림을 그릴 때에도 구체적인 계산이 필요합니다. 하지만 \LaTeX 에서 직접 그림을 그릴 때의 분명한 장점도 여럿 있습니다.

- 본문에서 사용하는 글꼴을 그대로 사용할 수 있기 때문에, 문서의 일관성을 유지할 수 있습니다.
- \TeX 의 수식 작성 기능을 그대로 사용할 수 있으므로, 수식을 포함한 그림을 더 쉽게 그릴 수 있습니다.
- 모든 개체의 좌표나 크기 등을 명시적으로 지정하므로, 그림을 이루는 각 요소를 정확하게 그리고 배치할 수 있습니다.

이번 절에서는 \LaTeX 에서 그림을 그릴 때 사용할 수 있는 도구를 소개합니다. 이들을 사용하는 방법은 이 책의 범위를 벗어나므로, 각각의 매뉴얼을 참조하시기 바랍니다.

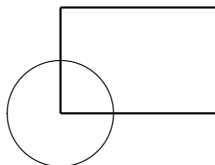
13.1.1 picture 환경

\LaTeX 은 간단한 그림을 그릴 수 있는 `picture` 환경을 제공합니다. 이 환경을 사용하면 패키지나 외부 프로그램의 도움 없이 \LaTeX 의 고유 기능만으로 그림을 그릴 수 있습니다. 각 요소를 배치할 곳을 `\put`으로 지정한 다음, 그 인자로 그리려는 요소의 세부 정보를 입력하는 방식을 사용합니다.

```

1 \setlength\unitlength{7mm}
2 \begin{picture}(4, 3)
3   \put(1, 1){\circle{2}}
4   \thicklines
5   \put(1, 1){\line(1, 0){3}}
6   \put(4, 1){\line(0, 1){2}}
7   \put(4, 3){\line(-1, 0){3}}
8   \put(1, 3){\line(0, -1){2}}
9 \end{picture}

```



하지만, 이 환경은 기능에 제약이 많아 오늘날 많이 쓰이지 않습니다. 선분과 화살표, 원, 타원, 베지에 곡선만을 그릴 수 있고, 선분이나 원을 실제로 그리는 것이 아니라 각각의 도형이 그려져 있는 글꼴을 사용해 글리프를 정확한 위치에 배치하는 방식을 사용합니다. 따라서 복잡한 그림을 그릴 때에는 많은 명령어를 사용해야 하고, 그릴 수 있는 도형의 크기나 선 굵기에도 많은 제약이 있습니다.

이 제약을 완화하기 위한 패키지를 `picture` 환경과 함께 사용할 수도 있습니다. `pict2e` 패키지를 사용하면 선분·화살표의 길이와 기울기, 원·타원의 크기에 있는 제약이 완화됩니다. 또, `curve2e` 패키지를 사용하면 선분과 화살표를 다른 방식으로 그릴 수 있으며, 곡선을 그리는 새로운 명령어를 사용할 수 있습니다.

`picture` 환경의 사용 방법은 [2, 17, 27] 등을 참조하시기 바랍니다.

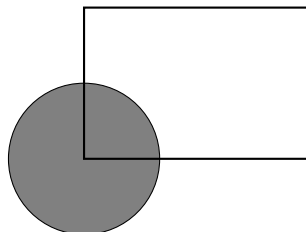
13.1.2 tikz 패키지

틸 탄타우(Till Tantau)가 개발한 `tikz` 패키지는 오늘날 \LaTeX 에서 그림을 그릴 때 가장 많이 사용되는 패키지입니다. PGF(portable graphics format) 엔진을 기반으로 그래픽을 그리며, `tikz` 패키지는 이를 사용해 그림을 그리는 명령어를 제공합니다. 이 패키지는 그림을 ‘어떻게’ 그리는지가 아니라 ‘무엇을’ 그리는지에 집중하여, 편리한 기능을 다른 도구들보다 더 많이 제공합니다.

```

1 \begin{tikzpicture}
2   \path (0, 0) rectangle (4, 3); %
   bounding box
3   \filldraw[fill=gray] (1, 1) circle
   [radius=1];
4   \draw[thick] (1, 1) rectangle (4, 3);
5 \end{tikzpicture}

```



PGF 엔진은 직선과 곡선을 직접 그리므로, 앞서 설명한 `picture` 환경에 있었던 제약 없이 자유롭게 그래픽을 그릴 수 있습니다. 다양한 기능을 편리하게 사용할 수 있도록 명령어가 정의되어 있으며, 복잡한 기능은 추가 라이브러리를 통해 사용할 수도 있습니다. 이 책에 있는 대부분의 그림도 이 패키지를 사용해 작성되었습니다.

이 패키지는 컴파일 엔진에 관계없이 잘 작동하기 때문에, 그래픽을 생성하는 패키지는 이 패키지를 기반으로 작동하기도 합니다. 대표적인 것은 함수의 그래프를 그리는 `pgfplots` 패키지나, 제11장에서 설명한 가환 도표를 그리는 `tikz-cd` 패키지입니다.

`tikz` 패키지의 소개와 사용 방법은 패키지 안내서[48]를 참조하시기 바랍니다.

13.1.3 다른 도구

\LaTeX 의 `picture` 환경과 `tikz` 패키지 외에도 다른 패키지나 프로그램을 사용하여 \LaTeX 내부에서 그림을 그릴 수 있습니다.

pstricks 패키지

`pstricks` 패키지는 PostScript 언어를 직접 사용해 그림을 그릴 수 있도록 하는 패키지입니다. PostScript를 직접 사용하는 만큼 강력한 기능을 사용할 수 있습니다.

하지만, PDF 파일을 바로 생성하는 \pdfTeX 과 \LuaTeX 에서는 사용할 수 없고, DVI 문서를 생성한 뒤 이를 PS 문서로 변환하는 방식으로 사용해야 한다는 불편함이 있습니다. PDF 문서가 필요하다면 이 PS 파일을 한 번 더 PDF로 변환해야 합니다. 또는, `pst-pdf` 패키지나 `pdftricks` 패키지를 대신 사용할 수도 있습니다.

`pstricks` 패키지에 관한 자세한 내용은 패키지 안내서[54]를 참조하시기 바랍니다.

xy 패키지

`xy` 패키지는 그래프와 도표 등을 그리는 데 특화된 패키지로, 몇 가지 특정한 형태의 그림을 편리하게 그릴 수 있는 기능을 제공합니다. 행렬 기능, 그래프 기능, 다각형 기능,

격자 기능, 매듭 기능 등을 제공하며, 격자·타원·호·곡선 등도 다양한 형태로 그릴 수 있습니다. 그림을 작성하는 코드가 난해하다는 단점이 있지만, 이 패키지를 사용하면 자연과학과 공학 분야에서 사용하는 다양한 모식도와 도표를 그릴 때 유용한 강력한 기능을 이용할 수 있습니다.

xy 패키지의 소개와 사용 방법은 패키지 안내서[41, 42]를 참조하시기 바랍니다.

METAPOST

METAPOST는 도널드 커누스 박사가 \TeX 과 함께 개발한 프로그램으로, \TeX 의 글꼴을 개발하는 프로그램인 METAFONT를 기반으로 만들어졌습니다. 그림을 작성하는 코드를 METAFONT 소스 파일(.mp 파일)에 작성한 뒤 mpost 명령어를 실행하여 PNG나 EPS 파일을 얻습니다. 이를 \LaTeX 문서로 불러들임으로써 문서에 그림을 넣을 수 있습니다.

METAPOST에 대한 소개와 자세한 사용 방법은 프로그램의 매뉴얼[19]을 참조하시기 바랍니다.

Asymptote

Asymptote는 2차원 및 3차원 그림을 그릴 수 있는 프로그램으로, C++ 언어와 비슷한 방식의 문법을 사용합니다. 스크립트 기능과 프로그래밍 기능을 지원함으로써 복잡한 그림을 더 편리하게 그릴 수 있습니다. \LaTeX 소스 파일과 별개의 Asymptote 소스 파일을 작성한 뒤 이를 컴파일 해 얻는 EPS 또는 PDF 파일을 문서 파일에 삽입하거나, asymptote 패키지를 불러온 뒤 그림의 코드를 \LaTeX 소스 파일에 바로 작성한 후 일련의 컴파일 과정(latex → asy → latex)을 거쳐 원하는 그림을 문서에 넣을 수도 있습니다.

Asymptote에 대한 소개와 자세한 사용 방법은 프로그램의 매뉴얼[18]을 참조하시기 바랍니다.

13.2 외부 그림 파일 삽입하기

이번 절에서는 \LaTeX 문서에 외부 그림 파일을 삽입하는 방법을 알아봅니다. 이때에는 graphics 패키지 또는 graphicx 패키지를 사용하며, 이 책에서는 더 흔히 사용되는 graphicx 패키지를 사용한 방법을 설명하겠습니다. 두 패키지는 옵션을 지정하는 문법에만 차이가 있으며, 같은 기능을 제공합니다.

\LaTeX 문서에 외부 그림 파일을 삽입할 때에는 `\includegraphics` 제어 문자열을 사용합니다.

`\includegraphics[<options>]{<file>}`

<options>: 그림을 삽입할 때의 옵션(들)

<file>: 삽입할 그림 파일의 이름

문서에 넣으려는 그림 파일을 준비한 뒤, *<file>* 인자로 그 파일의 디렉토리명과 이름을 입력하면 됩니다. 이때 삽입하려는 그림은 EPS, PDF, PNG, 또는 JPEG 형식이어야 합니다. DVI 파일을 생성하는 \LaTeX 엔진에서는 EPS 형식의 그림 파일만을 삽입할 수 있습니다.

```
1 \LaTeX\ 문서에는 아래와 같이 그림을 삽입할 수 있
  습니다. \par
2 \begin{center}
3   \includegraphics{Pictures/sample.pdf}
4 \end{center}
```

\LaTeX 문서에는 아래와 같이 그림을 삽입할 수 있습니다.



삽입된 그림은 문자와 동일하게 취급되어 배치됩니다. 따라서, 위 예시와 같이 center 환경 안에서 `\includegraphics` 제어 문자열을 사용하면 그림이 가운데에 배치됩니다. 제14장에서 설명하는 figure 환경을 사용하면 그림을 유동 개체로 취급되도록 할 수도 있습니다.

문서에 삽입한 그림 파일은 소스 파일을 컴파일 할 때마다 필요합니다. 문서를 수정할 경우를 대비하여, 그림 파일이 이동·삭제되지 않도록 주의하십시오.

13.2.1 그림 삽입 옵션

`\includegraphics`의 선택 인자를 통해 그림을 삽입할 때 크기를 바꾸거나 그림을 회전하는 등 여러 옵션을 지정할 수 있습니다. 여기의 옵션은 키-값 문법을 사용하며, 자주 쓰이는 옵션 키로는 표 13.1의 항목을 사용할 수 있습니다. [8]에서는 더 다양한 옵션을 확인할 수 있습니다.

그림을 삽입할 때 `width`와 `height` 옵션 중 하나만을 지정하면 그림의 종횡비를 유지한 채로 지정된 값에 맞추어 크기가 조정되며, 두 값을 모두 지정하면 그림의 원래 종횡비가 무시됩니다. 이때 `keepaspectratio = true`를 옵션으로 추가하면 종횡비를 유지하되 지정된 크기를 넘지 않도록 그림의 크기가 조절됩니다.

예를 들어, 다음 코드는 `sample.pdf` 문서의 2페이지를 폭 5 cm로 삽입합니다.

```
\includegraphics[page=2, width=5cm]{sample.pdf}
```

| 표 13.1 | graphicx 패키지: 그림을 삽입할 때의 주요 옵션

| 옵션 키 | 설명 |
|-----------------|--|
| width | 그림을 삽입할 때의 폭을 지정합니다. |
| height | 그림을 삽입할 때의 높이를 지정합니다. |
| keepaspectratio | 그림의 종횡비를 유지한 채로 삽입할지의 여부를 결정합니다. 값으로 true 또는 false를 지정합니다. |
| scale | 그림을 확대·축소할 배율을 지정합니다. |
| angle | 그림을 반시계 방향으로 회전할 각도를 도(°) 단위로 지정합니다. 회전의 기준점은 그림의 좌하단 꼭짓점입니다. |
| page | 여러 페이지의 PDF 문서를 삽입할 경우, 사용할 페이지를 지정합니다. 지정하지 않으면 1페이지가 사용됩니다. |
| draft | 그림을 문서에 실제로 삽입하지 않고, 그림과 같은 크기의 상자를 대신 그린 뒤 상자 안에 파일의 이름을 조판합니다. |

13.2.2 기타 설정

graphics 패키지와 graphicx 패키지는 문서에 그림을 삽입할 때 편리하게 사용할 수 있는 명령어로 `\graphicspath`와 `\DeclareGraphicsExtension`을 제공합니다. [8]에서는 `\DeclareGraphicsRule` 제어 문자열도 확인할 수 있습니다. 이는 자동 파일 변환 작업 등에 유용하게 사용할 수 있습니다.

그림 파일의 기본 경로 지정하기

`\graphicspath` 제어 문자열은 그림 파일을 검색할 디렉토리를 설정하는 역할을 합니다. `\includegraphics`의 `<file>` 인자에 파일의 확장자가 입력되지 않은 경우에 지정된 디렉토리에서 파일을 검색합니다.

`\graphicspath{<directories>}`

`<directories>`: 그림 파일을 검색할 디렉토리의 목록. 각 디렉토리는 중괄호 `{...}` 안에 작성합니다.

디렉토리를 구분할 때에는 운영 체제에 상관없이 빗금 `/`을 사용합니다. 디렉토리가 여러 개 입력된 경우, 먼저 입력된 디렉토리에서 지정된 파일을 검색하고, 여기에 파일이 존재하지 않는 경우 다음 디렉토리에서 파일을 검색합니다. 입력된 디렉토리의 하위 디렉토리는 검사하지 않습니다.

예를 들어, 다음 코드는 그림 파일을 작업 폴더 하위의 pictures 폴더에서 검색하도록 선언합니다.

```
\graphicspath{{pictures/}}
```

또, 다음 코드를 선언하였다면 그림을 작업 폴더 하위의 pictures/pdf 폴더에서 먼저 검색한 후, 파일이 존재하지 않으면 pictures/png 폴더에서 검색합니다.

```
\graphicspath{{pictures/pdf/}{pictures/png/}}
```

그림 파일의 기본 확장자 지정하기

\includegraphics의 $\langle file \rangle$ 인자에 삽입하려는 그림 파일의 확장자가 포함되어 있는 경우, \LaTeX 은 정확히 그 파일을 불러옵니다. 하지만 확장자가 지정되어 있지 않다면 \DeclareGraphicsExtensions의 인자로 지정된 확장자를 가지는 파일을 검색합니다.

```
\DeclareGraphicsExtensions{ $\langle extensions \rangle$ }
```

$\langle extensions \rangle$: 검색할 그림 파일의 확장자. 여러 확장자를 지정하려는 경우 각 확장자는 쉼표로 구분합니다.

예를 들어,

```
\DeclareGraphicsExtensions{.pdf,.png}
```

를 선언하였다면 \includegraphics{image}는 image.pdf 파일을 먼저 검색한 뒤, 이 파일이 존재하지 않으면 image.png 파일을 검색합니다. 반면, 기본 확장자 선언 여부에 관계없이 \includegraphics{profile.jpg}를 입력하면 항상 profile.jpg 파일이 삽입됩니다.

제 14 장

표와 그림의 배치

LaTeX은 기본적으로 표와 그림 등 개체를 문자와 동일하게 취급합니다. 문단 안에 입력된 개체는 문단 중간에 배치되며, 문단과 문단 사이에 입력된 개체는 별도의 줄에 배치됩니다. center 환경을 사용하면 이러한 개체를 판면 가운데에 배치할 수 있습니다.

하지만, 표나 그림은 보통 넓은 공간을 차지하기 때문에 문서의 레이아웃을 쉽게 망가뜨리고, 이로 인해 지면의 많은 공간이 비워져 버릴 수도 있습니다. 이러한 문제를 해결하려면 개체를 배치할 때 다른 방법을 사용해야 합니다. 이번 장에서는 이때 사용할 수 있는 두 가지 개체 배치 방법에 대해 알아봅니다. 또, 개체에 캡션과 번호를 붙이는 방법도 다룹니다.

14.1 유동 개체로 취급하여 배치

유동 개체로써 표와 그림을 배치하는 것은 — 쉽게 말하자면 — 아래아한글의 ‘자리 차지’, Word의 ‘위/아래’ 방식으로 개체를 배치하는 것과 비슷합니다.

유동 개체(floating object)는 일반적인 텍스트와는 별개로 취급되어 배치됩니다. 유동 개체 환경 안에 작성된 그림이나 표는 여유 공간이 있는 경우에 한해 현재 페이지에 배치되며, 그렇지 않은 경우에는 환경 뒤쪽의 내용이 먼저 조판됩니다. 현재 페이지에 배치되지 않은 개체는 다음 페이지에서 배치를 시도합니다.

14.1.1 기본 사용 방법

LaTeX 표준 클래스에서는 유동 개체가 두 가지로 구분됩니다. 첫 번째 유형인 표는 table 환경 안에 작성하며, 두 번째 유형인 그림은 figure 환경 안에 작성합니다. 이 환경 안에 배치된 개체는 현재 페이지의 위쪽에 배치되며, 공간이 적절하지 않은 경우

현재 페이지의 아래쪽에 배치합니다. 두 곳에 모두 개체를 배치할 수 없다면 개체는 다음 페이지로 넘어가고, 환경 뒤쪽에 입력된 내용이 먼저 조판됩니다.

예를 들어, 159쪽과 160쪽에 있는 표 11.1과 11.2를 배치할 때에는 코드를 다음과 같이 작성하였습니다. (간단한 설명을 위해 일부 코드는 변형 및 생략하였습니다.)

```

1 ... 이 표에 있는 옵션을 입력하면 화살표의 머리, 꼬리, 선의 모양이 모두 변경됩니다.
2
3 \begin{table}
4   \caption{\Lpkg{tikz-cd}~패키지: 화살표의 모양} % 표 11.1
5   \label{table:tikzcdarrowoptions}
6   \begin{tabularx}{...}{...} ... \end{tabularx}
7 \end{table}
8
9 한편, 머리, 꼬리, 선의 모양 중 일부만을 변경하는 명령어도 있으며, ... 더 다양한
  화살표를 만들 수도 있습니다.
10
11 \begin{table}
12   \caption{\Lpkg{tikz-cd}~패키지: 화살표의 부분별 모양} % 표 11.2
13   \label{table:tikzcdpartialarrowoptions}
14   \begin{tabularx}{...}{...} ... \end{tabularx}
15 \end{table}
16
17 \Lpkg{tikz-cd}~패키지가 제공하는 화살표의 머리\cdot 꼬리\cdot 선의 모양 옵션
  외에, ...

```

첫 번째 table 환경의 앞 부분인 ‘…… 머리, 꼬리, 선의 모양이 모두 변경됩니다.’까지의 내용을 조판한 후 판면의 남은 공간을 계산해 보았을 때, 표 11.1을 배치할 공간이 남아 있었으므로 이 표는 해당 페이지의 위쪽에 배치되었습니다. 반면, 이 표를 배치하고 두 번째 table 환경의 앞 부분인 ‘…… 화살표를 만들 수도 있습니다.’까지의 내용을 조판한 후 판면의 남은 공간을 계산했을 때에는 표 11.2를 배치할 공간이 없었기 때문에, 이 표는 다음 페이지에 배치되었습니다.

예시에서 확인할 수 있듯, 판면의 여유 공간과 개체의 크기에 따라 개체는 현재 페이지에 배치될 수도, 다음 페이지에 배치될 수도 있습니다. 만약 103쪽의 표 8.4처럼 개체가 많이 큰 경우에는 별도의 페이지에 개체가 배치되기도 합니다.

개체 배치 순서와 개체의 종류

LaTeX은 같은 종류의 유동 개체는 반드시 소스 파일에 입력된 순서에 맞춰 배치합니다. 소스 파일에 그림 1과 그림 2가 삽입되어 있을 때, 현재 페이지에 그림 1이 배치될 공

표 14.1 | 유동 개체의 배치 허용 위치 인자

| 인자 | 설명 |
|----|--|
| h | 개체를 현재 위치(here), 즉 페이지 중간에 배치합니다. |
| t | 개체를 현재 페이지의 위쪽(top)에 맞추어 배치합니다. |
| b | 개체를 현재 페이지의 아래쪽(bottom)에 맞추어 배치합니다. |
| p | 개체를 유동 개체 전용 페이지(page)에 배치합니다. |
| ! | 개체를 배치하기 위해 \LaTeX 이 사용하는 내부 인자를 일부 무시합니다. |

간이 없다면 그림 2를 현재 페이지에 배치할 수 있더라도 두 그림 모두 다음 페이지에, 순서를 지켜서 배치됩니다.

반면, 다른 종류의 개체 간에는 배치 순서가 달라질 수 있습니다. 소스 파일에 그림 1, 표 1, 그림 2가 이 순서대로 작성되어 있을 때, 현재 페이지에 그림 1이 배치될 공간은 없지만 표 1은 배치할 수 있다면 표 1이 먼저 배치됩니다. 그림 1은 다음 페이지나 그 뒤 페이지에 배치됩니다.

14.1.2 배치 허용 위치 지정하기

유동 개체 환경은 선택 인자를 입력받으며, 여기에는 h, t, b, p, !를 조합해 ‘유동 개체를 배치하도록 허용할 곳’을 지정해 줍니다. 기본값은 `tbp`이며, 각 문자의 의미는 표 14.1에서 확인할 수 있습니다. 개체가 배치되는 위치는 인자가 지정된 순서에 영향을 받지 않습니다. 즉, 다음 두 코드는 표 개체를 같은 곳에 배치합니다.

```

1 \begin{table}[!htp] ... \end{table}
2 \begin{table}[tp!h] ... \end{table}

```

예를 들어, 기본값인 `tbp`는 ‘페이지의 위쪽(t)에 배치를 시도하며, 공간이 부족한 경우 페이지의 아래쪽(b)에 배치를 시도하고, 이때에도 공간이 부족하다면 개체를 전용 페이지(p)에 배치함’을 뜻합니다. `ht`를 인자로 지정한 경우에는 ‘현재 위치(h)에 배치를 시도하며, 공간이 없다면 개체를 페이지의 위쪽(t)에 배치를 시도함’을 뜻합니다.

배치 허용 위치 지정 인자는 ‘이곳에 배치하는 것을 허용한다’를 뜻하며, ‘항상 이곳에 개체를 배치한다’를 뜻하지 않습니다. \LaTeX 은 유동 개체를 배치하는 데 여러 내부 인자를 고려하므로, 원하는 대로 개체가 배치되지 않을 수도 있습니다. 느낌표 `!`를 선택 인자에 포함시키면 \LaTeX 이 개체를 배치할 때 내부 인자를 일부 무시하게 되어, 의도하는 위치에 개체가 조판될 가능성이 높아집니다. (다만, 느낌표를 사용하더라도 항상

개체가 원하는 곳에 배치되는 것은 아닙니다.) 내부 인자에 대한 내용은 [33, pp. 510–512]에서 확인할 수 있습니다.

예를 들어, 배치 허용 위치 인자로 ‘!tp’가 지정된 경우, 이는

내부 인자를 일부 무시하고(!) 개체를 배치한다. 개체는 페이지의 위쪽(t)에 우선 배치를 시도하며, 여기에 개체를 배치할 수 없는 경우에는 전용 페이지(p)에 개체를 배치한다.

를 의미합니다.

이번 절의 나머지 부분에서는 유동 개체에 관해 더 자세한 내용을 다루므로, 일반적인 문서를 작성할 때에는 여기까지만의 내용을 알고 있어도 충분합니다.

유동 개체의 배치 과정

L^AT_EX의 유동 개체는 다음의 과정에 따라 배치됩니다. 이 과정을 반드시 알 필요는 없지만, 개체가 원하는 위치에 잘 배치되지 않는 경우 다음 과정을 참조하면 도움이 될 수 있습니다.

- 1 | 배치 허용 위치 인자에 !가 포함되어 있는지 확인합니다. 참인 경우, 이하의 모든 과정에서 내부 인자를 일부 무시하고 진행합니다.
- 2 | 배치 허용 위치 인자에 h가 포함되어 있는지 확인합니다. 참인 경우, 유동 개체 환경 이전에 작성된 내용을 조판하고, 그 아래쪽에 개체를 배치할 수 있는지 계산합니다. 가능한 경우 개체를 배치하며, 그렇지 않은 경우 다음 단계로 넘어갑니다. 만약 h 외에 t, b, p 중 다른 인자가 지정되지 않은 경우, t가 지정되어 있다고 가정하고 진행합니다.
- 3 | 배치 허용 위치 인자에 t가 포함되어 있는지 확인합니다. 참인 경우, 페이지의 위쪽에 개체 배치를 시도합니다. 제약 사항과 충돌하지 않는 경우 개체를 이곳에 배치하며, 그렇지 않은 경우 다음 단계로 넘어갑니다.
- 4 | 배치 허용 위치 인자에 b가 포함되어 있는지 확인합니다. 참인 경우, 위와 같은 과정을 페이지 아래쪽에 시도합니다.
- 5 | 위의 과정을 통해 개체가 배치되지 않았다면, 이 개체는 ‘배치 대기열’에 들어갑니다.

배치 대기열에 개체가 있는 경우, 새로운 페이지로 넘어가기 전까지 모든 유동 개체는 위의 과정을 거치지 않고 바로 배치 대기열로 들어갑니다. 새로운 페이지가 시작되고

배치 대기열에 개체가 있는 경우, 이들을 먼저 최대한 배치한 후 문서의 나머지 부분을 처리합니다. 이 과정은 다음 순서로 진행됩니다.

- 1 | 대기열의 첫 번째 개체에 p가 배치 허용 위치 인자에 포함되어 있는지 확인합니다. 거짓인 경우 이 대기열이 유지되며, 계속하여 문서를 조판합니다. 이는 문서가 `\clearpage` 제어 문자열이나 문서의 끝을 만날 때까지 계속됩니다. 참인 경우, 유동 개체 전용 페이지를 만든 후 다음 과정이 진행됩니다.
- 2 | 우선 이 개체를 현재 페이지에 배치하기로 기억해 둡니다. 실제로는 전용 페이지가 유동 개체로 ‘충분히(이는 내부 인자로 결정됨)’ 채워지기 전까지는 배치되지 않습니다.
- 3 | 대기열의 끝에 도달할 때까지 대기열에 있는 나머지 개체에 p 인자가 포함되어 있는지 순서대로 검사합니다. p 인자가 없는 개체가 발견되면 그 앞 개체들까지만을 고려하여, 그렇지 않은 경우 대기열 전체의 개체에 대하여 다음 과정을 진행합니다.
- 4 | 개체들의 크기를 계산하여 전용 페이지가 충분히 채워진다면 그 개체들을 조판합니다. 이러한 방식으로 최대한 많은 개체들을 순서를 지켜 유동 개체 전용 페이지에 조판합니다. 충분히 채워지지 않는 페이지가 생기는 경우, 그 페이지에 배치하려던 개체들은 대기열에 남겨 두고, 앞 페이지까지만 개체를 조판합니다.
- 5 | 대기열에 남아 있는 개체들을 새로운 페이지의 판면 위쪽이나 아래쪽에, 앞 페이지의 3~4번 과정을 따라 배치합니다.

유동 개체의 수나 크기에 따라, 위의 과정을 모두 거치더라도 대기열에 유동 개체가 남아 있을 수 있습니다. 이들은 `\clearpage`, `\cleardoublepage`나 문서의 끝을 만나면 p 인자의 유무에 관계없이 이 과정을 거쳐 배치됩니다. report 및 book 클래스의 `\part`나 `\chapter` 제어 문자열에는 `\clear(double)page`가 포함되어 있으므로, 이 두 클래스를 사용하는 문서에서는 유동 개체가 현재 장에 반드시 배치됩니다.

float 패키지의 H 인자

안젤름 링나우(Anselm Lingnau)가 작성한 float 패키지를 불러 오면, 유동 개체 환경의 특별한 배치 허용 위치 인자로 H를 사용할 수 있습니다. 이는 ‘반드시 이 개체를 여기에 배치한다’, 즉 해당 개체를 유동 개체로 취급하여 배치하지 않음을 뜻합니다. 개체를 유동 개체로 취급할 필요가 없다면 유동 개체 환경을 사용하지 않으면 되지만, 개체에 캡션을 달아야 한다면 — 캡션을 다는 `\caption` 제어 문자열은 유동 개체 환경 안에

서만 사용할 수 있으므로 — 유동 개체 환경을 반드시 사용해야 합니다. 이때 H 인자를 유용하게 사용할 수 있습니다.

참고로, H 인자는 표 14.1의 인자들과 함께 사용할 수 없습니다. H 인자의 의미를 따져 보았을 때, 이 인자는 단독으로 사용하는 것이 자연스럽기도 합니다.

14.1.3 2단 편집 문서의 유동 개체

2단 편집 문서에서 table이나 figure 환경을 사용하면 하나의 단을 하나의 페이지처럼 생각하여 유동 개체가 배치됩니다. 즉, 유동 개체가 양쪽 단에 걸쳐진 형태로 배치될 수 없습니다. 별표 붙은 환경인 table* 환경과 figure* 환경을 사용하면 유동 개체를 2단에 걸쳐 배치할 수 있습니다.

양쪽 단에 걸쳐 유동 개체를 배치하는 경우, 개체는 판면의 위쪽이나 전용 페이지에만 배치되며, 페이지의 중간이나 아래쪽에는 배치되지 않습니다. 따라서, 배치 허용 위치 인자로 h나 b를 사용할 수 없습니다. float 패키지의 H 인자도 사용할 수 없습니다. 배치 허용 위치 인자로 t나 p가 지정되지 않는다면, 이 유동 개체는 (앞서 설명한 과정에 의해) \clearpage의 앞이나 문서의 끝에 배치될 것입니다.

14.2 판면 좌우에 배치

도널드 아르세노(Donald Arseneau)가 작성한 wrapfig 패키지를 사용하면 개체를 판면의 왼쪽이나 오른쪽에 배치하고, 그 주변을 글로 채울 수 있습니다. 쉽게 말하자면 아래아 한글의 ‘어울림’, Word의 ‘정사각형’ 방식으로 개체를 배치하는 것과 비슷합니다.

개체를 배치할 때에는 wrapfloat 환경을 사용합니다.

```
\begin{wrapfloat}{<type>}{<lines>}{<placement>}{<overhang>}{<width>}
```

<type>: 배치하려는 개체의 종류

<lines>: 개체를 감쌀 글줄의 수

<placement>: 개체를 배치할 방향

<overhang>: 개체를 페이지 바깥쪽으로 이동할 거리

<width>: 개체를 배치할 공간의 폭

```
\end{wrapfloat}
```

<type> 인자에는 개체의 종류, 즉 table이나 figure를 입력합니다. 14.4절의 방법을 통해 새로운 종류의 개체를 정의한 경우, 그것을 입력해도 됩니다. 한편, 표나 그림을 배치할 때에는 간단히 wraptable 환경과 wrapfigure 환경을 사용할 수 있습니다. 두 환경은 wrapfloat 환경에서 첫 번째 인자가 이미 지정된 것이라고 생각하면 됩니다.

표 14.2 | wrapfig 패키지: 개체의 배치 방향

| 인자 | 설명 |
|------|--|
| l, L | 개체를 문단의 왼쪽(left)에 배치합니다. |
| r, R | 개체를 문단의 오른쪽(right)에 배치합니다. |
| i, I | 양면 편집 문서에서, 개체를 안쪽(inner side, 책등이 있는 쪽)에 배치합니다. |
| o, O | 양면 편집 문서에서, 개체를 바깥쪽(outer side, 책등의 반대쪽)에 배치합니다. |

⟨lines⟩ 인자는 개체 주위를 몇 줄의 본문 글줄이 감쌀지를 지정합니다. 보통은 이 인자를 생략해도 페이지가 잘 만들어지지만, 수식이나 특수한 상자 등이 주변 본문에 있는 경우에는 완벽하지 않은 결과를 얻을 수 있습니다. 이때에는 이 인자를 사용하여 문서의 모습을 조정해 줄 수 있습니다. 이때, 별행수식 한 개는 3줄을 차지하는 것으로 계산합니다.

⟨placement⟩ 인자에는 l, L, r, R, i, I, o, O의 8개 인자 중 하나를 입력합니다. 소문자는 ‘현재 위치에 배치함’을 뜻하며, 대문자는 ‘유동 개체처럼, 현재 위치에 여유 공간이 있을 때에만 배치함’을 뜻합니다. 각 문자의 의미는 표 14.2에서 확인할 수 있습니다.

⟨overhang⟩ 인자에 대해서는 뒤에서 설명합니다. ⟨width⟩ 인자에는 특수한 값으로 0pt를 지정할 수 있습니다. 이때에는 환경 안에 입력된 개체의 크기에 맞추어 폭이 자동으로 결정됩니다.

예를 들어, 251쪽에 있는 그림 17.1을 삽입할 때에는 코드를 다음과 같이 작성하였습니다.

```

1 \section{책갈피} \label{sec:bookmark}
2
3 \begin{wrapfigure}[17]{r}{5.4cm}
4   \includegraphics[...]{...}
5   \caption{Adobe Acrobat의 책갈피 창}\label{fig:acrobatbookmark}
6 \end{wrapfigure}
7
8 많은 PDF~뷰어는 \term[bookmark]{책갈피}\index{책갈피} 기능을 ...
```

이는 그림을 판면의 오른쪽 끝에 맞추어 배치하고, 그림을 배치할 공간의 폭은 5.4cm로 함을 뜻합니다. 선택 인자 ⟨lines⟩를 생략하면 그림 14.1처럼 그림 아래쪽에 지나치게 넓은 공간이 만들어지므로, 선택 인자로 ‘17’을 지정하였습니다.

18.2.1 대체 문자열 지정

책갈피 창에는 일반 문자만을 사용할 수 있으며, 문자에 서식을 지정하거나 수식을 작성할 수 없습니다. 따라서 장절 표제에 복잡한 TeX 명령어나 수식을 입력했다면 책갈피 창에는 표제가 제대로 표시되지 않을 수 있습니다. 이때에는 `\texorpdfstring` 제어 문자열을 사용해 문서를 조판할 때 사용할 TeX 코드와 책갈피 항목에 사용할 내용을 구분하여 지정해 줄 수 있습니다.

```
\texorpdfstring{<TeX text>}{<PDF text>}
```

<TeX text>: 문서를 조판할 때 사용할 TeX 코드

<PDF text>: 책갈피를 만들 때 사용할 문자열

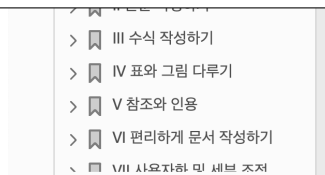


그림 18.1 | Adobe Acrobat의 책갈피 창

그림 14.1 | `<lines>` 인자를 지정하지 않았을 때 그림이 잘못 배치된 모습

14.2.1 개체를 수평 방향으로 이동하기

두 번째 선택 인자인 `<overhang>` 인자는 개체를 수평 방향으로 움직입니다. 값으로 0pt가 지정되었다면 개체의 한쪽 끝이 판면의 끝부분과 일치하지만, 양수 길이 값이 입력되었다면 개체는 판면의 바깥쪽으로 이동합니다. 예를 들어,

```
\begin{wraptable}{r}[1cm]{0pt} ... \end{wraptable}
```

를 입력한 경우, 이 환경 안의 표는 오른쪽 여백을 1cm 침범한 형태로 배치됩니다.

`<overhang>` 인자에는 특별한 길이 변수로 `\width`를 사용할 수 있습니다. 이 변수는 환경 안에 있는 개체의 폭을 나타냅니다. 예를 들어 `.5\width`를 인자로 지정하였다면, 이 개체는 폭의 0.5배만큼 판면 바깥쪽으로 이동합니다.

수평 방향 이동 거리의 기본값

길이 변수
→ 277쪽

`<overhang>` 인자의 기본값은 `\wrapoverhang` 길이 변수에 저장되어 있습니다. 초기 설정은 0pt이며, 문서에 삽입하는 개체를 일정한 간격으로 이동시키려면 이 인자를 유용하게 사용할 수 있습니다. 예를 들어, `\wrapoverhang`의 값을 다음과 같이 지정해 둔다면, 개체가 여백글 영역을 차지한 채로 배치될 것입니다.

```
1 \setlength\wrapoverhang\marginparwidth
2 \addtolength\wrapoverhang\marginparsep
```


14.2.2 주의 사항

wrapfloat 환경을 사용할 때에는 크게 두 가지 사항을 주의해야 합니다. 첫째로, 목록 형태의 문단 — list 환경을 사용해 만들어진 문단 — 주변에는 이 환경을 사용하여 개체를 배치할 수 없습니다. 즉, 6.1절에서 설명한 환경과 제7장에서 설명한 정리형 환경에서는 이 패키지의 기능을 사용할 수 없습니다.

둘째로, 이 개체는 유동 개체와는 다른 방식으로 배치되기 때문에, 같은 종류의 개체라도 의도한 순서와 다른 순서로 배치될 수 있습니다. 따라서 컴파일 결과 개체들이 올바른 순서로 배치되었는지 확인하는 것이 좋습니다. 개체들의 순서가 뒤죽박죽이 되었다면 이들이 올바르게 배치되도록 문서의 코드를 수정해 주어야 합니다.

이외의 주의 사항에 대해서는 패키지 안내서[1]를 참조하시기 바랍니다.

14.3 캡션과 개체의 번호

유동 개체 환경이나 \wrapfloat 환경 안에 개체를 입력하였다면, 그 개체에 번호와 캡션을 다는 것은 간단합니다. 환경 안에서 \caption(*) 제어 문자열을 입력하고 캡션의 내용을 인자로 지정하면 됩니다.

```
\caption[⟨short caption⟩]{⟨caption⟩}
\caption*{⟨caption⟩}
```

⟨short caption⟩: 표·그림 목차에서 사용할 캡션

⟨caption⟩: 캡션의 내용

컴파일 과정을 거치면 개체 번호와 캡션이 함께 조판되며, 번호 앞에는 개체의 종류에 따라 ‘Table’과 ‘Figure’ 또는 ‘표’와 ‘그림’이 출력됩니다.

표·그림 목차
→ 56쪽

이 제어 문자열은 장절 표제 명령어들과 마찬가지로 선택 인자를 입력받습니다. 표 목차나 그림 목차에는 다른 내용을 사용하려면 그 내용을 선택 인자로 지정하면 됩니다. 선택 인자를 지정하지 않았다면 필수 인자로 지정한 캡션이 표·그림 목차에 표시됩니다. 한편, 캡션에 번호를 붙이지 않고 싶다면 대신 별표 붙은 \caption*을 사용하면 됩니다. 캡션에 번호가 붙지 않은 개체는 표·그림 목차에도 표시되지 않습니다.

14.3.1 하위 번호 만들기

악셀 소머펠트(Axel Sommerfeldt)의 subcaption 패키지는 개체의 하위 번호를 만드는 기능을 제공합니다. 유동 개체 배치 환경은 하나의 큰 번호를 만들며, 이 환경 안에서

`\subcaptionbox(*)` 제어 문자열을 사용하면 이 번호에 대한 하위 번호가 만들어집니다.

**`\subcaptionbox[⟨short caption⟩]{⟨caption⟩}[⟨width⟩][⟨align⟩]{⟨object⟩}`
`\subcaptionbox*{⟨caption⟩}[⟨width⟩][⟨align⟩]{⟨object⟩}`**

⟨short caption⟩: 표·그림 목차에서 사용할 캡션

⟨caption⟩: 캡션의 내용

⟨width⟩: 개체를 배치할 공간의 폭

⟨align⟩: ⟨width⟩ 인자가 지정된 경우, 공간 안에서 개체를 정렬할 방향

⟨object⟩: 삽입하려는 개체

⟨short caption⟩과 ⟨caption⟩ 인자는 `\caption`의 두 인자와 동일한 역할을 합니다. 명령어의 인자에서 알 수 있듯, 하위 번호가 붙은 개체의 캡션은 별도의 `\caption` 제어 문자열을 사용하지 않고, `\subcaptionbox`의 인자로 지정해 줍니다.

```
1 \begin{figure}
2   \subcaptionbox{첫 번째 그림}
   {\includegraphics{...}}\quad
3   \subcaptionbox{두 번째 그림}
   {\includegraphics{...}}
4   \caption{두 개의 그림을 배치한 예시}
5 \end{figure}
```



(a) 첫 번째 그림



(b) 두 번째 그림

그림 1: 두 개의 그림을 배치한 예시

⟨width⟩ 인자는 개체를 배치할 공간의 폭을 지정합니다. 이 인자를 생략하면 ⟨object⟩ 인자에 입력된 내용의 크기를 바탕으로 폭이 설정됩니다. 캡션은 이 개체의 폭을 기준으로 조판되므로, 개체가 지나치게 작은 경우 ⟨width⟩의 값을 적당히 넓은 값으로 지정해 주어야 좋은 결과를 얻을 수 있습니다.

⟨width⟩ 인자가 지정되었다면 ⟨align⟩ 인자도 지정할 수 있습니다. 공간에 비해 개체의 크기가 작은 경우 개체를 정렬할 방향을 이 인자로 지정해 줍니다. 개체를 왼쪽에 정렬시키려면 l을, 가운데에 정렬시키려면 c를, 오른쪽에 정렬시키려면 r를 입력하면 되며, 일반적인 문단처럼 양쪽 정렬을 원하는 경우에는 s를 입력합니다. 기본값은 c입니다.

대체 인터페이스

subcaption 패키지는 `\subcaptionblock` 제어 문자열 외에도, subcaptionblock 환경이나 subcaptiongroup 환경을 사용해 개체를 배치하는 방법도 제공합니다. 이에 대해서는 패키지 안내서[46]를 참조하시기 바랍니다.

14.4 새로운 종류의 개체 만들기

float 패키지를 통해 표나 그림이 아닌, 새로운 종류의 개체를 만들 수도 있습니다. 예를 들어, 프로그램 코드를 자주 제시하는 문서라면 program을 새로운 개체로 정의할 수 있습니다.

14.4.1 기본 사용 방법

새로운 종류의 개체를 만들 때에는 `\newfloat` 제어 문자열을 사용합니다.

```
\newfloat{<type>}{<placement>}{<extension>}[<parent counter>]
```

`<type>`: 새로운 개체의 종류

`<placement>`: 이 유동 개체 환경의 배치 허용 위치 인자의 기본값

`<extension>`: 이 개체들의 목차를 만들 때 생성되는 보조 파일의 확장자

`<parent counter>`: 개체 번호를 종속시킬 카운터

예를 들어, 다음 코드를 입력하면 새로운 개체로 program을 만들 수 있습니다.

```
\newfloat{program}{tbp}{lop}
```

이 선언을 통해 program 환경과 program* 환경이 만들어지며, 프로그램 코드를 유동 개체로 배치하는 데 이 환경을 사용할 수 있습니다. 또, wrapfloat 환경의 `<type>` 인자로 ‘program’을 지정할 수 있게 됩니다.

`<placement>` 인자에는 표 14.1의 h, t, b, p 중 원하는 것들을 입력합니다. 위의 예시에서는 표준 클래스에서 `\table`과 `\figure` 환경의 기본값인 ‘tbp’를 입력하였습니다.

