

SURVEY METHODOLOGY

SURVEY METHODOLOGY

This is the Subtitle

Robert M. Groves

Universitat de les Illes Balears

Floyd J. Fowler, Jr.

University of New Mexico



A JOHN WILEY & SONS, INC., PUBLICATION

Copyright ©2007 by John Wiley & Sons, Inc. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services please contact our Customer Care Department with the U.S. at 877-762-2974, outside the U.S. at 317-572-3993 or fax 317-572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print, however, may not be available in electronic format.

Library of Congress Cataloging-in-Publication Data:

Survey Methodology / Robert M. Groves . . . [et al.].
p. cm.—(Wiley series in survey methodology)
“Wiley-Interscience.”
Includes bibliographical references and index.
ISBN 0-471-48348-6 (pbk.)
1. Surveys—Methodology. 2. Social sciences—Research—Statistical methods. I. Groves, Robert M. II. Series.

HA31.2.S873 2007
001.4'33—dc22 2004044064
Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

To my parents

CONTRIBUTORS

MASAYKI ABE, Fujitsu Laboratories Ltd., Fujitsu Limited, Atsugi, Japan

L. A. AKERS, Center for Solid State Electronics Research, Arizona State University,
Tempe, Arizona

G. H. BERNSTEIN, Department of Electrical and Computer Engineering, University
of Notre Dame, Notre Dame, South Bend, Indiana; formerly of Center for Solid
State Electronics Research, Arizona State University, Tempe, Arizona

CONTENTS IN BRIEF

PART I SUBMICRON SEMICONDUCTOR MANUFACTURE

1 The Submicrometer Silicon MOSFET	3
2 First Edited Book Sample Chapter Title G. Alvarez and R. K. Watts	5
3 Second Edited Book Sample Chapter Title George Smeal, Ph.D., Sally Smith, M.D. and Stanley Kubrick	7
4 Home	13
5 Overview	15
6 Environment Setup	17
7 Basic Syntax	19
8 Variable Type	21
9 Basic Operator	23
10 Decision Making	25
11 Loop	27
12 Numbers	29
13 Strings	31
	vii

14	Lists	33
15	Tuples	35
16	Dictionary	37
17	Date Time	39
18	Functions	41
19	Modules	43
20	Files I/O	45
21	Exceptions	47
22	Classess/Object	49
23	Reg Expression	51
24	CGI Programming	53
25	Databases Access	55
26	Networking	57
27	Sending Email	59
28	Python Multithread Programming	61
29	XML Processing	71
30	GUI Programming	81

CONTENTS

List of Figures	xvii
List of Tables	xix
Foreword	xxi
Preface	xxiii
Acknowledgments	xxv
Acronyms	xxvii
Glossary	xxix
List of Symbols	xxxi
Introduction	xxxiii
<i>Catherine Clark, PhD.</i>	
References	xxxiii

PART I SUBMICRON SEMICONDUCTOR MANUFACTURE

1	The Submicrometer Silicon MOSFET	3
1.1	Here is a normal section	3
		ix

1.1.1	This is the subsection	3
1.2	Tips On Special Section Heads	4
1.3	This Version of Section Head will be sent Contents	4
1.4	This show how to explicitly break lines in Table of Contents	4
1.5	How to get lower case in section head: pH	4
1.6	How to use a macro that has both upper and lower case parts: V_{Txyz}	4
1.7	Equation	4
2	First Edited Book Sample Chapter Title	5
	G. Alvarez and R. K. Watts	
2.1	Here is a normal section	5
3	Second Edited Book Sample Chapter Title	7
	George Smeal, Ph.D., Sally Smith, M.D. and Stanley Kubrick	
3.1	Sample Section	7
3.2	Example, Figure and Tables	8
3.2.1	Side by Side Tables and Figures	8
3.3	Algorithm	9
	Problems	10
	Exercises	10
3.4	Summary	11
4	Home	13
4.1	Sample Section	13
4.2	Example, Figure and Tables	13
4.3	Algorithm	13
4.4	Summary	14
5	Overview	15
5.1	Sample Section	15
5.2	Example, Figure and Tables	15
5.3	Algorithm	15
5.4	Summary	16
6	Environtment Setup	17
6.1	Sample Section	17
6.2	Example, Figure and Tables	17