`<extension>` 인자에는 표 목차나 그림 목차처럼, 개체 목차를 만들 때 생성되는 보조 파일의 확장자를 입력합니다. 표 목차의 확장자는 `lot`(list of tables), 그림 목차의 확장자는 `lof`(list of figures)이므로, 이와 비슷한 형태로 입력하면 됩니다. 단, 다음 파일들은 L^AT_EX에서 사용하는 확장자이므로 이 인자로 지정해서는 안 됩니다.

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| aux | bcf | bib | bbl | blg | bst | cls | dtx | dvi |
| fmt | idx | ilg | ind | ins | ist | lof | log | lot |
| ltx | out | pdf | ptc | sty | tex | toc | xdv | xdy |

위의 예시에서는 ‘list of programs’라는 의미로 ‘lop’를 사용하였습니다.

카운터
→ 269쪽

선택 인자인 `<parent counter>`에는 개체 번호를 종속시킬 카운터를 입력합니다. 이 인자를 지정하지 않는 경우, 개체 번호는 다른 카운터에 영향을 받지 않습니다. 표준 클래스의 관행을 따르려는 경우, article 클래스 문서에서는 이 인자를 생략하고, book

및 report 클래스에서는 이 인자로 chapter를 지정하면 됩니다.

개체의 이름

새로운 종류의 개체를 만들었다면, `\floatname` 제어 문자열을 사용하여 그 개체에 이름을 붙여 주어야 합니다. 이 이름은 캡션을 달 때 개체 번호 앞에 표시됩니다.

`\floatname{<type>}{<name>}`

<type>: 이름을 붙이려는 개체의 종류

<name>: 이 개체에 붙일 이름

예를 들어, 영어 문서에서는 `\floatname{program}{Program}`이라고 입력하면 되며, 한국어 문서에서는 `\floatname{program}{프로그램}`이라고 입력하면 됩니다.

새로 만들어진 개체에 이름이 지정되어 있지 않은 경우, `\newfloat`의 *<type>* 인자에 지정된 내용이 개체의 이름으로 사용됩니다.

14.4.2 개체 목차 만들기

새로운 종류의 개체를 만들었다면, 표·그림 목차처럼 해당 종류의 개체들의 캡션을 모아 개체 목차를 출력할 수 있습니다. 이때에는 `\listof` 제어 문자열을 사용합니다.

`\listof{<type>}{<title>}`

<type>: 목차를 출력하려는 개체의 종류

<title>: 개체 목차의 제목

예를 들어, `\listof{program}{프로그램 목차}`라고 입력하면 program 개체들의 캡션을 모은 목차가 만들어집니다.

14.4.3 스타일 기능

이 패키지는 유동 개체의 종류별로 스타일을 지정하는 기능을 제공합니다.¹ 표준 클래스의 기본 환경처럼 개체 주변에 특별한 서식을 적용하지 않을 수도 있고, 위아래에 패션을 그리거나 상자를 칠 수도 있습니다.

유동 개체의 스타일에 대한 설명은 패키지 안내서[28]를 참조하시기 바랍니다.

¹ 이 기능은 제7장에서 설명한 수학적 문단의 스타일과 비슷한 기능입니다. 스타일 기능을 사용하는 방법도 `amsmath` 패키지와 `float` 패키지가 비슷합니다.

V

참조와 인용

제 V편에서는 참조 및 인용과 관련된 기능을 알아봅니다. 다른 장과 절, 표와 그림의 상호 참조, 인용한 문헌의 목록 생성 등의 내용을 다룹니다. 또, 생성된 PDF 파일에서 문서를 편리하게 탐색할 수 있도록 하이퍼링크와 책갈피를 넣는 방법을 알아봅니다.

제15장 상호 참조 · 207

제16장 참고 문헌 · 229

제17장 PDF 파일의 링크와 속성 · 247

제 15 장

상호 참조

\LaTeX 은 상호 참조 기능을 제공합니다. 이 기능을 사용하면 다른 장이나 절, 그림, 표, 수식 등을 참조할 수 있습니다. 또, 상호 참조 시 생성되는 문구를 자동화할 수도 있습니다. 이번 장에서는 \LaTeX 에서 여러 대상을 상호 참조하는 방법과 함께 사용할 수 있는 패키지를 알아봅니다.

15.1 \LaTeX 의 상호 참조 기능

상호 참조의 기본 메커니즘은 참조하려는 대상에 가상의 이름(식별자)을 붙이는 것입니다. 이때 사용하는 제어 문자열은 `\label`입니다.

```
\label{<identifier>}
```

`<identifier>`: 참조할 대상에 붙일 식별자

이때, 각 대상에 붙이는 식별자는 유일해야 합니다. 식별자에는 대부분의 문자를 사용할 수 있지만, 쉼표와 공백 문자, 대괄호는 사용하지 않는 것이 좋습니다.

참조하려는 대상에 이름을 붙였다면, 그 대상을 참조할 곳에서 `\ref`나 `\pageref` 제어 문자열을 사용합니다.

```
\ref{<identifier>}
```

```
\pageref{<identifier>}
```

`<identifier>`: 참조하려는 대상에 붙은 식별자

전자는 대상에 붙은 번호(장 번호, 수식 번호 등)를 출력하며, 후자는 그에 대응하는 `\label`이 있는 위치(페이지 번호)를 출력합니다.

예를 들어, 이 책에서는 이번 장을 시작할 때 다음 코드를 입력하였습니다.

```
\chapter{상호 참조} \label{chap:crossref}
```

다른 곳에서 `\ref{chap:crossref}`를 입력하면 이번 장의 번호인 '15'가 출력되고, `\pageref{chap:crossref}`를 입력하면 이번 장이 시작되는 페이지 번호인 '207'이 출력됩니다.

```
1 \section{다항식의 연산}
  \label{sec:polynomial}
2 이번 장에서는 다항식의 사칙 연산과 인수 분해, 나머지
  정리에 대해 알아봅니다. ...
3
4 \section{방정식과 부등식}
  \label{sec:equations}
5 \ref{sec:polynomial}절에서 알아본 다항식의 연
  산을 바탕으로 방정식과 부등식의 해를 구하는 방법을
  알아봅니다. ...
```

1 다항식의 연산

이번 장에서는 다항식의 사칙 연산과 인수 분해, 나머지 정리에 대해 알아봅니다. ...

2 방정식과 부등식

1절에서 알아본 다항식의 연산을 바탕으로 방정식과 부등식의 해를 구하는 방법을 알아봅니다. ...

설명의 편의를 위해, 이제부터 이번 장에서 예시로 사용하는 식별자는 '번호 종류:번호'의 꼴로 나타내겠습니다. 예를 들어, `sec:3`은 3절을, `fig:2.1`은 그림 2.1을, `thm:1.1`은 정리 1.1을 가리키는 식별자입니다. 실제 문서를 작성할 때에는 참조 대상의 번호를 알지 못하므로, 식별자에는 대상의 이름이나 내용 등을 적는 것이 좋습니다.

식별자를 붙이기 위한 명령어의 위치

`\label`은 입력된 위치에 따라 식별자를 붙이는 대상이 달라집니다. 이름을 붙일 대상에 따라 명령어를 다음의 위치에 입력하면 됩니다.

- 장과 절의 표제를 참조할 때에는 `\chapter`, `\section` 등의 뒤에 입력합니다.
- 수식을 참조할 때에는 `equation`, `multline`, `align`, `gather` 등의 환경 안에 입력합니다. `align`이나 `gather` 환경처럼 수식 각 줄에 번호가 붙는 경우, 각 줄을 따로 참조할 수 있습니다.
- 표나 그림을 참조할 때에는 `table`이나 `figure` 환경 안의 `\caption` 제어 문자열 뒤에 입력합니다.
- 번호 목록의 항목을 참조할 때에는 `\item`의 뒤에 입력합니다. 구분점 목록이나 사전식 목록의 항목은 참조할 수 없습니다.

- `\newtheorem`을 사용해 만든 환경 안에 `\label`을 입력하면, 그 정의나 정리 등이 작성된 수학적 문단을 참조할 수 있습니다. `\newtheorem*`로 정의된 번호 없는 문단은 참조할 수 없습니다.

카운터
→ 269쪽

이외에도, 직접 정의한 카운터에 대해 `\refstepcounter`를 사용한 후 `\label`을 입력하면 그 카운터의 번호를 참조할 수 있습니다.¹

예를 들어, 다음과 같이 정의된 `\exercise` 제어 문자열을 생각해 봅시다.

```
1 \newcounter{exercise}
2 \newcommand\exercise{\par\medskip\refstepcounter{exercise}
   \noindent\textbf{연습문제 \theexercise}\quad}
```

이 제어 문자열은 `exercise` 카운터의 값을 1 증가시킨 뒤 그 값을 출력합니다. 이 제어 문자열의 정의에 `\refstepcounter`가 포함되어 있으므로, 문서를 작성할 때 `\exercise` 뒤에 `\label`을 입력하면 이 명령어는 이 연습문제를 참조하게 됩니다.

컴파일 시의 주의 사항

상호 참조 기능을 사용하는 경우, 컴파일을 두 번 해야 올바른 결과물을 얻습니다. 첫 번째 컴파일 시에는 `\label`의 정보가 보조 파일에 저장되고, 이 내용을 바탕으로 두 번째 컴파일 시 `\ref`나 `\pageref`를 입력한 자리에 올바른 값이 출력됩니다. 참조한 대상의 번호·위치가 변경되었거나 컴파일 후

LaTeX Warning: Label(s) may have changed. Rerun to get cross-references right.

라는 메시지가 표시되면 컴파일을 두 번 하여 올바른 번호가 나타나도록 해야 합니다.

보조 파일에 정보가 없는 식별자가 `\ref`나 `\pageref`에 사용되었다면 — 즉, 새로운 식별자를 만든 후 컴파일을 한 번만 했거나 `\label`로 정의되지 않은 식별자를 `\ref`나 `\pageref`의 인자로 지정했다면 — 컴파일 과정에서 경고 메시지가 출력되며, 참조하려던 대상의 번호 대신 ‘??’가 출력됩니다.

참고 | Overleaf를 사용하거나, `latexmk`를 사용하여 컴파일 하는 경우에는 참조 정보가 변경된 후 컴파일을 한 번만 하더라도 올바른 결과물을 얻습니다.

¹ 자세히 설명하자면, `\label`은 가장 최근에 `\refstepcounter` 제어 문자열이 사용된 카운터의 값을 참조합니다. 위에서 설명한 장절 표제 명령어, 수식 환경, 유동 개체 환경의 `\caption` 제어 문자열, 목록 환경의 `\item` 제어 문자열, 정리형 환경은 모두 `\refstepcounter` 제어 문자열을 내부적으로 사용하므로 `\label`을 사용했을 때 그 번호가 참조되는 것입니다.

15.1.1 자동 조사

한국어의 조사는 ‘이/가’, ‘을/를’, ‘로/으로’ 등과 같이 앞에 오는 글자의 종성 유무에 따라 달라집니다. 참조 기능을 사용하면 `\ref`나 `\pageref`가 어떤 숫자를 출력할지 알 수 없기 때문에, 이 뒤의 조사를 올바르게 입력하기 위해서는 계산된 숫자를 확인한 뒤 조사를 수정해야 합니다.

kotex 패키지는 이러한 불편을 없애기 위해 **자동 조사** 기능을 제공합니다. 변화하는 조사의 앞에 역슬래시를 붙이면 되며, 이 앞에 오는 문자를 읽는 방법에 따라 올바른 조사가 자동으로 선택됩니다. 이 기능을 사용할 수 있는 조사는 다음과 같습니다.

```
\은   \는   \이   \가   \을   \를   \와   \과
\로   \으로 \라   \이라
```

다음 예시에서 `table:varioreftext`는 표 15.1을, `table:kotexvarioreftext`는 표 15.2를 참조하는 식별자입니다. 자동 조사 기능을 사용하면 이처럼 참조된 대상의 번호와 어울리는 조사를 얻을 수 있습니다.

1 영어 문구에 대한 설명은

표~`\ref{table:varioreftext}`\를, 한국어 문
구에 대한 설명은
표~`\ref{table:kotexvarioreftext}`\을 참조
하시기 바랍니다.

영어 문구에 대한 설명은 표 15.1을, 한국어
문구에 대한 설명은 표 15.2를 참조하시기
바랍니다.

15.1.2 번호의 출력 형식 지정

`\labelformat` 제어 문자열을 사용하면 참조된 대상의 번호가 출력되는 형식을 변경할 수 있습니다.

```
\labelformat{<counter>}{<format>}
```

<counter>: 출력 형식을 지정할 카운터의 이름(번호의 종류)

<format>: 원하는 출력 형식. 번호가 들어갈 자리에는 ‘#1’을 입력합니다.

다만, 페이지 번호를 저장하는 `page` 카운터는 형식을 지정하더라도 `\pageref`의 출력 결과에 영향을 미치지 않습니다.

예를 들어, 한국어 문서에서는 다음을 입력해 번호의 형식을 지정할 수 있습니다.

```

1 \labelformat{part}{제#1편}
2 \labelformat{chapter}{제#1장}
3 \labelformat{section}{#1절}
4 \labelformat{subsection}{#1절}
5 \labelformat{table}{표~#1}
6 \labelformat{figure}{그림~#1}

```

이는 다음의 결과를 얻게 합니다.

1 앞서 설명한 내용은 \ref{table:3.1}\와
 \ref{fig:3.3}에서도 확인할 수 있습니다.

앞서 설명한 내용은 표 3.1과 그림 3.3에서
 도 확인할 수 있습니다.

참고 | 2019년 9월 이전 버전의 L^AT_EX을 사용하는 경우에는 `varioref` 패키지를 불러와야 이 기능을 사용할 수 있습니다.

첫 글자의 대문자화

대소문자를 구분하는 언어에서는 문장의 첫 글자를 대문자로 작성합니다. 이러한 경우를 위한 제어 문자열로 `\Ref`가 정의되어 있습니다. `\ref` 대신 `\Ref`를 사용하면 `<format>`에 입력된 내용 중 첫 번째 글자가 대문자로 조판됩니다. (한국어를 포함해, 대소문자의 구분이 없는 언어에서는 `\ref`든 `\Ref`든 같은 결과를 출력합니다.)

예를 들어, 전처리부에

```

1 \labelformat{table}{table~#1}
2 \labelformat{figure}{figure~#1}

```

가 입력되어 있다면 다음의 결과를 얻습니다.

1 \Ref{fig:3} illustrates the
 aforementioned process.
 2 \Ref{table:2} describes possible
 options related to the process.

Figure 3 illustrates the aforementioned process. Table 2 describes possible options related to the process.

`<format>` 인자의 첫 글자를 올바르게 대문자화하기 위해, 첫 글자로 확장 로마자가 사용될 때에는 입력에 주의해야 합니다. L^AT_EX의 명령어를 사용해 확장 로마자를 입력하는 경우(예: ‘é’를 ‘\e’로 입력하는 것)에는 해당 문자를 중괄호로 감싸 주어야 합니다. 레거시 엔진에서는 문자를 직접 입력하더라도 중괄호로 감싸야 합니다.

예를 들어, 튀르키예어에서 그림(figure)을 뜻하는 단어는 ‘şekil’이므로, 그림 번호의 형식을 지정하려면 다음 중 하나의 방법을 사용해야 합니다.

```

1 \labelformat{figure}{\c s}ekil~#1}    % 모든 엔진에서
2 \labelformat{figure}{\s}ekil~#1}      % 레거시 엔진에서
3 \labelformat{figure}{\s}ekil~#1}      % 유니코드 엔진에서

```

이렇게 하면 `\Ref{fig:3.7}`은 ‘Şekil 3.7’으로 올바르게 조판됩니다. 필요한 중괄호를 생략하는 경우에는 컴파일 과정에서 오류가 발생합니다.

참고 | 레거시 엔진에서 `kotex-utf` 패키지를 불러온 경우에는 `\Ref` 제어 문자열이 정상적으로 동작하지 않습니다. 레거시 엔진에서 라벨의 대문자화 기능과 한국어 조판 기능을 동시에 사용해야 한다면 `ckutf80`이나 `cjk-ko` 패키지를 대신 사용하시기 바랍니다.

15.2 참조 문구를 변경해 주는 패키지

LaTeX에서 기본으로 제공하는 `\ref`와 `\pageref`는 단순히 숫자만을 출력합니다. 앞서 소개한 `\labelformat`을 사용하더라도 필요한 문구를 직접 지정해 주어야 합니다. 패키지를 사용하면 참조한 대상이 있던 위치에 따라 ‘앞 페이지의’, ‘다음 페이지의’ 등의 추가 정보를 출력하거나, 참조 대상의 종류에 따라 ‘표 1’, ‘3절’ 등의 이름을 언어에 맞게 자동으로 붙여 주고, 여러 대상을 명령어 하나로 간편하게 참조할 수 있습니다. 이번 장에서는 이러한 기능을 제공하는 패키지를 소개합니다.

15.2.1 amsmath 패키지: 수식 참조

`amsmath` 패키지는 수식을 참조할 때 사용하는 `\eqref` 제어 문자열을 정의합니다. 이는 `\ref`와 동일하게 사용하며, 수식의 번호를 소괄호 안에 조판해 줍니다. 이렇게 입력한 수식 번호는 앞뒤 내용의 글꼴과 관계없이 기본 글꼴로 조판됩니다.

```

1 다음 등식 \eqref{eq:cauchyriemann}\을 ‘코
  시-리만 방정식’이라고 부른다.
2 \begin{equation}
3   \frac{\partial u}{\partial x} =
   \frac{\partial v}{\partial y}, \quad
   \frac{\partial u}{\partial y} =
   -\frac{\partial v}{\partial x}
   \label{eq:cauchyriemann}
4 \end{equation}

```

다음 등식 (5)를 ‘코시-리만 방정식’이라고 부른다.

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \quad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} \quad (5)$$

수식을 참조할 때에는 자동으로 계산된 번호가 붙은 수식뿐 아니라, `\tag` 제어 문자열을 사용해 임의의 기호를 붙인 수식도 참조할 수 있습니다. 이때에는 `\ref`나

`\eqref`로 참조했을 때 수식의 번호 대신 지정했던 기호가 출력됩니다.

```

1 단위원에서의 삼각형을 생각하면 다음 등식을 쉽게 얻을 수 있다.
2 \begin{equation}
3   \cos^2 x + \sin^2 x = 1 \label{eq:trig}
4 \end{equation}
5 식~\eqref{eq:trig}의 양변을  $(\cos^2 x)$ 로 나누면 아래의 식~\eqref{eq:trig}을 얻는다.
6 \begin{equation}
7   1 + \tan^2 x = \sec^2 x
8   \tag{\theequation'}
9   \label{eq:trig}
10 \end{equation}

```

단위원에서의 삼각형을 생각하면 다음 등식을 쉽게 얻을 수 있다.

$$\cos^2 x + \sin^2 x = 1 \quad (2)$$

식 (2)의 양변을 $\cos^2 x$ 로 나누면 아래의 식 (2')을 얻는다.

$$1 + \tan^2 x = \sec^2 x \quad (2')$$

참고 | `\labelformat`과 `\eqref`는 서로 기능이 충돌하므로, `\labelformat`으로 equation 카운터의 참조 형식을 변경했다면 `eqref`가 제대로 작동하지 않습니다.

여러 줄짜리 수식의 참조

`amsmath` 패키지에서 정의하는 `align` 환경이나 `gather` 환경 등은 수식을 여러 줄로 조판하며, 각 줄마다 번호를 따로 붙일 수 있습니다. 이때에는 각 줄에 `\label`을 입력함으로써 수식을 줄별로 참조할 수 있습니다.

```

1 쌍곡선함수의 정의는 다음과 같다.
2 \begin{align}
3   \cosh x &= \frac{e^x + e^{-x}}{2} \\
4   \sinh x &= \frac{e^x - e^{-x}}{2} \\
5 \end{align}
6 위의 두 식 \eqref{eq:cosh}와 \eqref{eq:sinh}를 잘 정리하면 아래와 같이 역함수를 명시적인 형태로 나타낼 수 있다. ...

```

쌍곡선함수의 정의는 다음과 같다.

$$\cosh x = \frac{e^x + e^{-x}}{2} \quad (3)$$

$$\sinh x = \frac{e^x - e^{-x}}{2} \quad (4)$$

위의 두 식 (3)과 (4)를 잘 정리하면 아래와 같이 역함수를 명시적인 형태로 나타낼 수 있다. ...

`split` 환경 등은 수식 모드 안에서 사용하는 하위 환경이므로 — 즉 환경 안에서 수식의 번호가 부여되지 않으므로 — 참조 기능을 사용할 수 없습니다. 이 환경이 사용된 바깥 수식 환경에 `\label`을 입력해야 합니다.

15.2.2 varioref 패키지: 참조 대상의 위치에 따른 문구

대상을 참조할 때에는 번호 뿐 아니라 그 위치까지 함께 설명해 주면 독자가 참조된 대상을 찾아가는 데 도움이 됩니다. 가장 간단한 방법은 다음과 같이 `\fullref` 제어 문자열을 정의하여 사용하는 것입니다.

```
\newcommand*\fullref[1]{\pageref{#1}쪽의 \ref{#1}}
```

이 제어 문자열을 정의하고 `\labelformat`을 적절히 사용하면 다음 결과를 얻습니다.

`\fullref{sec:5.1}`에서 설명했듯, ...

221쪽의 5.1절에서 설명했듯, ...

하지만, 참조 대상이 같은 페이지나 앞뒤 페이지에 있을 때에는 단순히 페이지 번호를 출력하는 대신 ‘현재 페이지’, ‘다음 쪽’ 등의 다른 문구를 사용하는 것이 자연스럽습니다. 프랭크 미텔바흐(Frank Mittelbach)가 작성한 `varioref` 패키지는 이러한 기능을 제공합니다. 이 패키지를 불러올 때에는 다음과 같이 `nospace` 옵션을 지정해야 합니다.

```
\usepackage[nospace]{varioref}
```

참고 | `nospace` 옵션은 이전 버전의 패키지를 사용하는 문서와의 호환을 위해 만들어진 것으로, 새로운 문서를 작성할 때에는 항상 이 옵션을 사용하는 것이 좋습니다. 여기에서는 이 옵션을 사용한다고 가정하고 설명할 것입니다. `nospace` 옵션에 대한 자세한 내용은 패키지 안내서[30]를 참조하시기 바랍니다.

대상 참조하기

패키지를 불러왔다면, 대상을 참조할 때 `\ref` 대신 `\vref` 제어 문자열을, `\Ref` 대신 `\Vref` 제어 문자열을, `\pageref` 대신 `\vpageref` 제어 문자열을 사용합니다.

```
\vref[⟨fallback text⟩]{⟨identifier⟩}
\Vref[⟨fallback text⟩]{⟨identifier⟩}
\vpageref[⟨fallback text⟩]{⟨identifier⟩}
```

⟨fallback text⟩: 참조 대상이 같은 페이지에 있을 때 출력할 내용

⟨identifier⟩: 참조할 대상의 식별자

참조한 대상이 현재 페이지에 있을 때에는 페이지 정보가 특별히 출력되지는 않지만, *⟨fallback text⟩* 인자를 지정해 두면 이 내용이 대신 출력됩니다. 한 페이지에 참조할 대상이 여럿 있는 경우 ‘above’나 ‘below’ 등을 선택 인자로 지정하면 적절합니다. 예를 들어, `thm:4.4`가 현재 페이지에 있는 식별자라면 다음의 결과를 얻습니다.

표 15.1 | varioref 패키지: 참조 대상의 위치에 따른 문구

| 참조 대상의 위치 | <code>\vref{#1}</code> 의 문구 | <code>\vpageref{#1}</code> 의 문구 |
|-----------|---|---|
| 같은 페이지 | <code>\ref{#1}</code> | (특별한 문구가 출력되지 않음) |
| 한 페이지 앞 | <code>\ref{#1}</code> on the preceding page
<code>\ref{#1}</code> on the previous page | on the preceding page
on the previous page |
| 한 페이지 뒤 | <code>\ref{#1}</code> on the following page
<code>\ref{#1}</code> on the next page | on the following page
on the next page |
| 왼쪽 페이지 | <code>\ref{#1}</code> on the preceding page
<code>\ref{#1}</code> on the facing page | on the preceding page
on the facing page |
| 오른쪽 페이지 | <code>\ref{#1}</code> on the next page
<code>\ref{#1}</code> on the facing page | on the next page
on the facing page |
| 이외의 위치 | <code>\ref{#1}</code> on page <code>\pageref{#1}</code> | on page <code>\pageref{#1}</code> |

| | |
|--|---|
| <code>\</code> According to Theorem~ <code>\ref[above]</code>
<code>{thm:4.4}</code> , we get the following result. | According to Theorem 4.4 above, we get
the following result. |
|--|---|

Theorem 4.4가 다음 페이지에 있었다면 위의 코드는

According to Theorem 4.4 on the next page, we get the following result.

를 출력할 것입니다.

참조 명령어가 사용된 곳에 따라 출력되는 영어 문구를 표 15.1에 정리하였습니다. 이때 같은 표현이 반복적으로 사용되는 것을 피하기 위해 두 가지 표현이 돌아가며 사용됩니다.

문서를 양면으로 편집하는 경우, 참조한 대상이 마주보는 페이지에 있다면 마주보지 않는 앞뒤 페이지에 있을 때와는 다른 문구가 사용됩니다. 예를 들어, sec:6.2가 214쪽에 있는 식별자라면 215쪽에서는 문서의 편집 방식에 따라 다음의 결과를 얻습니다.

| | |
|--|---------------------------------------|
| <code>\</code> % One-sided document
<code>\</code> See Section~ <code>\vref{sec:6.2}</code> . | See Section 6.2 on the previous page. |
| <code>\</code> % Two-sided document
<code>\</code> See Section~ <code>\vref{sec:6.2}</code> . | See Section 6.2 on the facing page. |

위의 코드가 213쪽에 있었다면 편집 방식에 관계없이 ‘See Section 6.2 on the next page.’가 출력되었을 것입니다.

여러 대상 참조하기

여러 대상을 한 번에 참조하여 범위로 나타내려면 `\vrefrange`와 `\vpagerefrange` 제어 문자열을 사용합니다. 참조할 대상의 식별자를 두 개 입력해야 한다는 점만 제외하면, `\vref`나 `\vpageref`와 동일하게 사용합니다. (대문자 버전인 `\Vrefrange`는 존재하지 않습니다.)

```
\vrefrange[⟨fallback text⟩]{⟨first item⟩}{⟨last item⟩}
\vpagerefrange[⟨fallback text⟩]{⟨first item⟩}{⟨last item⟩}
```

⟨fallback text⟩: 참조 대상이 같은 페이지에 있을 때 출력할 내용

⟨first item⟩: 참조하려는 첫 번째 항목의 식별자

⟨last item⟩: 참조하려는 마지막 항목의 식별자

두 대상이 같은 페이지에 있다면 페이지 번호 부분은 `\vpageref`와 동일한 결과를 출력하며, 그렇지 않은 경우에는 첫째 대상이 있는 페이지 번호부터 마지막 대상이 있는 페이지 번호까지의 범위가 출력됩니다.

예를 들어, `lem:4.5`와 `lem:4.6`이 각각 107쪽과 108쪽에 있다면 다음을 얻습니다.

```
1 For the proof, see
  Lemmas~\vrefrange{lem:4.5}{lem:4.6}.
```

For the proof, see Lemmas 4.5 to 4.6 on
pages 107–108.

만약 두 보조정리가 같은 페이지에 있고 위 코드가 한 페이지 앞에 있었다면 ‘Lemmas 4.5 to 4.6 on the next page’가 출력되었을 것입니다.

언어 설정 및 문구 사용자화

패키지를 불러올 때 옵션으로 문서의 언어를 입력하면 그 언어에 맞는 문구가 사용됩니다(언어를 지정하지 않으면 영어가 사용됩니다). `babel` 패키지를 사용한다면 이 패키지를 불러올 때 사용한 언어 옵션을 `varioref` 패키지의 옵션으로 지정해 주면 됩니다. 예를 들어, 프랑스어 문서라면 다음과 같이 입력합니다.

```
1 \usepackage[french]{babel}
2 \usepackage[nospace, french]{varioref}
```

여러 언어를 사용하는 문서라면 `babel` 패키지에서 옵션을 지정할 때처럼 주 언어를 마지막 옵션으로 지정해 줍니다. `babel` 패키지의 언어 변경 기능과 연동되어 자동으로 적절한 언어의 문구가 사용됩니다. 이때 `babel` 패키지를 더 먼저 불러와야 합니다. 위의 코드가 전처리부에 있다면 다음 예시를 조판할 수 있습니다.

표 15.2 | kotex-varioref 패키지: 참조 대상의 위치에 따른 문구

| 참조 대상의 위치 | <code>\vref{#1}</code> 의 문구 | <code>\vpageref{#1}</code> 의 문구 |
|-----------|--|---------------------------------|
| 같은 페이지 | 현재 페이지(의) <code>\ref{#1}</code> | 현재 페이지 |
| 한 페이지 앞 | 앞 페이지(의) <code>\ref{#1}</code> | 앞 페이지 |
| 한 페이지 뒤 | 다음 페이지(의) <code>\ref{#1}</code> | 다음 페이지 |
| 이외의 위치 | <code>\pageref{#1}</code> 페이지(의) <code>\ref{#1}</code> | <code>\pageref{#1}</code> 페이지 |

| | |
|---|---|
| <code>\Le chapitre~\vref{ch:1} donnera une introduction.</code> | Le chapitre 1 à la page 1 donnera une introduction. |
|---|---|

한편, 이 패키지는 한국어를 지원하지 않으므로, 한국어 문서에서는 KTUG가 제공하는 kotex-varioref 패키지를 함께 불러와 사용해야 합니다. 전처리부에 다음 코드를 입력해 두 패키지를 불러옵니다.

```
1 \usepackage[nospace]{varioref}
2 \usepackage{kotex-varioref}
```

한국어는 서양 언어와 문장 구조가 다르므로, 이 패키지는 자연스러운 문장을 출력하기 위해 `\labelformat`으로 편·장·절, 그림·표, 각주 번호의 출력 형식을 변경합니다.

kotex-varioref 패키지가 사용하는 표현은 표 15.2에서 확인할 수 있습니다. 조사 ‘의’는 `\UITrue` 선언을 하면 출력되며, `\UIfalse` 선언을 하면 출력되지 않습니다. 후자가 기본값입니다.

| | |
|--|--|
| 1 <code>\vref{sec:1.2}</code> 의 내용을 참고하여, ... <code>\</code>
2 <code>\UITrue</code> ... 내용을 <code>\vref{table:1.4}</code> 에 정리 하였다. | 앞 페이지 1.2절의 내용을 참고하여, ...
... 내용을 다음 페이지의 표 1.4에 정리하였다. |
|--|--|

다만, varioref 패키지에서 기본으로 지원하는 언어와는 달리, 이 패키지는 양면 편집 시 ‘마주보는 페이지’ 등의 문구를 지원하지 않으며, 한 가지 고정된 표현만 사용됩니다. 이 점이 마음에 들지 않는다면 kotex-varioref 패키지 대신 다음의 방법을 사용할 수도 있습니다.

먼저, 예시 문서 15.1을 varioref-korean.sty 파일로 저장합니다. 이후, 이 파일과 작업 파일을 같은 폴더에 둔 후 전처리부에

```
\usepackage[nospace, setlabel]{varioref-korean}
```

| 예시 문서 15.1 | varioref-korean.sty

```

1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{varioref-korean}[2024/03/31 Korean localisation of varioref
  package]
3
4 \newif\if@setlabelformat \@setlabelformatfalse
5 \DeclareOption{setlabel}{\@setlabelformattrue}
6 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{varioref}}
7 \ProcessOptions\relax
8 \RequirePackage{varioref}
9
10 \newif\ifUI \UItrue
11 \def\kvr@ui{\ifUI 의\else\fi}
12 \def\kvr@eitneun{\ifUI 에 있는\else\fi}
13
14 \def\reftextfaceafter{\reftextvario{오른쪽 페이지}{옆 쪽}\kvr@ui}
15 \def\reftextfacebefore{\reftextvario{왼쪽 페이지}{옆 쪽}\kvr@ui}
16 \def\reftextafter{\reftextvario{다음 쪽}{뒤쪽}\kvr@ui}
17 \def\reftextbefore{\reftextvario{이전 쪽}{앞쪽}\kvr@ui}
18 \def\reftextcurrent{현재 페이지\kvr@ui}
19 \def\reftextfaraway#1{\pageref{#1}쪽\kvr@ui}
20 \def\reftextpagerange#1#2{\pageref{#1}~{\pageref{#2}쪽\kvr@eitneun}
21 \def\reftextlabelrange#1#2{\ref{#1}~{\ref{#2}}}
22
23 \def\vrefformat#1#2{\vpageref[#1]{#2} \ref{#2}}
24 \def\Vrefformat#1#2{\vpageref[#1]{#2} \ref{#2}}
25 \def\fullrefformat#1{\reftextfaraway{#1} \ref{#1}}
26 \def\vreffrangeformat#1#2#3{\vpagerefrange[#1]{#2}{#3}
  \reftextlabelrange{#2}{#3}}
27
28 \if@setlabelformat
29   \labelformat{part}{제#1편}
30   \labelformat{chapter}{제#1장}
31   \labelformat{section}{#1절}
32   \labelformat{figure}{그림~#1}
33   \labelformat{table}{표~#1}
34   \labelformat{footnote}{각주~#1}
35 \fi
36
37 \endinput

```

을 입력하여 `varioref-korean` 패키지를 불러오면 됩니다. 여기에서 `setlabel` 옵션은 `\labelformat`을 사용하여 번호의 출력 형식을 바꾸어 주는 역할을 합니다. 다음 절에서 설명할 `cleveref` 패키지는 `\labelformat`과 충돌을 일으키므로 해당 패키지와 이 패키지를 함께 사용할 때에는 이 옵션을 사용하지 않음으로써 충돌을 피할 수 있습니다. `nospace` 등 이 패키지에 지정된 다른 옵션은 `varioref` 패키지에 그대로 전달됩니다.

`varioref` 패키지는 참조 문구를 출력할 때 쓰이는 각종 표현을 사용자화 할 수 있습니다. 예시 문서 15.1의 코드도 사용자화 기능을 이용한 것입니다. 하지만, 이 책에서는 이 패키지의 사용자화 기능에 대해서는 자세히 설명하지 않겠습니다. 자세한 내용은 패키지의 안내서[30]를 참조하시기 바랍니다.

15.2.3 cleveref 패키지: 참조 대상의 종류에 따른 문구

토비 쿠빗(Toby Cubitt)이 작성한 `cleveref` 패키지는 참조된 대상의 종류가 무엇인지에 따라 그 앞에 ‘chapter’, ‘section’, ‘fig.’, ‘table’, ‘eq.’ 등의 단어를 붙여 줍니다. 여러 항목을 한 번에 참조하는 경우 단어의 축약형·단수형·복수형을 적절하게 사용하는 등, 이 패키지는 앞서 소개한 `\labelformat`보다 더 다양한 기능을 제공합니다.

이 패키지에서 대상을 참조할 때에는 `\cref`와 `\Cref` 제어 문자열을 사용합니다. 패키지를 불러오기만 하면 별도의 설정을 하지 않더라도, 참조된 대상의 이름이 번호 앞에 붙은 채로 출력됩니다. 같은 방법으로, 페이지 번호를 참조할 때에는 `\cpageref`와 `\Cpageref` 제어 문자열을 사용합니다. 참조된 대상의 페이지 번호 앞에 ‘page’ 또는 ‘Page’가 붙은 채로 출력됩니다.

```
\Cref{eq:8.3} is useful when
integrating complex-valued functions,
and \cref{eq:8.4} is a special case of
\cref{eq:8.3}. See \cref{sec:8.4} on
\cpageref{sec:8.4} for the proof.
```

Equation 8.3 is useful when integrating complex-valued functions, and eq. 8.4 is a special case of eq. 8.3. See section 8.4 on page 231 for the proof.

영어 문장은 첫째 단어로 약어를 사용하지 않으므로, 위 예시에서 `\Cref{eq:8.3}`은 번호 앞에 ‘Equation’을, `\cref{eq:8.3}`은 번호 앞에 ‘eq.’를 붙였습니다.

참고 | `\labelformat`으로 참조 대상의 번호 출력 형식을 변경한 경우 이 패키지의 명령어는 정상적으로 작동하지 않습니다.

여러 대상 참조하기

`varioref` 패키지와 마찬가지로, `cleveref` 패키지도 여러 대상을 참조하는 기능을 제공합니다. `\crefrange`와 `\Crefrange` 제어 문자열은 두 식별자를 인자로 입력받고, 이들이 가리키는 대상의 번호를 범위의 형태로 조판해 줍니다. 두 식별자가 가리키는 대상은 종류가 같아야 합니다. 마찬가지로, `\cpagerefrange`와 `\Cpagerefrange` 제어 문자열도 두 식별자를 인자로 입력받고, 이들이 가리키는 대상의 페이지 번호를 범위의 형태로 조판해 줍니다.

`1 On \crefrange{ch:4}{ch:6}, we will discuss about the solutions of the first- and the second-order ODEs.`

On chapters 4 to 6, we will discuss about the solutions of the first- and the second-order ODEs.

한편, `\cref`나 `\Cref`, `\cpageref`나 `\Cpageref`를 사용해 여러 대상을 참조할 수도 있습니다. 이때에는 종류가 다른 대상도 함께 참조할 수 있습니다. 여러 식별자를 쉼표로 구분하여 인자로 지정하면(쉼표 뒤에 공백 문자를 입력하면 식별자의 일부로 간주됩니다), 같은 종류의 대상끼리 모아지고 번호가 오름차순으로 정렬되어 최적의 참조 정보가 출력됩니다.

`1 Refer to \cref{sec:1,sec:2,fig:3,table:1,sec:3,fig:1}.`

Refer to sections 1 to 3, figs. 1 and 3, and table 3.

수학적 문단의 참조

수학적 문단
→ 81쪽

`cleveref` 패키지는 수학적 문단을 참조하는 기능을 제공하지만, 추가 설정을 해 주어야 합니다. `\crefname`와 `\Crefname` 제어 문자열을 사용하여, 각 환경이 사용하는 카운터와 이름의 정보를 패키지에 알려 주어야 합니다.²

```
\crefname{<type>}{<singular>}{<plural>}
\Crefname{<type>}{<singular>}{<plural>}
```

<type>: 번호 앞에 출력할 단어를 지정할 정리형 환경의 이름
<singular>: 번호 앞에 출력할 단어의 단수 형태
<plural>: 번호 앞에 출력할 단어의 복수 형태

² 두 제어 문자열은 정리형 환경의 번호뿐 아니라 다른 번호의 형식을 사용자화 하는 데에도 사용되지만, 이 책에서는 이 사례에 한정하여 다루겠습니다.

`\crefname`의 인자로 지정된 단어는 `\cref` 등으로 참조 정보를 출력할 때 사용되고, `\Crefname`의 인자로 지정된 단어는 `\Cref` 등으로 참조 정보를 출력할 때 사용됩니다.

예를 들어, 영어 문서에서 `definition` 환경과 `theorem` 환경이 정의되어 있다면 전처리부에는 다음 코드가 있어야 합니다.

```
1 \usepackage{amsmath} % Not necessary
2 \newtheorem{definition}{Definition}
3 \newtheorem{theorem}{Theorem}
4
5 \usepackage{cleveref}
6 \crefname{definition}{Definition}{Definitions}
7 \crefname{theorem}{Theorem}{Theorems}
```

`\Crefname`을 따로 지정하지 않은 종류의 번호는 `\crefname`에 지정한 단어의 첫 글자를 자동으로 대문자화 하여 사용합니다. (위 예시에서는 인자로 작성한 단어가 이미 대문자화되어 있기 때문에 `\cref`든 `\Cref`든 같은 결과를 출력하겠지만…….)

```
1 \Crefrange{def:1.1}{def:1.3} are used
throughout this book. Also,
\cref{thm:1.1} shows the relation
between the three concepts.
```

Definitions 1.1 to 1.3 are used throughout this book. Also, Theorem 1.1 shows the relation between the three concepts.

만약 번호 앞에 출력할 단어로 축약된 형태를 사용한다면 다음과 같이, `\Crefname`을 사용해 축약되지 않은 형태를 별도로 지정해 주어야 합니다.

```
1 \usepackage{amsmath} % Not necessary
2 \newtheorem{definition}{Definition}
3 \newtheorem{theorem}{Theorem}
4
5 \usepackage{cleveref}
6 \crefname{definition}{Defn.}{Defns.}
7 \Crefname{definition}{Definition}{Definitions}
8 \crefname{theorem}{Thm.}{Thms.}
9 \Crefname{theorem}{Theorem}{Theorems}
```

```
1 \Crefrange{def:1.1}{def:1.3} are used
throughout this book. Also,
\cref{thm:1.1} shows the relation
between the three concepts.
```

Definitions 1.1 to 1.3 are used throughout this book. Also, Thm. 1.1 shows the relation between the three concepts.

여러 환경이 번호를 공유한다면 `amsmath` 패키지를 반드시 불러와야 합니다. 이 패키지가 없으면 여러 환경이 번호를 공유하더라도 `cleveref` 패키지가 각 환경을 구분할 수 없습니다. 다음은 `theorem` 환경과 `lemma` 환경이 번호를 공유할 때의 예시입니다.

```

1 \usepackage{amsmath}
2 \newtheorem{theorem}{Theorem}
3 \newtheorem{lemma}[theorem]{Theorem}
4
5 \usepackage{cleveref}
6 \crefname{theorem}{Theorem}{Theorem}
7 \crefname{lemma}{Lemma}{Lemma}

```

1 \Cref{lem:1.3} is the key idea for the proof of \cref{thm:1.4}.

Lemma 1.3 is the key idea for the proof of Theorem 1.4.

varioref 패키지와의 상호 작용

`varioref` 패키지와 `cleveref` 패키지를 함께 사용하는 경우, `varioref` 패키지를 더 먼저 불러와야 합니다. `cleveref` 패키지는 `\vref`, `\vrefrange` 등의 제어 문자열을 재정의 하여 두 패키지의 기능을 결합해 줍니다. `\vref` 등이 여러 식별자를 한 번에 입력받을 수 있게 되며, 참조 대상의 표지도 번호 앞에 자동으로 붙여 줍니다. 다만, 원래의 제어 문자열에서 사용할 수 있던 선택 인자는 사용할 수 없게 됩니다.

1 See \vref{sec6,sec8} for applications.

See sections 6 and 8 on pages 34 and 42 for applications.

1 You may refer to \vrefrange{ch:24}{ch:26}.

You may refer to chapters 24 to 26 on pages 259–341.

`cleveref` 패키지는 추가로, `varioref` 패키지에서는 제공하지 않았던 `\Vrefrange`, `\Vpageref`, `\Vpagerefrange`, `\Fullref` 제어 문자열을 정의합니다. 앞서 설명했던 대문자 버전 명령어들과 마찬가지로, 이들은 문장 첫머리에서 사용할 수 있도록 첫 글자가 대문자로 표시됩니다.

언어 설정

표나 그림, 장과 절을 일컫는 단어는 언어마다 다르므로, 이 패키지도 언어 설정 기능을 제공합니다. `varioref` 패키지와 마찬가지로, 문서의 언어를 패키지를 불러올 때 옵션으로 지정해 주면 됩니다. `babel` 패키지를 사용한다면 이 패키지를 불러올 때 지정한 언어

옵션을 `cleveref` 패키지의 옵션에 그대로 입력하면 됩니다. 반면, `polyglossia` 패키지를 사용하는 경우에는 `\setmainlanguage` 등의 제어 문자열로 문서의 언어 설정을 한 후 특별한 옵션 없이 `cleveref` 패키지를 불러오면 언어 설정이 자동으로 반영됩니다.

예를 들어, `babel` 패키지를 사용하여 프랑스어 문서를 작성하는 경우에는 전처리부에서 패키지를 다음과 같이 불러옵니다.

```
1 \usepackage[french]{babel}
2 \usepackage[french]{cleveref}
```

한편, `polyglossia` 패키지를 사용하는 경우에는 다음과 같이 불러옵니다.

```
1 \usepackage{polyglossia}
2 \setmainlanguage{french}
3 \usepackage{cleveref}
```

적절한 설정을 갖추었다면 다음 결과를 얻을 수 있습니다.

```
1 Le \vref{ch:1} donnera une
   introduction.
```

Le chapitre 1 donnera une introduction.

이 패키지도 한국어를 지원하지 않습니다. 따라서, 한국어 문서에서는 `cleveref` 패키지의 사용자화 기능을 사용해야 합니다. 예시 문서 15.2의 코드를 `cleveref-korean.sty` 라는 이름으로 작업 폴더에 저장한 뒤

```
\usepackage{cleveref-korean}
```

을 전처리부에 입력해 `cleveref-korean` 패키지를 불러 옵니다. `cleveref` 패키지에서 제공하는 옵션은 이 패키지를 불러올 때 지정하면 됩니다. `varioref-korean` 패키지와 함께 사용하려면 다음과 같이 입력하면 됩니다.

```
1 \usepackage[nospace]{varioref-korean}
2 \usepackage{cleveref-korean}
```

이 코드는 실험적인 것으로, 간혹 제대로 작동하지 않을 수도 있습니다.

이상의 설정을 갖추었다면 다음과 같이 한국어 문서와 어울리는 결과를 얻을 수 있습니다.

```
1 \Cref{ch:3}에서 알아본 극한과 연속의 내용을 바탕
   으로, \cref{ch:4}에서는 함수의 미분을 알아본다.
```

제3장에서 알아본 극한과 연속의 내용을 바탕으로, 제4장에서는 함수의 미분을 알아본다.

| 예시 문서 15.2 | cleveref-korean.sty

```

1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{cleveref-korean}[2024/03/31 Korean localisation of cleveref
  package]
3
4 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{cleveref}}
5 \ProcessOptions\relax
6 \RequirePackage{cleveref}
7
8 \newcommand\crefrangeconjunction{\~{}}
9 \newcommand\crefpairconjunction{와\space}
10 \newcommand\crefmiddleconjunction{, }
11 \newcommand\creflastconjunction{, }
12 \newcommand\crefpairgroupconjunction{와\space}
13 \newcommand\crefmiddlegroupconjunction{, }
14 \newcommand\creflastgroupconjunction{, 그리고 }
15
16 \crefname{part}{편}{편}
17 \crefname{chapter}{장}{장}
18 \crefname{section}{절}{절}
19 \crefname{appendix}{부록}{부록}
20 \crefname{table}{표}{표}
21 \crefname{figure}{그림}{그림}
22 \crefname{equation}{식}{식}
23 \crefname{footnote}{각주}{각주}
24 \crefname{page}{쪽}{쪽}
25
26 \crefformat{part}{제#2#1#3편}
27 \crefformat{chapter}{제#2#1#3장}
28 \crefformat{section}{#2#1#3절}
29 \crefformat{page}{#2#1#3쪽}
30
31 \crefrangeformat{part}{제#3#1#4\~{}}#5#2#6편}
32 \crefrangeformat{chapter}{제#3#1#4\~{}}#5#2#6장}
33 \crefrangeformat{section}{#3#1#4\~{}}#5#2#6절}
34 \crefrangeformat{page}{#3#1#4\~{}}#5#2#6쪽}
35
36 \crefmultiformat{part}{제#2#1#3}{편과 제#2#1#3편}{, #2#1#3}{, #2#1#3편}
37 \crefmultiformat{chapter}{제#2#1#3}{장과 제#2#1#3장}{, #2#1#3}{, #2#1#3장}
38 \crefmultiformat{section}{#2#1#3}{절과 #2#1#3절}{, #2#1#3}{, #2#1#3절}
39 \crefmultiformat{page}{#2#1#3}{쪽과 #2#1#3쪽}{, #2#1#3}{, #2#1#3쪽}
40
41 \crefrangemultiformat{part}{제#3#1#4\~{}}#5#2#6}
42 {편과 제#3#1#4\~{}}#5#2#6편}{, #3#1#4\~{}}#5#2#6}{, #3#1#4\~{}}#5#2#6편}

```



```

43 \crefrangemultiformat{chapter}{제#3#1#4\~{}}#5#2#6}
44 {장과 제#3#1#4\~{}}#5#2#6장}{, #3#1#4\~{}}#5#2#6}{, #3#1#4\~{}}#5#2#6장}
45 \crefrangemultiformat{section}{#3#1#4\~{}}#5#2#6}
46 {절과 #3#1#4\~{}}#5#2#6절}{, #3#1#4\~{}}#5#2#6}{, #3#1#4\~{}}#5#2#6절}
47 \crefrangemultiformat{page}{#3#1#4\~{}}#5#2#6}
48 {쪽과 #3#1#4\~{}}#5#2#6쪽}{, #3#1#4\~{}}#5#2#6}{, #3#1#4\~{}}#5#2#6쪽}
49
50 \@ifpackageloaded{varioref}{%
51 \newif\ifkcr@faraway \kcr@farawayfalse
52 \def\reftextfaraway#1{\pageref{#1}쪽\kvr@ui\global\kcr@farawaytrue}
53 \def\reftextpagerange#1#2{\pageref{#1}\~{}}\pageref{#2}쪽
   \kvr@eitneun\global\kcr@farawaytrue}
54
55 \def\cref@vref#1#2{%
56 \leavevmode%
57 \begingroup%
58 \def\reftextcurrent{%
59 \begingroup\UIfalse%
60 \cref@vpageref{\reftextcurrent}[]{#2}\endgroup%
61 \ifkcr@faraway\kvr@eitneun\fi\global\kcr@farawayfalse%
62 \@setcref@space\@cref{#1}{#2}%
63 \endgroup}%
64 \def\cref@vrefrange#1#2#3{%
65 \pagerefrange{#2}{#3}\@setcref@space%
66 \@@setcrefrange{#1}{#2}{#3}{}}%
67 \def\cref@fullref#1#2{%
68 \begingroup\UIfalse\@cref{fullpageref}{#2}\endgroup%
69 \kvr@eitneun\@setcref@space\@cref{#1}{#2}%
70 }
71
72 \endinput

```

몇 가지 패키지 옵션과 사용자화

cleveref 패키지는 참조 항목을 표시하는 방법을 조정하는 패키지 옵션을 제공합니다. 먼저, capitalize 옵션과 noabbrev 옵션은 참조된 번호 앞뒤에 출력되는 문구를 변경합니다.

- capitalize 옵션 번호 앞에 붙는 단어의 첫 글자를 대문자로 조판하도록 합니다. 즉, `\cref{sec:1}`과 `\Cref{sec:1}`은 모두 ‘Section 1’을 조판합니다.

이 옵션을 사용한다고 항상 `\cref`와 `\Cref`의 출력이 같아지는 것은 아닙니다. ‘Figure’나 ‘equation’ 등의 단어는 문장 첫머리에 오지 않을 때에는 축약될 수 있

으므로, `\cref{fig:3}`은 ‘Fig. 3’을, `\Cref{fig:3}`은 ‘Figure 3’을 출력합니다.

- **noabbrev 옵션** 축약형 단어를 사용하지 않도록 합니다. 예를 들어, 영어 문서에서는 ‘figure’의 약어로 ‘fig.’를, ‘equation’의 약어로 ‘eq.’를 사용하는 것이 기본 설정이지만, 이 옵션을 지정하면 약어를 사용하지 않게 됩니다.

또, `sort`, `compress`, `sort&compress`, `nosort` 옵션은 `\cref`나 `\Cref`로 여러 대상을 한 번에 참조했을 때의 표시 방법(정렬 여부, 범위 형태로의 표시 여부)을 변경합니다. 각 옵션의 설명과 함께 예시로

```
\cref{sec:4,sec:1,sec:2,sec:3}
```

을 조판한 결과를 아래 목록에 제시하였습니다.

- **sort 옵션** 번호를 정렬하되, 연속된 번호를 범위 형태로 나타내지는 않습니다.
예시: sections 1, 2, 3, and 4
- **compress 옵션** 입력된 번호를 정렬하지 않은 상태에서 연속된 번호만 범위의 형태로 나타냅니다. 참조 항목의 번호는 식별자가 입력된 순서대로 조판됩니다.
예시: sections 4 and 1 to 3
- **sort&compress 옵션** 기본 설정으로, 번호를 정렬한 후 연속된 번호는 범위의 형태로 나타냅니다.
예시: sections 1 to 4
- **nosort 옵션** 번호를 정렬하지도, 범위로 나타내지도 않습니다.
예시: sections 4, 1, 2, and 3

이외에도, 예시 문서 15.2처럼 명령어를 사용하여 문구가 표시되는 방식을 완전히 사용자화 할 수도 있습니다. 이에 대해서는 패키지 안내서[14, p. 11]를 참조하십시오.

15.2.4 hyperref 패키지: 참조 대상의 종류에 따른 문구

`hyperref` 패키지는 PDF 파일에 하이퍼링크를 만드는 데 널리 쓰이는 패키지입니다. 한편, 이 패키지는 상호 참조 기능도 제공합니다. 하이퍼링크 기능에 관한 내용은 제17장에서 알아보며, 이번 절에서는 상호 참조 기능에 대해 알아보겠습니다.

하이퍼링크
→ 247쪽

이 패키지는 `cleveref`나 `kotex` 패키지 등의 몇 가지를 제외하고는 가장 마지막으로 불러와야 합니다. 따라서 `varioref`, `hyperref`, `cleveref` 패키지를 함께 사용할 때에는 이 순서로 불러와야 합니다.

대상 참조하기

hyperref 패키지도 cleveref 패키지처럼 참조 대상의 종류에 따라 앞에 문구를 붙여 주는 기능을 제공합니다. 대상을 참조할 때 \ref 대신 \autoref 제어 문자열을 사용하면 됩니다. 또, \pageref 대신 \autopageref 제어 문자열을 사용하면 페이지 번호 앞에 ‘page’가 조판됩니다.

1 See \autoref{fig:3} on
\autopageref{fig:3}.

See Figure 3 on page 138.

참고 | \eqref나 \cref 등과 마찬가지로, \autoref도 \labelformat과 기능이 충돌합니다. 따라서 \autoref를 사용하려는 경우에는 \labelformat으로 번호의 형식을 변경하지 않아야 합니다.

번호 앞에 붙는 문구를 사용자화 하는 방법은 패키지 매뉴얼[40, p. 26]을 참조하시기 바랍니다. \autoref는 ‘3절’처럼 번호 뒤에 문구를 붙이는 기능을 제공하지 않으므로 한국어 문서에서는 예시 문서 15.3과 같은 편법을 사용해야 합니다.

한국어 문서에서의 hyperref 패키지 사용

hyperref 패키지는 varioref 패키지의 제어 문자열을 일부 재정의하므로, 한국어 문서에서 kotex-varioref 패키지나 varioref-korean 패키지(예시 문서 15.1 참조)를 불러오더라도 \vref 등이 올바르게 동작하지 않습니다. 이때에는 예시 문서 15.3의 코드로 hyperref-korean 패키지를 만든 후 이를 hyperref 패키지 대신 불러옵니다. hyperref 패키지를 불러올 때 지정하던 옵션은 이 패키지에 그대로 지정하여 사용할 수 있습니다. 또, 이 패키지는 \autoref가 번호 앞뒤에 출력하는 단어도 한국어로 변경해 줍니다.

| 예시 문서 15.3 | hyperref-korean.sty

```

1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{hyperref-korean}[2024/03/31 Korean localisation of hyperref
package]
3
4 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{hyperref}}
5 \ProcessOptions\relax
6 \RequirePackage{hyperref}
7
8 \def\equationautorefname{수식}%
9 \def\footnoteautorefname{각주}%

```

```

10 \def\itemautorefname{항목}%
11 \def\figureautorefname{그림}%
12 \def\tableautorefname{표}%
13 \def\partautorefname#1#2#3#4#5#6#7#8{제#2#3#4#5#6#7#8편}%
14 \def\appendixautorefname{부록}%
15 \def\chapterautorefname#1#2#3#4#5#6#7#8{제#2#3#4#5#6#7#8장}%
16 \def\sectionautorefname#1#2#3#4#5#6#7#8{#2#3#4#5#6#7#8절}%
17 \def\subsectionautorefname#1#2#3#4#5#6#7#8{#2#3#4#5#6#7#8절}%
18 \def\subsubsectionautorefname#1#2#3#4#5#6#7#8{#2#3#4#5#6#7#8절}%
19 \def\paragraphautorefname#1#2#3#4#5#6#7#8{#2#3#4#5#6#7#8문단}%
20 \def\subparagraphautorefname#1#2#3#4#5#6#7#8{#2#3#4#5#6#7#8문단}%
21 \def\FancyVerbLineautorefname#1#2#3#4#5#6#7#8{#2#3#4#5#6#7#8행}%
22 \def\theoremautorefname{정리}%
23 \def\pageautorefname#1#2#3#4#5#6#7#8{#2#3#4#5#6#7#8쪽}%
24
25 \ifpackageloaded{varioref}{
26   \AtBeginDocument{%
27     \ifundefined{hyper@nopath@varioref}{
28       \renewcommand\Vref@star[2][]{%
29         \begingroup
30           \let\T@pageref\@pagerefstar
31           \vpageref[#1]{#2} \Ref*{#2}%
32         \endgroup}%
33       \renewcommand\Vr@f[2][]{%
34         \begingroup
35           \let\T@pageref\@pagerefstar
36           \hyperref[{#2}]{\vpageref[#1]{#2} \Ref*{#2}}%
37         \endgroup}%
38       \renewcommand\vr@f[2][]{%
39         \begingroup
40           \let\T@pageref\@pagerefstar
41           \hyperref[{#2}]{\vpageref[#1]{#2} \ref*{#2}}%
42         \endgroup}%
43       \renewcommand\vref@star[2][]{%
44         \begingroup
45           \let\T@pageref\@pagerefstar
46           \vpageref[#1]{#2} \ref*{#2}%
47         \endgroup}%
48     }{}%
49   }
50 }
51
52 \endinput

```

제 16 장

참고 문헌

thebiblio-
graphy 환경
→ 62쪽

논문이나 책을 작성할 때에는 다른 문헌을 참조한 후 본문에 인용 표시를 하고, 문서의 끝 부분에 인용한 문헌의 정보를 모아 작성해 둡니다. 5.4.1절에서는 thebibliography 환경을 사용해 이 목록을 직접 작성하는 방법을 알아보았습니다. 이번 장에서는 보조 프로그램을 사용해 이 목록을 자동으로 생성하는 방법을 알아보겠습니다.

16.1 참고 문헌 관리 방법

문서의 규모가 작다면 보조 프로그램 없이 참고 문헌 목록을 직접 작성해도 괜찮습니다. 하지만 인용한 문헌이 많고 작성한 글을 여러 학회에 제출한다면 문헌의 목록을 관리하는 것이 어려워집니다. 새로운 문헌을 추가할 때마다 문헌의 올바른 순서를 알아야 하고, 저자와 제목, 출판 연도 등에 올바른 서식을 지정해 주어야 하기 때문입니다.

이러한 작업을 편리하게 할 수 있도록, \TeX 배포판에는 \LaTeX 과 함께 사용할 수 있는 보조 프로그램으로 Bib \TeX 과 biber가 있습니다. 이번 절에서는 \LaTeX 에서 참고 문헌을 관리하는 방법을 간략히 설명합니다.

16.1.1 문헌 정보 파일

가장 먼저, .bib 확장자의 문헌 정보 파일이 필요합니다. 이 파일에는 인용할 문헌의 저자, 제목, 출판 연도 등의 정보를 일정한 형식에 맞추어 저장합니다. 또, 각 문헌을 인용할 때 사용할 식별자를 지정해 둡니다. \LaTeX 소스 파일에서는 이 문헌 파일을 불러온 뒤, 식별자(identifier)를 사용해 문헌을 인용하고 인용한 문헌의 목록을 조판하도록 설정합니다. 이때 인용 표지나 목록의 형식은 \LaTeX 소스 파일에서 원하는 대로 변경할 수 있습니다.

| 예시 문서 16.1 | latex.bib

```

1 @book{texbook,
2   author = {Donald E. Knuth},
3   publisher = {Addison-Wesley Professional},
4   title = {The {\TeX}book},
5   year = {1986}
6 }
7 @book{latexdps,
8   author = {Leslie Lamport},
9   edition = {Second},
10  publisher = {Addison-Wesley},
11  title = {{\LaTeX}: A Document Preparation System},
12  year = {1994}
13 }
14 @book{tlc2,
15   author = {Frank Mittelbach and Michel Goossens},
16   edition = {Second},
17   publisher = {Addison-Wesley},
18   title = {The {\LaTeX} Companion},
19   year = {2004}
20 }

```

예시 문서 16.1은 문헌 정보 파일의 한 예시입니다. 각 문헌의 정보는 골뱅이표 @를 입력한 뒤에 작성하고, 골뱅이표 뒤의 ‘book’은 문헌의 종류가 책을 뜻합니다. 문헌의 정보는 중괄호 안에 작성하며, 각 문헌의 첫째 줄에 적힌 ‘texbook’이나 ‘latexdps’, ‘tlc2’는 \LaTeX 소스 파일에서 `\cite` 제어 문자열로 문헌을 인용할 때 사용하는 식별자입니다. 나머지 내용은 제목이나 저자 등 각 문헌의 세부 정보입니다. 문헌 정보 파일을 작성하는 방법은 16.2절에서 설명합니다.

16.1.2 정렬 프로그램

보조 프로그램에는 `BibTeX`과 `biber`가 있습니다. 이들은 목록에 조판할 문헌 정보를 받은 뒤, 각 문헌을 올바른 순서로 정렬하고, 문헌의 정보에 올바른 서식을 지정해 줍니다.

- `BibTeX` \LaTeX 과 함께 사용하는 가장 기본적인 프로그램입니다. 오랫동안 사용된 만큼 대부분의 경우에는 안정적으로 작동하지만, 그만큼 개발된 시기가 오래되어 ASCII 범위의 문자만을 제대로 처리할 수 있는 등 내용을 처리하는 데 제약이 많습니다. 8비트 문자를 처리할 수 있도록 확장된 `BibTeX8`도 있지만, 유니코드를 지원하지 않아 현재에는 잘 사용되지 않습니다.

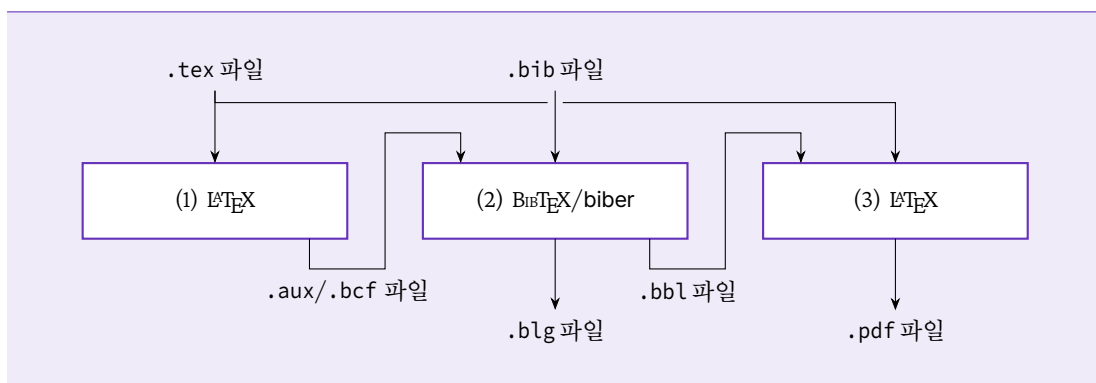


그림 16.1 | 문헌 목록 생성 시의 파일 처리 과정

- **biber** 유니코드를 지원하는 프로그램으로, 유럽의 언어뿐 아니라 수많은 언어를 잘 지원합니다. **biblatex** 패키지와 함께 사용하며, **BibTeX** 등 기존 프로그램에 비해 더 다양하고 정교한 기능을 사용할 수 있습니다.

상황이나 필요에 따라 원하는 프로그램을 사용하면 되지만, 특별한 제약이 없다면 다양한 문자를 자유롭게 처리할 수 있는 **biber**를 사용하는 것이 좋습니다.

16.1.3 파일 처리 과정

필요한 파일과 프로그램이 모두 준비되었다면, 서너 번의 단계를 거쳐 파일을 처리해 줍니다. 이렇게 함으로써 완성된 PDF 문서를 얻을 수 있습니다.

- 1 | **.tex** 파일을 **L&ATeX**으로 컴파일 합니다. 이 파일에는 사용하려는 **.bib** 파일을 지정해 주어야 합니다. 현재 문서에서 인용한 문헌의 정보와 문헌 정보 파일이 **.aux** 파일 또는 **.bcf** 파일에 기록됩니다.
- 2 | **BibTeX**으로 **.aux** 파일을 처리하거나 **biber**로 **.bcf** 파일을 처리합니다. **.bib** 파일에 저장된 내용을 바탕으로 참고 문헌 목록을 만든 뒤 이를 기록한 **.bbl** 파일을 생성합니다.
- 3 | **.tex** 파일을 **L&ATeX**으로 컴파일 합니다. 앞 단계에서 생성된 **.bbl** 파일을 불러와 문헌 목록을 조판하고 인용 표지를 업데이트 합니다. **BibTeX**을 사용했다면 컴파일을 두 번 해야 합니다.

Overleaf 등의 환경에서는 컴파일 과정을 자동으로 처리해 주므로, 컴파일 명령을 한 번만 내려 주면 됩니다.

16.2 문헌 정보 파일 작성하기

이제, 문헌 정보 파일을 작성하는 방법을 알아보겠습니다. 이 파일은 플레인 텍스트 형식이므로, 메모장이나 텍스트 편집기, 또는 각종 TeX 문서 편집기 등의 프로그램으로 작성할 수 있습니다. 파일을 저장할 때 .bib 확장자만 붙여 주면 됩니다.

앞의 예시에서 보았듯, 하나의 문헌 정보 파일에는 여러 문헌의 정보가 저장되어 있습니다. 각 문헌의 정보는 다음의 형식으로 작성합니다.

```

1 @⟨entry type⟩{⟨identifier⟩,
2   ⟨key⟩ = ⟨value⟩,
3   ...
4   ⟨key⟩ = ⟨value⟩
5 }
```

⟨entry type⟩는 이 문헌의 종류가 무엇인지 지정합니다. ⟨identifier⟩에는 앞서 설명한 식별자를 적습니다. 중괄호 안에는 각 문헌의 정보를 ‘⟨key⟩ = ⟨value⟩’의 형식으로 작성합니다. 이 정보 한 쌍을 ‘필드’라고 부릅니다. ⟨value⟩를 작성할 때에는 내용을 중괄호 {...}나 큰따옴표 "..."로 감싸야 합니다. 다만, ‘2024’처럼 숫자만 있을 때에는 중괄호나 큰따옴표로 감싸지 않아도 됩니다.

식별자를 입력할 때에는 사용할 수 있는 문자에 제약이 있습니다. 정렬 프로그램으로 BibTeX을 사용할 것이라면 식별자에 문자 " # % , = { } ~ \와 공백 문자를 제외한 ASCII 문자만을 사용할 수 있습니다. biber를 사용할 것이라면 제약이 완화되어, 문자 " # % , = { } ' () ~ \와 공백 문자를 제외한 임의의 문자 — 즉, 한글과 한자 등을 포함한 아무 문자 — 를 식별자에 사용할 수 있습니다.

이번 절에서는 BibTeX이나 biber/biblatex에서 사용하기 위한 문헌 정보 파일을 작성하는 방법을 알아봅니다. 16.2.1절과 16.2.2절에서는 BibTeX의 문헌 정보 파일에 관한 내용을, 16.2.3절에서는 biber/biblatex의 문헌 정보 파일에 관한 내용을 설명합니다.

16.2.1 BibTeX의 문헌 정보 파일: 기본 기능

BibTeX에서 사용할 수 있는 문헌의 종류에 대한 설명은 표 16.1에서, 필드의 종류에 대한 설명은 표 16.2에서 확인할 수 있습니다. 문헌의 종류에 따라 필수로 정보를 입력해야 하는 필드, 선택적으로 입력할 수 있는 필드, 사용되지 않는 필드가 구분되며, 이 정보도 표 16.1에서 확인할 수 있습니다. 표에 언급되지 않은 필드는 대부분 사용되지 않고 무시되지만, 설정에 따라 부가 정보로써 문헌 목록에 조판될 수도 있습니다.

표 16.1 | BibTeX: 문헌의 종류

| 문헌의 종류 | 설명 |
|---------------|--|
| article | 학술지나 잡지 등에 실린 글.
필수 필드: author, journal, title, year
선택 필드: month, note, number, pages, volume |
| book | 출판사를 통해 출간된 책이나 단행본.
필수 필드: author 또는 editor, publisher, title, year
선택 필드: address, edition, month, note, number 또는 volume, series |
| booklet | 출판사를 통하지 않고 출판된 작업물.
필수 필드: title 선택 필드: address, author, howpublished, month, note, year |
| inbook | 책의 한 편이나 장 등 일부분.
필수 필드: author 또는 editor, chapter 또는 pages, publisher, title, year
선택 필드: address, edition, month, note, number 또는 volume, series, type |
| incollection | 책의 일부분으로, 별도의 제목이 달려 있는 것.
필수 필드: author, booktitle, publisher, title, year
선택 필드: address, chapter, edition, editor, month, note, number 또는 volume, pages, series, type |
| inproceedings | 학회의 프로시딩에 속한 논문.
필수 필드: author, booktitle, title, year
선택 필드: address, editor, month, note, number 또는 volume, organization, pages, publisher, series |
| manual | 기술적인 문서나 사용 설명서 등.
필수 필드: title 선택 필드: address, author, edition, month, note, organization, year |
| masterthesis | 석사 학위 논문.
필수 필드: author, school, title, year 선택 필드: address, month, note, type |
| misc | 다른 모든 종류에 속하지 않는 문헌. 선택 필드 중 하나 이상의 정보를 입력해야 함.
필수 필드: 없음 선택 필드: author, howpublished, month, note, title, year |
| phdthesis | 박사 학위 논문.
필수 필드: author, school, title, year 선택 필드: address, month, note, type |
| proceedings | 학회의 프로시딩.
필수 필드: title, year
선택 필드: address, editor, month, note, number 또는 volume, organization, publisher, series |
| techreport | 학교나 기관 등에서 출판한 보고서.
필수 필드: author, institution, title, year
선택 필드: address, month, note, number, type |
| unpublished | 저자와 제목이 있지만 정식으로 출판되지 않은 글.
필수 필드: author, note, title 선택 필드: month, year |

| 표 16.2 | Bibtex: 정보 필드의 종류

| 필드의 종류 | 설명 |
|--------------|--|
| address | 문헌을 출판한 출판사나 학회, 단체가 위치한 주소를 적습니다. |
| annote | 문헌에 대한 주석을 작성합니다. 기본 스타일에서는 사용되지 않지만, 이 필드를 사용하는 스타일을 사용한다면 여기의 내용을 바탕으로 주석이 작성됩니다. |
| author | 저자(들)의 이름을 적습니다. 이름을 작성하는 형식은 뒤쪽의 내용을 참조하시기 바랍니다. |
| booktitle | 책의 일부분이 인용될 때(inbook 등의 문헌에서), 그 책 자체의 제목을 작성합니다. |
| chapter | 장이나 절, 편 등의 번호를 적습니다. |
| crossref | 다른 항목의 정보를 가져올 때, 그 항목의 키를 입력합니다. |
| edition | 책의 판 번호를 적습니다. 'Second'와 같이 첫 글자가 대문자인 서수로 적어야 합니다. |
| editor | 편집자(들)의 이름을 작성합니다. author 필드와 마찬가지로, 뒤쪽에 설명된 형식을 갖추어 이름을 작성해야 합니다. |
| howpublished | 문헌이 출간된 방법을 작성합니다. 첫 글자는 대문자로 작성해야 합니다. |
| institution | 연구 보고서에서, 연구를 지원한 단체의 이름을 적습니다. |
| journal | 저널의 이름을 작성합니다. |
| key | 문헌 목록을 정렬하고 인용 표지를 만들 때, 저자나 편집자의 이름, 단체의 이름 대신 여기에 지정된 문자열을 사용합니다. |
| month | 문헌이 출간 또는 작성된 월을 적습니다. mar과 같은 세 글자 약어를 중괄호나 큰따옴표로 감싸지 않고 입력해야 합니다. |
| note | 독자에게 도움을 줄 수 있는 추가 정보를 작성합니다. 첫 글자는 대문자로 작성해야 합니다. |
| number | 저널이나 잡지의 호수, 보고서의 일련번호 등을 적습니다. |
| organization | 회의를 지원한 단체나 매뉴얼을 출판한 단체의 이름을 작성합니다. |
| pages | 페이지의 번호나 그 범위를 적습니다. '48--65'나 '5, 12, 33--51' 등의 형식으로 입력합니다. |
| publisher | 출판사의 이름을 작성합니다. |
| school | 학위논문이 작성된 학교의 이름을 적습니다. |
| series | 여러 권의 책이 하나의 시리즈로 출판된 경우, 그 시리즈의 이름을 작성합니다. |
| title | 문헌의 제목을 작성합니다. |
| type | 'Research Note'와 같이, 연구 보고서의 형식을 적습니다. |
| volume | 저널이나 여러 권에 걸친 책의 권 번호를 작성합니다. |
| year | 문헌이 출간 또는 작성된 연도를 적습니다. 'about 1984'와 같이 내용의 마지막 네 글자가 숫자이면 문제가 없지만, 일반적으로는 네 자리 숫자만 적는 것이 좋습니다. |

인명이나 글의 제목을 작성할 때에는 몇 가지 규칙을 따라야 합니다. 또, 한 문헌의 정보를 다른 문헌에서 참조해 사용할 수도 있습니다. 이 방법에 대해서 자세히 알아보겠습니다.

인명 작성하기

인명을 작성할 때에는 성과 나머지 부분을 구분해서 작성합니다. 대부분의 경우에는 이름을 그대로 입력해도 문제가 없지만, 성이 여러 단어로 구성된 경우에는 성을 반드시 앞에 적고, 성과 나머지 부분을 쉼표로 구분해야 합니다. 다음은 몇 가지 예시입니다.

```
1 author = {Albert Einstein}
2 author = {Einstein, Albert}
3 author = {Ludwig van Beethoven}
4 author = {van Beethoven, Ludwig}
5 author = {Lloyd Webber, Andrew}
```

마지막 예시에서 인명을 ‘Andrew Lloyd Webber’로 입력한다면 BibTeX은 이름을 ‘Andrew Lloyd’, 성을 ‘Webber’로 해석합니다. 따라서, 위와 같이 성이 앞에 오도록 입력해야 합니다. 더 복잡한 형태의 이름을 작성하는 방법은 16.2.2절에서 설명합니다.

인명을 적을 때, 원문에서 이름이나 미들네임을 축약했다면 그 표기를 살려야 합니다. 같은 사람이 작성한 두 문헌에서의 인명 표기가 다르면 문헌 정보 파일에도 이를 반영해야 합니다. 다만, 축약되지 않은 이름이 명확한 경우에는 축약된 부분을 대괄호 [...] 안에 적어줄 수 있습니다. 예를 들어, 한 문헌에서는 ‘D. E. Knuth’라고, 다른 문헌에서는 ‘Donald E. Knuth’라고 작성되어 있다면 첫 번째 문헌의 저자 정보를 다음 둘 중 하나로 입력합니다.

```
1 author = {D. E. Knuth}
2 author = {D[onald] E. Knuth}
```

여러 사람의 이름을 나열할 때에는 각 사람의 이름을 ‘and’로 구분하며, 일부 인명을 생략할 때에는 ‘and others’로 목록을 끝내면 됩니다. 이렇게 입력된 내용은 스타일에 따라 ‘et al.’ 등의 문구로 자동으로 조판됩니다. 이는 인명을 나열할 때뿐 아니라 단체의 이름 등을 여럿 적어야 할 때에도 동일하게 적용됩니다. ‘and’를 중괄호로 감싸면 각 이름을 구분하는 것으로 인식되지 않습니다.

BibTeX은 확장 로마자를 제대로 처리하지 못하므로 이들을 L^AT_EX의 명령어를 사용해 입력해야 하며, 또 그 명령어를 중괄호로 감싸 주어야 합니다. 예를 들어, ‘Gödel’은 ‘G{"o}del’ 또는 ‘G{"{o}}del’로 입력해야 합니다.

글의 제목 작성하기

제목을 작성할 때에는 원문의 대소문자 표기를 보존하여 작성해야 합니다. 책의 제목은 보통 각 단어의 첫 글자를 대문자로 적으며, 논문의 제목은 첫 단어의 첫 글자만을 대문자로 적습니다. 단, 첫 번째 단어의 첫 글자나 쌍점 뒤 첫 단어의 첫 글자는 대문자로 적어 주어야 합니다.

문헌 정보를 문서에 조판할 때 사용하는 스타일에 따라 대소문자 표기는 자동으로 바뀔 수 있습니다. 만약 고유 명사를 입력하거나 특수한 표기를 사용하는 경우에는 대소문자가 자동으로 바뀌지 않도록 중괄호로 한 번 감싸 주어야 합니다. 예를 들어, 글의 제목에 ‘Korea’가 포함되어 있다면 이 단어 전체나 첫 번째 글자를 중괄호로 감싸 ‘{Korea}’ 또는 ‘{K}orea’로 입력해야 합니다.

L^AT_EX의 제어 문자열을 사용해 특수한 문자열을 입력할 때에도 그 제어 문자열을 중괄호로 감싸 주어야 합니다. 예를 들어, ‘The T_EXbook’은 다음과 같이 입력합니다.

```
title = {The {\TeX}book}
```

날짜 작성하기

B_IB_TE_X은 문헌의 출판 날짜를 기록하는 필드로 year과 month 필드를 제공합니다. 연도는 아라비아 숫자로, 월(月)은 세 글자 약어로 중괄호나 따옴표 없이 입력합니다. 예를 들어, 2018년 7월에 출판된 문헌이라면

```
year = 2018, month = jul
```

이라고 입력합니다. 만약 일(日)을 입력해야 한다면 month 필드에 ‘<month> # "<day>"’ (일을 월의 뒤에 표기) 또는 ‘"<day>~" # <month>’ (일을 월의 앞에 표기)의 형식으로 입력합니다. 즉, 2003년 3월 7일에 출판된 문헌은 날짜 정보를 작성할 때

```
year = 2003, month = mar # "~7"
```

이라고 입력해야 합니다. 이 문법은 B_IB_TE_X의 상용구 기능을 사용한 것으로, 이 기능에 대한 자세한 설명은 239쪽을 참조하시기 바랍니다.

16.2.2 B_IB_TE_X의 문헌 정보 파일: 심화 기능

16.2.1절의 내용만으로도 B_IB_TE_X에서 사용하기 위한 문헌 정보 파일을 작성할 수 있습니다. 하지만 문헌의 양이 많고 정보가 복잡해진다면 B_IB_TE_X의 여러 심화 기능을 사용하는 것이 좋습니다.

복잡한 인명 작성하기

엄밀히는, BibTeX은 사람의 이름을 네 개의 부분으로 나누어 처리합니다. 이름 부분, von 부분, 성 부분, Jr 부분으로, 각각 *⟨first⟩*, *⟨von⟩*, *⟨last⟩*, *⟨jr⟩*로 나타내겠습니다.

- 이름 부분 이름(first name)과 미들네임(middle name)을 적습니다.
- von 부분 'von', 'de', 'van', 'van der' 등 소문자로 쓰는 접두어를 적습니다.
- 성 부분 성(last name)을 적습니다.
- Jr 부분 'Jr.', 'Sr.' 등을 적습니다. 앞에 쉼표(,)를 적어야 합니다.

서구권에서 인명을 적을 때에는 이름을 먼저 적기도 하고, 성을 적고 쉼표를 적은 뒤 그 뒤에 이름을 적기도 합니다. 따라서, 이에 따라 BibTeX이 이름을 올바르게 해석할 수 있도록 적절한 처리를 해 주어야 합니다.

- 이름을 먼저 적는 경우 '*⟨first⟩ ⟨von⟩ ⟨last⟩*'의 형식으로 적습니다.
 - 소문자로 적은 부분이 있다면, 이 부분은 von 부분으로 처리됩니다. 이 부분의 앞뒤는 각각 이름 부분과 성 부분으로 처리됩니다.
 - von 부분이 없다면, 마지막 단어는 성으로, 나머지는 이름으로 처리됩니다. 성이 여러 단어인 경우에는 중괄호로 묶어 주어야 합니다.
- 성을 먼저 적는 경우 '*⟨von⟩ ⟨last⟩, ⟨first⟩*'의 형식으로 적습니다.
 - 소문자로 적은 부분이 있다면, 이 부분은 von 부분으로 처리됩니다.
 - 쉼표를 기준으로 성 부분과 이름 부분이 구분됩니다.
- Jr 부분이 있는 경우 '*⟨von⟩ ⟨last⟩, ⟨jr⟩, ⟨first⟩*'의 형식으로 적습니다.
 - 소문자로 적은 부분이 있다면, 이 부분은 von 부분으로 처리됩니다.
 - 쉼표를 기준으로 성 부분과 Jr 부분, 이름 부분이 구분됩니다.

문헌 정보 파일에 인명을 작성할 때 어떤 문헌은 성을 앞에 적고 어떤 문헌은 이름을 앞에 적더라도, 문헌 목록을 조판할 때에는 지정한 스타일에 따라 일정하게 조판됩니다. 또, 'Jr.'와 'Sr.' 앞뒤의 쉼표도 자동으로 처리됩니다.

다른 문헌의 정보를 상호 참조하기

crossref 필드를 사용하면 다른 문헌의 정보를 그대로 가져와 사용할 수 있습니다. [37]에서 인용한 아래 예시를 사용해 설명하겠습니다.

```
1 @INPROCEEDINGS{no-gnats,
2   crossref = "gg-proceedings",
```

```

3   author = "Rocky Gneisser",
4   title = "No Gnats Are Taken for Granite",
5   pages = "133-139"
6 }
7 @PROCEEDINGS{gg-proceedings,
8   editor = "Gerald Ford and Jimmy Carter",
9   title = "The Gnats and Gnus 1988 Proceedings",
10  booktitle = "The Gnats and Gnus 1988 Proceedings"
11 }

```

식별자가 no-gnats인 문헌은 gg-proceedings의 여러 필드 중 no-gnats에 없는 필드의 정보를 상속받습니다. 이 경우에는 editor와 booktitle 필드의 정보를 상속받아 다음과 동일한 내용을 가지게 됩니다.

```

1 @INPROCEEDINGS{no-gnats,
2   author = "Rocky Gneisser",
3   editor = "Gerald Ford and Jimmy Carter",
4   title = "No Gnats Are Taken for Granite",
5   booktitle = "The Gnats and Gnus 1988 Proceedings",
6   pages = "133-139"
7 }

```

이처럼, 두 문헌 사이에 여러 정보가 겹치는 경우에는 상호 참조 기능을 사용해 같은 내용을 여러 번 작성하는 것을 피할 수 있습니다.

인용 표지에 원하는 문자열 사용하기

문헌 참조 형식 중에는 저자의 이름을 사용하는 방식이 있습니다. 문헌 인용 표지에 축약된 저자의 이름과 출판 연도를 사용하여, 저자가 Donald E. Knuth이고 출판연도가 1984년이라면 '[Knu84]'처럼 표시하는 식입니다. 저자의 이름이 지정되지 않았다면 editor 필드나 organization 필드의 정보를 대신 사용합니다.

하지만, key 필드를 사용하면 문헌 인용 표지에 저자·편집자·단체 이름 대신 원하는 문자열을 사용할 수 있습니다. 예컨대 organization 필드의 정보로 라벨이 생성되는 다음 상황을 생각해 봅시다.

```

1 organization = "Korean \TeX\ Users Group",
2 year = "2007"

```

이 문헌을 본문에서 인용하면 라벨이 '[Kor07]'로 출력됩니다. 이때, 단체의 이름 앞쪽을 딴 'Kor' 대신 단체의 약어 'KTUG'를 사용하려면 key = "KTUG"라고 입력하면 됩니다.

문헌을 인용하면 인용 표지로 ‘[KTUG07]’이 출력됩니다.

상용구 정의하기

학회의 이름이나 학회지의 이름 등 자주 쓰이는 문구는 상용구로 정의해 둬으로써 문헌 정보 파일을 편리하게 작성할 수 있습니다. 문헌 정보 파일에서 다음과 같은 형식으로 @string 명령어를 사용하면 상용구를 정의할 수 있습니다.

```
@string{<abbreviation> = <full text>}
```

여기에서 <abbreviation>에는 단축어를, <full text>에는 원래의 문구를 작성합니다. 상용구를 사용할 때에는 <abbreviation> 부분에 입력한 문자열을 중괄호나 큰따옴표로 감싸지 않고 입력해야 합니다. 여기에 다른 내용을 덧붙이려면 해시 기호 #를 사용해야 합니다.

예를 들어, 문헌 정보 파일에

```
@string{KTUG = {Korean \TeX\ Users Group}}
```

라고 입력해 둔다면 다음 두 줄의 코드는 같은 정보를 organization 필드에 저장하는 역할을 합니다.

```
1 organization = KTUG
2 organization = {Korean \TeX\ Users Group}
```

월(月)을 입력할 때 사용하는 jan, feb, mar 등의 문자열도 상용구로써 정의되어 있습니다.

전처리부 기능

@preamble 명령어는 BibTeX의 처리 결과 만들어지는 .bbl 파일의 첫머리에 코드를 삽입하는 역할을 합니다. 이를 통해 문헌을 처리할 때에만 사용할 명령어를 정의할 수 있습니다. 이 기능의 자세한 사용 방법은 [37]을 참조하시기 바랍니다.

주석 작성하기

@comment 문헌 안에 작성된 내용은 주석으로 처리되어, 문헌 정보 처리 시 무시됩니다. 또, BibTeX은 문헌 정보 바깥 부분에 있는 모든 내용을 무시합니다. 한편, TeX에서 주석을 작성할 때 사용하는 퍼센트 기호 %는 문헌 정보 파일에서 주석을 작성하는 데 사용할 수 없습니다.