6.3	Algorithm	17
6.4	Summary	18
7	Basic Syntax	19
7.1	Sample Section	19
7.2	Example, Figure and Tables	19
7.3	Algorithm	19
7.4	Summary	20
8	Variabel Type	21
8.1	Sample Section	21
8.2	Example, Figure and Tables	21
8.3	Algorithm	21
8.4	Summary	22
9	Basic Operator	23
9.1	Sample Section	23
9.2	Example, Figure and Tables	23
9.3	Algorithm	23
9.4	Summary	24
10	Desicion Making	25
10.1	Sample Section	25
10.2	Example, Figure and Tables	25
10.3	Algorithm	25
10.4	Summary	26
11	Loop	27
11.1	Sample Section	27
11.2	Example, Figure and Tables	27
11.3	Algorithm	27
11.4	Summary	28
12	Numbers	29
12.1	Sample Section	29
12.2	Example, Figure and Tables	29
12.3	Algorithm	29
12.4	Summary	30

13	Strings	31
13.1	Sample Section	31
13.2	Example, Figure and Tables	31
13.3	Algorithm	31
13.4	Summary	32
14	Lists	33
14.1	Sample Section	33
14.2	Example, Figure and Tables	33
14.3	Algorithm	33
14.4	Summary	34
15	Tuples	35
15.1	Sample Section	35
15.2	Example, Figure and Tables	35
15.3	Algorithm	35
15.4	Summary	36
16	Dictionary	37
16.1	Sample Section	37
16.2	Example, Figure and Tables	37
16.3	Algorithm	37
16.4	Summary	38
17	Date Time	39
17.1	Sample Section	39
17.2	Example, Figure and Tables	39
17.3	Algorithm	39
17.4	Summary	40
18	Functions	41
18.1	Sample Section	41
18.2	Example, Figure and Tables	41
18.3	Algorithm	41
18.4	Summary	42
19	Modules	43
19.1	Sample Section	43

19.2	Example, Figure and Tables	43
19.3	Algorithm	43
19.4	Summary	44
20	Files I/O	45
20.1	Sample Section	45
20.2	Example, Figure and Tables	45
20.3	Algorithm	45
20.4	Summary	46
21	Exceptions	47
21.1	Sample Section	47
21.2	Example, Figure and Tables	47
21.3	Algorithm	47
21.4	Summary	48
22	Clasess/Object	49
22.1	Sample Section	49
22.2	Example, Figure and Tables	49
22.3	Algorithm	49
22.4	Summary	50
23	Reg Expression	51
23.1	Sample Section	51
23.2	Example, Figure and Tables	51
23.3	Algorithm	51
23.4	Summary	52
24	CGI Programming	53
24.1	Sample Section	53
24.2	Example, Figure and Tables	53
24.3	Algorithm	53
24.4	Summary	54
25	Databases Access	55
25.1	Sample Section	55
25.2	Example, Figure and Tables	55
25.3	Algorithm	55

25.4	Summary	56
26	Networking	57
26.1	Sample Section	57
26.2	Example, Figure and Tables	57
26.3	Algorithm	57
26.4	Summary	58
27	Sending Email	59
27.1	Sample Section	59
27.2	Example, Figure and Tables	59
27.3	Algorithm	59
27.4	Summary	60
28	Python Multithread Programming	61
28.1	Memulai Thread Baru	62
28.2	Modul Threading	63
28.3	Membuat Thread Menggunakan Threading Modul	64
28.4	Sinkronisasi Thread	65
28.5	Multithreaded Antrian Prioritas	67
29	XML Processing	71
29.1	Arsitektur Parsing XML dan API	71
29.2	Parsing XML dan API SAX	73
29.3	Parsing XML dan API DOM	76
29.4	Membangun Parsing Document XML menggunakan Python	78
30	GUI Programming	81
30.1	Tkinter Pemrograman	81
30.2	Tkinter Widget	83
30.3	Manajemen Geometri	88
30.4	Manfaat Tkinter	89
A	Alternate Reference Styles	93
	References	95
	References	97