16.2.3 biber/biblatex의 문헌 정보 파일

biber와 biblatex을 사용해 문헌 정보를 처리할 때에는 더 다양하고 세분화된 정보를 사용할 수 있습니다. 문헌 정보 파일은 같은 방법으로 작성하지만, 문헌의 종류를 분류하는 기준이 일부 다르고, 정보를 작성하는 방법이 다른 필드도 몇 가지 있습니다. biber/biblatex에서 지원하는 문헌의 종류는 표 16.3에서 확인할 수 있습니다. biber/biblatex이 지원하는 필드의 종류는 아주 많으므로 이 책에서 설명하지 않을 것이며, [24]를 참조하시기 바랍니다.

BibTeX과 biber/biblatex의 호환

biber/biblatex은 BibTeX이 사용하는 문헌 정보 파일과의 호환을 위해, 표 16.1에는 있지만 표 16.3에는 없는 종류의 문헌도 처리할 수 있습니다. masterthesis와 phdthesis는 thesis로 처리되며, techreport는 report로 처리됩니다. 단, BibTeX에서의 inbook 문헌은 별도의 제목과 저자가 없다면 book으로 변경해 주어야 합니다.

표 16.2에는 있지만 biber/biblatex은 지원하지 않는 대부분의 정보 필드도 원활하게 처리됩니다. 다만, edition 필드는 기수로 적힌 단어를 아라비아 숫자로 바꾸어 적어야 하고, month 필드는 월을 아라비아 숫자로 입력해야 합니다. 날짜를 적을 때에는 year와 month 필드를 사용하는 것보다는 date 필드에 'YYYY-MM-DD'의 형식으로 입력하는 것이 좋습니다.

16.3 L^AT_EX 문서 작성하기

문헌 정보 파일이 준비되었다면 L^AT_EX 문서에서 명령어를 사용해 이 파일을 불러온 후, 문헌 목록과 인용 표지를 어떤 스타일로 조판할지 지정해 주어야 합니다. 이후 본문에서는 문헌 정보 파일에 작성된 문헌을 인용할 수 있습니다. 이번 절에서는 문헌 정보 파일을 불러올 때, 스타일을 지정할 때, 문헌을 인용할 때 사용하는 명령어에 대해 알아봅니다.

참고 문헌을 처리할 때 쓰이는 ‘스타일’은 다음을 결정하는 역할을 합니다.

- 문헌 인용 표지로 사용하는 문자열(문헌의 번호, 저자 이름과 출판 연도, 문헌의 제목 등, 괄호 표시 여부)
- 문헌 목록에서 문헌들의 정렬 순서(본문에서 인용한 순서에 따라서, 저자 이름과 연도에 따라서, 제목에 따라서 등)

| 표 16.3 | biber/biblatex: 문헌의 종류

| 문헌의 종류 | | 설명 |
|---------|----------------|---|
| 기사 | article | 저널, 잡지, 신문이나 정기 간행물에 속한 글로, 제목이 달려 있는 것.
필수 필드: author, date 또는 year, journaltitle, title
선택 필드: annotator, commentator, editor, eid, journaltitleaddon, origlanguage, pages, version |
| | book | 한 명의 저자가 작성했거나 여러 명의 저자가 전체에 기여한 단권의 책.
필수 필드: author, date 또는 year, title
선택 필드: chapter, eid, mainsubtitle, maintitle, maintitleaddon, pages, pagetotal, part, volume |
| 책(단행본) | mvbook | book의 다른 모든 조건을 만족하되 여러 권으로 된 문헌.
필수 필드: author, date 또는 year, title 선택 필드: pagetotal |
| | inbook | 제목이 붙어 있으며 완결된 내용을 갖춘, 책의 일부분.
필수 필드: author, booktitle, date 또는 year, title
선택 필드: bookauthor, booksubtitle, booktitleaddon, chapter, eid, mainsubtitle, maintitle, maintitleaddon, pages, part, volume |
| | bookinbook | inbook의 조건을 만족하면서, 원래는 독립된 책으로 출간되었던 것. |
| | suppbook | book의 보조적인 자료. inbook과는 달리 서문이나 발문 등의 요소를 가리킴. 표준 스타일에서는 inbook과 동일하게 처리됨. |
| booklet | | 책과 비슷한 형태의 문헌이지만 공식적인 출판사나 후원 단체가 없는 것.
필수 필드: author, date 또는 year, title
선택 필드: chapter, eid, howpublished, location, pages, pagetotal, type |
| 모음집 | collection | 각각의 저자가 완결된 내용으로 작성한 글들을 한 권으로 묶은 모음집.
필수 필드: date 또는 year, editor, title
선택 필드: chapter, eid, mainsubtitle, maintitle, maintitleaddon, pages, pagetotal, part, volume |
| | mvcollection | collection의 다른 모든 조건을 만족하되 여러 권으로 된 문헌.
필수 필드: date 또는 year, editor, title 선택 필드: pagetotal |
| | incollection | 모음집을 이루는, 완결되고 독립된 한 부분. 저자와 제목이 명시되어 있음.
필수 필드: author, booktitle, date 또는 year, editor, title
선택 필드: booksubtitle, booktitleaddon, chapter, eid, mainsubtitle, maintitle, maintitleaddon, pages, part, volume |
| | suppcollection | collection의 보조적인 자료. suppbook의 collection 버전. 표준 스타일에서는 incollection과 동일하게 처리됨. |
| dataset | | 주로 원시 데이터로 구성된 데이터 집합 또는 이와 비슷한 모음.
필수 필드: author 또는 editor, date 또는 year, title
선택 필드: edition, location, number, organization, publisher, series, type, version |

다음 페이지에서 계속

표 16.3, 앞 페이지에서 계속됨

| 문헌의 종류 | | 설명 |
|--------|----------------|--|
| manual | | 기술적인 또는 기타 문서. 반드시 인쇄된 형태일 필요는 없음.
필수 필드: author 또는 editor, date 또는 year, title
선택 필드: chapter, edition, eid, isbn, location, number, organization, pages, pagetotal, publisher, series, type, version |
| misc | | 다른 어느 종류에도 맞지 않는 문헌.
필수 필드: author 또는 editor, date 또는 year, title
선택 필드: howpublished, location, month, organization, type, version |
| online | | 웹 페이지 등 인터넷 상의 자료.
필수 필드: author 또는 editor, date 또는 year, doi 또는 eprint 또는 url, title
선택 필드: month, organization, version |
| patent | | 특허 또는 특허 신청 정보.
필수 필드: author, date 또는 year, number, title
선택 필드: holder, location, month, type, version |
| 간행물 | periodical | 저널의 특별호 등 정기 간행물의 특정 호수.
필수 필드: date 또는 year, editor, title |
| | suppperiodical | periodical의 보조적인 자료. suppbok의 periodical 버전. 표준 스타일에서는 article과 동일하게 처리됨. |
| 임시문헌 | proceedings | 단권으로 이루어진 학회 프로시딩.
필수 필드: date 또는 year, title
선택 필드: chapter, eid, mainsubtitle, maintitle, maintitleaddon, pages, pagetotal, part, volume |
| | mvproceedings | proceedings의 다른 모든 조건을 만족하되 여러 권으로 된 문헌.
필수 필드: date 또는 year, title 선택 필드: pagetotal |
| | inproceedings | incollection처럼, 학회 프로시딩을 이루는 문서 하나.
필수 필드: author, booktitle, date 또는 year, title
선택 필드: booksubtitle, booktitleaddon, chapter, eid, mainsubtitle, maintitle, maintitleaddon, pages, pagetotal, part, volume |
| 참고 자료 | reference | 백과사전이나 사전과 같이, 단권으로 이루어진 참고 자료. 표준 스타일에서는 collection과 동일하게 처리됨. |
| | mvreference | reference의 다른 모든 조건을 만족하되 여러 권으로 된 문헌. 표준 스타일에서는 mvcollection과 동일하게 처리됨. |
| | inreference | 참고 자료에 속한 하나의 문서. 표준 스타일에서는 incollection과 동일하게 처리됨. |

다음 페이지에서 계속

표 16.3, 앞 페이지에서 계속됨

| 문헌의 종류 | 설명 |
|-------------|---|
| report | 대학교나 다른 기관에서 출판한 기술적 보고서, 연구 보고서, 또는 기타 논문.
필수 필드: author, date 또는 year, institution, title, type
선택 필드: chapter, eid, isrn, location, month, note, number, pages, pagetotal, version |
| software | 컴퓨터 소프트웨어. 기본 스타일에서는 misc와 동일하게 취급됨. 표준 스타일에서는 misc와 동일하게 처리됨. |
| thesis | 교육 기관에서 학위를 따기 위해 작성한 학위 논문.
필수 필드: author, date 또는 year, institution, title, type
선택 필드: chapter, eid, isbn, location, month, pages, pagetotal |
| unpublished | 저자와 제목이 있지만 공식적으로 출판되지는 않은, 원고나 대본 등의 문헌.
필수 필드: author, date 또는 year, title
선택 필드: eventdate, eventtitle, eventtitleaddon, howpublished, isbn, location, month, type, venue |
| custom[a-f] | 특별한 스타일을 위해 사용할 수 있는 예비 문헌. 표준 스타일에서는 misc와 동일하게 취급됨. |

표의 각 칸에 나열한 필드 외에도, 다음 필드들은 여러 문헌의 공통 선택 필드로써 사용됩니다.

- 모든 문헌: doi, eprint, eprintclass, eprinttype, language(patent 문헌 제외), note, pubstate, subtitle, titleaddon, url, urldate 필드
- 기사 및 간행물: editor[a-c], issn, issue, issuesubtitle, issuetitle, issuetitleaddon, month, number, series, volume 필드
- 책: afterword, annotator, commentator, edition, editor, editor[a-c], foreword, introduction, isbn, location, number, origlanguage, publisher, series, translator, volumes 필드
- 모음집: afterword, annotator, commentator, edition, editor[a-c], foreword, introduction, isbn, location, number, origlanguage, publisher, series, translator, volumes 필드
- 프로시딩: editor, eventdate, eventtitle, eventtitleaddon, isbn, location, month, number, organization, publisher, series, venue, volumes 필드

이외에도, 문서의 설정에 따라 여기에 언급되지 않은 필드가 추가로 사용될 수 있습니다.

■ 문헌 목록에서의 인용 표지 조판 여부

■ 각 문헌의 세부 정보를 조판할 때의 규칙

Bib_{TEX}은 위의 각 항목이 설정된 스타일 파일을 제공하며, biber/biblatex을 사용하는 경우에는 biblatex 패키지의 옵션을 조정함으로써 스타일을 변경할 수 있습니다.

문헌 정보 파일과 스타일을 지정하고 필요한 문헌을 인용하여 L^AT_EX 소스 파일을 작성하였다면, 아래와 같이 그림 16.1에서 설명한 3단계의 컴파일 과정을 거치면 됩니다.

latex → bibtex 또는 biber → latex

16.3.1 BibTeX을 사용하는 경우

BibTeX을 사용하는 경우에는 문헌 정보 파일을 지정할 때 `\bibliography` 제어 문자열을, 스타일을 지정할 때 `\bibliographystyle` 제어 문자열을 사용합니다. `\bibliography` 제어 문자열이 입력된 곳에는 참고 문헌 목록도 조판됩니다.

`\bibliography{<files>}`

`<files>`: 사용할 문헌 정보 파일의 이름. 확장자(.bib)는 제외하고 입력합니다.

`\bibliographystyle{<style>}`

`<style>`: 사용할 문헌 참조 스타일의 이름

여러 문헌 정보 파일을 동시에 사용할 수도 있으며, 이때에는 각 파일을 쉼표로 구분하여 `<files>` 인자에 입력해 줍니다. 사용할 문헌 정보 파일을 지정했다면, 이 파일들에 저장되어 있는 문헌을 인용할 수 있게 됩니다. `<style>` 인자에는 L^AT_EX의 표준 참조 스타일

abbrv acm alpha apalake ieetr plain siam
unsrt

를 비롯하여, 200개 이상의 스타일 중 원하는 것을 입력하면 됩니다. [49]에서 사용할 수 있는 모든 스타일과 조판 예시를 확인할 수 있습니다.

예를 들어, 문헌을 plain 스타일을 사용해 표시하며, mybib.bib 파일을 문헌 정보 파일로 사용하려면 전처리부에는

```
\bibliographystyle{plain}
```

를 입력하고, 본문의 문헌 목록을 출력하려는 부분에서는

```
\bibliography{mybib}
```

을 입력해 주면 됩니다.

16.3.2 biber/biblatex을 사용하는 경우

biber/biblatex을 사용하는 경우에는 문헌 정보 파일을 지정할 때 `\addbibresource` 제어 문자열을 사용하며, 스타일은 biblatex 패키지의 옵션으로 지정합니다. 참고 문헌 목록을 조판할 곳에서는 `\printbibliography` 제어 문자열을 입력해 줍니다.

`\bibresource[⟨options⟩]{⟨file⟩}`

⟨options⟩: 문헌 정보 파일을 불러올 때 사용할 옵션

⟨file⟩: 불러올 문헌 정보 파일

`\printbibliography[⟨options⟩]`

⟨options⟩: 문헌 목록에 대한 옵션

`\addbibresource` 제어 문자열은 `\bibliography`와는 달리 하나의 파일만을 지정할 수 있으며, 파일 이름에 확장자를 포함해야 합니다. 문헌 정보 파일을 여러 개 사용하려는 경우에는 이 제어 문자열을 여러 번 사용하면 됩니다.

문헌 정보 파일을 불러올 때의 옵션, 문헌 목록에 대한 옵션, 스타일에 관련된 패키지 옵션은 `biblatex` 패키지 안내서[24]를 참조하시기 바랍니다.

예를 들어, `mybib.bib` 파일을 문헌 정보 파일로 사용하려면 전처리부에

```
1 \usepackage{biblatex}
2 \addbibresource{mybib.bib}
```

을 입력하고, 참고 문헌 목록을 조판할 곳에 `\printbibliography`를 입력하면 됩니다.

16.3.3 문헌 인용하기

참고 문헌 목록에 나열되는 문헌은 본문에서 `\cite`를 사용하여 인용한 문헌들입니다. 문헌을 인용하려는 곳에서 `\cite` 제어 문자열을 입력하면 해당 문헌의 인용 표지가 조판됩니다.

`\cite[⟨note⟩]{⟨key⟩}`

⟨note⟩: 인용할 문헌에 관한 추가 정보

⟨key⟩: 인용하려는 문헌의 식별자

예를 들어, 식별자가 `texbook`인 문헌을 인용하려면 `\cite{texbook}`을 입력합니다. 스타일에 따라 `[1]`, `[Knu86]` 등이 표시될 것입니다.

선택 인자를 지정하면 문헌에서 인용하려는 페이지 등의 추가 정보를 함께 표시할 수 있습니다. 예를 들어, `\cite[2--5]{texbook}`라고 입력하면 `[1, pp. 2-5]`나 `[Knu86, pp. 2-5]`가 표시될 것입니다.

여러 문헌을 동시에 인용할 수도 있으며, 이 경우 ⟨key⟩에 각 문헌의 식별자를 쉼표로 구분해 입력하면 됩니다. 예를 들어, 식별자가 `texbook`인 문헌과 `latexdps`인 문헌을

동시에 인용하려면 `\cite{texbook, latexdps}`를 입력합니다.

본문에서 인용하지 않았더라도, 중요한 문헌은 참고 문헌 목록에 따로 추가해 줄 수도 있습니다. 이때에는 `\nocite` 제어 문자열을 사용합니다.

`\nocite{<keys>}`

`<keys>`: 목록에 추가할 문헌들의 식별자

`\cite`와 마찬가지로, 각 문헌의 식별자는 쉼표를 사용해 구분합니다. 인자로 별표 `*`를 입력하면 문헌 정보 파일에 있는 모든 문헌이 목록에 추가됩니다.

제 17 장

PDF 파일의 링크와 속성

PDF 문서에는 다양한 메타데이터를 저장할 수 있으며, 문서를 읽는 사람의 편의를 위해 하이퍼링크나 책갈피 등 다양한 편의 기능을 추가할 수 있습니다. `hyperref` 패키지는 `LaTeX`에서 이를 가능하게 해 줍니다. 이 패키지를 불러오는 것만으로 필요한 부분에 하이퍼링크가 만들어지고 책갈피가 생성되지만, 직접 이들 요소를 편집하거나 메타데이터를 추가할 수도 있습니다. 이번 장에서는 이러한 방법을 알아봅니다.

참고 | `hyperref` 패키지는 `\ref`나 `\cite` 등 많은 `LaTeX`의 제어 문자열을 재정의하므로, 몇 가지 패키지(`kotex`, `cleveref` 등)를 제외하고는 가장 마지막 순서로 불러와야 합니다.

17.1 하이퍼링크

하이퍼링크 `hyperlink`는 문서의 탐색을 돕는 대표적인 기능입니다. 문자열이나 이미지 등의 대상에 하이퍼링크가 설정되면 PDF 파일 뷰어에서 그 대상을 클릭 또는 터치할 수 있게 됩니다. 이 동작을 통해 연결된 곳으로 바로 이동할 수 있습니다. 패키지를 불러올 때 특별한 설정을 하지 않았다면 하이퍼링크가 설정된 부분에는 색깔 상자가 표시되며, 이 상자는 문서를 종이에 출력할 때에는 인쇄되지 않습니다.

17.1.1 자동으로 생성되는 하이퍼링크

`hyperref` 패키지를 불러 오면 참조 기능이 사용된 부분에 자동으로 하이퍼링크가 만들어집니다. 대표적으로, 15.2.4절에서 설명했던 `\ref`나 `\pageref` 등을 사용해 상호 참조한 번호에 하이퍼링크가 생성됩니다. 이 하이퍼링크는 참조 대상이 있는 위치로 연결됩니다. 이외에도 각주, 목차, 인용 표지, 색인의 페이지 번호 등에 하이퍼링크가

설정됩니다. 상호 참조를 하되 하이퍼링크를 만들고 싶지 않을 때에는 `\ref`, `\Ref`, `\pageref` 대신 `\ref*`, `\Ref*`, `\pageref*`를 사용하면 됩니다. 한편, 문서의 하이퍼링크 자동 설정 여부는 패키지를 불러올 때 옵션을 지정함으로써 변경할 수 있습니다. 자세한 내용은 17.3절을 참조하십시오.

제15장에서 설명한 `\varioref` 패키지나 `\cleveref` 패키지도 `hyperref` 패키지와 함께 사용할 수 있습니다. `\vref`나 `\cref` 등의 제어 문자열로 조판한 참조 번호와 페이지 번호에 하이퍼링크가 삽입되며, 별표 붙은 제어 문자열을 사용하면 하이퍼링크가 만들어지지 않습니다. 각 패키지에서 정의하는 별표 붙은 제어 문자열은 다음과 같습니다.

- `varioref` 패키지 `\vref*`, `\Vref*`, `\vpageref*`, `\vpagerefrange*`
- `cleveref` 패키지 `\cref*`, `\Cref*`, `\crefrange*`, `\Creffrange*`
- `hyperref` 패키지 `\autoref*`, `\autopageref*`

이외에, `varioref` 패키지와 `cleveref` 패키지를 동시에 불러왔다면 본래 `varioref` 패키지에서 제공하던 명령어를 확장한

`\vrefrange*` `\Vrefrange*` `\fullref*` `\Fullref*`

도 정의됩니다.

17.1.2 하이퍼링크 직접 만들기

패키지가 자동으로 생성해 주는 하이퍼링크 외에, 독자가 문서를 편리하게 탐색할 수 있도록 하이퍼링크를 직접 추가할 수도 있습니다.

문서 내의 하이퍼링크

`\hyperref` 제어 문자열은 `\label`로 만든 식별자가 있는 곳으로 이동하는 하이퍼링크를 만듭니다. 이때 식별자의 번호 대신 원하는 문자열에 링크를 설정할 수 있습니다.

```
\hyperref[<identifier>]{<text>}
```

<identifier>: 하이퍼링크를 통해 이동할 대상의 식별자

<text>: 하이퍼링크를 설정할 문자열

다음 예시에서 ‘앞 장’을 클릭 또는 탭 하면 참고 문헌을 다루었던 제16장으로 연결됩니다.

1 `\hyperref[chap:bibliography]{앞~장}`에서는 `\LaTeX`에서 참고 문헌 목록을 만드는 방법을 알아보았습니다.

앞 장에서는 `LATEX`에서 참고 문헌 목록을 만드는 방법을 알아보았습니다.

상호 참조 가능한 대상 외에도, 문서 내 임의의 위치로 이동하는 하이퍼링크를 만들 수도 있습니다. 상호 참조할 때 참조하려는 대상에 `\label`로 식별자를 만들고 `\ref`로 그 대상을 참조했던 것처럼, `\hypertarget` 제어 문자열로 이동하려는 위치에 식별자를 붙여 두고 `\hyperlink` 제어 문자열로 하이퍼링크를 만듭니다.

`\hypertarget{<identifier>}{<text>}`

`<identifier>`: 현재 위치에 붙일 식별자
`<text>`: 현재 위치에 표시할 텍스트

`\hyperlink{<identifier>}{<text>}`

`<identifier>`: 이동할 위치에 붙인 식별자
`<text>`: 하이퍼링크를 삽입할 텍스트

제15장에서 알아본 `\label`의 식별자와 마찬가지로, `\hypertarget`에 사용하는 식별자도 고유해야 합니다.

이번 장의 첫 번째 문단에는 다음과 같이 코드를 작성하여 식별자를 붙여 두었습니다.

`\hypertarget{pdfdocument}{PDF~문서}`에는 다양한 메타데이터를 저장할 ...

아래 예시에서 ‘이번 장’에는 하이퍼링크가 만들어지며, PDF 뷰어에서 이 부분을 클릭 또는 탭 하면 해당 위치로 연결됩니다.

1 PDF~문서에 하이퍼링크를 설정하는 방법은 `\hyperlink{pdfdocument}{이번 장}`에서 다릅니다.

PDF 문서에 하이퍼링크를 설정하는 방법은 이번 장에서 다릅니다.

문서 외부로의 하이퍼링크

문서 외부로의 하이퍼링크를 만들 때에는 `\href` 또는 `\url` 제어 문자열을 사용합니다. `\href` 제어 문자열은 하이퍼링크를 설정할 텍스트를 따로 지정할 수 있으며, `\url` 제어 문자열은 참조 대상의 URL을 조판하면서 여기에 하이퍼링크를 설정해 줍니다.

`\href{<url>}{<text>}`

`\url{<url>}`

`<url>`: 이동할 대상의 URL

`<text>`: 링크를 삽입할 내용

‘http:’나 ‘https:’, ‘ftp:’ 등으로 시작하는 URL을 입력하면 그 웹 페이지로 열리는 하이퍼링크가 만들어집니다. ‘mailto:’를 입력한 뒤 전자 우편 주소를 작성하면 해당 주소로 전자 우편을 보낼 수 있는 화면이 열리는 하이퍼링크가 생성됩니다. URL을 작성할 때에는 TeX의 특수 문자 중 URL에 자주 쓰이는 문자인 #, %, _를 제어 문자열을 사용하지 않고 바로 입력할 수 있습니다. 단, \href나 \url이 다른 제어 문자열이나 환경의 인자로 사용되면 이들 문자를 바로 입력할 수 없습니다.

- 1 \TeX 을 설치하려면 \url{https://tug.org}를 방문하십시오. \par
- 2 이 매뉴얼에 관한 문의 사항이 있다면 \href{mailto:applemincho@kaist.ac.kr}{여기}를 클릭하여 저자에게 전자 우편을 보내시면 됩니다.

TeX을 설치하려면 <https://tug.org>를 방문하십시오.

이 매뉴얼에 관한 문의 사항이 있다면 여기를 클릭하여 저자에게 전자 우편을 보내시면 됩니다.

하이퍼링크를 통해 특정 동작을 수행하도록 할 수도 있습니다. ‘file:’을 입력한 뒤 연결하려는 파일의 경로와 이름을 입력하면 해당 파일이 열리는 하이퍼링크가 삽입됩니다. 이때 PDF 문서를 열도록 하는 경우, 해시 기호 # 뒤에 앵커의 이름을 입력하면 앵커
→ 252쪽 해당 위치가 표시되도록 할 수도 있습니다. 예를 들어,

```
... \href{file:manual.pdf#chapter.3}{매뉴얼의 제3장}을 ...
```

를 입력했을 때, 만들어진 하이퍼링크는 현재 폴더에 있는 manual.pdf 파일의 제3장을 표시합니다.

이외에도, ‘run:’을 입력한 후 파일의 이름을 입력하면 해당 파일을 여는 하이퍼링크가 만들어집니다. PDF 파일은 file: 명령어를 사용하여 하이퍼링크를 만들어도 잘 열리지만, 그 외의 확장자를 가지는 파일은 run: 명령어를 사용해 하이퍼링크를 만드는 것이 좋습니다.

예를 들어, 다음 코드는 현재 폴더 하위의 images 폴더에 있는 roadmap.png 파일을 여는 하이퍼링크를 만듭니다.

```
... \href{run:images/roadmap.png}{로드맵 이미지}를 ...
```

URL을 적절한 글꼴로 조판하되, 하이퍼링크를 설정하고 싶지 않다면 \nolinkurl 제어 문자열을 \url 대신 사용할 수 있습니다. 앞의 예시와 달리, 아래 예시에서는 URL을 클릭 또는 탭 하더라도 웹 페이지로 연결되지 않습니다. (PDF 뷰어에 따라 프로그램 자체에서 URL을 인식하고 웹 페이지로 연결해 줄 수도 있습니다.)

`\nolinkurl{https://latex-project.org}`에서는 `\LaTeX\` 관련 매뉴얼을 제공합니다.

`https://latex-project.org`에서는 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 관련 매뉴얼을 제공합니다.

17.2 책갈피

많은 PDF 뷰어는 책갈피 bookmark 기능을 지원합니다. 문서의 목차가 책갈피로 만들어지며, PDF 뷰어의 책갈피 창에서 표제를 클릭하면 해당 표제가 있는 위치로 바로 이동할 수 있습니다. 그림 17.1은 이 안내서를 Adobe Acrobat에서 열었을 때의 책갈피 창을 보여 줍니다.

더 다양한 옵션을 사용할 수 있는 책갈피 기능이 필요하다면 하이코 오버디엑(Heiko Oberdiek)이 작성한 bookmark 패키지[35]를 참조하시기 바랍니다.

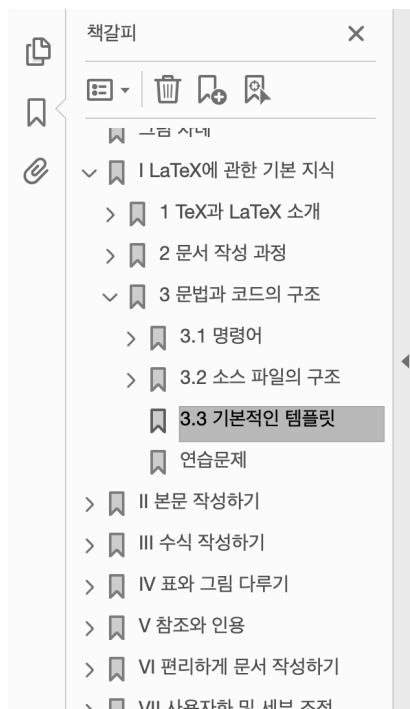


그림 17.1 | Adobe Acrobat의 책갈피 창

17.2.1 대체 문자열 지정

책갈피 창에는 일반 문자만을 사용할 수 있으며, 문자에 서식을 지정하거나 수식을 작성할 수 없습니다. 따라서 장절 표제에 복잡한 $\text{T}_{\text{E}}\text{X}$ 명령어나 수식을 입력했다면 책갈피 창에는 표제가 제대로 표시되지 않을 수 있습니다. 이때에는 `\texorpdfstring` 제어 문자열을 사용해 문서를 조판할 때 사용할 $\text{T}_{\text{E}}\text{X}$ 코드와 책갈피 항목에 사용할 내용을 구분하여 지정해 줄 수 있습니다.

`\texorpdfstring{ $\langle\text{T}_{\text{E}}\text{X text}\rangle$ }{ $\langle\text{PDF text}\rangle$ }`

$\langle\text{T}_{\text{E}}\text{X text}\rangle$: 문서를 조판할 때 사용할 $\text{T}_{\text{E}}\text{X}$ 코드

$\langle\text{PDF text}\rangle$: 책갈피를 만들 때 사용할 문자열

예를 들어, 절 표제에 $a^2 + b^2 = c^2$ 라는 수식을 작성해야 한다면 코드를 다음과 같이 작성할 수 있습니다.

```
\section{피타고라스 정리: \texorpdfstring{(a^2+b^2 = c^2)}}
{a\texttt{twosuperior}\ + b\texttt{twosuperior}\ = c\texttt{twosuperior}}
```

이렇게 코드를 작성하여 얻은 PDF 문서의 책갈피에는 수식 부분이 ‘ $a^2 + b^2 = c^2$ ’로 표시될 것입니다.

17.2.2 직접 책갈피 만들기

원하는 위치에 직접 책갈피를 만들 수도 있습니다. 이때 사용할 수 있는 가장 기본적인 제어 문자열은 `\pdfbookmark`입니다.

```
\pdfbookmark[<level>]{<text>}{<anchor>}
```

`<level>`: 책갈피 항목의 수준

`<text>`: 책갈피로 표시할 내용

`<anchor>`: 파일 내부에서 사용할 앵커

책갈피 항목의 수준을 지정하지 않으면 0이 사용되며, 값이 커질수록 하위 항목이 만들어집니다. `\chapter`, `\section`, `\subsection`의 수준은 0, 1, 2이며, `\part`의 수준은 article 클래스에서는 0, report나 book 클래스에서는 -1입니다. 앵커의 이름을 지정하는 `<anchor>` 인자는 각 책갈피마다 고유하게 지정되어야 합니다.

한편, 선택 인자를 입력받지 않는 세 가지 제어 문자열도 있습니다.

```
\currentpdfbookmark{<text>}{<anchor>}
```

```
\subpdfbookmark{<text>}{<anchor>}
```

```
\belowpdfbookmark{<text>}{<anchor>}
```

`<text>`: 책갈피로 표시할 내용

`<anchor>`: 파일 내부에서 사용할 앵커

`\currentpdfbookmark` 제어 문자열을 사용하면 현재 수준(주로 마지막으로 만들어진 책갈피와 같은 수준)의 책갈피가 만들어집니다. `\subpdfbookmark` 제어 문자열은 현재 수준을 기준으로 하위 수준의 책갈피가 만들며, 현재 수준을 1단계 증가시킵니다. `\belowpdfbookmark` 제어 문자열도 현재 수준보다 한 단계 낮은 책갈피를 만들지만, 현재 수준의 값을 변경하지 않습니다.

17.2.3 책갈피의 앵커

각각의 책갈피 항목에는 앵커 `anchor`라는 식별자가 있습니다. 이는 파일 내부에서 각 책갈피 항목을 구분하는 역할을 하며, 앞서 설명했던 `\href` 등의 제어 문자열을 사용해

책갈피가 만들어진 지점으로 이동하는 하이퍼링크를 만드는 데 사용할 수 있습니다.

LaTeX으로 만들어진 PDF 문서는 앵커의 이름이 일정한 형식으로 지정됩니다. `\part`, `\chapter`나 `\section` 등의 장절 표제에 의해 만들어진 앵커는 `'<heading>.<number>'`의 형식을 따릅니다. 이때 `<heading>`은 `part`, `chapter`, `section`처럼 해당 표제를 만들 때 사용한 명령어의 이름이며, `<number>`는 3, 4.1, 5.2.3 등 해당 표제의 번호입니다. 예를 들어, `article` 클래스 문서에서 5.3절의 앵커 이름은 `subsection.5.3`이며, `book` 클래스 문서에서 제 III 편의 앵커 이름은 `part.3`입니다.

17.2.2절의 명령어를 사용해 책갈피와 앵커를 직접 만든 경우에는 `'<name>.<level>'`의 형식을 따릅니다. `<name>`은 제어 문자열의 마지막 인자로 지정했던 문자열이며, `<level>`은 책갈피 항목의 수준을 나타내는 값입니다. 예를 들어,

```
\pdfbookmark[2]{연습용 책갈피}{myanchor}
```

를 입력해 책갈피를 만들었다면, 이 책갈피의 앵커는 `myanchor.2`입니다.

이외에도, Adobe Acrobat 등의 프로그램에서는 책갈피 항목의 속성을 표시하도록 하여 앵커의 이름을 직접 확인할 수도 있습니다.

17.3 패키지의 옵션

`hyperref` 패키지는 여러 옵션을 제공함으로써 하이퍼링크나 북마크 기능의 설정을 변경할 수 있도록 합니다. 대부분의 옵션은 키-값 문법을 사용하여 지정하며, 다음 두 가지 위치에서 지정해 줄 수 있습니다.

- 패키지를 불러올 때 언제나처럼 `\usepackage` 제어 문자열의 선택 인자로 옵션을 지정하면 됩니다.

```
\usepackage[<options>]{hyperref}
```

- 패키지를 불러온 후 `\hypersetup` 제어 문자열의 인자로 옵션을 지정하면 됩니다. `document` 환경 안에서도 사용할 수 있지만, 일반적으로는 전처리부에서 사용합니다.

```
\hypersetup{<options>}
```

이번 절에서는 `hyperref` 패키지에서 제공하는 옵션 중 자주 쓰이는 일부를 설명합니다. 패키지가 제공하는 모든 옵션은 패키지 안내서[40]를 참조하시기 바랍니다.

17.3.1 자동 하이퍼링크 설정 범위

다음 옵션들은 `hyperref` 패키지를 불러왔을 때 자동으로 하이퍼링크를 설정할 범위를 변경합니다.

- `implicit` 옵션 `LaTeX`의 제어 문자열을 재정의할지를 결정합니다. 값으로 `true`를 지정하면 이들을 재정의하지 않게 되어, 상호 참조한 번호, 인용 표지 등 앞에서 설명한 각종 대상에 하이퍼링크가 자동으로 설정되지 않으며, 원하는 곳에서 직접 하이퍼링크를 만들어 주어야 합니다. 기본값은 하이퍼링크가 자동으로 만들어지도록 하는 `false`입니다.
- `hyperfootnotes` 옵션 각주 번호의 하이퍼링크 설정 여부를 결정합니다. 값으로 `true`를 지정하면 링크를 설정하고, `false`를 지정하면 링크를 설정하지 않습니다. 기본값은 `true`입니다.
- `hyperindex` 옵션 색인에 있는 페이지 번호의 하이퍼링크 설정 여부를 결정합니다. 값으로 `true`를 지정하면 링크를 설정하고, `false`를 지정하면 링크를 설정하지 않습니다. 기본값은 `true`입니다.
- `linktoc` 옵션 목차의 표제와 페이지 번호 중 어느 부분에 링크를 설정할지 결정합니다. 값으로는 하이퍼링크를 설정하지 않는 `none`, 표제에만 링크를 설정하는 `section`, 페이지 번호에만 링크를 설정하는 `page`, 표제와 페이지 번호 모두 링크를 설정하는 `all`을 지정할 수 있습니다. 기본값은 `section`입니다.

17.3.2 하이퍼링크 표시 방법

`hyperref` 패키지를 불러 오면 하이퍼링크가 설정된 곳에 색깔 상자가 그려집니다. 다음 옵션들은 PDF 파일에 하이퍼링크를 표시하는 방식을 변경합니다.

- `colorlinks` 옵션 하이퍼링크가 설정된 부분에 색깔 상자를 그리는 대신 글자의 색상을 변경합니다. `colorlinks=false`를 입력하면 기본 설정과 같이 색깔 상자를 사용하여 하이퍼링크를 표시합니다.
- `hidelinks` 옵션 하이퍼링크가 설정된 부분에 색상 표시가 나타나지 않습니다. 즉, 글자의 색도 변경되지 않으며, 색깔 상자도 그려지지 않습니다. 이 옵션은 값을 입력받지 않습니다.
- `frenchlinks` 옵션 하이퍼링크가 설정된 부분의 문자를 작은 대문자 스타일로 표시합니다.

하이퍼링크에 색상 표시를 하는 경우, 옵션을 통해 이때 사용되는 색상도 변경할 수 있습니다. 이에 대한 설명은 패키지의 안내서를 참조하시기 바랍니다.

17.3.3 책갈피 관련 옵션

다음 옵션들은 책갈피 기능과 관련된 것들입니다.

- **bookmarks** 옵션 책갈피를 생성할지의 여부를 설정합니다. 값을 true로 지정하면 책갈피를 만들고, false로 지정하면 책갈피를 만들지 않습니다. 기본값은 true입니다.
- **bookmarksnumbered** 옵션 책갈피의 이름에 장절 표제의 번호를 넣을지 결정합니다. 값을 true로 지정하면 번호를 포함하고, false로 지정하면 번호를 포함하지 않습니다. 기본값은 false입니다.
- **bookmarkstype** 옵션 책갈피를 생성할 때 기준으로 삼을 목차를 지정합니다. 값으로는 사용할 목차 파일의 확장자를 입력하면 됩니다. 기본값은 장절 표제의 목차 정보를 사용하는 toc이며, lot를 입력하면 표 목차를, lof를 입력하면 그림 목차를 바탕으로 책갈피가 만들어집니다.

이 책에서는 패키지를 불러올 때 다음의 옵션을 지정하였습니다.

```
1 \usepackage[
2   linktoc = all, hidelinks, bookmarksnumbered = true
3 ]{hyperref}
```

17.3.4 PDF 파일의 메타데이터

hyperref 패키지는 PDF 파일의 제목, 저자, 기타 정보 등 메타데이터를 문서에 삽입하는 기능을 제공합니다. PDF 파일 뷰어는 문서의 제목이 설정되어 있다면 파일의 이름 대신 제목을 보여 주며, 속성 창을 열면 문서의 정보를 확인할 수 있습니다.

- **pdftitle** 옵션 문서의 제목을 지정합니다.
- **pdfauthor** 옵션 문서의 저자를 지정합니다.
- **pdfsubject** 옵션 문서의 주제를 지정합니다.
- **pdfkeywords** 옵션 문서의 키워드를 지정합니다. 각 키워드는 쉼표를 사용하여 구분합니다.

- **pdflang 옵션** RFC 3066 태그를 사용하여 문서의 언어를 지정합니다. 예를 들어, 한국어 문서는 ko-KR, 영어 문서는 en-US를 값으로 지정하면 됩니다.

이 책은 전처리부에 다음과 같이 입력하여 메타데이터를 저장하였습니다.

```
1 \hypersetup{
2   pdftitle={레이텍 입문부터 활용까지},
3   pdfauthor={강우현},
4   pdfsubject={Typesetting},
5   pdfkeywords={LaTeX, Korean, document},
6   pdflang=ko-KR
7 }
```


VI

편리하게 문서 작성하기

제 VI편에서는 문서를 편리하게 작성할 수 있도록 돕는 명령어를 설명합니다. 특히, 매크로를 정의하고 카운터를 다루는 등의 고급 기능을 알아봅니다. '논리적 구조'에 집중하여 문서를 작성하기 위해서는 이번 장의 기능을 자주 사용하게 될 것입니다.

제18장 매크로 정의하기 · 259

제19장 카운터와 길이 변수 · 269

제20장 문서 파일 분리하기 · 283

제 18 장

매크로 정의하기

이번 장에서는 ‘매크로’를 정의하는 방법을 알아봅니다. 매크로_{macro}는 긴 명령어를 짧게 줄여 쉽게 실행할 수 있도록 한 것으로, \LaTeX 으로 문서를 작성할 때에는 수많은 매크로를 사용하게 됩니다.

반복적인 작업을 매크로를 사용해 처리하면 문서를 쉽고 빠르게 작성할 수 있습니다. 예를 들어, (제3장에서 소개했듯) 이 책에서는 패키지의 이름을 작성할 때 `\Lpkg`라는 매크로를 사용하였습니다.

1 문서의 언어 설정에는 `\Lpkg{babel}`~패키지나 `\Lpkg{polyglossia}`~패키지를 사용합니다.

문서의 언어 설정에는 `babel` 패키지나 `polyglossia` 패키지를 사용합니다.

이 매크로는 다음과 같이 정의되어 있습니다. 즉, 인자로 입력된 패키지의 이름을 산세리프체로 조판하는 역할을 합니다.

```
\newcommand*\Lpkg[1]{\sffamily #1}
```

패키지의 이름을 더 강조하려면 `\Lpkg`의 정의를 다음과 같이 고칠 수 있습니다.

```
1 \usepackage{xcolor}
2 \newcommand*\Lpkg[1]{\colorbox{black!8}{\strut\sffamily\;#1\;}}
```

그러면 본문에서 패키지 이름을 입력한 곳을 일일이 찾아가서 고칠 필요 없이, 모든 패키지의 이름에 새로운 서식이 일괄 적용됩니다.

1 문서의 언어 설정에는 `\Lpkg{babel}`~패키지나 `\Lpkg{polyglossia}`~패키지를 사용합니다.

문서의 언어 설정에는 `babel` 패키지나 `polyglossia` 패키지를 사용합니다.

요컨대, 매크로는 문서의 코드를 줄여 주어 가독성을 높이고, 통일된 서식을 지정할 수 있게 해 줍니다. 이번 장에서는 매크로를 어떻게 정의하는지 알아봅니다.

18.1 제어 문자열 형태의 매크로

L^AT_EX의 매크로는 제어 문자열의 형태인 것과 환경의 형태인 것이 있습니다. 이번 절에서는 먼저 제어 문자열 형태의 매크로를 정의하는 방법을 알아봅니다.

제어 문자열 형태의 매크로를 정의할 때에는 `\newcommand(*)` 제어 문자열을 사용합니다. 간단한 매크로를 정의하는 것에서 시작해 이 제어 문자열을 사용하는 방법을 알아보겠습니다. 별표 붙은 것과 별표 없는 것의 차이도 뒤에서 설명할 것입니다.

```
\newcommand(*){<cname>}[<param no.>][<default arg.>]{<definition>}
```

<cname>: 정의할 제어 문자열

<param no.>: 사용할 매개 변수의 개수

<default arg.>: 첫 번째 매개 변수의 기본값

<definition>: 제어 문자열의 정의

18.1.1 세 가지 종류의 매크로

간단한 매크로

가장 먼저, 인자를 입력받지 않는 간단한 형태의 매크로를 정의해 봅시다. 문서에 내용을 입력해 주는 단축어 정도로 간단할 수도 있고, 문서의 레이아웃을 바꾸기 위해 선언형 명령어를 복합해 둔 것이 될 수도 있습니다.

수학 문서에서 가장 흔하게 사용하는 매크로는 ‘ \mathbb{R} ’과 같은 수 집합을 출력하는 것입니다. 수 집합은 자주 언급되는 대상이므로 ‘ \mathbb{R} ’처럼 짧은 명령어로 정의해 두면 문서를 편리하게 작성할 수 있습니다.

이처럼 간단한 매크로는 선택 인자를 지정할 필요 없이, 정의할 제어 문자열과 그 정의만을 입력해 주면 됩니다. 앞선 예시에서 ‘ \mathbb{R} ’는 `\mathbb{R}`로 입력하므로 다음과 같이 입력하면 됩니다.

```
\newcommand\R{\mathbb{R}}
```

이렇게 입력한 후 수식 모드에서 `\R`를 입력하면 실수 집합을 쉽게 입력할 수 있습니다.

1 실수 집합은 흔히 `\(\mathbb{R}\)`로 나타낸다.

실수 집합은 흔히 \mathbb{R} 로 나타낸다.

인자를 입력받는 매크로

인자를 입력받는 매크로를 정의하면 하나의 매크로를 여러 곳에 사용할 수 있습니다. 하나의 매크로는 최대 9개의 인자를 입력받을 수 있으며, 입력받으려는 인자의 수를 $\langle param no. \rangle$ 에 입력하면 됩니다.

$\langle definition \rangle$ 부분에 정의를 입력할 때, 각 인자는 매개 변수인 #1~#9를 사용해 나타냅니다. 매크로가 사용될 때 #1은 첫 번째 인자로, #2는 두 번째 인자로 치환됩니다. 정의한 매크로를 사용할 때에는 다른 \LaTeX 의 제어 문자열처럼 중괄호 $\{ \dots \}$ 를 사용해 인자를 지정합니다. 문자 하나나 제어 문자열 하나를 인자로 지정할 때에는 중괄호를 생략해도 된다는 규칙, 인자 사이에는 공백 문자를 입력해도 된다는 규칙도 동일하게 적용됩니다.

하나의 예시로, 분수 꼴의 미분 연산자를 쉽게 입력하기 위한 $\backslash dif$ 매크로를 정의해 봅시다. 다음 코드

```
\newcommand*\dif[2]{\frac{d\#1}{d\#2}}
```

를 입력하여 $\backslash dif$ 매크로를 정의하였다면 다음과 같이 사용할 수 있습니다. #1 자리에 f 가, #2 자리에 x 가 사용되어 미분 연산자가 입력되었음을 알 수 있습니다.

```
1 \((f(x) = \sin x)\)라고 했을 때
2 \[ \dif fx = \cos x \]
3 이다.
```

$f(x) = \sin x$ 라고 했을 때

$$\frac{df}{dx} = \cos x$$

이다.

선택 인자를 입력받는 매크로

한편, \LaTeX 의 많은 매크로는 선택 인자를 입력받기도 합니다. $\langle default arg. \rangle$ 인자를 지정하면 첫 번째 인자가 선택 인자로 바뀌며, 해당 인자를 생략한 경우에는 $\langle default arg. \rangle$ 에 입력된 내용이 첫 번째 인자의 기본값으로 사용됩니다.

앞서 정의했던 $\backslash dif$ 의 정의를 다음과 같이 바꾸어 봅시다.

```
\newcommand*\dif[3][\frac{d^{\#1}}{d^{\#2}}]{d^{\#3}}{\#1}}
```

기존의 정의에서 선택 인자를 입력받아 미분한 횟수를 나타내도록 한 것입니다. 선택 인자를 생략하면 특별한 표시가 나타나지 않습니다. 따라서 다음과 같이 사용할 수 있습니다.

```

1 \ (f(x) = \sin x\ )라고 했을 때
2 \ [ \dif fx = \cos x \text{이고} \quad \quad \quad
   \dif[2]fx = -\sin x \ ]
3 이다.

```

$f(x) = \sin x$ 라고 했을 때

$$\frac{df}{dx} = \cos x \text{이고} \quad \frac{d^2f}{dx^2} = -\sin x$$

이다.

18.1.2 기존 제어 문자열 재정의하기

`\newcommand(*)`를 사용할 때에는 $\langle csname \rangle$ 인자에 ‘새로운’ 제어 문자열, 즉 기존의 정의가 없는 제어 문자열을 입력해야 합니다. 기존에 이미 정의되어 있던 제어 문자열을 입력하면 컴파일 시에 오류가 발생하며, 정의도 변경되지 않습니다.

기존 매크로의 정의를 바꾸어 주려면 `\renewcommand(*)`를 대신 사용해야 합니다. 이 제어 문자열은 `\newcommand(*)`와 동일한 방법과 동일한 규칙으로 사용하며, $\langle csname \rangle$ 인자에는 이미 정의되어 있는 제어 문자열이 입력되어야 합니다. 오히려 정의되어 있지 않은 제어 문자열을 입력하면 오류가 발생합니다. 단, `\relax`는 재정의할 수 없습니다.

하지만, 자신이 정의하지 않은 제어 문자열 — 즉 \TeX , \LaTeX 이나 클래스, 패키지에서 제공하는 제어 문자열 — 은 되도록 재정의하지 않는 것이 좋습니다. \LaTeX 내부에서, 또는 다른 패키지에서 해당 제어 문자열을 중요하게 사용하고 있을 수도 있기 때문입니다. 특히, `\if`, `\or`, `\box`, `\def`, `\let`, `\the`, `\char`, `\null`, `\skip`, `\span`, `\break`, `\empty` 등은 절대로 재정의하면 안 됩니다.

매크로를 정의할 때에는 `\providecommand(*)` 제어 문자열을 사용할 수도 있습니다. 이는 `\newcommand(*)`처럼 새로운 매크로를 정의하지만, $\langle csname \rangle$ 인자에 이미 정의되어 있는 제어 문자열이 입력되면 매크로의 정의를 변경하지 않고 무시합니다.

18.1.3 세부 사항

이제, 앞서 설명하지 않았던 세부 사항을 설명하겠습니다. 정의 명령어의 별표 있는 것과 없는 것의 차이, 이름의 규칙, 기타 주의 사항 등을 여기에서 확인할 수 있습니다.

긴 인자

앞서 알아본 세 종류의 제어 문자열에는 별표 있는 것과 별표 없는 것이 모두 존재합니다. 별표의 유무는 ‘긴 인자’를 허용하는지를 나타냅니다. 긴 인자(long argument)는 빈

줄(또는 문단 나눔이나 `\par` 제어 문자열)을 포함하는 인자를 뜻합니다. 별표 없는 제어 문자열은 긴 인자를 허용하며, 별표 있는 제어 문자열은 긴 인자를 허용하지 않습니다.

인자가 없는 간단한 매크로를 정의할 때에는 `\newcommand`와 `\newcommand*` 중 어느 것을 사용하더라도 기능적으로 차이가 없습니다. 하지만, 인자가 있는 매크로를 정의할 때에는 그렇지 않습니다. 앞 예시에서 사용했던 것처럼 일반적으로는 `\newcommand*`를 사용하는 것이 좋으며, 특별히 긴 인자를 허용하고 싶을 때에만 `\newcommand`를 사용하는 것이 좋습니다. 인자를 지정하다가 실수를 하였더라도 더 쉽게 오류를 찾을 수 있기 때문입니다. 또, 같은 동작을 하는 환경을 정의할 수 있다면 긴 인자를 허용하는 제어 문자열 대신 환경을 정의하는 것이 효율적입니다.

매크로 이름의 규칙

`\newcommand(*)`나 `\providecommand(*)` 제어 문자열을 사용하여 매크로를 정의하려면 정의할 매크로의 이름을 $\langle csname \rangle$ 인자로 지정해 주어야 합니다. 여기에 입력한 제어 문자열을 $\langle name \rangle$ 이라고 할 때, 이 제어 문자열은 다음 조건을 만족해야 합니다.

- 이미 정의되어 있지 않아야 합니다.
- $\langle name \rangle$ 의 첫 세 글자가 ‘end’이면 안 됩니다.
- `\end $\langle name \rangle$` 이 정의되어 있지 않아야 합니다.

두 번째와 세 번째 규칙은 \LaTeX 이 내부적으로 환경을 정의하는 방식에서 기인합니다. 한편, `\renewcommand(*)`를 사용할 때에는 첫 번째 규칙 대신 다음 규칙이 사용됩니다.

- 이미 정의되어 있어야 합니다. 다만, `\relax`는 재정의할 수 없습니다.

고급 기능을 원한다면

자신만의 매크로 패키지를 만들고 싶다면 \LaTeX 의 `\newcommand`만으로는 기능이 부족할 수 있습니다. 플레인 \TeX 이나 다른 패키지, \LaTeX 3의 명령어를 사용하면 더 다양한 매크로를 정의할 수 있습니다. 여기에서 소개하는 기능에 대한 자세한 설명은 이 책의 범위를 벗어나므로, 관련 내용은 인용된 문헌을 참조하시기 바랍니다.

\TeX 은 제어 문자열을 정의하는 명령어로 `\def`를 제공하며, 다른 제어 문자열의 정의를 복사하는 `\let`도 사용할 수 있습니다. `\def`는 선택 인자 기능은 제공하지 않지만, 패턴 매칭 방식으로 인자를 입력받는 제어 문자열을 정의할 수 있습니다. 이들에 대한 자세한 사항은 [26]에서 확인할 수 있습니다.

참고 | `\def`와 `\let`은 제어 문자열이 이미 정의되어 있는지 검사하지 않으므로, 사용 시 각별히 주의해야 합니다. 일반적인 문서에서 매크로를 정의할 때에는 사용을 권장하지 않으며, 많은 매크로를 복잡하게 정의하고 사용하는 클래스나 패키지 파일에서만 사용하는 것이 좋습니다.

\LaTeX 3에서는 `\def`를 강화하여, 선택 인자 기능까지 추가한 `\NewDocumentCommand` 제어 문자열을 사용할 수 있습니다. 매크로의 정의를 복사할 때에는 `\let`과 비슷한 기능을 하는 `\NewCommandCopy`를 사용할 수 있습니다. 이들에 대한 자세한 내용은 [16]에서 확인할 수 있습니다.

다른 프로그래밍 언어처럼 조건문을 사용할 수도 있습니다. \TeX 의 `\if ... \fi` 구문은 \LaTeX 에서 사용할 수 있는 가장 원시적인 조건문이며, `ifthen` 패키지[10]를 불러오면 \TeX 의 조건문을 \LaTeX 의 문법에 익숙한 형태로 사용할 수 있습니다. 매크로 기능과 조건문, 19.1절의 카운터 기능을 재귀적으로 사용하면 반복문도 구현할 수 있습니다.

18.2 환경 형태의 매크로

환경 형태의 매크로를 정의할 때에는 `\newenvironment(*)` 제어 문자열을 사용합니다. 환경의 정의에 두 개의 인자를 사용하는 것을 제외하면, `\newcommand(*)`와 동일한 문법으로 사용합니다.

```
\newenvironment(*){<envname>}  
[<param no.>][<default arg.>]{<definition@begin>}{<definition@end>}
```

`<envname>`: 정의할 환경의 이름
`<param no.>`: 사용할 매개 변수의 개수
`<default arg.>`: 첫 번째 매개 변수의 기본값
`<definition@begin>`: 환경 시작 부분에서의 정의
`<definition@end>`: 환경 끝 부분에서의 정의

주의할 점으로, `<definition@end>` 인자에는 매개 변수 #1~#9를 사용할 수 없습니다. 정의한 환경을 사용할 때 `\begin{<envname>}`은 `<definition@begin>`에 입력한 내용으로, `\end{<envname>}`은 `<definition@end>`에 입력한 내용으로 치환됩니다.

`\newcommand*`와 마찬가지로, `\newenvironment*`를 사용해 정의한 환경은 긴 인자를 허용하지 않습니다. 반면, 환경 안의 내용은 인자가 아니므로, 여기의 내용을 작성할 때에는 별표의 유무에 관계 없이 빈 줄을 입력할 수 있습니다.

예를 들어, 과목을 소개하기 위한 `subjectinfo` 환경은 `tabular` 환경을 사용하여 다음과 같이 정의할 수 있습니다.


```

1 \newenvironment*{subjectinfo}[3]
2   {%
3     \par
4     \begin{tabular}{@{}p{\linewidth}@{}}
5       \hline
6       \textbf{#1} {\footnotesize #2}\hfill #3 \\ \relax
7     }
8     {%
9       \\ \hline
10      \end{tabular}\par
11   }

```

환경을 시작할 때에는 tabular 환경을 사용하여 폭이 현재 글줄의 길이인 1열짜리 표를 만들고(4행), 표의 위쪽에 가로줄을 그린 뒤(5행), 과목의 정보를 일정한 형식으로 조판하도록 하였습니다(6행). 세 개의 인자는 각각 과목 이름, 과목 코드, 강의 시간: 실험 시간:학점 수를 뜻합니다. 환경을 끝낼 때에는 표 아래쪽에 가로줄을 그려준 뒤(9행) tabular 환경을 끝냅니다(10행). 이렇게 정의한 환경은 다음과 같이 사용합니다.

```

1 \begin{subjectinfo}{선형대수학}{MAS212}
2   {3:0:3}
3 \end{subjectinfo}
4 \medskip
5 \begin{subjectinfo}{해석학~I}{MAS241}
6   {3:2:4}
7 \end{subjectinfo}

```

선형대수학 MAS212 3:0:3
 벡터 공간과 선형 변환을 바탕으로 하여,
 선형대수학과 관련된 다양한 내용을 다룹니다.

해석학 I MAS241 3:1:4
 실수 공간의 완비성을 다루고, 이에 따라
 실수 열의 극한, 급수의 수렴성, 실수 값 함
 수의 극한과 미분, 적분 등을 다룹니다.

환경 재정의하기

이미 존재하는 환경을 다시 정의하려면 `\renewenvironment(*)`를 동일한 방식으로 사용하면 됩니다. `\renewcommand`를 사용할 때와 마찬가지로, 재정의하려는 환경이 다른 곳에서 사용되고 있지는 않은지 주의하십시오.

18.3 유용한 매크로

자주 사용하는 기호나 문법은 매크로로 정의해 두면 문서를 작성할 때 요긴하게 사용할 수 있습니다. 이번 절에는 정의해 두면 편리한 매크로를 모아 두었습니다.

수 집합

수학 문서에서 정수 집합 \mathbb{Z} , 유리수 집합 \mathbb{Q} , 실수 집합 \mathbb{R} , 복소수 집합 \mathbb{C} 등은 빈번하게 사용됩니다. 수 집합을 칠판 볼드체 대신 볼드체를 사용하여 나타내기도 하며, 그런 경우에는 아래의 `\numsetfont`의 정의를 바꾸어 사용하면 됩니다.

```
1 \newcommand*\numsetfont{\mathbb}
2 \newcommand*\Z{\numsetfont Z} \newcommand*\Q{\numsetfont Q}
3 \newcommand*\R{\numsetfont R} \newcommand*\C{\numsetfont C}
```

연산자와 함수

특정 연산자는 기준이 되는 대상을 아래 첨자로 나타내기도 합니다. 차원을 나타내는 `dim`이 그 예로, X 의 \mathbb{R} -벡터 공간으로서의 차원을 $\dim_{\mathbb{R}} X$ 로 표기하는 식입니다. 특정 스칼라 체를 자주 사용한다면 이를 제어 문자열로 정의해 간단하게 입력할 수 있습니다.

```
1 \newcommand*\dimr{\dim_{\R}}
2 \DeclareMathOperator{\codim}{codim}
3 \newcommand*\codimr{\codim_{\R}}
```

```
1 \begin{gather*}
2 \quad \dim X, \quad \dimr X \\
3 \quad \codim X, \quad \codimr X \\
4 \end{gather*}
```

| | |
|-------------|-------------------------|
| $\dim X,$ | $\dim_{\mathbb{R}} X$ |
| $\codim X,$ | $\codim_{\mathbb{R}} X$ |

해석학 기호

변수가 한없이 커질 때의 극한, 미분과 편미분 기호, $e^{i\pi x}$ 꼴의 지수함수는 자주 사용됩니다.

```
1 \newcommand*\limn{\lim_{n\to\infty}}
2 \newcommand*\limN{\lim_{N\to\infty}}
3 \newcommand*\limm{\lim_{m\to\infty}}
4 \newcommand*\limi{\lim_{i\to\infty}}
5 \newcommand*\limj{\lim_{j\to\infty}}
6 \newcommand*\limsupn{\limsup_{n\to\infty}}
7 \newcommand*\liminfn{\liminf_{n\to\infty}}
8
9 \newcommand*\dif[2]{\frac{d\#1}{d\#2}}
10 \newcommand*\dl[3]{\frac{d\#1}{d\#2}\bigg|_{\#3}}
11 \newcommand*\pa{\partial}
```

```

12 \newcommand*{\bpa}{\bar\partial}
13 \newcommand*{\pdif}[2]{\frac{\pa#1}{\pa#2}}
14 \newcommand*{\pdl}[3]{\pdif{#1}{#2}\big|_{#3}}
15
16 \newcommand*{\eitx}{e^{2\pi itx}}
17 \newcommand*{\emitx}{e^{-2\pi itx}}

```

```

1 \begin{gather*}
2 \quad \Lambda \limn f_n = \limn \Lambda f_n \\
3 \quad \nabla f = \pdif fx{\hat{\mathrm x}} + \\
4 \quad \pdif fy{\hat{\mathrm y}} \\
5 \quad \intt f(\theta) \, d\theta \\
6 \quad \hat f(t) = \intr f(x) \emitx \, dx \\
7 \end{gather*}

```

$$\Lambda \lim_{n \rightarrow \infty} f_n = \lim_{n \rightarrow \infty} \Lambda f_n$$

$$\nabla f = \frac{\partial f}{\partial x} \hat{x} + \frac{\partial f}{\partial y} \hat{y}$$

$$df = \left. \frac{\partial f}{\partial x} \right|_y dx + \left. \frac{\partial f}{\partial y} \right|_x dy$$

$$h(x) = e^{-2\pi itx} dx$$

대형 연산자

합과 곱, 적분 등을 나타내는 대형 연산자는 특정 범위에서 자주 사용됩니다.

```

1 \newcommand*{\sumn}{\sum_{n=1}^{\infty}}
2 \newcommand*{\sumi}{\sum_{i=1}^n}
3 \newcommand*{\sumz}{\sum_{-\infty}^{\infty}}
4 \newcommand*{\prodn}{\prod_{n=1}^{\infty}}
5 \newcommand*{\prodi}{\prod_{i=1}^n}
6 \newcommand*{\cupn}{\bigcup_{n=1}^{\infty}}
7 \newcommand*{\cupi}{\bigcup_{i=1}^n}
8 \newcommand*{\capn}{\bigcap_{n=1}^{\infty}}
9 \newcommand*{\capi}{\bigcap_{i=1}^n}
10 \newcommand*{\intr}{\int_{-\infty}^{\infty}}
11 \newcommand*{\intrp}{\int_0^{\infty}}
12 \newcommand*{\intt}{\int_{-\pi}^{\pi}}

```

```

1 \begin{gather*}
2 \quad \sumn \frac{1}{n^2} = \frac{\pi^2}{6} \\
3 \quad \quad \quad A = \cupi A_i \\
4 \quad \intt f(\theta) \, d\theta

```

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad A = \bigcup_{i=1}^n A_i$$

$$\int_{-\pi}^{\pi} f(\theta) d\theta$$

제 19 장

카운터와 길이 변수

LaTeX은 변수를 사용하여 문서의 여러 요소를 편리하게 조판할 수 있습니다. 정수 값을 저장하는 변수는 ‘카운터’라고 부르며, 길이 값을 저장하는 ‘길이 변수’도 있습니다. 이번 장에서는 이 두 가지 변수에 대해 알아봅니다.

19.1 정수 값과 카운터

카운터(counter)는 정수 값을 기록하는 변수를 가리킵니다. 문서 내에서 표시되는 각종 번호 — 장절 표제, 그림과 표 등의 개체, 수식, 페이지의 번호 — 는 모두 카운터를 통해 관리됩니다. 또, 문서의 레이아웃과 관련된 일부 내부 변수도 카운터로써 관리됩니다.

19.1.1 카운터 기본 사용 방법

먼저, 카운터를 사용하는 기본 방법을 알아보겠습니다. 새로운 카운터를 만들고, 기존 카운터에 저장된 값을 변경하는 방법을 알아봅시다.

참고 | 이번 절에서 설명하는 제어 문자열 중 `\newcounter`, `\setcounter`, `\addtocounter`, `\stepcounter`, `\refstepcounter`는 전역 제어 문자열입니다. 전역 제어 문자열을 사용하는 경우, 특정 영역 안에서 사용하였더라도 그 효과는 영역 바깥까지 유효합니다.

새로운 카운터를 만들 때에는 `\newcounter` 제어 문자열을 사용합니다.

```
\newcounter{<ctrname>}[<parent counter>]
```

`<ctrname>`: 만들려는 카운터의 이름

`<parent counter>`: 현재 카운터를 종속시킬 카운터

카운터의 종속
→ 274쪽

`\newcounter`는 $\langle ctrname \rangle$ 인자를 이름으로 하는 카운터를 만들며, 이 카운터에는 기본값으로 0이 저장됩니다. $\langle ctrname \rangle$ 은 이미 정의되어 있는 다른 카운터의 이름과 겹치면 안 됩니다. 선택 인자인 $\langle parent\ counter \rangle$ 를 지정하면 이 카운터를 $\langle parent\ counter \rangle$ 카운터에 종속시킬 수 있습니다.

카운터에 저장된 값은 `\setcounter`나 `\addtocounter` 제어 문자열을 사용해 바꿀 수 있습니다. `\setcounter`는 카운터의 값을 지정된 값으로 바꾸고, `\addtocounter`는 카운터의 현재 값에 지정된 값을 더합니다.

```
\setcounter{\langle ctrname \rangle}{\langle integer \rangle}  
\addtocounter{\langle ctrname \rangle}{\langle integer \rangle}
```

$\langle ctrname \rangle$: 값을 바꾸려는 카운터의 이름
 $\langle integer \rangle$: 새로 저장할/원래의 값에 더할 정수 값

다음은 temp 카운터를 만들고, 그 값을 변경시키는 예시입니다.

| 코드 | 카운터에 저장된 값 |
|---------------------------------------|------------|
| <code>\newcounter{temp}</code> | 0 |
| <code>\setcounter{temp}{5}</code> | 5 |
| <code>\addtocounter{temp}{3}</code> | 8 |
| <code>\addtocounter{temp}{-20}</code> | -12 |

한편, 카운터로 번호를 매길 때에는 값을 보통 1씩 증가시킵니다. 이때에는 위의 제어 문자열 대신 `\stepcounter` 또는 `\refstepcounter` 제어 문자열을 사용합니다. `\refstepcounter`를 사용해 값을 1 증가시켰다면, 이후 `\label`을 사용하여 해당 카운터의 새로운 값에 식별자를 붙일 수 있습니다.

```
\stepcounter{\langle ctrname \rangle}  
\refstepcounter{\langle ctrname \rangle}
```

$\langle ctrname \rangle$: 값을 1 증가시킬 카운터의 이름

카운터에 저장된 값 사용하기

카운터에 저장된 값은 두 가지 방법으로 사용할 수 있습니다. 첫 번째는 저장된 값 자체를 다른 명령어의 인자 등으로 사용하는 것이고, 두 번째는 값을 문자의 형태로 문서에 조판하는 것입니다.

카운터에 저장된 값을 명령어의 인자 등으로 사용할 때에는 `\value` 제어 문자열을 사용합니다. 값을 사용하려는 카운터의 이름을 `\value`의 인자로 지정하면 됩니다. 예를 들어, firstcounter 카운터에 저장된 값을 secondcounter 카운터에 저장하려면

```
\setcounter{secondcounter}{\value{firstcounter}}
```

를 입력하면 됩니다.

카운터에 저장된 값을 문서에 조판할 때에는 카운터의 이름 앞에 ‘\the’를 붙입니다. 예를 들어, temp 카운터가 정의되어 있다면 \thetemp는 이 카운터에 저장되어 있는 값을 출력합니다. 특별한 설정을 하지 않은 경우, 이 명령어를 사용하면 카운터에 저장된 값을 아라비아 숫자로 출력합니다.

카운터 사용 예시

다음은 카운터 기능을 응용하여 정리를 작성하기 위한 theorem 환경을 직접 정의한 것입니다. (이는 \newtheorem 제어 문자열을 통해 만들어지는 환경의 정의와는 다릅니다.)

```
1 \newcounter{theorem}
2 \newenvironment{theorem}
3   {\par\addvspace{\topsep}%
4    \refstepcounter{theorem}%
5    \noindent\textbf{Theorem \thetheorem.}\enspace}
6   {\par\addvspace{\topsep}}
```

1행에서 정리의 번호를 계산하는 데 사용할 theorem 카운터를 정의했습니다. 2~6행에서는 theorem 환경을 정의합니다. 환경을 시작할 때에는 앞 문단과의 간격을 약간 띄우고(3행), theorem 카운터의 값을 1증가시킨 후(4행), 정리의 번호를 볼드체로 조판합니다(5행). 환경을 끝낼 때에는 환경을 시작했을 때처럼 다음 문단과 약간의 간격을 띄워 줍니다(6행). 이 환경을 사용하면 다음의 결과를 얻습니다.

```
1 We can use the following theorem to check if a subset of a group is a
  subgroup.
2 \begin{theorem}\label{thm:subgroupcrit}
3   Let  $(H)$  be a subset of a group  $(G)$ . If  $(ab^{-1} \in H)$  for any
    $(a,b \in H)$ , then  $(H)$  is a subgroup of  $(G)$ .
4 \end{theorem}
5 Here is a similar theorem for rings:
6 \begin{theorem}\label{thm:subringcrit}
7   Let  $(S)$  be a subset of a ring  $(R)$ . If  $(a-b, ab \in S)$  for any  $(a,b$ 
    $\in S)$ , then  $(S)$  is a subring of  $(R)$ .
8 \end{theorem}
```

We can use the following theorem to check if a subset of a group is a subgroup.

Theorem 1. Let H be a subset of a group G . If $ab^{-1} \in H$ for any $a, b \in H$, then H is a subgroup of G .

Here is a similar theorem for rings:

Theorem 2. Let S be a subset of a ring R . If $a - b, ab \in S$ for any $a, b \in S$, then S is a subring of R .

theorem 카운터의 값을 증가시킬 때 `\refstepcounter`를 사용했으므로, theorem 환경 안에서 `\label`을 사용하는 경우, 그 식별자는 정리의 번호를 가리킵니다. 위의 예시에서 작성한 `subgroupcrit`를 참조하면 다음의 결과를 얻습니다.

```
1 ... By Theorem~\ref{thm:subgroupcrit},
the given subset  $(2\mathbb{Z})$  is a
subgroup of  $(\mathbb{Z})$ .
```

...By Theorem 1, the given subset $2\mathbb{Z}$ is a subgroup of \mathbb{Z} .

19.1.2 카운터 번호의 출력 형식

사실 카운터의 값은 여러 가지 형식으로 조판할 수 있으며, `\the<ctrname>`은 카운터 번호를 출력할 ‘기본 형식’을 저장하고 있습니다. 이번 절에서 설명하는 명령어를 사용해 `\the<ctrname>`을 재정의하면 `<ctrname>` 카운터의 기본 출력 형식이 변경됩니다.

표 19.1에 있는 제어 문자열의 인자로 카운터의 이름을 입력하면 그 카운터의 번호를 각각의 형식으로 출력할 수 있습니다.

표 19.1 | 카운터 값 표시 형식

| 제어 문자열 | 설명 | 표시 형식 예시 | 표현 가능 범위 |
|------------------------|------------------------|--|---------------------------|
| <code>\arabic</code> | 아라비아 숫자 | -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 | $-2^{31} \sim 2^{31} - 1$ |
| <code>\Roman</code> | 로마 숫자 대문자 ¹ | I II III IV V VI VII VIII IX X XI XII XIII | $0 \sim 2^{31} - 1$ |
| <code>\roman</code> | 로마 숫자 소문자 ¹ | i ii iii iv v vi vii viii ix x xi xii xiii xiv | $0 \sim 2^{31} - 1$ |
| <code>\Alph</code> | 로마자 대문자 ² | A B C D E F G H I J K L M N O P Z | 1~26 |
| <code>\alph</code> | 로마자 소문자 ² | a b c d e f g h i j k l m n o p q r s z | 1~26 |
| <code>\fnsymbol</code> | 기호 ² | * † ‡ § ¶ ** †† ‡‡ | 1~9 |

¹ 0이나 음수는 표시되지 않으며, 지나치게 큰 값은 올바르게 표시되지 않을 수 있습니다.

² 표현 가능 범위 바깥의 값을 표시하려는 경우 컴파일 과정에서 오류가 발생합니다. 다만 0은 오류를 일으키지 않습니다.

temp 카운터에 5가 저장되어 있는 경우, 각 제어 문자열을 사용하면 다음의 결과를 얻습니다.

| 코드 | 조판되는 내용 |
|---|---------|
| <code>\arabic{temp}</code> | 5 |
| <code>\Roman{temp} \& \roman{temp}</code> | V & v |
| <code>\Alph{temp} \& \alph{temp}</code> | E & e |
| <code>\fnsymbol{temp}</code> | ¶ |

kotex 패키지를 불러 오면 한국어 문서에서 자주 사용되는 번호 형식을 추가로 사용할 수 있습니다. 표 19.2의 제어 문자열을 사용하면 되며, 이들도 값을 출력하려는 카운터의 이름을 인자로 지정하면 됩니다. 표현 가능 범위 바깥의 값은 표시되지 않으며, 그 값이 0이 아닌 경우에는 컴파일 과정에서 오류도 발생합니다.

표 19.2 | kotex 패키지: 카운터 값 표시 형식

| 제어 문자열 | 설명 | 표시 형식 예시 | 표현 가능 범위 |
|------------------------|------------------------|---|----------|
| <code>\jaso</code> | 자음자 | ㄱ ㄴ ㄷ ㄹ ㅁ ㅂ ㅅ ㅇ ㅈ ㅊ ㅋ ㅌ ㅍ ㅎ | 1~14 |
| <code>\ojaso</code> | 자음자 원문자 | ㄱ ㄴ ㄷ ㄹ ㅁ ㅂ ㅅ ㅇ ㅈ ㅊ ㅋ ㅌ ㅍ ㅎ | 1~14 |
| <code>\pjaso</code> | 자음자 괄호 문자 | (ㄱ) (ㄴ) (ㄷ) (ㄹ) (ㅁ) (ㅂ) (ㅅ) (ㅇ) (ㅈ) (ㅊ) (ㅋ) (ㅌ) (ㅍ) (ㅎ) | 1~14 |
| <code>\gana</code> | 음절자 | 가 나 다 라 마 바 사 아 자 차 카 타 파 하 | 1~14 |
| <code>\ogana</code> | 음절자 원문자 | 가 나 다 라 마 바 사 아 자 차 카 타 파 하 | 1~14 |
| <code>\pgana</code> | 음절자 괄호 문자 | (가) (나) (다) (라) (마) (바) (사) (아) (자) (차) (카) (타) (파) (하) | 1~14 |
| <code>\oeng</code> | 로마자 원문자 | ① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩ ⑪ ⑫ ⑬ ⑭ ⑮ …… ⑳ | 1~26 |
| <code>\peng</code> | 로마자 괄호 문자 | (a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l) (m) …… (z) | 1~26 |
| <code>\hRoman</code> | 로마 숫자 대문자 ¹ | I II III IV V VI VII VIII IX X XI XII | 1~12 |
| <code>\hroman</code> | 로마 숫자 소문자 ¹ | i ii iii iv v vi vii viii ix x xi xii | 1~12 |
| <code>\onum</code> | 숫자 원문자 | ① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩ ⑪ ⑫ ⑬ ⑭ ⑮ | 1~15 |
| <code>\pnum</code> | 숫자 괄호 문자 | (1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) | 1~15 |
| <code>\hnum</code> | 순우리말 기수사 | 하나 둘 셋 넷 다섯 여섯 일곱 여덟 …… 스물넷 | 1~24 |
| <code>\Hnum</code> | 순우리말 서수사 | 첫째 둘째 셋째 넷째 다섯째 여섯째 …… 스물넷째 | 1~24 |
| <code>\hNum</code> | 한자어 기수사 | 일 이 삼 사 오 육 칠 팔 구 십 십일 …… 이십사 | 1~24 |
| <code>\hanjanum</code> | 한자 | 一 二 三 四 五 六 七 八 九 十 十一 …… 二十四 | 1~24 |

¹ `\Roman`과 `\roman`과는 달리, 이 제어 문자열을 사용하면 로마 숫자가 전각 문자로 출력됩니다.

temp 카운터에 8이 저장되어 있는 경우, 각 제어 문자열을 사용하면 다음의 결과를

엄습니다.

| 코드 | 조판되는 내용 |
|---|---------|
| <code>\jaso{temp}, \gana{temp}</code> | ㅇ, 아 |
| <code>\ojaso{temp}, \pgana{temp}</code> | ㉚, (아) |
| <code>\oeng{temp}, \pnun{temp}</code> | ㉜, (8) |
| <code>\hNum{temp}, \hanjanum{temp}</code> | 팔, 八 |

표 19.1과 19.2의 명령어를 사용하여 `\the<ctrname>`을 재정의하면 카운터 번호의 기본 출력 형식이 변경됩니다. 예를 들어, 다음과 같이 입력하면 `\thetemp`를 입력했을 때 temp 카운터의 번호가 대문자 로마 숫자로 출력됩니다.

```
\renewcommand\thetemp{\Roman{temp}}
```

문서에서 번호가 일관된 형식으로 표시되도록 하기 위해, 카운터의 값을 조판할 때에는 두 표의 명령어를 직접 사용하는 것보다는 `\the<ctrname>`을 사용하는 것이 좋습니다.

19.1.3 카운터의 종속

카운터 사이에 상하 관계를 만들 수도 있습니다. 두 카운터 사이에 상하 관계가 설정 되면, 상위 카운터의 값이 1 증가할 때 하위 카운터의 값은 0으로 초기화됩니다. 이를 상위 카운터에 하위 카운터가 종속되어 있다고 부르겠습니다.

예를 들어, book 클래스에서 장의 번호는 chapter 카운터에, 절의 번호는 section 카운터에 저장되어 있습니다. 이때, section 카운터는 chapter 카운터에 종속되어 있으므로, 장 번호가 1 증가하면 절 번호는 0으로 초기화됩니다.

종속 관계의 설정과 해제

앞서 알아보았듯, `\newcounter` 제어 문자열을 사용하면 카운터를 새로 만들 때 종속 관계를 설정할 수 있습니다. 이미 만들어진 카운터의 종속 관계를 설정하거나 해제할 수도 있으며, 이때에는 `\counterwithin(*)`과 `\counterwithout(*)` 제어 문자열을 사용합니다.

```
\counterwithin(*){<ctrname>}{<parent counter>}
\counterwithout(*){<ctrname>}{<parent counter>}
```

`<ctrname>`: 종속 관계를 설정·해제하려는 카운터의 이름

`<parent counter>`: 상위 카운터의 이름

별표 붙은 제어 문자열을 사용하면 `\the<ctrname>`이 재정의되지 않으며, 별표 없는 제어 문자열을 사용하면 `\the<ctrname>`이 재정의됩니다. `\counterwithin`은

```
\the<parent counter>.\arabic{<ctrname>}
```

으로, `\counterwithout`은

```
\arabic{<ctrname>}
```

으로 `\the<ctrname>`를 재정의합니다.

예를 들어, 표·그림·수식 번호를 저장하는 `table`, `figure`, `equation` 카운터는 `article` 클래스에서 다른 카운터에 종속되어 있지 않습니다. 이때

```
\counterwithin{equation}{section}
```

를 입력하면 수식 번호가 절 번호에 종속됩니다. 더불어, 원래 ‘1, 2, 3……’으로 조판되었던 번호는 (절 번호가 1이라면) ‘1.1, 1.2, 1.3……’으로 조판됩니다. 한편, `report` 클래스와 `book` 클래스에서는 세 카운터가 장 번호에 종속되어 있습니다. 이때 위와 동일한 코드를 입력하면 (장 번호가 1인 경우) ‘1.1, 1.2, 1.3……’로 조판되었을 번호가 ‘1.1.1, 1.1.2, 1.1.3……’로 조판됩니다. 반대로,

```
\counterwithout{equation}{chapter}
```

를 입력하면 수식 번호가 장 번호에 종속되지 않도록 바꿉니다. 본래 ‘1.1, 1.2, 1.3……’처럼 장 번호와 함께 조판되었을 번호가 ‘1, 2, 3……’로 장 번호 없이 조판됩니다.

참고 | `\counterwithin(*)`과 `\counterwithout(*)`은 전역 제어 문자열입니다.

19.1.4 표준 클래스의 기본 카운터

\LaTeX 표준 클래스는 몇 가지 카운터를 정의하여 번호를 저장하고 조판할 때 사용합니다. 이를 표 19.3에 정리하였습니다. 이 표에 있는 것들 외에도, 이 책에서 설명하지는 않지만 \LaTeX 이 문서를 조판할 때 내부적으로 사용하는 정수 값을 저장하는 카운터도 있습니다. 이 카운터들은 문서의 레이아웃을 바꾸는 데 사용됩니다.

이외에도, 여러 패키지는 필요에 따라 카운터를 정의하고 사용합니다. 예를 들어, `amsmath` 패키지는 행렬의 최대 크기를 저장하는 `MaxMatrixCols` 카운터를, 수식의 하위 번호를 저장하는 `subequation` 카운터를 정의합니다. 이러한 정보는 각 패키지의 매뉴얼에서 확인할 수 있습니다.

| 표 19.3 | L^AT_EX의 번호 카운터

| 종류 | 카운터 이름 | 설명 |
|-----------------|---------------|---|
| 표제 번호 | part | 편(part)의 번호를 저장합니다. |
| | chapter | 장(chapter)의 번호를 저장합니다. article 클래스에서는 정의되어 있지 않습니다. |
| | section | 절(section)의 번호를 저장합니다. report 및 book 클래스에서는 chapter 카운터에 종속되어 있습니다. |
| | subsection | 절(subsection)의 번호를 저장합니다. section 카운터에 종속되어 있습니다. |
| | subsubsection | 절(subsubsection)의 번호를 저장합니다. subsection 카운터에 종속되어 있습니다. |
| | paragraph | 문단(paragraph)의 번호를 저장합니다. subsubsection 카운터에 종속되어 있습니다. |
| | subparagraph | 문단(subparagraph)의 번호를 저장합니다. paragraph 카운터에 종속되어 있습니다. |
| 개체 번호 | figure | 그림의 번호를 저장합니다. report 및 book 클래스에서는 chapter 카운터에 종속되어 있습니다. |
| | table | 표의 번호를 저장합니다. report 및 book 클래스에서는 chapter 카운터에 종속되어 있습니다. |
| | equation | 수식의 번호를 저장합니다. report 및 book 클래스에서는 chapter 카운터에 종속되어 있습니다. |
| 페이지 번호 | page | 페이지 번호를 저장합니다. |
| | totalpages | 현재까지의 전체 페이지 수를 저장합니다. |
| 각주 번호 | footnote | 각주 번호를 저장합니다. report 및 book 클래스에서는 chapter 카운터에 종속되어 있습니다. |
| | mpfootnote | minipage 환경 내에서의 각주 번호를 저장합니다. |
| 번호 목록의
항목 번호 | enumi | 번호 목록에서 1단계 번호를 저장합니다. |
| | enumii | 번호 목록에서 2단계 번호를 저장합니다. enumi 카운터에 종속되어 있습니다. |
| | enumiii | 번호 목록에서 3단계 번호를 저장합니다. enumii 카운터에 종속되어 있습니다. |
| | enumiv | 번호 목록에서 4단계 번호를 저장합니다. enumiii 카운터에 종속되어 있습니다. |

19.2 길이 값과 길이 변수

길이 변수는 길이 값을 기록하는 변수를 가리키며, 제어 문자열의 형태로 되어 있습니다. 길이 변수는 페이지나 여백의 크기 등 문서의 레이아웃을 설정하는 데 주로 쓰입니다.

19.2.1 길이 변수 기본 사용 방법

카운터와 마찬가지로, 길이 변수를 사용하는 기본 방법을 먼저 살펴봅시다. 이들은 (종속 기능을 제외하면) 카운터와 동일한 방법으로 사용됩니다.

참고 | 카운터 관련 명령어와 달리, 길이 변수 관련 명령어는 전역 효과를 가지지 않습니다. 영역 안에서 사용된 명령어는 그 영역 안에서만 유효합니다.

새로운 길이 변수를 선언할 때에는 `\newlength` 제어 문자열을 사용합니다.

`\newlength{<cname>}`

`<cname>`: 새로 선언할 길이 변수 이름

`\newlength`를 사용하면 `<cname>`에 입력된 제어 문자열이 길이를 저장하는 변수로써 선언되며, 길이의 초기값으로 0 pt가 저장됩니다. 길이 변수의 이름은 매크로를 정의할 때와 같은 조건을 만족해야 합니다.

길이 변수에 저장된 값은 `\setlength`와 `\addtolength` 제어 문자열을 사용해 바꿀 수 있습니다. `\setlength`는 길이 변수의 값을 지정된 값으로 바꾸고, `\addtolength`는 길이 변수의 현재 값에 지정된 값을 더합니다.

`\setlength{<cname>}{<length>}`

`\addtolength{<cname>}{<length>}`

`<cname>`: 값을 바꾸려는 길이 변수

`<length>`: 새로 저장할/원래의 값에 더할 길이 값

다음은 `\mylength` 제어 문자열을 길이 변수로 정의하고, 그 값을 변경시키는 예시입니다.

코드

```
\newlength\mylength
\setlength\mylength{2in}
\addtolength\mylength{3in}
\addtolength\mylength{-254mm}
```

길이 변수에 저장된 값

```
0 pt
2 in = 144.54 pt
5 in = 361.35 pt
-5 in = -361.35 pt
```

길이 변수에 저장된 값 사용하기

길이 변수에 저장된 값은 두 가지 방법으로 사용할 수 있습니다. 첫 번째는 저장된 값 자체를 다른 명령어의 인자 등으로 사용하는 것이고, 두 번째는 값을 문자의 형태로 문서에 조판하는 것입니다.

길이 변수에 저장된 값을 명령어의 인자 등으로 사용할 때에는 길이 변수 제어 문자열을 바로 입력하면 됩니다. 예를 들어, `\firstlength` 길이 변수에 저장된 값을 `\secondlength` 길이 변수에 저장하려면

```
\setlength\secondlength\firstlength
```

라고 입력하면 됩니다.

(흔한 일은 아니지만) 길이 변수에 저장된 값을 문서에 조판할 때에는 길이 변수 앞에 `\the` 제어 문자열을 입력합니다. 예를 들어, `\mylength` 길이 변수가 정의되어 있다면 `\the\mylength`는 이 변수에 저장되어 있는 값을 출력합니다. 이때 길이 값은 포인트(pt) 단위로 변환되어 출력됩니다.

19.3 값 지정하기

이번 절에서는 \LaTeX 이 정수 값과 길이 값을 해석하는 여러 가지 방법을 설명합니다. 많은 경우 정수 값은 10진수 표현으로, 길이 값은 실수 값 + 단위의 형태로 입력하면 되지만, 더 다양한 방법을 사용할 수도 있습니다.

19.3.1 정수 값

\LaTeX 에서 정수 값은 ‘숫자로 입력한 값, 문자 코드, 카운터에 저장한 값’의 세 가지 형태로 나타납니다.

숫자로 입력한 값

숫자로 입력하는 정수 값은 진법에 따라 세 가지 방법으로 쓸 수 있습니다.

- 10진수 특별한 문법 없이 아라비아 숫자로 원하는 값을 입력하면 됩니다.
- 8진수 원하는 값을 8진수로 표현한 뒤, 앞에 작은따옴표 '를 붙입니다.
- 16진수 원하는 값을 16진수로 표현한 뒤, 앞에 큰따옴표 "를 붙입니다. 10~15의 값은 로마자 A~F로 나타내며, 대·소문자는 구분하지 않습니다.

이러한 10진수, 8진수, 16진수 표현 앞에는 부호로 +나 -를 붙일 수 있습니다. +는 생략할 수 있습니다.