LIST OF FIGURES

3.1	Short figure caption.	8
3.2	Oscillograph for memory address access operations, showing 500 ps address access time and superimposed signals of address access in 1 kbit memory plane.	8
3.3	This caption will go on the left side of the page. It is the initial caption of two side-by-side captions.	8
3.4	This caption will go on the right side of the page. It is the second of two side-by-side captions.	8

LIST OF TABLES

3.1	Small Table	8
3.2	Effects of the two types of $\alpha\beta \sum_B^A$ scaling proposed by Dennard and co-workers ^{a,b}	8
3.3	Table Caption	9
3.4	Table Caption	9

FOREWORD

This is the foreword to the book.

PREFACE

This is an example preface. This is an example preface. This is an example preface.
This is an example preface.

R. K. WATTS

Durham, North Carolina
September, 2007

ACKNOWLEDGMENTS

From Dr. Jay Young, consultant from Silver Spring, Maryland, I received the initial push to even consider writing this book. Jay was a constant “peer reader” and very welcome advisor during this year-long process.

To all these wonderful people I owe a deep sense of gratitude especially now that this project has been completed.

G. T. S.

ACRONYMS

ACGIH	American Conference of Governmental Industrial Hygienists
AEC	Atomic Energy Commission
OSHA	Occupational Health and Safety Commission
SAMA	Scientific Apparatus Makers Association

GLOSSARY

NormGibbs	Draw a sample from a posterior distribution of data with an unknown mean and variance using Gibbs sampling.
pNull	Test a one sided hypothesis from a numerically specified posterior CDF or from a sample from the posterior
sintegral	A numerical integration using Simpson's rule

SYMBOLS

- A Amplitude
- $\&$ Propositional logic symbol
- a Filter Coefficient

- \mathcal{B} Number of Beats

INTRODUCTION

CATHERINE CLARK, PHD.
Harvard School of Public Health
Boston, MA, USA

The era of modern began in 1958 with the invention of the integrated circuit by J. S. Kilby of Texas Instruments [1]. His first chip is shown in Fig. I. For comparison, Fig. I.2 shows a modern microprocessor chip, [4].
This is the introduction. This is the introduction. This is the introduction. This is the introduction. This is the introduction. This is the introduction.

$$ABC\mathcal{D}\mathcal{E}\mathcal{F}\alpha\beta\Gamma\Delta\sum_{def}^{abc} \tag{I.1}$$

REFERENCES

1. J. S. Kilby, "Invention of the Integrated Circuit," *IEEE Trans. Electron Devices*, **ED-23**, 648 (1976).
2. R. W. Hamming, *Numerical Methods for Scientists and Engineers*, Chapter N-1, McGraw-Hill, New York, 1962.
3. J. Lee, K. Mayaram, and C. Hu, "A Theoretical Study of Gate/Drain Offset in LDD MOSFETs" *IEEE Electron Device Lett.*, **EDL-7**(3). 152 (1986).

PART I

SUBMICRON SEMICONDUCTOR MANUFACTURE

CHAPTER 1

THE SUBMICROMETER SILICON MOSFET

The sheer volume of answers can often stifle insight...The purpose of computing is insight, not numbers.

—Hamming [2]

1.1 Here is a normal section

Here is some text.

1.1.1 This is the subsection

Here is some normal text. Here is some normal text. Here is some normal text. Here is some normal text. Here is some normal text. Here is some normal text. Here is some normal text. Here is some normal text. Here is some normal text. Here is some normal text.

1.1.1.1 This is the subsubsection Here is some text after the subsubsection. Here is some text after the subsubsection. Here is some text after the subsubsection. Here is some text after the subsubsection.

This is the paragraph Here is some normal text. Here is some normal text. Here is some normal text. Here is some normal text. Here is some normal text.

1.2 Tips On Special Section Heads

Here are some things you can do for a special section head.

1.3 Break Long Section heads with double backslash

Here is some normal text. Here is some normal text. Here is some normal text.

1.4 Here is a Section Title

See this section head for information on how to explicitly break lines in table of contents.

1.5 How to get lower case in section head: pH

Here is some normal text. Here is some normal text. Here is some normal text.