다음은 여러 가지 방법으로 temp 카운터에 정수 값을 저장하였을 때, 그 값의 10진수 표현을 나타낸 것입니다.

| 코드 | 카운터에 저장되는 값 |
|---------------------------------------|-------------|
| <code>\setcounter{temp}{10}</code> | 10 |
| <code>\setcounter{temp}{'341}</code> | 225 |
| <code>\setcounter{temp}{-'127}</code> | -87 |
| <code>\setcounter{temp}{"8C}</code> | 140 |
| <code>\setcounter{temp}{-"6f}</code> | 111 |

문자 코드

문자 코드는 다음의 두 가지 형태로 입력할 수 있습니다. `<char>` 자리에 문자를 입력하면 그 문자의 문자 코드 값이 사용됩니다. 그레이브 악센트 ``` 앞에는 +나 - 부호를 붙일 수 있습니다.

``<char>` ``\<char>`

\TeX 이 특수하게 해석하는 문자(역슬래시, 중괄호, 퍼센트 기호)의 경우, 기호 앞에 역슬래시를 붙이지 않으면 \TeX 이 코드를 잘못 해석할 수 있기 때문에 반드시 두 번째 방법을 사용해야 합니다. 여기에서 사용하는 역슬래시는 제어 문자열을 만드는 것이 아니라, ‘특수 문자를 특수하게 해석하지 않음’을 \TeX 에게 알려 주는 역할을 합니다.

다음은 여러 가지 방법으로 temp 카운터에 문자 코드를 저장한 것입니다.

| 코드 | 카운터에 저장되는 값 |
|--------------------------------------|-------------|
| <code>\setcounter{temp}{`a}</code> | 97 |
| <code>\setcounter{temp}{`\a}</code> | 97 |
| <code>\setcounter{temp}{-`P}</code> | -80 |
| <code>\setcounter{temp}{-`\P}</code> | -80 |
| <code>\setcounter{temp}{`\%}</code> | 37 |

카운터에 저장된 값

19.1절에서 설명했듯, 카운터에 저장된 값을 사용할 때에는 `\value`를 사용합니다. 예를 들어, temp 카운터에 7이 저장되어 있다면 그 값은 `\value{temp}`를 입력하여 사용합니다.

19.3.2 길이 값

L^AT_EX에서 길이 `length` 값은 ‘숫자와 단위로 입력한 값, 길이 변수에 저장된 값’의 두 가지 형태로 나타납니다.

숫자와 단위로 입력한 값

숫자와 단위로 길이 값을 입력할 때에는 부호, 정수 또는 실수, 단위를 순서대로 입력해 줍니다. 부호로는 +나 -를 입력하며, 생략할 수도 있습니다. 정수는 19.3.1절의 방법으로 입력하면 되며, 실수는 소수의 형태로 나타냅니다. 단위로는 표 19.4에 있는 것들을 사용할 수 있습니다.

표 19.4 | T_EX에서 사용할 수 있는 길이의 단위

| 구분 | 단위 | 이름 | 설명 |
|----------------|----|------------------------|---------------------|
| 고정된 단위 | pt | point | |
| | pc | pica | 1 pc = 12 pt |
| | in | inch | 1 in = 72.27 pt |
| | bp | big point | 72 bp = 1 in |
| | cm | centimeter | 2.54 cm = 1 in |
| | mm | millimeter | 10 mm = 1 cm |
| | dd | didot point | 1157 dd = 1238 pt |
| | cc | cicero | 1 cc = 12 dd |
| | sp | scaled point | 65536 sp = 1 pt |
| 글꼴에
상대적인 단위 | ex | | 대략 현재 글꼴의 소문자 x의 높이 |
| | em | | 대략 현재 글꼴의 대문자 M의 폭 |
| | mu | math unit ¹ | 18 mu = 1 em |

¹ mu는 `\mspace` 제어 문자열에서만 사용할 수 있습니다.

예를 들어, `3pt`, `4.3cm`, `-2ex`, `"1Cbp`, `-3.5em` 등은 모두 길이 값으로 유효한 코드입니다.

길이 값으로 0을 지정하려는 경우에도 단위를 생략하면 안 됩니다. `0pt`, `0cm`, `0em` 등 단위를 꼭 붙여 주어야 하며, 단위를 입력하지 않았다면 컴파일 과정에서 오류 메시지가 출력됩니다.

길이 변수에 저장된 값

길이 변수에 저장되어 있는 길이 값을 하나의 단위처럼 사용할 수도 있습니다. 이때에는 부호, 정수 또는 실수, 길이 변수를 순서대로 입력해 줍니다.

예를 들어, 문단 사이의 간격을 저장하고 있는 길이 변수 `\parskip`의 2배 길이는 `2\parskip`으로, -1.35배는 `-1.35\parskip`으로 입력하면 됩니다. 정수 값 1은 생략할 수 있습니다. 따라서 `-\parskip`이나 `\parskip`도 길이 값으로 유효한 코드입니다.

고정 길이와 가변 길이

LaTeX의 길이 값은 고정 길이 `rigid length`와 가변 길이 `rubber length`로 나뉩니다. 길이 값을 지정해야 하는 대부분의 상황(페이지의 크기, 판면의 크기, 표의 열의 폭을 지정할 때 등)에서는 고정 길이가 사용되지만, 공백을 만들 때에는 더 유연한 레이아웃을 위해 가변 길이를 사용할 수 있습니다.

가변 길이를 숫자와 단위로 지정할 때에는 다음의 문법을 사용합니다. `<natural length>`, `<stretchability>`, `<shrinkability>` 부분에 원하는 길이 값을 입력하면 됩니다. 이들은 각각 기준 길이, 늘어날 길이의 최댓값, 줄어들 길이의 최댓값을 뜻합니다.

`<natural length>` plus `<stretchability>` minus `<shrinkability>`

‘plus `<stretchability>`’와 ‘minus `<shrinkability>`’는 필요에 따라 생략할 수 있습니다.

예를 들어, `12pt plus 6pt minus 3pt, 5mu plus 5mu`는 모두 가변 길이 값으로 유효한 코드입니다.

길이 변수에 값을 저장할 때에는 고정 길이뿐 아니라 가변 길이를 저장할 수도 있습니다. `\addtolength` 제어 문자열을 사용할 때에는 세 가지 성분이 독립적으로 계산됩니다.

한편, 가변 길이가 저장된 길이 변수를 사용할 때, 변수 앞에 정수나 실수 값이 입력되면 고정 길이로 변환됩니다. +나 - 부호만을 붙이면 가변 길이로써 사용됩니다. 예를 들어, `\mylength`에 `1 in plus .5 in minus .25 in`가 저장되어 있을 때, `2\mylength`는 `2 in`를, `-\mylength`는 `-1 in plus -.5 in minus -.25 in`를 나타냅니다.

참고 | 고정 길이를 입력받는 자리에 가변 길이를 입력하는 경우, `<stretchability>`와 `<shrinkability>` 성분은 무시되고 `<natural length>` 성분만 사용됩니다.

19.3.3 값 사이의 연산

calc 패키지를 사용하면 정수 값이나 길이 값을 지정할 때 덧셈, 뺄셈, 곱셈, 나눗셈의 산술 연산을 사용할 수 있습니다. 각각 +, -, *, / 기호를 사용하며, 연산의 우선 순위를 지정하는 데 (와)를 사용할 수 있습니다.

예를 들어, 다음 코드는 temp 카운터에 subcounter 카운터의 값을 2배 한 뒤 4를 더한 값을 저장합니다.

```
\setcounter{temp}{\value{subcounter}*2+4}
```

calc 패키지는 두 개의 인자를 입력받는 \maxof, \minof 제어 문자열도 제공합니다. 이름에서 알 수 있듯, 이들은 인자로 지정된 두 값 중 최댓값, 최솟값을 각각 반환합니다.

정수 값의 연산

곱셈 및 나눗셈을 할 때에는 실수를 사용할 수 있습니다. 이때에는 \real 제어 문자열을 사용하며, 실수의 소수 표현을 인자로 지정해 줍니다. 실수 값이 연산에 포함된 경우에는 *나 / 연산자의 뒤에 실수 값을 입력해야 합니다. 이때, 계산 결과가 실수인 경우에는 그 값을 정수로 버림하며, 여러 연산을 연쇄적으로 하는 경우에는 중간 과정마다 버림 처리가 일어납니다.

예를 들어, 다음 코드는 temp 카운터에 $[5 \times 2.7] = 13$ 을 저장합니다.

```
\setcounter{temp}{5 * \real{2.7}}
```

길이 값의 연산

길이 값의 곱셈·나눗셈 연산 시에는 반드시 *나 / 연산자의 앞에 길이 값을, 연산자 뒤에 곱해 주거나 나눌 수를 입력해야 합니다. 즉, 4in * 3은 유효하지만 3 * 4in은 유효하지 않습니다.

곱하거나 나눌 수를 입력할 때, 길이 간의 비율을 유리수로 나타낼 때에는 \ratio 제어 문자열을 사용하며, 분자와 분모 자리에 들어가는 길이 값을 각각 인자로 지정해 주면 됩니다. 실수 값을 사용할 때에는 앞서 설명한 바와 같이 \real 제어 문자열을 사용합니다.

예를 들어, 다음 코드는 \mylength에 $4\text{ in} \times (3\text{ cm}/5\text{ in}) = 2.4\text{ cm}$ 을 저장합니다.

```
\setlength\mylength{4in * \ratio{3cm}{5in}}
```

calc 패키지의 더 다양한 기능은 패키지 안내서[51]를 참조하시기 바랍니다.

제 20 장

문서 파일 분리하기

큰 규모의 문서를 작성할 때에는 하나의 문서 파일에 모든 내용을 작성하는 것보다, 여러 개의 파일에 나누어 내용을 작성하고 이를 합쳐서 컴파일하는 것이 편리합니다. 이번 장에서는 \LaTeX 에서 소스 파일을 분리해 문서를 작성하는 방법을 알아보겠습니다.

20.1 \backslash input 제어 문자열

파일을 분리해 문서를 작성하는 가장 간단한 방법은 \backslash input을 사용하는 것입니다. \backslash input은 하나의 인자를 입력받으며, 여기에 파일의 경로와 이름을 입력하면 그 파일의 내용이 현재 파일의 현재 위치에 입력됩니다. 경로나 파일 이름에는 공백 문자를 사용하지 않는 것이 좋지만, 만약 공백 문자가 포함되어 있다면 해당 파일은 큰따옴표로 감싸 주어야 합니다.

예를 들어, 현재 작업 중인 폴더에 main.tex 파일과 equation.tex 파일이 있고, 두 파일의 내용이 아래와 같다고 가정해 봅시다.

main.tex 파일:

```
1 \documentclass[a4paper]{article}
2 \usepackage{mathtools,amssymb}
3 \begin{document}
4 The time-independent Schrödinger
  equation is given as follows:
5 \input{equation}
6 \end{document}
```

equation.tex 파일:

```
1 \begin{equation}
2 \frac{1}{2m} \left( i\hbar \frac{d}{dx} \right)^2 \psi + U\psi =
  \frac{1}{2m}(p)^2 \psi + U\psi =
  E\psi
3 \end{equation}
```

이후 main.tex 파일을 컴파일하면 다음의 내용이 작성된 파일을 얻습니다.

The time-independent Schrödinger equation is given as follows:

$$\frac{1}{2m} \left(i\hbar \frac{d}{dx} \right)^2 \psi + U\psi = \frac{1}{2m} (p)^2 \psi + U\psi = E\psi \quad (1)$$

주의할 점으로, `\input`으로 불러오는 파일에는 다음 세 가지 코드를 사용하면 안 됩니다.

```
\documentclass \begin{document} \end{document}
```

`\input`은 `\input`으로 불러오는 소스 파일에 중첩하여 사용할 수도 있습니다. 예를 들어, 아래와 같이 `main.tex` 파일을 비롯해 여러 파일을 분리해 작성하고, 각 장의 내용을 담은 소스 파일은 `main.tex` 파일이 저장된 폴더 하위의 `chapters` 폴더에, 각 절의 내용을 담은 소스 파일은 `sections` 폴더에 분리해 둘 수도 있습니다. 전처리부에 작성할 내용이 많은 경우 `preamble.tex` 파일을 따로 작성해서 불러올 수도 있습니다.

main.tex:

```
1 \documentclass[a4paper]{book}
2 \input{preamble}
3 \title{Sample Document}
4 \author{Anonymous}
5 \begin{document}
6 \frontmatter
7 \maketitle
8 \input{chapters/preface}
9 \tableofcontents
10 \mainmatter
11 \input{chapters/chapter1}
12 \input{chapters/chapter2}
13 \appendix
14 \input{chapters/appendix}
15 \end{document}
```

chapters 폴더 > chapter1.tex:

```
1 \chapter{The First Chapter}
2 \input{sections/section1-1}
3 \input{sections/section1-2}
4 \input{sections/section1-3}
```

chapters 폴더 > chapter2.tex:

```
1 \chapter{The Second Chapter}
2 \input{sections/section2-1}
3 \input{sections/section2-2}
```

chapters 폴더 > appendix.tex:

```
1 \chapter{The Appendix}
2 \input{sections/sectionA-1}
3 \input{sections/sectionA-2}
4 \input{sections/sectionA-3}
```

\endinput 제어 문자열

`\input`으로 불러 오는 파일에는 `\endinput` 제어 문자열을 입력할 수 있습니다. 이 제어 문자열이 입력된 경우, 이 이후의 코드는 읽어들여지지 않습니다. 디버깅을 위해 일부 코드를 주석 처리하듯이 사용할 수도 있고, 소스 파일의 맨 마지막에 입력하여

파일 끝부분에 있을 수 있는 의도하지 않은 문자가 삽입되는 것을 방지할 수도 있습니다. \endinput은 다음 절에서 설명하는 \include로 불러오는 파일에도 사용할 수 있습니다.

20.2 \include 제어 문자열

\input 대신 \include 제어 문자열을 사용할 수도 있습니다. \include를 사용하는 경우에는 불러온 파일의 코드를 삽입하기 전후에 페이지가 나뉩니다. 따라서, book이나 report 클래스 문서에서 하나의 장을 하나의 파일에 작성하고, 이를 \include로 불러오는 방식으로 사용할 수 있습니다.

\input을 사용할 때와 마찬가지로, \include로 불러 오는 파일에는 \document나 \begin{document}, \end{document}를 입력하면 안 됩니다. 한편, \input과는 달리 \include는 전처리부에서 사용할 수 없으며, 불러온 파일 안에 \include를 중첩하여 사용할 수 없습니다(\input은 사용할 수 있습니다).

일부 파일만 불러 오기

전처리부에서 \includeonly 제어 문자열을 사용하면 \include한 파일 중 일부만을 실제 컴파일에 사용하도록 할 수 있습니다. \includeonly 제어 문자열의 인자로 컴파일에 사용할 파일들의 이름을 입력하면 됩니다. 일부 파일만을 불러오더라도 불러오지 않은 파일에 있던 내용의 상호 참조 정보 등은 유지됩니다. 따라서, 삽입되지 않은 파일에 작성된 대상을 참조하더라도 페이지 번호나 대상의 번호는 올바르게 표시됩니다. 이 기능은 큰 규모의 문서를 빠르게 컴파일 할 때 유용하게 사용할 수 있습니다.

예를 들어, 앞의 예시에서 일부 명령어를 수정하여 다음과 같이 사용할 수 있습니다.

```

1 \documentclass[a4paper]{book}
2 \input{preamble}
3 \includeonly{chapters/chapter2,chapters/appendix}
4 \title{Sample Document}
5 \author{Anonymous}
6
7 \begin{document}
8 \frontmatter
9 \maketitle
10 \include{chapters/preface}
11 \tableofcontents
12
```

```
13 \mainmatter
14 \include{chapters/chapter1}
15 \include{chapters/chapter2}
16 \appendix
17 \include{chapters/appendix}
18
19 \end{document}
```

이 문서를 컴파일하면 제2장과 부록 부분만 실제 컴파일에 사용되며, 서문과 제1장은 컴파일 되지 않습니다. 제3행을 주석 처리하면 모든 파일이 포함된 채로 컴파일 됩니다. `\input`으로 불러온 파일인 `preamble.tex`은 영향을 `\includeonly`의 영향을 받지 않으며, 항상 포함된 채로 컴파일 됩니다.

VII

부록

부록에서는 \LaTeX 의 문법 외적인 내용을 설명합니다. \TeX 배포판을 컴퓨터에 설치하고 관리하는 방법과, 컴파일 과정에서 발생한 문제를 해결하는 방법을 다룹니다.

제A장 \TeX 배포판 사용하기 · 289

제B장 도움말과 추가 자료 · 295

부록 A

TeX 배포판 사용하기

이 책의 첫머리에서는 Overleaf를 사용해 L^ATeX 문서를 작성하는 방법을 설명하였지만, 복잡한 문서를 만든다면 Overleaf 등의 온라인 플랫폼을 사용하는 것이 불편할 것입니다. 이때에는 컴퓨터에 TeX를 설치해 문서를 작성하는 것이 더 편리합니다.

이번 장에서는 Overleaf 등의 온라인 플랫폼을 통해 TeX를 사용하지 않고, 컴퓨터에 TeX를 설치해 사용하는 방법을 알아보겠습니다. TeX를 설치하고 사용할 때에는 (1) TeX에 관련된 모든 프로그램과 파일을 관리하는 프로그램인 TeX 배포판과 (2) 문서를 편집하고 확인할 때 사용하는 편집기 겸 뷰어가 필요합니다. 이들을 설치하고 관리하는 방법을 알아보시다.

A.1 TeX 배포판 설치

TeX 배포판에는 여러 종류가 있으며, 그중 TeX Live와 MikTeX이 자주 사용됩니다. 두 배포판 모두 Windows, macOS, 리눅스, 유닉스에서 사용할 수 있습니다. 두 TeX 배포판에는 몇 가지 차이점이 있으며, 그 특징을 그림 A.1에 정리하였습니다. 두 배포판 중 어느 것을 설치하더라도 조판 결과는 동일하므로, 자신이 사용하기에 편리한 것을 사용하시면 됩니다.

TeX 배포판에 관한 자세한 정보는 [3]을 참조하십시오.

A.1.1 TeX Live와 MacTeX

TUG의 TeX Live 페이지(tug.org/texlive)에서 안내에 따라 설치 프로그램을 내려받은 후, 해당 프로그램을 실행하면 됩니다. TeX Live를 설치할 때에는 인터넷 연결이 필요하며,

TeX Live의 특징

- 처음 설치할 때 필요한 프로그램과 패키지를 모두 설치합니다. 인터넷을 통해 설치가 진행되며, 5 GB의 패키지를 내려받으며 설치에 2시간 이상 소요됩니다.
- 컴파일 과정에서 내부적으로 사용하는 프로그래밍 언어인 Perl을 함께 설치합니다.
- MikTeX에 비해 컴파일 속도가 빠릅니다.
- 매년 새로운 버전이 출시됩니다. 여러 버전을 한 컴퓨터에 동시에 설치할 수도 있고, 단순히 업데이트 과정을 통해 최신 버전만을 사용할 수도 있습니다.
- Windows에서 사용자의 계정 이름이나, 작업한 파일의 이름 또는 경로에 한글이 포함되어 있으면 설치 및 문서 컴파일 과정에서 오류가 발생할 수 있습니다.

MikTeX의 특징

- 처음 설치할 때, 필수적인 패키지만을 내려받고 그 외 필요한 패키지는 문서를 컴파일하다 필요하면 그때그때 내려받습니다. 필요한 패키지를 내려받을 때는 인터넷 연결이 필요합니다.
- 컴파일 과정에서 내부적으로 사용하는 프로그래밍 언어인 Perl이 기본적으로 설치되지 않습니다. 이전에 컴퓨터에 설치한 적이 없다면 별도 설치가 필요합니다.
- TeX Live에 비해 컴파일 속도가 느립니다.
- 새로운 버전이 출시되면 업데이트 과정을 통해 새 버전을 사용할 수 있습니다.
- 사용자의 계정 이름이나, 작업한 파일의 이름 또는 경로에 한글이 포함되어 있더라도 문제가 발생하지 않습니다.

| 그림 A.1 | TeX Live와 MikTeX의 차이점

1시간 이상의 긴 시간이 소요될 수 있습니다. 설치가 완료되면 프로그램 목록에 기본 편집기인 TeXworks와 관리 프로그램인 TeXLive Manager가 나타날 것입니다.

macOS에서는 TeX Live의 macOS용 버전인 MacTeX을 설치하면 됩니다. TUG의 MacTeX 페이지(tug.org/mactex)에서 안내를 확인할 수 있습니다. 설치 과정이 완료되면 Launchpad에 기본 편집기인 TeXShop과 관리 프로그램인 TeX Live utility가 나타날 것입니다.

A.1.2 MikTeX

MikTeX 홈페이지의 Download 페이지(miktex.org/download)에서 운영 체제에 맞는 설치 파일을 내려받습니다. 내려받은 프로그램을 실행하면 컴퓨터에 MikTeX이 설치됩니다. 설치 과정이 완료되면 프로그램 목록에 기본 편집기인 TeXworks와 관리 도구인 MikTeX Console이 나타납니다.

A.1.3 편집기 및 뷰어

TeX 문서는 플레인 텍스트 형식이므로, 어떤 텍스트 편집기를 사용하더라도 문제 없이 소스 코드를 작성할 수 있습니다. 운영 체제의 기본 프로그램인 메모장(Windows)이나

텍스트 편집기(macOS)를 사용할 수도 있고, 프로그래밍을 할 때 사용하는 VSCode, Xcode, Vim 등을 사용할 수도 있습니다. 또, 컴파일 과정을 거쳐 얻은 PDF 파일은 아무 PDF 뷰어를 통해 확인할 수 있습니다.

한편, \TeX works, \TeX Shop, \TeX studio, \TeX maker 등 \TeX 문서 작성에 특화된 다양한 편집기 겸 뷰어도 있습니다. 이 프로그램들은 \TeX 을 사용해 문서를 작성할 때에는 ‘코드 작성 → 컴파일 → 생성된 문서 확인’의 과정을 하나의 프로그램 안에서 할 수 있게 해 줍니다. 편집기에서 코드를 작성한 후, 터미널을 열어 복잡한 명령어를 입력할 필요 없이 프로그램 안에 있는 ‘조판’ 버튼을 누르면 바로 컴파일이 시작됩니다. 프로그램의 뷰어를 통해 PDF 파일도 같이 볼 수 있으며, 컴파일이 완료되면 실시간으로 업데이트 되기 때문에 편리하게 문서를 작성할 수 있습니다. 이러한 프로그램에서는 편집기와 뷰어가 연동되어, PDF 파일 상의 특정 지점을 클릭하면 편집기에서 해당 부분의 코드를 보여 주고, 그 반대의 경우도 가능합니다.

A.2 \TeX 배포판 관리

이번 절에서는 \TeX 배포판과 패키지 파일을 관리하는 방법을 다룹니다. 컴퓨터에 설치한 \TeX 배포판에 따라 방법이 달라지므로, 이를 기준으로 구분하여 방법을 설명하겠습니다.

A.2.1 \TeX Live

컴퓨터에 설치한 \TeX 배포판이 \TeX Live인 경우에 대해 설명하겠습니다. \TeX Live Shell이라는 앱을 사용합니다.

Windows에서는 시작 메뉴의 앱 목록에서 ‘ \TeX Live 2023 → TLShell \TeX Live Manager’를 클릭해서, 그 외의 운영 체제에서는 터미널을 열고 `tlshell`을 입력해 관리 프로그램인 \TeX Live Shell을 열 수 있습니다.

이 프로그램에서는 현재 컴퓨터에 설치된 패키지와 설치되지 않은 패키지, 또 업데이트할 수 있는 패키지의 목록을 확인할 수 있습니다. 화면 가운데 ‘Package List’ 부분에서 목록에 표시할 조건을 선택한 후, 패키지 이름 왼쪽의 동그라미를 클릭한 뒤 업데이트하거나 삭제, 설치할 수 있습니다. 창 오른쪽의 ‘Update all’을 클릭해 간단히 모든 패키지를 업데이트할 수도 있습니다. 가능한 경우 관리 프로그램 자체의 업데이트도 할 수 있습니다.

새로운 버전의 TeX Live 사용하기

TeX Live는 매년 새로운 버전이 공개되며, 이전 버전은 일정 시점이 되면 더 이상 패키지의 업데이트를 제공하지 않습니다. 따라서, 새로운 버전의 TeX Live를 사용하려면 TeX Live Shell을 통해 업데이트를 설치하는 것이 아니라, TUG에서 새로운 버전의 TeX Live를 새로 설치해야 합니다(설치 과정은 부록 A.1을 참조하세요). 만약 이전 버전의 TeX Live가 필요하지 않다면 다음 절의 안내에 따라 해당 버전을 삭제할 수 있습니다.

TeX Live 삭제하기

더 이상 TeX를 사용하지 않는다면, TeX 배포판을 삭제해 공간 낭비를 줄이는 것이 좋습니다. 또, TeX Live는 새로운 버전이 설치되더라도 이전 버전이 사라지지 않기 때문에, 불필요하다면 오래된 버전의 배포판을 삭제해 주는 것이 좋습니다. 먼저, Windows에서는 아래의 과정을 통해 TeX Live를 삭제할 수 있습니다.

1 | 시작 메뉴를 열고, 설치된 앱 목록에서 ‘TeX Live 2024 → Uninstall TeX Live’를 클릭합니다. 컴퓨터에 설치된 버전에 따라 연도가 다르게 표시될 수 있습니다.

2 | 화면의 안내에 따라 삭제 과정을 완료합니다.

다른 운영 체제에서 TeX Live를 삭제하려면 터미널을 열고,

```
> tlmgr uninstall --all
```

를 실행합니다. 이후에는 자동으로 TeX Live가 삭제됩니다.

A.2.2 MacTeX

macOS에서는 앞 절에서 다룬 TeX Live Shell을 사용할 수도 있고, 대신 TeX Live Utility를 사용할 수도 있습니다. Launchpad에서 아이콘을 클릭하여 이 앱을 실행합니다.

‘Updates’ 탭에서는 현재 업데이트할 수 있는 패키지를 확인할 수 있으며, 목록에서 Control-클릭 후 나타나는 메뉴에서 ‘Update All Packages’를 클릭해 사용 가능한 모든 업데이트를 설치할 수도 있습니다. ‘Packages’ 탭에서는 CTAN에 현재 등록된 모든 패키지의 목록을 확인할 수 있습니다. 패키지를 더블 클릭하면 정보를 확인할 수 있으며, 패키지 문서가 존재한다면 여기에서 바로 읽을 수도 있습니다. TeX Live Utility 자체의 업데이트도 할 수 있으며, 메뉴 막대에서 ‘TeX Live Utility → Check for Updates...’를 클릭해 진행할 수 있습니다.

MacTeX도 핵심 부분은 TeX Live를 통해 관리되므로, 새로운 연도의 버전이 출시

되면 이전 버전의 패키지 업데이트 제공이 중단됩니다. 새로운 업데이트를 받으려면 부록 A.1의 내용을 참고해 TUG에서 새로운 버전의 MacTeX을 설치하십시오.

macOS에서 MacTeX을 삭제하려면 다음의 과정을 따릅니다.

- 1 | 먼저, TeX Live를 삭제합니다. Finder에서 ‘이동 → 폴더로 이동...’을 클릭합니다.
- 2 | `/usr/local/texlive`를 입력하고 리턴 키를 누릅니다.
- 3 | 삭제하려는 버전의 폴더를 선택하고 삭제합니다. 이중 `texmf-local` 폴더에는 여러 버전의 TeX Live에서 공유하는 파일이 저장되어 있으며, 컴퓨터에서 TeX을 완전히 제거하려면 `texmf-local` 폴더까지 삭제합니다. 삭제 과정에서 시스템이 관리자의 암호를 요구할 수 있습니다.
- 4 | 이제 TeX과 관련된 앱을 모두 삭제합니다. TeX Live를 업데이트하려는 것이라면 이 과정은 진행하지 마십시오. Finder의 왼쪽 사이드바에서 ‘응용 프로그램’을 클릭합니다.
- 5 | TeX 폴더를 선택하고 삭제합니다.

A.2.3 MiKTeX

컴퓨터에 설치한 TeX 배포판이 TeX Live가 아닌 MiKTeX이라면, MiKTeX Console을 사용해 관리할 수 있습니다. 설치된 앱 목록에서 ‘MiKTeX Console’을 클릭해 실행합니다.

- 1 | 맨 처음에는 사용자 모드로 진행할지, 관리자 모드로 진행할지 선택해야 합니다. 자신이 어떤 모드를 선택해야 하는지 모르겠다면 우선 사용자 모드를 선택해 진행하고, 막히는 것이 있다면 ‘File → Switch to MiKTeX administrator mode’를 선택해 관리자 모드로 전환하면 됩니다.
- 2 | 왼쪽에서 ‘Updates’를 선택하면 MiKTeX을 업데이트할 수 있습니다. ‘Check for updates’를 클릭해 업데이트를 확인하고, 가능한 업데이트가 있다면 ‘Update now’를 클릭해 업데이트를 진행합니다.
- 3 | ‘Documentation’을 선택하면 다양한 문서를 읽을 수 있습니다. 원하는 문서를 검색하고 더블클릭하면 됩니다.
- 4 | ‘Packages’를 선택하면 패키지의 목록을 확인할 수 있습니다. 현재 컴퓨터에 설치된 패키지를 확인하고, 필요한 경우 설치·삭제·업데이트할 수 있습니다.
- 5 | 컴퓨터에서 MiKTeX을 삭제하거나 설정을 초기화하려면 ‘Cleanup’을 선택합니다. 이곳의 안내에 따라 진행합니다.

MiKTeX은 연도에 따른 버전 구분 없이 모든 패키지의 업데이트가 제공됩니다. TeX

Live와 달리 재설치 등을 통한 업그레이드 과정을 거치지 않으며, 필요한 경우 Mi \TeX Console에서 \TeX 배포판의 업데이트를 진행하면 됩니다.

A.3 \TeX 의 명령줄 인터페이스

\TeX 문서를 컴파일 할 때에는 명령줄 인터페이스를 사용할 수 있습니다. 운영 체제의 터미널(Windows라면 PowerShell, macOS라면 터미널 등)을 실행한 후, 컴파일에 사용할 엔진과 컴파일 하려는 파일의 이름을 입력하면 됩니다. 예를 들어, \LaTeX 엔진으로 ~/Documents/mydocument.tex 파일을 컴파일 하려면

```
$ latex ~/Documents/mydocument.tex
```

를 입력한 후 `return` 키를 눌러 명령을 실행하면 됩니다. 파일 이름에 마침표가 없으면 '.tex' 확장자를 생략해도 됩니다. 사용하는 엔진에 따라 latex 대신 pdflatex, xelatex, lualatex을 입력하면 됩니다. 일본어 문서를 작성하는 경우, uplatex을 사용할 수도 있습니다.

latex 또는 uplatex 명령어를 사용해 .tex 파일을 컴파일한 경우에는 PDF 파일 대신 DVI 파일을 얻습니다. 이때에는 dvipdfmx 명령어를 사용해 DVI 파일을 PDF 파일로 변환할 수 있습니다.

```
$ dvipdfmx ~/Documents/mydocument.dvi
```

위 명령어를 입력하면 mydocument.dvi 파일을 mydocument.pdf 파일로 변환할 수 있습니다. 마찬가지로 파일 이름에 마침표가 없는 경우에는 '.dvi' 확장자를 생략해도 됩니다.

부록 B

도움말과 추가 자료

이번 장에서는 각종 문제를 해결할 때 참고할 수 있는 도움말과, 이 책에서 사용한 용어에 대한 정보를 제공합니다.

B.1 문제 해결

LaTeX 문서를 작성하다 보면 크고 작은 문제가 발생하기도 합니다. 이번 절에서는 컴파일 과정에서 어떤 오류 메시지가 나타날 수 있는지 알아보고, 문제를 해결하는 방법을 알아봅니다.

B.1.1 오류 및 경고 메시지

소스 파일에 잘못된 부분이 있다면, 컴파일 과정에서 오류 메시지와 경고 메시지가 표시됩니다. **오류** error 메시지는 주로 소스 파일의 문법에 문제가 있는 경우 발생합니다. 많은 오류는 오타에 의해 발생하며, 주의해서 사용해야 하는 특수 문자를 잘못 입력한 경우에도 발생할 수 있습니다. **경고** warning 메시지는 오류에 비해서는 덜 심각한 문제가 있을 때 발생합니다. 소스 파일의 문법에는 문제가 없지만 만족스럽지 못한 조판 결과가 만들어질 때에 주로 경고 메시지가 표시됩니다.

오류 메시지는 컴파일 과정을 중단시키고 사용자에게 어떤 동작을 취할지 물어보지만, 경고 메시지는 그렇지 않습니다. 또, 오류 메시지는 항상 느낌표 ‘!’로 시작합니다.

전처리부 관련 메시지

다음은 전처리부와 관련하여 볼 수 있는 메시지입니다. 주로 클래스나 패키지 관련 코드를 잘못 입력했을 때 발생합니다.

- ! LaTeX error: File '`\langle class \rangle.cls`' not found.
문서 클래스를 지정할 때, `\langle class \rangle`라는 이름의 클래스가 존재하지 않음을 뜻합니다.
클래스 이름을 확인하십시오.
- LaTeX warning: Unused global option(s): [`\langle options \rangle`].
클래스를 불러올 때 지정한 옵션 중 클래스에서도 사용되지 않고 불러온 패키지에서도 사용되지 않은 것이 있음을 뜻합니다.
- ! LaTeX error: `\usepackage` before `\documentclass`.
전처리부에서 `\documentclass`가 입력되기 전에 `\usepackage`가 입력되었음을 뜻합니다. `\documentclass`를 빠뜨리지 않았는지 확인하십시오.
- ! LaTeX error: File '`\langle package \rangle.sty`' not found.
패키지를 불러올 때, `\langle package \rangle`라는 이름의 패키지가 존재하지 않음을 뜻합니다.
패키지 이름을 확인하십시오.
- ! LaTeX error: Unknown option '`\langle option \rangle`' for package '`\langle package \rangle`'.
패키지를 불러올 때 입력한 `\langle option \rangle` 옵션을 `\langle package \rangle` 패키지에서 지원하지 않음을 뜻합니다.
- ! LaTeX error: Option clash for package `\langle package \rangle`.
같은 패키지를 다른 옵션을 지정하여 여러 번 불러왔음을 뜻합니다. 패키지를 여러 번 불러오지는 않았는지, 다른 패키지에서 자신이 불러오려는 패키지를 이미 불러오지는 않았는지 확인하십시오.
- ! LaTeX error: Missing `\begin{document}`.
document 환경 안에 입력해야 할 내용을 전처리부에 입력했음을 뜻합니다.
- Missing Character: There is no `\langle char \rangle` ("`\langle charcode \rangle`") in font `\ nullfont!`
전처리부에서 문자를 입력했음을 뜻합니다. 위의 오류 메시지와 함께 출력됩니다.
- ! LaTeX error: Can be used only in preamble.
전처리부에서만 사용할 수 있는 제어 문자열을 document 환경 안에서 사용했음을 뜻합니다. `\documentclass`, `\usepackage`, `\DeclareTextSymbol` 등의 제어 문자열이 전처리부에서 사용되지 않았는지 확인하십시오.

특수 문자 관련 메시지

LaTeX의 10가지 특수 문자를 잘못 입력했을 때 발생하는 메시지를 정리하였습니다. 특수 문자 중 일부는 작동 방식에 의해 이상한 결과물을 만들거나, 다른 오류를 일으킬 수

있습니다.

■ ! Too many }'s.

여는 중괄호보다 더 많은 닫는 중괄호를 입력했음을 뜻합니다. 여는 중괄호를 빠뜨리지 않았는지, 닫는 중괄호를 필요 이상으로 입력하지 않았는지 확인하십시오.

■ ! Missing } inserted.

위와 반대로, 닫는 중괄호를 빠뜨렸음을 뜻합니다. 여는 중괄호를 필요 이상으로 입력하지 않았는지, 닫는 중괄호를 빠뜨리지 않았는지 확인하십시오.

■ ! You can't use `macro parameter character #' in *mode* mode.

해시 기호 '#'를 적절하지 않은 곳에 입력했음을 뜻합니다. 해시 기호를 문서에 입력하려던 것이면 \#을 입력하십시오.

■ ! Misplaced alignment tab character &.

앰퍼샌드 '&'를 적절하지 않은 곳에 입력했음을 뜻합니다. 앰퍼샌드를 문서에 입력하려던 것이면 \&를 입력하십시오.

명령어 관련 메시지

■ ! Paragraph ended before *cs* was complete.

제어 문자열의 인자를 지정할 때, 중괄호를 닫지 않거나 인자에 빈 줄이 포함되면 발생합니다. 중괄호의 개수가 맞는지, 또 인자에 빈 줄이 포함돼 있지는 않은지 확인하세요.

■ ! Missing { inserted. 또는 ! Missing } inserted.

중괄호가 잘못 사용되었을 때 발생합니다.

■ ! Undefined control sequence.

자주 보게 될 오류 메시지로, 입력한 제어 문자열이 정의되어 있지 않음을 뜻합니다. 제어 문자열에 오타는 없는지, 대소문자가 틀리진 않았는지, 제어 문자열 뒤의 띄어쓰기는 잘 되었는지 확인하세요. 제어 문자열을 사용하기 위해 불러와야 하는 패키지를 불러오지 않았을 때에도 이 오류가 발생합니다.

■ ! LaTeX Error: Environment *env* undefined.

입력한 이름의 환경이 정의되지 않았을 때 발생합니다. 환경의 이름을 입력할 때 오타가 있었거나, 환경을 사용하기 위해 불러와야 할 패키지를 불러오지 않아할 때 주로 발생합니다.

- ! LaTeX Error: `\begin{<env_1>}` on input line `<line>` ended by `\end{<env_2>}`.

시작한 환경의 이름과 끝낸 환경의 이름이 다를 때 발생합니다. 여러 개의 환경을 중첩해 사용할 때 환경을 열고 닫는 순서를 헷갈리거나, `\begin` 또는 `\end`의 짝이 맞지 않으면 발생합니다.

모드 관련 메시지

- ! Missing \$ inserted.

수식 모드에서만 사용할 수 있는 `^`나 `_`, 또는 수식 모드 전용 제어 문자열을 텍스트 모드에서 입력했음을 뜻합니다. 수식 모드를 시작한 후 수식 모드를 끝내지 않았을 때에도 이 오류가 발생합니다.

- ! `<cs>` allowed only in math mode.

수식 모드에서만 허용되는 제어 문자열 `<cs>`를 텍스트 모드에서 입력했을 때 발생합니다.

- ! Command `<cs>` invalid in math mode.

텍스트 모드에서만 사용할 수 있는 제어 문자열 `<cs>`를 수식 모드에서 입력했을 때 발생합니다. 해당 제어 문자열과 동일한 동작을 하는 수식 모드용 명령어를 사용하거나, 해당 제어 문자열을 `\text`의 인자로 사용하세요.

- LaTeX Warning: Command `<cs>` invalid in math mode on input line `<line>`.

텍스트 모드에서만 허용되는 제어 문자열 `<cs>`를 수식 모드에서 입력했을 때 발생합니다.

- ! LaTeX Error: Bad math environment delimiter.

`\(, \), \[, \]`를 잘못 사용하면 발생합니다. 짝은 맞는지, 수식 모드 안에서 또 수식 모드를 시작하진 않았는지 확인하세요.

문자 입력 관련 메시지

- ! LaTeX error: Unicode character `<char>` (U+`<ucharcode>`) not set up for use with LaTeX.

레거시 엔진에서 소스 파일에 입력한 문자 `<char>`을 조판하기 위한 적절한 설정이 갖추어지지 않았을 때 발생합니다. 예를 들어, 소스 파일에 한글을 입력했는데

kotex 패키지를 불러오지 않은 경우 등이 해당합니다.

- Missing character: There is no $\langle char \rangle$ (U+ $\langle ucharcode \rangle$) in font $\langle font \rangle$!

글꼴 $\langle font \rangle$ 에 문자 $\langle char \rangle$ 이 존재하지 않을 때 발생합니다. 올바른 문자를 입력하거나, 다른 글꼴을 사용하십시오.

- LaTeX Font Warning: Font shape $\langle font \rangle$ undefined using $\langle font \rangle$ instead on input line $\langle line \rangle$.

지정한 글꼴을 사용할 수 없는 경우에 발생합니다. 현재 글꼴이 제공하지 않는 패밀리·웨이트·스타일의 조합을 사용했을 때 자주 발생합니다. 다른 모양의 글꼴을 사용하십시오.

- LaTeX Font Warning: Some font shapes were not available, defaults substituted.

사용할 수 없는 글꼴을 지정한 곳이 소스 파일에 있을 때, 컴파일 완료 후 표시됩니다. 다른 모양의 글꼴을 사용하십시오.

- There's no line here to end.

문단과 문단 사이에 추가 간격을 주기 위해 문단 끝에 $\backslash\backslash$ 을 입력하면 발생합니다. 이러한 경우에는 $\backslash\backslash$ 대신 \backslashbigskip 등의 제어 문자열을 대신 사용하는 것이 적절합니다.

수식 관련 메시지

- ! Double subscript. 또는 ! Double superscript.

위 첨자나 아래 첨자를 적절한 중괄호 없이 여러 번 사용했을 때 발생합니다.

- ! Missing \backslashright . inserted.

수식 모드에서 \backslashleft 를 입력하고 \backslashright 를 입력하지 않았을 때 발생합니다.

- ! Extra \backslashmiddle . 또는 ! Extra \backslashright .

수식 모드에서 \backslashleft 가 입력되지 않았는데 \backslashmiddle 또는 \backslashright 가 입력되었을 때 발생합니다.

- ! LaTeX Error: Too many columns in eqnarray environment.

eqnarray 환경에서 &가 한 줄에서 3개 이상 사용되면 발생합니다.

표 관련 메시지

- ! LaTeX Error: Illegal character in array arg.
tabular 또는 array 환경의 인자에 사용할 수 없는 문자를 입력했을 때 발생합니다.
- ! Misplaced \noalign.
\hline 제어 문자열 앞에 \\을 입력하지 않은 경우에 발생합니다. 가로선을 긋기 위해 \hline을 사용하기 전에는 줄을 나누어 주어야 합니다.
- ! Misplaced \omit.
\multicolumn을 입력한 칸에 다른 내용이 더 입력되어 있을 때 발생합니다. 해당 칸에 입력하려는 내용은 \multicolumn의 인자로만 지정해야 합니다.
- ! Extra alignment tab has been changed to \cr.
생성한 열의 수보다 더 많은 열에 내용을 입력했을 때 발생합니다. &를 지나치게 많이 사용하지 않았는지, \\을 잊지 않았는지 확인하십시오.

참조 기능 관련 메시지

- LaTeX Warning: Reference '*identifier*' on page *page* undefined on input line *line*.
식별자 *identifier*가 \label로 정의된 적이 없을 때 발생합니다. 새로운 식별자를 만들고 컴파일을 한 적이 없는 경우에도 발생할 수 있습니다.
- LaTeX Warning: Label '*identifier*' multiply defined.
이미 정의된 식별자를 한 번 더 정의하려 할 때 발생합니다.
- LaTeX Warning: Citation '*identifier*' on page *page* undefined on input line *line*.
식별자 *identifier*가 문헌 정보 파일에서 정의된 적이 없을 때 발생합니다. 본문에 새로운 문헌을 인용하고 BibTeX이나 biber를 실행한 적이 없는 경우에도 발생할 수 있습니다.
- LaTeX Warning: There were undefined references.
정의되지 않은 식별자가 문서에서 사용되었을 때, 컴파일 완료 후 표시됩니다.
- LaTeX Warning: There were multiply-defined labels.
같은 이름의 식별자가 여러 번 정의된 적이 있을 때, 컴파일 완료 후 표시됩니다.

- LaTeX Warning: Label(s) may have changed. Rerun to get cross-references right.

상호 참조 정보에 변경 사항이 생겼을 때, 컴파일 완료 후 표시됩니다. 컴파일을 한 번 더 하면 경고 메시지가 사라집니다.

파일 관련 메시지

- \! LaTeX Error: File '*filename*' not found.
존재하지 않는 파일을 \input이나 \include, \includegraphics 등으로 불러오려 한 경우 표시됩니다. 올바른 파일 이름을 입력했는지 확인해 보기 바랍니다.
- LaTeX Warning: File '*filename*' not found on input line *line*.
존재하지 않는 파일을 \input이나 \include, \includegraphics 등으로 불러오려 한 경우 표시됩니다. 올바른 파일 이름을 입력했는지 확인해 보기 바랍니다.
- \! Unable to load picture or PDF file '*filename*'.
존재하지 않는 그림 또는 PDF 파일을 \includegraphics로 불러오려 한 경우 표시됩니다.

특정 명령어의 메시지

- ! LaTeX Error: No \title given.
\maketitle을 입력하였지만 \title을 지정하지 않았을 때 발생합니다.
- LaTeX Warning: No \author given.
\maketitle을 입력하였지만 \author를 지정하지 않았음을 뜻합니다.
- ! LaTeX Error: Something's wrong--perhaps a missing \item.
목록 환경(enumerate 등)에서 \item이 사용되지 않았을 때 발생합니다.
- ! LaTeX Error: Lonley \item--perhaps a missing list environment.
\item 제어 문자열이 목록 작성 환경 바깥에서 사용되었을 때 발생합니다.
- ! LaTeX Error: Too deeply nested.
itemize 또는 enumerate 환경을 다섯 번 이상, description 환경을 일곱 번 이상 중첩하여 사용했을 때 발생합니다.
- ! LaTeX Error: \verb illegal in argument.
\verb 제어 문자열을 다른 명령어의 인자 자리에 사용하였을 때 발생합니다. 이 제어 문자열은 정의가 복잡하여, 다른 제어 문자열의 인자로 지정할 수 없습니다.

- ! LaTeX Error: \verb ended by end of line.

\verb 제어 문자열을 시작하고 끝내지 않았을 때 발생합니다. 그대로 조판하려는 문자열의 범위를 문법에 맞게 지정했는지 확인하세요.

오버풀과 언더풀

가장 쉽게 발생하는 경고 메시지입니다. 여기에서는 이 메시지가 왜 표시되는지 간략하게만 설명하겠습니다.

- Overfull \hbox (<width> too wide) ...

T_EX이 문서의 내용을 조판할 때, 적절한 줄바꿈 지점을 찾지 못해 판면 바깥까지 문자가 조판되었을 때 발생합니다. 오류가 발생한 곳의 조판 결과를 보면 해당 부분의 내용이 오른쪽 여백 부분을 침범했음을 확인할 수 있습니다.

- Underfull \hbox (badness <num>) ...

위와 마찬가지로, T_EX이 적절한 줄바꿈 지점을 찾지 못했을 때 발생합니다. 오버풀과 다르게, 문자가 판면을 벗어나지는 않지만 단어 사이의 간격이 지나치게 넓어집니다.

- Overfull \vbox (<height> too high) ...

T_EX이 문서의 내용을 조판할 때, 적절한 페이지 나눔 지점을 찾지 못했을 때 발생합니다. 오류가 발생한 곳의 조판 결과를 보면 해당 부분의 내용이 아래쪽 여백을 침범했음을 확인할 수 있습니다.

- Underfull \vbox (badness <num>) ...

위와 마찬가지로, T_EX이 적절한 페이지 나눔 지점을 찾지 못했을 때 발생합니다. 오버풀과 다르게, 문서의 내용이 페이지의 여백을 침범하지는 않지만 문단과 문단 사이의 간격이 지나치게 넓어집니다.

B.1.2 커뮤니티의 도움 받기

L^AT_EX으로 문서를 작성하다가 원인을 도저히 알 수 없는 문제가 발생했을 때, 특정 기능은 어떻게 구현하는지 궁금할 때, T_EX과 L^AT_EX의 깊은 내용이 궁금할 때 등 많은 경우에는 인터넷 검색을 통해 궁금증을 해결할 수 있습니다. 이번 절에서는 인터넷 검색을 통해 궁금증을 해결할 때 참고할 사항을 몇 가지 알려드리겠습니다.

- 기초적인 내용이 아니라면 영어로 검색한다. 대부분의 경우, 영어로 검색했을 때 양질의 답변을 많이 얻을 수 있습니다. 국내에서도 많은 사람들이 사용하는 Word,

아래아한글, 포토샵 등과 달리, \TeX 은 비교적 한정된 분야의 사람들이 사용하므로 한국어로 검색하는 경우 좋은 정보를 많이 찾기 어렵습니다. 영어가 어렵더라도 영어로 정보를 찾는 습관을 들이는 것이 좋습니다. 다만, 아주 기초적인 내용은 한국어로 작성된 글도 몇몇 있으므로 기본적인 내용은 한국어로 검색해도 좋은 결과를 얻을 수 있습니다.

- 특정 기능을 구현하는 방법, 특정 주제에 관한 궁금증은 질문-답변 형식의 웹사이트를 사용한다. 구체적인 기능을 어떻게 구현하는지 알고 싶거나, 특정 주제에 관한 질문은 Stack Exchange나 KTUG의 질문 게시판 등 질문-답변 형식의 웹사이트를 사용하는 것이 좋습니다. 질문 및 답변 웹사이트에서는 일반적으로 답변 작성자가 예시 코드를 제시하므로, 이를 참고해 더 쉽게 해당 기능을 사용하는 방법을 알 수 있습니다.
- 패키지의 사용법은 패키지 매뉴얼을 통해 확인한다. 이 패키지가 어떤 기능을 제공하는지, 어떻게 사용하는지 궁금하다면 해당 패키지 매뉴얼을 읽는 것이 좋습니다. 패키지 매뉴얼은 CTAN이라는 웹사이트에서 확인할 수 있습니다. 컴퓨터 터미널의 명령 창에 `texdoc <pkg name>`을 입력하거나, \TeX 배포판의 관리 도구를 사용해 확인할 수도 있습니다. 예를 들어, kotex 패키지의 문서는 터미널에 `texdoc kotex`을 입력해 확인할 수 있습니다.
- 질문을 올릴 때에는 최소 동작 예제를 첨부한다. 문서에 문제가 있어 질문을 올리든, 특정 기능을 구현하는 방법이 궁금해서 질문을 올리든, 답변 작성자가 참고할 수 있는 최소 동작 예제(Minimal Working Example, MWE)를 꼭 함께 첨부해야 합니다. 문제가 있었다면 그 문제가 발생하는 문서를 첨부하고, 기능을 구현하는 방법이 궁금하다면 최소한의 내용과 함께 자신이 원하는 출력 결과 이미지를 첨부하는 것이 좋습니다. 자신이 작성하던 문서 자체를 그대로 올릴 필요는 없습니다.

최소 동작 예제를 첨부하는 것은 질문-답변 사이트의 기본적인 예절입니다. 답변 작성자는 자신의 시간을 할애해 여러분에게 답변을 제공하는 것이므로, 답변 작성자를 배려해 질문을 작성하는 것이 예의입니다.

B.2 용어의 한국어 표기

\TeX 과 \LaTeX , 문서 조판, 수식 조판 등에는 다양한 개념과 용어가 사용됩니다. 이 책에서는 이러한 용어를 한국어로 번역·음역하여 사용하였습니다. 이 책에서 사용한 표기를

이번 절에 정리하였습니다.

- argument: 인자
 - long argument: 긴 인자
- bibliography: 참고 문헌
- bookmark: 책갈피
- cases: 경우 (식을 위아래로 나열하고 식의 왼쪽을 중괄호로 묶은 형태의 구조)
- character: 문자
- class: 클래스
- command: 명령어
- comment: 주석
- commutative diagram: 가환 도표
- control sequence: 제어 문자열
 - control symbol: 제어 기호
 - control word: 제어 단어
- counter: 카운터
- cross reference: 상호 참조
- dash: 대시
- delimiter: 구분 기호
- diacritic: 악센트
 - grave accent: 그레이브 악센트
 - tilde accent: 틸데 악센트
- displayed formula: 별행수식
- documentation: 안내서
- ellipsis: 줄임표
- environment: 환경
- font: 글꼴
 - blackboard bold: 칠판 볼드체
 - calligraphic font: 흘림체
 - fraktur: 독일 활자체
 - script font: 필기체
- series: 시리즈
- shape: 셰이프
- typeface: 타입페이스
- footer: 바닥글
- footnote: 각주
- float: 유동 개체
- group: 영역
- header: 머리글
- hyphen: 하이픈
- hyphenation: 분철
- index: 색인
- inline formula: 행중수식
- length: 길이
 - rigid length: 고정 길이
 - rubber length: 가변 길이
- ligature: 합자
- lining figure: 대문자 숫자
- log-like functions: 로그형 함수
- macro: 매크로
- margin note: 여백글
- mode: 모드
 - math mode: 수식 모드
 - text mode: 텍스트 모드
- old-style figure: 소문자 숫자
- one-sided editing: 단면 편집
- operator: 연산자
- package: 패키지
- parameter: 매개 변수
- preamble: 전처리부
- preface: 머리말

- style: 스타일
 - display style: 별행수식 스타일
 - text style: 행중수식 스타일
- subscript: 아래 첨자
- superscript: 위 첨자
- \TeX distribution: \TeX 배포판
- theorem-like environment: 정리형 환경
- two-sided editing: 양면 편집
- underscore: 밑줄 문자

후기

2022년 11월 3일에 이 파일을 처음 만들고 약 1년 반 만에 이 책이 완성되었습니다. 2022년 여름에 학과 행사로 L^AT_EX 세미나가 열렸을 때, 그 보조 자료로 간략하게 만들었던 L^AT_EX 가이드북을 발전시켜 책으로 내자는 생각을 하였고, 그 해 11월에 책을 작성하기 시작하였습니다.

이 책을 작성하기까지, 책의 내용이 수도 없이 갈아엎어졌습니다. 내용을 구성하는 것과 그 순서를 정하는 것에서 시작하여, 어떤 표현을 사용할지, 내용은 어떤 방향으로 풀어가야 할지, 책의 디자인은 어떻게 할지 많은 고민을 하였습니다. 특히, 기본적인 기능을 설명하는 제II편과 제III편은 못해도 다섯 번은 썼던 글을 지우고 처음부터 다시 썼던 것 같습니다.

결국 긴 시간이 흘러 책을 완성하였지만, 이 책은 사실 완벽한 것이 아닙니다. 이 책에 넣고 싶었지만 아직 넣지 못한 내용이 한가득이며, 이는 또다시 긴 시간 후에 추가된 채로 공개될 것입니다. 다루지 못한 내용을 몇 가지만 나열해 보자면 다음과 같습니다.

캡션의 스타일화, 각주·미주·측주 기능, 목차의 사용자화, *MakeIndex*와 *xindy*를 사용한 색인 생성, 용어 목록 생성, 매크로 정의와 관련된 고급 기능, 패키지와 클래스 파일을 직접 작성하는 방법, 문서의 레이아웃 사용자화, 다양한 서식을 지정하도록 하는 패키지, 다양한 글꼴을 사용하는 방법, 문서의 언어 설정, 상자와 패선 만들기, 공백과 채움선 넣기, 줄 나눔과 페이지 나눔 조정하기, 수식 모드의 스타일, 수식 기호 간의 간격 세부 조정, 유용한 클래스와 패키지 소개

이러한 내용을 넣지 못한 것은 저의 군 입대를 앞두고 시간이 부족했기 때문입니다. 많은 내용을 책에 넣으려다 보니 집필 시간이 무한정 늘어났고, 제가 입대하기 전까지 이 모든 내용을 책에 넣는 것은 불가능하다고 판단했습니다. 따라서 현재까지 작성된

내용만을 포함하여 이 문서를 공개하기로 결정하였습니다. 입대를 하루 앞둔 지금 시점에서 저는 군 생활을 무사히 마친 뒤, 빠진 내용을 잘 다듬어 개정된 문서를 여러분께 보여 드리고 싶을 따름입니다.

도움 주신 분들

이 책을 작성하는 데 도움을 주신 분들이 많습니다. \LaTeX 가이드북 집필을 동아리 프로젝트로 진행할 것을 제안한 이재현 님의 조언 덕분에, 이 책은 KAIST 수학문제연구회의 많은 회원들의 도움을 받아 작성되었습니다. 오타 검토에서부터 시작해 내용 제안, 표현 검토, 예시 및 연습문제 준비 등 다방면에 도움을 주셨습니다. 이 프로젝트를 진행하는 동안 도움 주셨던 김다인, 권도형, 권순용, 노희윤, 문강연, 박현수, 심우용, 어세훈, 이재현, 이종민, 이태영, 장지연, 조현준, 최은수, 하준안 님께 감사의 말씀을 드립니다.

전역 후, 더 나은 \LaTeX 안내서를 만들기 위해 열심히 노력하겠습니다. 이 책을 읽어 주시는 독자 여러분께도 감사의 말씀 드립니다. 이 책을 읽으며 생긴 문의 사항은 kangwh.2003@gmail.com으로 전자 우편을 보내 주시면 됩니다.

2024년 4월 29일

강우현

참고 문헌

- [1] Donald Arseneau. *The wrapfig package*. Version 3.6. Jan. 31, 2023. URL: <http://mirrors.ctan.org/macros/latex/contrib/wrapfig/wrapfig-doc.pdf> (visited on 04/26/2024).
- [2] Karl Berry. *L^AT_EX 2_ε: An unofficial reference manual*. Sept. 2023. URL: <https://latexref.xyz/dev/latex2e.pdf> (visited on 04/29/2024).
- [3] Karl Berry, ed. *The T_EX Live Guide—2024*. Mar. 2024. URL: <https://tug.org/texlive/doc/texlive-en/texlive-en.pdf> (visited on 04/29/2024).
- [4] Javier Bezos. *Customizing lists with the enumitem package*. Version 3.9. June 20, 2019. URL: <http://mirrors.ctan.org/macros/latex/contrib/enumitem/enumitem.pdf> (visited on 04/18/2024).
- [5] Javier Bezos and Johannes L. Braams. *Babel User Guide: Localization and internalization*. Version 24.4. Apr. 20, 2024. URL: <http://mirrors.ctan.org/macros/latex/required/babel/base/babel.pdf> (visited on 04/28/2024).
- [6] Johannes Braams. *The alltt environment*. L^AT_EX Project team. Jan. 29, 2021. URL: <http://mirrors.ctan.org/macros/latex/base/alltt.pdf> (visited on 04/25/2024).
- [7] C_T_EX 宏集手册. Chinese. Version 2.5.10. July 14, 2022. URL: <http://mirrors.ctan.org/language/chinese/ctex/ctex.pdf> (visited on 04/28/2024).
- [8] D. P. Carlisle. *Packages in the ‘graphics’ bundle*. The L^AT_EX Project. Mar. 5, 2021. URL: <http://mirrors.ctan.org/macros/latex/required/graphics/grfguide.pdf> (visited on 04/16/2024).
- [9] David Carlisle. *The enumerate package*. Version 3.00. L^AT_EX Project team. July 4, 2023. URL: <http://mirrors.ctan.org/macros/latex/required/tools/enumerate.pdf> (visited on 04/29/2024).
- [10] David Carslie. *The ifthen package*. L^AT_EX Project Team. Apr. 13, 2022. URL: <http://mirrors.ctan.org/macros/latex/base/ifthen.pdf> (visited on 04/16/2024).

- [11] François Charette et al. *Polyglossia: Modern multilingual typesetting with Xe₂LaTeX and LuaLaTeX*. Version 2.1. Mar. 7, 2024. URL: <http://mirrors.ctan.org/macros/unicodetex/latex/polyglossia/polyglossia.pdf> (visited on 04/28/2024).
- [12] CTAN: *Class*. Mar. 29, 2024. URL: <https://ctan.org/topic/class> (visited on 03/29/2024).
- [13] CTAN: *Packages*. Apr. 12, 2024. URL: <https://ctan.org/pkg> (visited on 04/02/2024).
- [14] Toby Cubitt. *The cleveref package*. Version 0.21.4. Mar. 27, 2018. URL: <http://mirrors.ctan.org/macros/latex/contrib/cleveref/cleveref.pdf> (visited on 04/18/2024).
- [15] David S. Dummit and Richard M. Foote. *Abstract Algebra*. 3rd ed. John Wiley and Sons, 2004.
- [16] *LaTeX for authors — current version*. LaTeX Project Team. May 23, 2023. URL: <https://www.latex-project.org/help/documentation/usrguide.pdf> (visited on 04/16/2024).
- [17] Michel Goossens et al. *The LaTeX Graphics Companion*. 2nd ed. Addison-Wesley, 2007.
- [18] Andy Hammerlindl, John Bowman, and Tom Prince. *Asymptote: the Vector Graphics Language*. Version 2.89. 2024. URL: <http://mirrors.ctan.org/graphics/asymptote/doc/asymptote.pdf> (visited on 04/28/2024).
- [19] John D. Hobby. *METAPOST: A User's Manual*. Version 2.10. Feb. 18, 2024. URL: <http://mirrors.ctan.org/systems/doc/metapost/mpman.pdf> (visited on 04/16/2024).
- [20] Jobst Hoffmann. *The Listings Package*. Version 1.10a. Mar. 11, 2024. URL: <http://mirrors.ctan.org/macros/latex/contrib/listings/listings.pdf> (visited on 04/25/2024).
- [21] Morten Høgholm and Lars Madsen. *The mathtools package*. Version 1.30. LaTeX3 Project. Mar. 11, 2024. URL: <http://mirrors.ctan.org/macros/latex/contrib/mathtools/mathtools.pdf> (visited on 04/28/2024).
- [22] Vafa Khalighi. *The bidi Package: Bidirectional typesetting in plain TeX and LaTeX, using XeTeX engine*. Version 39.8. Aug. 2, 2023. URL: <http://mirrors.ctan.org/macros/xetex/latex/bidi/bidi-doc.pdf> (visited on 04/28/2024).
- [23] Dohyun Kim. *XeTeX-ko 간단 매뉴얼*. Korean. Version 4.3. Mar. 1, 2024. URL: <http://mirrors.ctan.org/macros/xetex/generic/xetexko/xetexko-doc.pdf> (visited on 04/28/2024).

- [24] Philip Kime, Moritz Wemheuer, and Philipp Lehman. *The biblatex Package*. Version 3.20. Mar. 21, 2024. URL: <http://mirrors.ctan.org/macros/latex/contrib/biblatex/doc/biblatex.pdf> (visited on 04/28/2024).
- [25] Donald E. Knuth. *The Art of Computer Programming*. Addison-Wesley Professional, 1968-.
- [26] Donald E. Knuth. *The T_EXbook*. Addison-Wesley Professional, 1986.
- [27] Leslie Lamport. *L^AT_EX: A Document Preparation System*. 2nd ed. Addison-Wesley, 1994.
- [28] Anselm Lingnau. *An Improved Environment for Floats*. Version 1.3d. Nov. 8, 2001. URL: <http://mirrors.ctan.org/macros/latex/contrib/float/float.pdf> (visited on 04/26/2024).
- [29] Frank Mittelbach. *The doc and shortvrb Packages*. Version 3.0m. L^AT_EX Project team. Jan. 14, 2024. URL: <http://mirrors.ctan.org/macros/latex/base/alltt.pdf> (visited on 04/25/2024).
- [30] Frank Mittelbach. *The varioref package*. L^AT_EX Project team. Jan. 9, 2022. URL: <http://mirrors.ctan.org/macros/latex/required/tools/varioref.pdf> (visited on 04/28/2024).
- [31] Frank Mittelbach and David Carlisle. *A new implementation of L^AT_EX's tabular and array environment*. Version 2.5g. L^AT_EX Project Team. Nov. 21, 2023. URL: <http://mirrors.ctan.org/macros/latex/required/tools/array.pdf> (visited on 04/16/2024).
- [32] Frank Mittelbach, Robin Fairbairns, and Werner Lemberg. *L^AT_EX font encodings*. L^AT_EX Project Team. Feb. 18, 2016. URL: <https://www.latex-project.org/help/documentation/encguide.pdf> (visited on 04/18/2024).
- [33] Frank Mittelbach, Ulrike Fischer, and Joseph Wright. *The L^AT_EX Companion*. 3rd ed. Addison-Wesley, 2023.
- [34] James Munkres. *Topology*. Pearson New International Edition. 2nd ed. Pearson, 2014.
- [35] Heiko Oerdiel. *The bookmark package*. Version 1.31. Dec. 10, 2023. URL: <http://mirrors.ctan.org/macros/latex/contrib/bookmark/bookmark.pdf> (visited on 04/29/2024).
- [36] Tobias Oetiker et al. *The Not So Short Introduction to L^AT_EX 2_ε*. L^AT_EX 2_ε in 139 minutes. Version 6.4. Mar. 9, 2021. URL: <http://mirrors.ctan.org/info/lshort/english/lshort.pdf> (visited on 04/26/2024).

- [37] Oren Patashnik. *BibTeXing*. Version 0.99d. Feb. 8, 1988. URL: <http://mirrors.ctan.org/biblio/bibtex/base/btxdoc.pdf> (visited on 04/26/2024).
- [38] Geoffrey M. Poore. *Geoffrey M. Poore*. Version 2.9. Dec. 18, 2023. URL: <http://mirrors.ctan.org/macros/latex/contrib/minted/minted.pdf> (visited on 04/25/2024).
- [39] Geoffrey M. Poore. *The fvextra package*. Version 1.6.1. Nov. 28, 2023. URL: <http://mirrors.ctan.org/macros/latex/contrib/fvextra/fvextra.pdf> (visited on 04/25/2024).
- [40] Sebastian Rahtz and Heiko Oberdiek. *Hypertext marks in L^AT_EX: a manual for hyperref*. Version 7.01h. The L^AT_EX3 Project. Jan. 20, 2024. URL: <http://mirrors.ctan.org/macros/latex/contrib/hyperref/doc/hyperref-doc.pdf> (visited on 03/29/2024).
- [41] Kristoffer H. Rose. *Xy-pic User's Guide*. Version 3.8.9. June 10, 2013. URL: <http://mirrors.ctan.org/macros/generic/diagrams/xy-pic/doc/xyguide.pdf> (visited on 04/16/2024).
- [42] Kristoffer H. Rose and Ross Moore. *Xy-pic Reference Manual*. Version 3.8.9. Oct. 6, 2013. URL: <http://mirrors.ctan.org/macros/generic/diagrams/xy-pic/doc/xyrefer.pdf> (visited on 04/16/2024).
- [43] Walter Rudin. *Principles of Mathematical Analysis*. 3rd ed. McGraw-Hill, 1976.
- [44] Rainer Schöpf, Bernd Raichle, and Chris Rowley. *A New Implementation of L^AT_EX's verbatim and verbatim* Environments*. L^AT_EX Project team. Nov. 16, 2023. URL: <http://mirrors.ctan.org/macros/latex/required/tools/verbatim.pdf> (visited on 04/25/2024).
- [45] Martin Schröder. *The ragged2e-package*. June 22, 2023. URL: <http://mirrors.ctan.org/macros/latex/contrib/ragged2e/ragged2e.pdf> (visited on 04/17/2024).
- [46] Axel Sommerfeldt. *The subcaption package*. Version 1.6. Aug. 13, 2023. URL: <http://mirrors.ctan.org/macros/latex/contrib/caption/subcaption.pdf> (visited on 04/26/2024).
- [47] Augusto Stoffel. *Commutative diagrams with TikZ*. Version 1.0. May 4, 2021. URL: <http://mirrors.ctan.org/graphics/pgf/contrib/tikz-cd/tikz-cd-doc.pdf> (visited on 04/14/2024).
- [48] Till Tantau. *The TikZ and pgf packages: Manual for Version 3.1.10*. Jan. 15, 2023. url: <http://mirrors.ctan.org/graphics/pgf/base/doc/pgfmanual.pdf> (visited on 04/16/2024).
- [49] *The Definitive Guide to BibTeX - bibtex.com*. Apr. 28, 2024. url: <https://www.bibtex.com> (visited on 04/28/2024).

- [50] *The LuaTeX-ja package*. Version 20240219.0. The LuaTeX-ja project team. Feb. 19, 2024. url: <http://mirrors.ctan.org/macros/luatex/generic/luatexja/doc/luatexja-en.pdf> (visited on 04/28/2024).
- [51] Kresten Krab Thorup, Frank Jensen, and Chris Rowley. *The calc package: Infix notation arithmetic in L^AT_EX*. Version 4.3. L^AT_EX Project team. July 8, 2023. url: <http://mirrors.ctan.org/macros/latex/required/tools/calc.pdf> (visited on 04/28/2024).
- [52] *User's Guide for the amsmath Package*. Version 2.1. American Mathematical Society and L^AT_EX Project. Feb. 18, 2020. url: <http://mirrors.ctan.org/macros/latex/required/amsmath/amslatex.pdf> (visited on 04/16/2024).
- [53] *Using the amsthm Package*. Version 2.20.3. American Mathematical Society. Sept. 2017. url: <http://mirrors.ctan.org/macros/latex/required/amsthm/doc/amsthmdoc.pdf> (visited on 04/16/2024).
- [54] Timothy Van Zandt. *PSTricks: PostScript macros for Generic TeX*. Dec. 10, 2007. url: <http://mirrors.ctan.org/graphics/pstricks/base/doc/pst-user.pdf> (visited on 04/14/2024).
- [55] Timothy Van Zandt. *The 'fancyvrb' package: Fancy Verbatims in L^AT_EX*. Ed. by Denis Girou, Sebastian Rahtz, and Herbert Voß. Princeton, USA, Jan. 20, 2024. url: <http://mirrors.ctan.org/macros/latex/contrib/fancyvrb/doc/fancyvrb-doc.pdf> (visited on 04/25/2024).
- [56] 김강수. 한국어 텍 koT_EX. Korean. July 2021. url: <https://mirrors.ctan.org/language/korean/kotex-utf/doc/kotex-doc.pdf> (visited on 04/16/2024).
- [57] 네이버 글꼴 모음. Korean. Apr. 17, 2024. url: <https://hangeul.naver.com/fonts/search?f=nanum> (visited on 04/17/2024).

용어 찾아보기

C

CTAN 10

K

KTUG 10

L

L^AT_EX 4, 9

 L^AT_EX2.09 4

 L^AT_EX 2 ϵ 4

 L^AT_EX3 프로그래밍 레이어 5

L^AT_EX 프로젝트 팀 4

LuaL^AT_EX 9

LuaL^AT_EX 9

P

PDFL^AT_EX 9

PDFT_EX 9

pL^AT_EX 9

T

T_EX 3

T_EX 배포판 7

TUG 10

U

upL^AT_EX 9

upT_EX 9

X

X_qL^AT_EX 9

X_qL^AT_EX 9

ㄱ

가운뎃점 42

가환 도표 151

각주 60

강조 70

개행 문자 27

공백 문자 27

관계 기호 102

근호 115

긴 인자 262

길이 280

 가변 길이 281

 고정 길이 281

길이 변수 277

ㄷ

대형 연산자 100

도널드 어빈 커누스 3

드러냄표 70

따옴표 40

ㅁ

목록 65

 구분점 목록 65

 번호 목록 65

 사전식 목록 65

목차 56

 그림 목차 57

 표 목차 57

| | |
|-------------|---------|
| 문자 코드 | 51 |
| 문장 부호 | 39 |
| 수식 | 106 |
| 밑줄 | 70, 115 |

ㅂ

| | |
|-------------|-----|
| 법 연산자 | 102 |
| 별행수식 | 97 |
| 부록 | 60 |
| 분수 | 114 |
| 연분수 | 114 |
| 붙임표 | 43 |

ㅅ

| | |
|--------------|----|
| 색인 | 64 |
| 선언 | 21 |
| 셰이프 | 71 |
| 수식 모드 | 96 |
| 수학적 문단 | 81 |
| 스타일 | 97 |
| 시리즈 | 71 |

ㅇ

| | |
|---------------|---------|
| 아포스트로피 | 40 |
| 앵커 | 252 |
| 엔 대시 | 40 |
| 엔진 | 7 |
| 레거시 엔진 | 10 |
| 유니코드 엔진 | 10 |
| 일본어 엔진 | 10 |
| 엠 대시 | 40 |
| 여백글 | 61 |
| 역슬래시 | 19, 113 |
| 영역 | 21 |
| 윗줄 | 115 |
| 음의 부호 | 40 |
| 이항 계수 | 114 |
| 이항 연산자 | 99 |
| 인용문 | 68 |
| 인자 | 20 |
| 선택 인자 | 20 |
| 일반 기호 | 107 |

ㅈ

| | |
|--------------|--------|
| 자동 조사 | 210 |
| 적분 기호 | 100 |
| 전처리부 | 23 |
| 정리형 환경 | 81 |
| 정수 | 278 |
| 제어 문자열 | 19 |
| 제어 기호 | 20 |
| 제어 단어 | 20 |
| 주석 | 22, 60 |
| 줄임표 | 110 |
| 영어 | 41 |
| 한국어 | 42 |
| 줄표 | 43 |

ㅊ

| | |
|----------------|-----|
| 참고 문헌 목록 | 62 |
| 책갈피 | 251 |
| 첨자 | 73 |
| 아래 첨자 | 116 |
| 위 첨자 | 116 |
| 초록 | 56 |

ㅋ

| | |
|--------------|-----|
| 카운터 | 269 |
| 종속 | 274 |
| 클래스 | 23 |
| 표준 클래스 | 23 |

ㅌ

| | |
|--------------|----|
| 텍스트 모드 | 96 |
|--------------|----|

ㅍ

| | |
|---------------|-----|
| 패밀리 | 71 |
| 패키지 | 24 |
| 패키지 안내서 | 25 |
| 표 | 167 |
| 표제 | 57 |
| 장 | 59 |
| 절 | 57 |
| 편 | 60 |
| 표지 | 55 |

ㅎ

| | | | |
|-------------|-----|------------|-----|
| 하이퍼링크 | 247 | 행중수식 | 97 |
| 하이픈 | 40 | 화살표 | 105 |
| 행렬 | 123 | 환경 | 22 |

명령어 찾아보기

| | | | | | |
|--------------------------|--------|-----------------------------|----------------|-----------------|-----|
| ? (?) | 107 | * (*) | 99 | [(]) | 107 |
| ! (!) | 107 | 표 환경에서 | 173 | \[| 29 |
| 표 환경에서 | 172 | + (+) | 99 |] (]) | 107 |
| . (.) | 108 | , (,) | 107 | \] | 29 |
| -- (-) | 40, 99 | \, | 40 | ` (‘) | 40 |
| ' | | ,, (,,) | 40 | ?` (¿) | 40 |
| 수식 모드 (‘) | 108 | / (/) | 108 | !` (¡) | 40 |
| 텍스트 모드 (‘) | 40 | ~ | 28 | \` | 38 |
| '' (") | 40 | \\ | 21, 30, 38, 41 | `` (") | 40 |
| _ | 30, 47 | align, alignat, eqnarray, | | () | 108 |
| \. | 38 | flalign, gather, multiline, | | 표 환경에서 | 168 |
| \' | 38 | split 환경에서 | 142 | | |
| \" | 38 | tikzcd 환경에서 | 152 | A | |
| \# | 30 | 경우 환경에서 | 126 | \AA (Å) | 38 |
| \\$ | 30, 47 | 표 환경에서 | 168 | \aa (å) | 38 |
| \% | 30 | 행렬 환경에서 | 124 | abstract 환경 | 56 |
| & | | verse 환경에서 | 67 | \acute (á) | 109 |
| align, alignat, flalign, | | : | 103 | \addbibresource | 244 |
| split 환경에서 | 142 | ; | 107 | \addtocounter | 270 |
| eqnarray 환경에서 | 144 | < (<) | 103 | \addtolength | 277 |
| tikzcd 환경에서 | 152 | 표 환경에서 | 172 | \AE (Æ) | 38 |
| 경우 환경에서 | 126 | << («) | 40 | \ae (æ) | 38 |
| 표 환경에서 | 168 | = (=) | 103 | \aleph (ℵ) | 108 |
| 행렬 환경에서 | 124 | \= | 38 | align 환경 | 144 |
| \& | 30 | > (>) | 103 | align* 환경 | 144 |
| ((()) | 107 | 표 환경에서 | 172 | alignat 환경 | 145 |
| \(| 29, 95 | >> (») | 40 | alignat* 환경 | 145 |
|) ()) | 107 | @ (표 환경) | 171 | \Alpha | 272 |
| \) | 29, 95 | \@ | 42 | \alpha | 272 |

| | | | | | |
|---|----------|--|---------|--|-----|
| <code>\alpha</code> (α) | 109 | <code>\beta</code> (β) | 109 | <code>\blacktriangledown</code> (▼) | 108 |
| <code>\amalg</code> (\amalg) | 99 | <code>\beth</code> (\beth) | 108 | <code>\blacktriangleleft</code> (◀) | 103 |
| <code>\and</code> | 56 | <code>\between</code> (\between) | 103 | <code>\blacktriangleright</code> (▶) | 103 |
| <code>\angle</code> (\angle) | 108 | <code>\bf</code> | 71, 138 | <code>Bmatrix</code> 환경 | 123 |
| <code>\appendix</code> | 60 | <code>\bfseries</code> | 72 | <code>bmatrix</code> 환경 | 123 |
| <code>\approx</code> (\approx) | 103 | <code>\bibitem</code> | 62 | <code>\bmod</code> (mod) | 102 |
| <code>\Approxcolon</code> ($\approx::$) | 105 | <code>\bibliography</code> | 244 | <code>\boldmath</code> | 138 |
| <code>\approxcolon</code> ($\approx:$) | 105 | <code>\bibliographystyle</code> | 244 | <code>\boldsymbol</code> | 137 |
| <code>\approxeq</code> (\approx) | 103 | <code>\Big</code> | 119 | <code>\bot</code> (\perp) | 108 |
| <code>\ar</code> | 156 | <code>\big</code> | 119 | <code>\bowtie</code> (\bowtie) | 103 |
| <code>\arabic</code> | 272 | <code>\bigcap</code> (\bigcap) | 100 | <code>\Box</code> (\square) | 108 |
| <code>\arccos</code> (arccos) | 101 | <code>\bigcirc</code> (\bigcirc) | 99 | <code>\boxdot</code> (\boxdot) | 99 |
| <code>\arcsin</code> (arcsin) | 101 | <code>\bigcup</code> (\bigcup) | 100 | <code>\boxminus</code> (\boxminus) | 99 |
| <code>\arctan</code> (arctan) | 101 | <code>\Bigg</code> | 119 | <code>\boxplus</code> (\boxplus) | 99 |
| <code>\arg</code> (arg) | 101 | <code>\bigg</code> | 119 | <code>\boxtimes</code> (\boxtimes) | 99 |
| <code>array</code> 환경 | 167, 169 | <code>\Biggl</code> | 120 | <code>\breve</code> (\breve) | 109 |
| <code>\arrow</code> | 152 | <code>\biggl</code> | 120 | <code>Bsmallmatrix</code> 환경 | 126 |
| <code>\ast</code> ($*$) | 99 | <code>\Biggm</code> | 120 | <code>bsmallmatrix</code> 환경 | 126 |
| <code>\asymp</code> (\asymp) | 103 | <code>\biggm</code> | 120 | <code>\bullet</code> (•) | 99 |
| <code>\author</code> | 55 | <code>\biggr</code> | 120 | <code>\Bumpeq</code> (\Bumpeq) | 103 |
| <code>\autopageref</code> | 227 | <code>\biggr</code> | 120 | <code>\bumpeq</code> (\bumpeq) | 103 |
| <code>\autopageref*</code> | 248 | <code>\Bigl</code> | 120 | | |
| <code>\autoref</code> | 227 | <code>\bigl</code> | 120 | | |
| <code>\autoref*</code> | 248 | <code>\Bigm</code> | 120 | | |
| | | <code>\bigm</code> | 120 | C | |
| B | | <code>\bigodot</code> (\bigodot) | 100 | <code>c</code> (표 환경) | 168 |
| <code>b</code> (표 환경) | 170 | <code>\bigoplus</code> (\bigoplus) | 100 | <code>\c</code> | 38 |
| <code>\b</code> | 38 | <code>\bigotimes</code> (\bigotimes) | 100 | <code>\cal</code> | 138 |
| <code>\backepsilon</code> (ϵ) | 103 | <code>\Bigr</code> | 120 | <code>\Cap</code> (\cap) | 99 |
| <code>\backmatter</code> | 54 | <code>\bigr</code> | 120 | <code>\cap</code> (\cap) | 99 |
| <code>\backprime</code> (\backprime) | 108 | <code>\bigl</code> | 120 | <code>\capitalacute</code> | 48 |
| <code>\backsim</code> (\backsim) | 103 | <code>\bigl</code> | 120 | <code>\capitalbreve</code> | 48 |
| <code>\backsimeq</code> (\backsimeq) | 103 | <code>\bigl</code> | 120 | <code>\capitalcaron</code> | 48 |
| <code>\backslash</code> | 30 | <code>\bigl</code> | 120 | <code>\capitalcedilla</code> | 48 |
| <code>\bar</code> (\bar) | 109 | <code>\bigl</code> | 120 | <code>\capitalcircumflex</code> | 48 |
| <code>\barwedge</code> (\barwedge) | 99 | <code>\bigl</code> | 120 | <code>\capitaldieresis</code> | 48 |
| <code>\Bbbk</code> (\mathbb{k}) | 108 | <code>\bigl</code> | 120 | <code>\capitaldotaccent</code> | 48 |
| <code>\because</code> (\because) | 103 | <code>\bigl</code> | 120 | <code>\capitalgrave</code> | 48 |
| <code>\begin</code> | 22 | <code>\bigl</code> | 120 | <code>\capitalhungarianumlaut</code> | 48 |
| <code>\begingroup</code> | 96 | <code>\bigl</code> | 120 | <code>\capitalmacron</code> | 48 |
| <code>\belowpdfbookmark</code> | 252 | <code>\bigl</code> | 120 | <code>\capitalnewtie</code> | 48 |
| | | <code>\blacklozenge</code> (◆) | 108 | | |
| | | <code>\blacksquare</code> (■) | 108 | | |
| | | <code>\blacktriangle</code> (▲) | 108 | | |

| | | | | | |
|---|---------|---|-----|--|---------|
| <code>\capitalgreek</code> | 48 | <code>\cong</code> (\cong) | 103 | <code>\Dashcolon</code> (\dashcolon) | 105 |
| <code>\capitalring</code> | 48 | <code>\coprod</code> (\coprod) | 100 | <code>\dashcolon</code> (\dashcolon) | 105 |
| <code>\capitaltie</code> | 48 | <code>\copyright</code> (\copyright) | 47 | <code>\dashleftarrow</code> (\dashleftarrow) .. | 106 |
| <code>\capitaltilde</code> | 48 | <code>\cos</code> (\cos) | 101 | <code>\dashrightarrow</code> (\dashrightarrow) .. | 106 |
| <code>\caption</code> | 201 | <code>\cosh</code> (\cosh) | 101 | <code>\dashv</code> (\dashv) | 103 |
| <code>\caption*</code> | 201 | <code>\cot</code> (\cot) | 101 | <code>\date</code> | 55 |
| <code>cases*</code> 환경 | 126 | <code>\coth</code> (\coth) | 101 | <code>\dblcolon</code> (\dblcolon) | 105 |
| <code>\cdot</code> (\cdot) | 99 | <code>\counterwithin</code> | 274 | <code>dcases</code> 환경 | 127 |
| <code>\cdots</code> (\cdots) | 107 | <code>\counterwithin*</code> | 274 | <code>dcases*</code> 환경 | 127 |
| <code>Center</code> 환경 | 69 | <code>\counterwithout</code> | 274 | <code>\ddag</code> (\ddag) | 47 |
| <code>center</code> 환경 | 68 | <code>\counterwithout*</code> | 274 | <code>\ddagger</code> (\ddagger) | 99 |
| <code>\centerdot</code> (\cdot) | 99 | <code>\Cpageref</code> | 219 | <code>\ddddot</code> (\ddddot) | 109 |
| <code>\Centering</code> | 69 | <code>\cpageref</code> | 219 | <code>\dddot</code> (\dddot) | 109 |
| <code>\centering</code> | 69 | <code>\Cpagerefrange</code> | 220 | <code>\ddot</code> (\ddot) | 109 |
| <code>\cfrac</code> | 114 | <code>\cpagerefrange</code> | 220 | <code>\ddots</code> (\ddots) | 107 |
| <code>\chapter</code> | 58 | <code>\Cref</code> | 219 | <code>\DeclareGraphicsExtensions</code>
191 | |
| <code>\chapter*</code> | 58 | <code>\cref</code> | 219 | <code>\DeclareGraphicsRule</code> .. | 190 |
| <code>\check</code> (\check) | 109 | <code>\Cref*</code> | 248 | <code>\def</code> | 263 |
| <code>\checkmark</code> (\checkmark) | 47 | <code>\cref*</code> | 248 | <code>\deg</code> (\deg) | 101 |
| <code>\chi</code> (χ) | 109 | <code>\Crefname</code> | 220 | <code>\Delta</code> (Δ) | 109 |
| <code>\circ</code> (\circ) | 99 | <code>\Creffrange</code> | 220 | <code>\delta</code> (δ) | 109 |
| <code>\circeq</code> (\circeq) | 103 | <code>\creffrange</code> | 220 | <code>description</code> 환경 | 65 |
| <code>\circlearrowleft</code> (\circlearrowleft) .. | 106 | <code>\creffrange*</code> | 248 | <code>\det</code> (\det) | 101 |
| <code>\circlearrowright</code> (\circlearrowright) .. | 106 | <code>\creffrange*</code> | 248 | <code>\DH</code> (\DH) | 38 |
| <code>\circledast</code> (\circledast) | 99 | <code>\csc</code> (\csc) | 101 | <code>\dh</code> (\dh) | 38 |
| <code>\circledcirc</code> (\circledcirc) | 99 | <code>\Cup</code> (\Cup) | 99 | <code>\diagdown</code> (\diagdown) | 108 |
| <code>\circleddash</code> (\circleddash) | 99 | <code>\cup</code> (\cup) | 99 | <code>\diagup</code> (\diagup) | 108 |
| <code>\circledR</code> (\circledR) | 47 | <code>\curlyeqprec</code> (\curlyeqprec) | 103 | <code>\Diamond</code> (\Diamond) | 108 |
| <code>\circledS</code> (\circledS) | 108 | <code>\curlyeqsucc</code> (\curlyeqsucc) | 103 | <code>\diamond</code> (\diamond) | 99 |
| <code>\cite</code> | 63, 245 | <code>\curlyvee</code> (\curlyvee) | 99 | <code>\diamondsuit</code> (\diamondsuit) | 108 |
| <code>\cline</code> | 179 | <code>\curlywedge</code> (\curlywedge) | 99 | <code>\digamma</code> (\digamma) | 109 |
| <code>\clubsuit</code> (\clubsuit) | 108 | <code>\currentpdfbookmark</code> .. | 252 | <code>\dim</code> (\dim) | 101 |
| <code>\Colonapprox</code> (\Colonapprox) | 105 | <code>\curvearrowleft</code> (\curvearrowleft) .. | 106 | <code>\displaylimits</code> | 130 |
| <code>\colonapprox</code> (\colonapprox) | 105 | <code>\curvearrowright</code> (\curvearrowright) .. | 106 | <code>displaymath</code> 환경 .. | 97, 141 |
| <code>\Colondash</code> (\Colondash) | 105 | | | <code>\div</code> (\div) | 99 |
| <code>\colondash</code> (\colondash) | 105 | D | | <code>\divideontimes</code> (\divideontimes) | 99 |
| <code>\Coloneq</code> (\Coloneq) | 105 | <code>\d</code> | 38 | <code>\DJ</code> (\DJ) | 38 |
| <code>\coloneq</code> (\coloneq) | 105 | <code>\dag</code> (\dag) | 47 | <code>\dj</code> (\dj) | 38 |
| <code>\Colonsim</code> (\Colonsim) | 105 | <code>\dagger</code> (\dagger) | 99 | <code>\dlar</code> | 156 |
| <code>\colonsim</code> (\colonsim) | 105 | <code>\daleth</code> (\daleth) | 108 | <code>document</code> 환경 | 27 |
| <code>\complement</code> (\complement) | 108 | <code>\dar</code> | 156 | | |

| | | |
|--|--|---|
| <code>\documentclass</code> 23 | <code>\eqslantgtr</code> (\gtrless) 103 | <code>\gamma</code> (γ) 109 |
| <code>\dot</code> ($\dot{}$) 109 | <code>\eqslantless</code> (\lessgtr) 103 | <code>gather</code> 환경 143 |
| <code>\dotemph</code> 70 | <code>equation</code> 환경 97, 141 | <code>gather*</code> 환경 143 |
| <code>\doteq</code> (\doteq) 103 | <code>equation*</code> 환경 141 | <code>\gcd</code> (\gcd) 101 |
| <code>\doteqdot</code> (\doteqdot) 103 | <code>\equiv</code> (\equiv) 103 | <code>\ge</code> (\geq) 103 |
| <code>\dotplus</code> ($\dot{+}$) 99 | <code>\eta</code> (η) 109 | <code>\genfrac</code> 128 |
| <code>\dots</code> (...) 41, 47, 107 | <code>\eth</code> (\eth) 108 | <code>\geq</code> (\geq) 103 |
| <code>\dotsb</code> (...) 107 | <code>\exists</code> (\exists) 108 | <code>\geqq</code> (\geqq) 103 |
| <code>\dotsc</code> (...) 107 | <code>\exp</code> (\exp) 101 | <code>\geqslant</code> (\gtrsim) 103 |
| <code>\dotsi</code> (...) 107 | <code>\extracolsep</code> 180 | <code>\gets</code> (\leftarrow) 106 |
| <code>\dotsm</code> (...) 107 | | <code>\gg</code> (\gg) 103 |
| <code>\dotso</code> (...) 107 | F | <code>\ggg</code> (\ggg) 103 |
| <code>\doublebarwedge</code> ($\overline{\wedge}$) 99 | <code>\fallingdotseq</code> (\fallingdotseq) 103 | <code>\gggtr</code> (\gggtr) 103 |
| <code>\doublecap</code> (\cap) 99 | <code>figure</code> 환경 193 | <code>\gimel</code> (\gimel) 108 |
| <code>\doublecup</code> (\cup) 99 | <code>figure*</code> 환경 198 | <code>\gnapprox</code> (\gtrapprox) 103 |
| <code>\Downarrow</code> (\Downarrow) 106 | <code>\Finv</code> (\Finv) 108 | <code>\gneq</code> (\gtr) 103 |
| <code>\downarrow</code> (\downarrow) 106 | <code>flalign</code> 환경 146 | <code>\gneqq</code> (\gtrneq) 103 |
| <code>\downdownarrows</code> (\downdownarrows) 106 | <code>flalign*</code> 환경 146 | <code>\gnsim</code> (\gtrsim) 103 |
| <code>\downharpoonleft</code> (\downharpoonleft) 106 | <code>\flat</code> (\flat) 108 | <code>\graphicspath</code> 190 |
| <code>\downharpoonright</code> (\downharpoonright) 106 | <code>\floatname</code> 204 | <code>\grave</code> ($\grave{}$) 109 |
| <code>\drar</code> 156 | <code>FlushLeft</code> 환경 69 | <code>\gtrapprox</code> (\gtrapprox) 103 |
| <code>drcases</code> 환경 127 | <code>flushleft</code> 환경 68 | <code>\gtrdot</code> (\gtrdot) 99 |
| <code>drcases*</code> 환경 127 | <code>FlushRight</code> 환경 69 | <code>\gtreqless</code> (\gtrlessgtr) 103 |
| | <code>flushright</code> 환경 68 | <code>\gtreqqless</code> (\gtrlessgtr) 103 |
| E | <code>\fnsymbol</code> 272 | <code>\gtrless</code> (\gtrless) 103 |
| <code>\ell</code> (ℓ) 108 | <code>\footnote</code> 61 | <code>\gtrsim</code> (\gtrsim) 103 |
| <code>\em</code> 70 | <code>\footnotesize</code> 73 | <code>\guillemetleft</code> (\llcorner) 47 |
| <code>\emph</code> 70 | <code>footnotesize</code> 환경 73 | <code>\guillemetright</code> (\ggcorner) 47 |
| <code>\emptyset</code> (\emptyset) 108 | <code>\forall</code> (\forall) 108 | <code>\guilsinglleft</code> (\lrcorner) 47 |
| <code>\end</code> 22 | <code>\frac</code> 114 | <code>\guilsinglright</code> (\rccorner) 47 |
| <code>\endgroup</code> 96 | <code>\frenchspacing</code> 42 | <code>\gvertneqq</code> (\gtrneqq) 103 |
| <code>\endinput</code> 284 | <code>\frontmatter</code> 54 | |
| <code>enumerate</code> 환경 65 | <code>\frown</code> (\frown) 103 | H |
| <code>\epsilon</code> (ϵ) 109 | <code>\Fullref</code> 222 | <code>\H</code> 38 |
| <code>\eqcirc</code> (\circ) 103 | <code>\fullref</code> 214 | <code>\hat</code> 30, 109 |
| <code>\Eqcolon</code> (\equiv) 105 | <code>\Fullref*</code> 248 | <code>\hbar</code> (\hbar) 108 |
| <code>\eqcolon</code> (\equiv) 105 | <code>\fullref*</code> 248 | <code>\hdots</code> (...) 107 |
| <code>eqnarray</code> 환경 144 | | <code>\hdotsfor</code> 125 |
| <code>eqnarray*</code> 환경 144 | G | <code>\heartsuit</code> (\heartsuit) 108 |
| <code>\eqref</code> 212 | <code>\Game</code> (\Game) 108 | <code>\hellipsis</code> (\hellipsis) 43 |
| <code>\eqsim</code> (\sim) 103 | <code>\Gamma</code> (Γ) 109 | <code>\hline</code> 179 |

| | | |
|--|---|--|
| <code>\hom</code> (hom) ······ 101 | <code>\iota</code> (ι) ······ 109 | <code>\Leftarrow</code> (\Leftarrow) ······ 106 |
| <code>\hookleftarrow</code> (\hookleftarrow) ··· 106 | <code>\it</code> ······ 71, 138 | <code>\leftarrow</code> (\leftarrow) ······ 106 |
| <code>\hookrightarrow</code> (\hookrightarrow) ··· 106 | <code>\item</code> ······ 64, 65 | <code>\leftarrowtail</code> (\leftarrowtail) ··· 106 |
| <code>\href</code> ······ 249 | <code>itemize</code> 환경 ······ 65 | <code>\leftharpoondown</code> (\leftharpoondown) · 106 |
| <code>\hslash</code> (\hslash) ······ 108 | <code>\itshape</code> ······ 72 | <code>\leftharpoonup</code> (\leftharpoonup) ··· 106 |
| <code>\Huge</code> ······ 73 | J | <code>\leftleftarrows</code> (\longleftrightarrow) ··· 106 |
| <code>\huge</code> ······ 73 | <code>\j</code> (\j) ······ 38 | <code>\Leftrightarrow</code> (\Leftrightarrow) ··· 106 |
| <code>Huge</code> 환경 ······ 73 | <code>\jmath</code> (\jmath) ······ 108 | <code>\leftrightharrows</code> (\leftrightharrows) · 106 |
| <code>huge</code> 환경 ······ 73 | <code>\Join</code> (\Join) ······ 103 | <code>\leftrightharpoons</code> (\leftrightharpoons) |
| <code>\Hwithstroke</code> (\mathbb{H}) ······ 38 | <code>\justifying</code> ······ 69 | 106 |
| <code>\hwithstroke</code> (\mathfrak{h}) ······ 38 | K | <code>\leftrightsquigarrow</code> (\leftrightsquigarrow) |
| <code>\hyperlink</code> ······ 249 | <code>\k</code> ······ 38 | 106 |
| <code>\hyperref</code> ······ 248 | <code>\kappa</code> (κ) ······ 109 | <code>\leftthreetimes</code> (\leftthreetimes) ····· 99 |
| <code>\hypersetup</code> ······ 253 | <code>\ker</code> (ker) ······ 101 | <code>\leq</code> (\leq) ······ 103 |
| <code>\hypertarget</code> ······ 249 | L | <code>\leqq</code> (\leqq) ······ 103 |
| I | <code>\l</code> (표 환경) ······ 168 | <code>\leqslant</code> (\leqslant) ······ 103 |
| <code>\i</code> (i) ······ 38 | <code>\L</code> (\mathbb{L}) ······ 38 | <code>\lessapprox</code> (\lessapprox) ······ 103 |
| <code>\idotsint</code> ($\int \dots \int$) ····· 100 | <code>\l</code> (\mathfrak{l}) ······ 38 | <code>\lessdot</code> (\lessdot) ······ 99 |
| <code>\iff</code> (\iff) ······ 106 | <code>\label</code> ······ 30, 207 | <code>\lesseqgtr</code> (\lesseqgtr) ······ 103 |
| <code>\iiint</code> (\iiint) ······ 100 | <code>\labelformat</code> ······ 210 | <code>\lesseqqgtr</code> (\lesseqqgtr) ······ 103 |
| <code>\iint</code> (\iint) ······ 100 | <code>\Lambda</code> (Λ) ······ 109 | <code>\lessgtr</code> (\lessgtr) ······ 103 |
| <code>\IJ</code> ($\mathbb{I}\mathbb{J}$) ······ 38 | <code>\lambda</code> (λ) ······ 109 | <code>\lesssim</code> (\lesssim) ······ 103 |
| <code>\ij</code> (\mathfrak{ij}) ······ 38 | <code>\land</code> (\wedge) ······ 99 | <code>\let</code> ······ 263 |
| <code>\Im</code> (\Im) ······ 108 | <code>\langle</code> (\langle) ······ 107 | <code>\lfloor</code> (\lfloor) ······ 107 |
| <code>\imath</code> (\imath) ······ 108 | <code>\lar</code> ······ 156 | <code>\lg</code> (\lg) ······ 101 |
| <code>\impliedby</code> (\impliedby) ····· 106 | <code>\LARGE</code> ······ 73 | <code>\lgroup</code> ($\{$) ······ 118 |
| <code>\implies</code> (\implies) ······ 106 | <code>\Large</code> ······ 73 | <code>\lhd</code> (\lhd) ······ 99 |
| <code>\in</code> (\in) ······ 103 | <code>\large</code> ······ 73 | <code>\lim</code> (lim) ······ 101 |
| <code>\include</code> ······ 285 | <code>LARGE</code> 환경 ······ 73 | <code>\liminf</code> (lim inf) ······ 101 |
| <code>\includegraphics</code> ······ 188 | <code>Large</code> 환경 ······ 73 | <code>\limits</code> ······ 130 |
| <code>\includeonly</code> ······ 285 | <code>large</code> 환경 ······ 73 | <code>\limsup</code> (lim sup) ······ 101 |
| <code>\indexspace</code> ······ 64 | <code>\LaTeX</code> ······ 6 | <code>\listof</code> ······ 204 |
| <code>\inf</code> (inf) ······ 101 | <code>\LaTeXe</code> ······ 6 | <code>\listoffigures</code> ······ 57 |
| <code>\infty</code> (∞) ······ 108 | <code>\lceil</code> (\lceil) ······ 107 | <code>\listoftables</code> ······ 57 |
| <code>\injlim</code> (inj lim) ······ 101 | <code>\ldots</code> (\dots) ······ 47, 107 | <code>\ll</code> (\ll) ······ 103 |
| <code>\input</code> ······ 283 | <code>\le</code> (\leq) ······ 103 | <code>\llcorner</code> (\llcorner) ······ 107 |
| <code>\int</code> (\int) ······ 100 | <code>\leadsto</code> (\leadsto) ······ 106 | <code>\Lleftarrow</code> (\Lleftarrow) ······ 106 |
| <code>\intercal</code> (\intercal) ······ 99 | <code>\left</code> ······ 118 | <code>\lll</code> (\lll) ······ 103 |
| <code>\intertext</code> ······ 146 | | <code>\llless</code> (\llless) ······ 103 |
| | | <code>\lmoustache</code> (\int) ······ 118 |

| | | |
|--|---|--|
| <code>\ln</code> (ln) 101 | <code>\mathring</code> (\grave{a}) 109 | <code>\newenvironment*</code> 264 |
| <code>\lnapprox</code> (\lesssim) 103 | <code>\mathscr</code> 136 | <code>\newfloat</code> 203 |
| <code>\lneq</code> (\lesseqgtr) 103 | <code>\mathsection</code> (§) 108 | <code>\newlength</code> 277 |
| <code>\lneqq</code> (\lesseqgtr) 103 | <code>\mathsterling</code> (£) 108 | <code>\newtheorem</code> 81 |
| <code>\lnot</code> (\neg) 108 | <code>\mathstrut</code> 132 | <code>\newtheorem*</code> 81 |
| <code>\lnsim</code> (\lesssim) 103 | matrix 환경 123 | <code>\newtheoremstyle</code> 87 |
| <code>\log</code> (log) 101 | matrix* 환경 124 | <code>\newtie</code> 48 |
| <code>\Longleftarrow</code> (\Longleftarrow) ··· 106 | <code>\max</code> (max) 101 | <code>\nexists</code> (\nexists) 108 |
| <code>\longleftarrow</code> (\longleftarrow) ··· 106 | <code>\maxof</code> 282 | <code>\NG</code> (I) 38 |
| <code>\Longleftrightharrow</code> (\Longleftrightarrow)
106 | <code>\mdseries</code> 72 | <code>\ng</code> (η) 38 |
| <code>\longleftrightharrow</code> (\longleftrightarrow)
106 | <code>\measuredangle</code> (\sphericalangle) 108 | <code>\ngeq</code> (\nlessgtr) 103 |
| <code>\longmapsto</code> (\longmapsto) 106 | <code>\mho</code> (Ω) 108 | <code>\ngeqq</code> (\nlessgtr) 103 |
| <code>\Longrightarrow</code> (\Longrightarrow) ··· 106 | <code>\mid</code> (\mid) 103 | <code>\ngeqslant</code> (\nlessgtr) 103 |
| <code>\longrightarrow</code> (\longrightarrow) ··· 106 | <code>\middle</code> 119 | <code>\ngtr</code> (\ngtr) 103 |
| <code>\looparrowleft</code> (\looparrowleft) ··· 106 | <code>\min</code> (min) 101 | <code>\ni</code> (\ni) 103 |
| <code>\looparrowright</code> (\looparrowright) ··· 106 | <code>\minof</code> 282 | <code>\nLeftarrow</code> (\nLeftarrow) 106 |
| <code>\lor</code> (\vee) 99 | <code>\mit</code> 138 | <code>\nLeftarrow</code> (\nLeftarrow) 106 |
| <code>\lowercase</code> 39 | <code>\mod</code> 102 | <code>\nLeftarrow</code> (\nLeftarrow) ··· 106 |
| <code>\lozenge</code> (\lozenge) 108 | <code>\models</code> (\models) 103 | <code>\nleq</code> (\nlessgtr) 103 |
| <code>\lrcorner</code> (\lrcorner) 107 | <code>\mp</code> (\mp) 99 | <code>\nleqq</code> (\nlessgtr) 103 |
| <code>\Lsh</code> (\Lsh) 106 | <code>\mu</code> (μ) 109 | <code>\nleqslant</code> (\nlessgtr) 103 |
| <code>\ltimes</code> (\ltimes) 99 | <code>\multicolumn</code> 175 | <code>\nless</code> (\nlessgtr) 103 |
| <code>\lVert</code> (\lVert) 107 | <code>\multimap</code> (\multimap) 103 | <code>\nmid</code> (\nmid) 103 |
| <code>\lvert</code> (\lvert) 107 | <code>\multirow</code> 177 | <code>\NoCaseChange</code> 39 |
| <code>\lvertneqq</code> (\lvertneqq) 103 | multiline 환경 142 | <code>\nocite</code> 246 |
| | multiline* 환경 142 | <code>\nolimits</code> 130 |
| M | N | <code>\nolinkurl</code> 250 |
| <code>m</code> (표 환경) 170 | <code>\nabla</code> (∇) 108 | <code>\nonfrenchspacing</code> 42 |
| <code>\mainmatter</code> 54 | <code>\natural</code> (\natural) 108 | <code>\normalfont</code> 72 |
| <code>\MakeLowercase</code> 38 | <code>\ncong</code> (\ncong) 103 | <code>\normalsize</code> 73 |
| <code>\maketitle</code> 55 | <code>\ne</code> (\neq) 103 | normalsize 환경 73 |
| <code>\MakeUppercase</code> 38 | <code>\nearrow</code> (\nearrow) 106 | <code>\not</code> 104 |
| <code>\maltese</code> (\maltese) 47 | <code>\neg</code> (\neg) 108 | <code>\notag</code> 147 |
| <code>\mapsto</code> (\mapsto) 106 | <code>\neq</code> (\neq) 103 | <code>\notin</code> (\notin) 103 |
| <code>\marginpar</code> 61 | <code>\newcommand</code> 260 | <code>\nparallel</code> (\nparallel) 103 |
| math 환경 95 | <code>\newcommand*</code> 260 | <code>\nprec</code> (\nprec) 103 |
| <code>\mathdollar</code> (\$) 108 | <code>\NewCommandCopy</code> 264 | <code>\npreceq</code> (\npreceq) 103 |
| <code>\mathellipsis</code> (...) 107 | <code>\newcounter</code> 269 | <code>\nrightarrow</code> (\nrightarrow) 106 |
| <code>\mathparagraph</code> (\mathparagraph) 108 | <code>\NewDocumentCommand</code> ··· 264 | <code>\nrightarrow</code> (\nrightarrow) 106 |
| | <code>\newenvironment</code> 264 | <code>\nshortmid</code> (\nshortmid) 103 |

`\nshortparallel` (\parallel) 103
`\nsim` (\approx) 103
`\nsubseteq` ($\not\subseteq$) 103
`\nsubseteqq` ($\not\subseteqq$) 103
`\nsucc` (\succ) 103
`\nsucceq` (\succeq) 103
`\nsupseteq` (\supseteq) 103
`\nsupseteqq` (\supseteqq) 103
`\ntriangleleft` (\triangleleft) 103
`\ntrianglelefteq` (\trianglelefteq) 103
`\ntriangleright` (\triangleright) 103
`\ntrianglerighteq` (\trianglerighteq) 103
`\nu` (ν) 109
`\nVDash` (\nVdash) 103
`\nVdash` (\nVdash) 103
`\nvDash` (\nvDash) 103
`\nvdash` (\nvdash) 103
`\narrow` (\nearrow) 106

O

`\O` (\mathcal{O}) 38
`\o` (\mathcal{o}) 38
`\odot` (\odot) 99
`\OE` (\mathcal{E}) 38
`\oe` (\mathcal{e}) 38
`\oint` (\oint) 100
`\oldstylenums` 74
`\Omega` (Ω) 109
`\omega` (ω) 109
`\ominus` (\ominus) 99
`\operatorname` 101
`\operatorname*` 101
`\oplus` (\oplus) 99
`\ordinarycolon` ($:$) 105
`\oslash` (\oslash) 99
`\otimes` (\otimes) 99
`\overline` 115
`\overset` 134
`\overunderset` 134
`\owns` (\ni) 103

P

`p` (표 환경) 168
`\P` (\mathbb{P}) 47
`\pageref` 207
`\pageref*` 248
`\par` 28
`\paragraph` 58
`\paragraph*` 58
`\parallel` (\parallel) 103
`\part` 58
`\part*` 58
`\partial` (∂) 108
`\perp` (\perp) 103
`\Phi` (Φ) 109
`\phi` (ϕ) 109
`\Pi` (Π) 109
`\pi` (π) 109
`picture` 환경 186
`\pitchfork` (\pitchfork) 103
`\pm` (\pm) 99
`pmatrix` 환경 123
`\pmb` 137
`\pmod` 102
`\pod` 102
`\pounds` (\pounds) 47
`\Pr` (\Pr) 101
`\prec` (\prec) 103
`\precapprox` (\precapprox) 103
`\preccurlyeq` (\preccurlyeq) 103
`\preceq` (\preceq) 103
`\precnapprox` (\precnapprox) 103
`\precneqq` (\precneqq) 103
`\precnsim` (\precnsim) 103
`\precsim` (\precsim) 103
`\prescript` 132
`\prime` (\prime) 108
`\printbibliography` 244
`\prod` (\prod) 100
`\projlim` ($\operatorname{projlim}$) 101
`proof` 환경 83
`\propto` (\propto) 103

`\providecommand` 262
`\providecommand*` 262
`\Psi` (Ψ) 109
`\psi` (ψ) 109
`psmallmatrix` 환경 126
`\put` 186

Q

`\qed` 83
`\qedhere` 84
`\qqquad` 117
`\quad` 117
`quotation` 환경 68
`quote` 환경 68
`\quotedblbase` („) 47

R

`r` (표 환경) 168
`\r` 38
`\RaggedLeft` 69
`\raggedleft` 69
`\RaggedRight` 69
`\raggedright` 69
`\rangle` (\rangle) 107
`\rar` 156
`\ratio` 282
`rcases` 환경 127
`rcases*` 환경 127
`\rceil` (\lceil) 107
`\Re` (\Re) 108
`\real` 282
`\Ref` 211
`\ref` 30, 207, 227
`\Ref*` 248
`\ref*` 248
`\refstepcounter` 270
`\relax` 21
`\renewcommand` 262
`\renewcommand*` 262
`\renewenvironment` 265
`\renewenvironment*` 265

| | | | | | |
|--|---------|--|---------|--|-----|
| <code>\restriction</code> (\restriction) | 106 | <code>\setlength</code> | 277 | <code>\sqsubseteq</code> (\sqsubseteq) | 103 |
| <code>\rfloor</code> (\rfloor) | 107 | <code>\setminus</code> | 99 | <code>\sqsupset</code> (\sqsupset) | 103 |
| <code>\rgroup</code> (\rangle) | 118 | <code>\sf</code> | 71 | <code>\sqsupseteq</code> (\sqsupseteq) | 103 |
| <code>\rhd</code> (\rhd) | 99 | <code>\sffamily</code> | 72 | <code>\square</code> (\square) | 108 |
| <code>\rho</code> (ρ) | 109 | <code>\sharp</code> (\sharp) | 108 | <code>\SS</code> ($\text{\text{SS}}$) | 38 |
| <code>\right</code> | 118 | <code>\shortintertext</code> | 146 | <code>\ss</code> ($\text{\text{ss}}$) | 38 |
| <code>\Rrightarrow</code> (\Rightarrow) | 106 | <code>\shortmid</code> (\mid) | 103 | <code>\stackbin</code> | 134 |
| <code>\rightarrow</code> (\rightarrow) | 106 | <code>\shortparallel</code> (\parallel) | 103 | <code>\stackrel{\cdot}{\sim}</code> | 133 |
| <code>\rightarrowtail</code> (\rightarrowtail) | 106 | <code>\shoveleft</code> | 142 | <code>\star</code> (\star) | 99 |
| <code>\rightharpoondown</code> (\rightharpoondown) | 106 | <code>\shoveright</code> | 142 | <code>\stepcounter</code> | 270 |
| <code>\rightharpoonup</code> (\rightharpoonup) | 106 | <code>\sideset</code> | 131 | subarray 환경 | 130 |
| <code>\rightleftarrows</code> (\rightleftarrows) | 106 | <code>\Sigma</code> (Σ) | 109 | subcaptionblock 환경 | 202 |
| <code>\rightleftharpoons</code> (\rightleftharpoons) | 106 | <code>\sigma</code> (σ) | 109 | <code>\subcaptionbox</code> | 202 |
| <code>\rightthreetimes</code> (\rightthreetimes) | 99 | <code>\sim</code> (\sim) | 43, 103 | <code>\subcaptionbox*</code> | 202 |
| <code>\risingdotseq</code> (\risingdotseq) | 103 | <code>\Simcolon</code> ($\sim::$) | 105 | subcaptiongroup 환경 | 202 |
| <code>\rm</code> | 71, 138 | <code>\simcolon</code> ($\sim:$) | 105 | subequations 환경 | 148 |
| <code>\rmfamily</code> | 72 | <code>\simeq</code> (\simeq) | 103 | <code>\subitem</code> | 64 |
| <code>\rmoustache</code> (\rangle) | 118 | <code>\sin</code> (\sin) | 101 | <code>\subparagraph</code> | 58 |
| <code>\Roman</code> | 272 | <code>\sinh</code> (\sinh) | 101 | <code>\subparagraph*</code> | 58 |
| <code>\roman</code> | 272 | <code>\sl</code> | 71 | <code>\subpdfbookmark</code> | 252 |
| <code>\Rrightarrow</code> (\Rightarrow) | 106 | <code>\slshape</code> | 72 | <code>\subsection</code> | 58 |
| <code>\Rsh</code> (\Rsh) | 106 | <code>\small</code> | 73 | <code>\subsection*</code> | 58 |
| <code>\rtimes</code> (\rtimes) | 99 | small 환경 | 73 | <code>\Subset</code> (\Subset) | 103 |
| <code>\rVert</code> (\rVert) | 107 | <code>\smallfrown</code> (\smallfrown) | 103 | <code>\subset</code> (\subset) | 103 |
| <code>\rvert</code> (\rvert) | 107 | <code>\smallint</code> (\int) | 100 | <code>\subseteq</code> (\subseteq) | 103 |
| S | | smallmatrix 환경 | 125 | <code>\subseteqeq</code> (\subseteqeq) | 103 |
| <code>\S</code> (\S) | 47 | smallmatrix* 환경 | 126 | <code>\subsetneq</code> (\subsetneq) | 103 |
| <code>\sb</code> | 116 | <code>\smallsetminus</code> (\smallsetminus) | 99 | <code>\subsetneqq</code> (\subsetneqq) | 103 |
| <code>\sc</code> | 71 | <code>\smallsmile</code> (\smallsmile) | 103 | <code>\substack</code> | 129 |
| <code>\scriptsize</code> | 73 | <code>\smile</code> (\smile) | 103 | <code>\subsubitem</code> | 64 |
| scriptsize 환경 | 73 | <code>\sp</code> | 116 | <code>\subsubsection</code> | 58 |
| <code>\scshape</code> | 72 | <code>\space</code> | 28 | <code>\subsubsection*</code> | 58 |
| <code>\searrow</code> (\searrow) | 106 | <code>\spadesuit</code> (\spadesuit) | 108 | <code>\succ</code> (\succ) | 103 |
| <code>\sec</code> (\sec) | 101 | <code>\sphericalangle</code> (\sphericalangle) | 108 | <code>\succapprox</code> (\succapprox) | 103 |
| <code>\section</code> | 58 | split 환경 | 143 | <code>\succcurlyeq</code> (\succcurlyeq) | 103 |
| <code>\section*</code> | 58 | <code>\splitdffrac</code> | 129 | <code>\succeq</code> (\succeq) | 103 |
| <code>\setcounter</code> | 270 | <code>\splitfrac</code> | 129 | <code>\succapprox</code> (\succapprox) | 103 |
| | | <code>\sqcap</code> (\sqcap) | 99 | <code>\succneqq</code> (\succneqq) | 103 |
| | | <code>\sqcup</code> (\sqcup) | 99 | <code>\succnsim</code> (\succnsim) | 103 |
| | | <code>\sqrt</code> | 115 | <code>\succsim</code> (\succsim) | 103 |
| | | <code>\sqsubset</code> (\sqsubset) | 103 | <code>\sum</code> (\sum) | 100 |

| | | | | |
|--|------------|--|---|--------|
| <code>\sup</code> (^{sup}) | 101 | 47, 49 | <code>\textendash</code> (—) | 47 |
| <code>\Supset</code> (\supseteq) | 103 | <code>\textbackslash</code> | <code>\textestimated</code> (€) | 49 |
| <code>\supset</code> (\supset) | 103 | <code>\textbaht</code> (฿) | <code>\texteuro</code> (€) | 49 |
| <code>\supseteq</code> (\supseteq) | 103 | <code>\textbar</code> () | <code>\textexclamdown</code> (¡) | 47 |
| <code>\supseteqq</code> (\supseteqq) | 103 | <code>\textbardbl</code> () | <code>\textfiguredash</code> (—) | 47 |
| <code>\supsetneq</code> (\supsetneq) | 103 | <code>\textbf</code> | <code>\textfiveoldstyle</code> (5) | 49 |
| <code>\supsetneqq</code> (\supsetneqq) | 103 | <code>\textbigcircle</code> (○) | <code>\textflorin</code> (f) | 49 |
| <code>\surd</code> (✓) | 108 | <code>\textblank</code> (b) | <code>\textfouroldstyle</code> (4) | 49 |
| <code>\swapnumbers</code> | 85 | <code>\textborn</code> (✶) | <code>\textfractionsolidus</code> (/) | |
| <code>\swarrow</code> (↙) | 106 | <code>\textbraceleft</code> { | 49 | |
| <code>\symbol</code> | 51 | <code>\textbraceright</code> } | <code>\textgravedbl</code> (¨) | 48 |
| T | | | | |
| <code>\t</code> | 38, 48 | <code>\textbrokenbar</code> () | <code>\textgreater</code> (>) | 47 |
| table 환경 | 193 | <code>\textbullet</code> (•) | <code>\textguarani</code> (G) | 49 |
| table* 환경 | 198 | <code>\textcapitalcompwordmark</code> () | <code>\texthorizontalbar</code> (—) | 47 |
| <code>\tableofcontents</code> | 56 | 49 | <code>\textinterrobang</code> (!) | 49 |
| tabular 환경 | 167, 169 | <code>\textcelsius</code> (°C) | <code>\textinterrobangdown</code> (‡) | 49 |
| tabular* 환경 | 180 | <code>\textcent</code> (¢) | 72 | |
| tabularx 환경 | 181 | <code>\textcentoldstyle</code> (¢) | <code>\textit</code> | 72 |
| <code>\tag</code> | 148 | <code>\textcircled</code> | <code>\textlangle</code> (⟨) | 49 |
| <code>\tag*</code> | 148 | <code>\textcircledP</code> (Ⓟ) | <code>\textlbrackdbl</code> (⟦) | 49 |
| <code>\tan</code> (tan) | 101 | <code>\textcolonmonetary</code> (¢) | <code>\textleaf</code> (🍃) | 49 |
| <code>\tanh</code> (tanh) | 101 | <code>\textcompwordmark</code> () | <code>\textleftarrow</code> (←) | 49 |
| <code>\tau</code> (τ) | 109 | <code>\textcopyleft</code> (©) | <code>\textless</code> (<) | 47 |
| <code>\tbinom</code> | 128 | <code>\textcopyright</code> (©) | <code>\textlira</code> (₺) | 49 |
| <code>\TeX</code> | 4 | <code>\textcurrency</code> (¤) | <code>\textlnot</code> (¬) | 49 |
| <code>\texorpdfstring</code> | 251 | <code>\textdagger</code> (†) | <code>\textlquill</code> (f) | 49 |
| <code>\text</code> | 117 | <code>\textdaggerdbl</code> (‡) | <code>\textmarried</code> (∞) | 49 |
| <code>\textacutedbl</code> (¨) | 48 | <code>\textdblhyphen</code> (=) | <code>\textmd</code> | 72 |
| <code>\textascendercompwordmark</code> | | <code>\textdblhyphenchar</code> (≡) | <code>\textmho</code> (℧) | 49 |
| () | 49 | <code>\textdegree</code> (°) | <code>\textminus</code> (—) | 40, 49 |
| <code>\textasciicircum</code> (˘) | 48 | <code>\textdied</code> (†) | <code>\textmu</code> (μ) | 49 |
| <code>\textasciibreve</code> (˘) | 48 | <code>\textdiscount</code> (ℳ) | <code>\textmusicalnote</code> (♪) | 49 |
| <code>\textasciicaron</code> (ˇ) | 48 | <code>\textdiv</code> (÷) | <code>\textnaira</code> (₦) | 49 |
| <code>\textasciicircum</code> | 30, 47 | <code>\textdivorced</code> (¶) | <code>\textnineoldstyle</code> (9) | 49 |
| <code>\textasciidieresis</code> (¨) | 48 | <code>\textdollar</code> (\$) | <code>\textnonbreakinghyphen</code> (-) | |
| <code>\textasciigrave</code> (˘) | 48 | <code>\textdollaroldstyle</code> (₵) | 47 | |
| <code>\textasciimacron</code> (˘) | 48 | <code>\textdong</code> (₫) | <code>\textnormal</code> | 72 |
| <code>\textasciitilde</code> | 30, 43, 47 | <code>\textdownarrow</code> (↓) | <code>\textnumero</code> (№) | 49 |
| <code>\textasteriskcentered</code> (*) | | <code>\texteightoldstyle</code> (8) | <code>\textohm</code> (Ω) | 49 |
| | | <code>\textellipsis</code> (...) | <code>\textonehalf</code> (½) | 49 |
| | | <code>\textemdash</code> (—) | <code>\textoneoldstyle</code> (1) | 49 |

| | | |
|--|--|---|
| <code>\textonequarter</code> ($\frac{1}{4}$) ···· 49 | <code>\textsl</code> ··········· 72 | <code>\tikzcdset</code> ········· 157 |
| <code>\textonesuperior</code> (¹) ··· 49 | <code>\textsterling</code> (£) ··· 47, 49 | <code>\tilde</code> ··········· 30, 109 |
| <code>\textopenbullet</code> (◦) ···· 49 | <code>\textsubscript</code> ······· 73 | <code>\times</code> (×) ········· 99 |
| <code>\textordfeminine</code> (^a) 47, 49 | <code>\textsuperscript</code> ······ 73 | <code>\tiny</code> ··········· 73 |
| <code>\textordmasculine</code> (^o) ··· 47, 49 | <code>\textsurd</code> ($\sqrt{}$) ······· 49 | tiny 환경 ········· 73 |
| <code>\textparagraph</code> (¶) ··· 47, 49 | <code>\textthreeoldstyle</code> (3) · 49 | <code>\title</code> ··········· 55 |
| <code>\textperiodcentered</code> (·) 47, 49 | <code>\textthreequarters</code> ($\frac{3}{4}$) · 49 | titlepage 환경 ······ 56 |
| <code>\textpertenthousand</code> (‰) 47, 49 | <code>\textthreequartersemdash</code> (—) ······· 49 | <code>\to</code> (→) ········· 106 |
| <code>\textperthousand</code> (‰) ··· 47, 49 | <code>\textthreesuperior</code> (³) · 49 | <code>\top</code> (T) ········· 108 |
| <code>\textpeso</code> (P) ········· 49 | <code>\texttildelow</code> (˜) ······ 48 | <code>\triangle</code> (Δ) ······· 108 |
| <code>\textpilcrow</code> (¶) ······· 49 | <code>\texttimes</code> (×) ······· 49 | <code>\triangledown</code> (∇) ····· 108 |
| <code>\textpm</code> (±) ········· 49 | <code>\texttrademark</code> (™) ··· 47, 49 | <code>\triangleleft</code> (◁) ······ 99 |
| <code>\textquestiondown</code> (¿) ··· 47 | <code>\texttt</code> ··········· 72 | <code>\trianglelefteq</code> (⊲) ··· 103 |
| <code>\textquotedbl</code> (") ······ 47 | <code>\texttwelveudash</code> (—) ··· 49 | <code>\triangleleq</code> (⊴) ······ 103 |
| <code>\textquotedblleft</code> (“) ··· 47 | <code>\texttwooldstyle</code> (2) ··· 49 | <code>\triangleright</code> (▷) ······ 99 |
| <code>\textquotedblright</code> (”) ··· 47 | <code>\texttwosuperior</code> (²) ··· 49 | <code>\trianglerighteq</code> (⊳) ··· 103 |
| <code>\textquoteleft</code> (‘) ······ 47 | <code>\textunderscore</code> (_) ···· 47 | <code>\tt</code> ··········· 71, 138 |
| <code>\textquoteright</code> (’) ····· 47 | <code>\textup</code> ··········· 72 | <code>\ttfamily</code> ········· 72 |
| <code>\textquotesingle</code> (') ····· 49 | <code>\textuparrow</code> (↑) ······· 49 | <code>\twoheadleftarrow</code> (↤) 106 |
| <code>\textquotestraightbase</code> (,) 49 | <code>\textvisiblespace</code> () ··· 47 | <code>\twoheadrightarrow</code> (↠) 106 |
| <code>\textquotestraightdblbase</code> (,,) ······· 49 | <code>\textwon</code> (₩) ········ 49 | |
| <code>\textrangle</code> (◁) ······· 49 | <code>\textyen</code> (¥) ········ 49 | U |
| <code>\textrbrackdbl</code> (⌋) ····· 49 | <code>\textzerooldstyle</code> (o) ··· 49 | <code>\u</code> ··········· 38 |
| <code>\textrecipe</code> (R) ······· 49 | <code>\tfrac</code> ··········· 128 | <code>\uar</code> ··········· 156 |
| <code>\textreferencemark</code> (※) · 49 | <code>\TH</code> (P) ··········· 38 | <code>\UIfalse</code> ········· 217 |
| <code>\textregistered</code> (®) 47, 49 | <code>\th</code> (p) ··········· 38 | <code>\UITrue</code> ········· 217 |
| <code>\textrightarrow</code> (→) ···· 49 | <code>\thanks</code> ··········· 56 | <code>\ular</code> ··········· 156 |
| <code>\textrm</code> ··········· 72 | <code>\the</code> ··········· 278 | <code>\ulcorner</code> (⌞) ······· 107 |
| <code>\textrquill</code> (}) ······· 49 | thebibliography 환경 · 62 | <code>\unboldmath</code> ······· 138 |
| <code>\textsc</code> ··········· 72 | theindex 환경 ······· 64 | <code>\underline</code> ········· 70, 115 |
| <code>\textsection</code> (§) ···· 47, 49 | <code>\theoremstyle</code> ······· 86 | <code>\underset</code> ········· 134 |
| <code>\textservicemark</code> (SM) ··· 49 | <code>\therefore</code> (∴) ······· 103 | <code>\unlhd</code> (⊲) ········· 99 |
| <code>\textsevenoldstyle</code> (7) · 49 | <code>\Theta</code> (Θ) ········· 109 | <code>\unrhd</code> (⊳) ········· 99 |
| <code>\textsf</code> ··········· 72 | <code>\theta</code> (θ) ········· 109 | <code>\Uparrow</code> (↑) ······· 106 |
| <code>\textsixoldstyle</code> (6) ··· 49 | <code>\thickapprox</code> (≈) ······ 103 | <code>\uparrow</code> (↑) ······· 106 |
| | <code>\thicksim</code> (∼) ······· 103 | <code>\Updownarrow</code> (↕) ····· 106 |
| | <code>\thmname</code> ········· 87 | <code>\updownarrow</code> (↕) ····· 106 |
| | <code>\thmnote</code> ········· 87 | <code>\upharpoonleft</code> (⌈) ····· 106 |
| | <code>\thmnumber</code> ······· 87 | <code>\upharpoonright</code> (⌋) ··· 106 |
| | tikzcd 환경 ······· 151 | <code>\uplus</code> (⊕) ········· 99 |

| | | | |
|--------------|--|--------------|---------------|
| L | | \로 210 | \으로 210 |
| \는 210 | | \를 210 | \은 210 |
| | | | \을 210 |
| 르 | | 오 | \이 210 |
| \라 210 | | \와 210 | \이라 210 |

클래스·패키지 찾아보기

A

alltt 패키지 77
 amscd 패키지 98
 amsfons 패키지 98
 amsmath 패키지 98, 141, 212
 centertags 옵션 150
 intllimits 옵션 131
 leqno 옵션 149
 namelimits 옵션 131
 nointlimits 옵션 131
 nonamelimits 옵션 131
 nosumlimits 옵션 131
 reqno 옵션 149
 sumlimits 옵션 131
 tbtags 옵션 150
 amssymb 패키지 98
 amsthm 패키지 81, 98
 array 패키지 169
 article 클래스 24
 asymptote 패키지 188

B

beamer 클래스 24
 biblatex 패키지 231
 book 클래스 24
 bookmark 패키지 251

C

calc 패키지 282

calrsfs 패키지 137
 cleveref 패키지 219
 capitalize 옵션 225
 compress 옵션 226
 noabbrev 옵션 226
 nosort 옵션 226
 sort 옵션 226
 sort&compress 옵션 226
 curve2e 패키지 186

E

enumerate 패키지 66
 enumitem 패키지 66

F

fancyvrb 패키지 77
 float 패키지 197, 203
 fvextra 패키지 77

G

graphics 패키지 188
 graphicx 패키지 188

H

hyperref 패키지 226, 247
 bookmarks 옵션 255
 bookmarksnumbered 옵션
 255
 bookmarkstype 옵션 255

colorlinks 옵션 254
 frenchlinks 옵션 254
 hidelinks 옵션 254
 hyperfootnotes 옵션 254
 hyperindex 옵션 254
 implicit 옵션 254
 linktoc 옵션 254
 pdfauthor 옵션 255
 pdfkeywords 옵션 255
 pdflang 옵션 256
 pdfsubject 옵션 255
 pdftitle 옵션 255

I

ifthen 패키지 264

K

kotex-varioref 패키지 217

L

listings 패키지 77

M

mathrsfs 패키지 136
 mathtools 패키지 98, 141
 memoir 클래스 24
 minted 패키지 77
 multirow 패키지 177

O

oblivoir 클래스 24

P

pdftricks 패키지 187

pgfplots 패키지 187

pict2e 패키지 186

pst-pdf 패키지 187

pstricks 패키지 187

R

ragged2e 패키지 69

report 클래스 24

S

shortvrb 패키지 77

stackrel 패키지 134

T

tabularx 패키지 181

textcomp 패키지 48

tikz 패키지 186

tikz-cd 패키지 151, 187

V

varioref 패키지 214

nospace 옵션 214

verbatim 패키지 77

W

wrapfig 패키지 198