1.6 How to use a macro that has both upper and lower case parts:

$$V_{Txyz}$$

See the top of this file where the definition and box were set.

1.7 Equation

For optimal vertical spacing, no blank lines before or after equations

$$\alpha\beta\Gamma\Delta \tag{1.1}$$

as you see here.

CHAPTER 2

FIRST EDITED BOOK SAMPLE CHAPTER TITLE

G. ALVAREZ AND R. K. WATTS

Carnegie Mellon University, Pittsburgh, Pennsylvania

2.1 Here is a normal section

Here is some text.

CHAPTER 3

SECOND EDITED BOOK SAMPLE CHAPTER TITLE

GEORGE SMEAL, PH.D.¹, SALLY SMITH, M.D.² AND STANLEY KUBRICK¹

¹AT&T Bell Laboratories Murray Hill, New Jersey

²Harvard Medical School, Boston, Massachusetts

3.1 Sample Section

Here is some sample text.

3.2 Example, Figure and Tables

EXAMPLE 3.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

illustration here

Figure 3.1 Short figure caption.

Figure 3.2 Oscillograph for memory address access operations, showing 500 ps address access time and superimposed signals of address access in 1 kbit memory plane.

Table 3.1 Small Table			
one	two	three	four
C	D	E	F

Table 3.2 Effects of the two types of $\alpha\beta \sum_B^A$ scaling proposed by Dennard and co-workers^{a,b}

Parameter	κ Scaling	κ, λ Scaling
Dimension	κ^{-1}	λ^{-1}
Voltage	κ^{-1}	κ^{-1}
Currant	κ^{-1}	λ/κ^2
Dopant Concentration	κ	λ^2/κ

^aRefs. 19 and 20.

^b $\kappa, \lambda > 1$.

3.2.1 Side by Side Tables and Figures

Space for figure...

Figure 3.3 This caption will go on the left side of the page. It is the initial caption of two side-by-side captions.

Space for second figure...

Figure 3.4 This caption will go on the right side of the page. It is the second of two side-by-side captions.

The command `\sidebyside{\}{\}` works similarly for tables:

1. This is the first item in the numbered list.
 2. This is the second item in the numbered list. This is the second item in the numbered list. This is the second item in the numbered list.
- This is the first item in the itemized list.
 - This is the first item in the itemized list. This is the first item in the itemized list. This is the first item in the itemized list.

This is the first item in the itemized list.

This is the first item in the itemized list. This is the first item in the itemized list. This is the first item in the itemized list.

PROBLEMS

3.1 For Hooker's data, Problem 1.2, use the Box and Cox and Atkinson procedures to determine a appropriate transformation of PRES in the regression of PRES on TEMP. find $\hat{\lambda}$, $\tilde{\lambda}$, the score test, and the added variable plot for the score. Summarize the results.

3.2 The following data were collected in a study of the effect of dissolved sulfur on the surface tension of liquid copper (Baes and Killogg, 1953).

$x = \text{Weight \% sulfur}$		$Y = \text{Decrease in Surface Tension}$ (dynes/cm), two Replicates	
0.	034	301	316
0.	093	430	422
0.	30	593	586

- a) Find the transformations of X and Y so that in the transformed scale the regression is linear.
- b) Assuming that X is transformed to $\ln(X)$, which choice of Y gives better results, Y or $\ln(Y)$? (Sclove, 1972).
- c) In the case of α_1 ?
- d) In the case of α_2 ?

3.3 Examine the Longley data, Problem 3.3, for applicability of assumptions of the linear model.

3.4 In the case of Γ_1 ?

3.5 In the case of Γ_2 ?

EXERCISES

3.1 For Hooker's data, Exercise 1.2, use the Box and Cox and Atkinson procedures to determine a appropriate transformation of PRES in the regression of PRES on

TEMP. find $\hat{\lambda}$, $\tilde{\lambda}$, the score test, and the added variable plot for the score. Summarize the results.

3.2 The following data were collected in a study of the effect of dissolved sulfur on the surface tension of liquid copper (Baes and Killogg, 1953).

$x = \text{Weight \% sulfur}$	$Y = \text{Decrease in Surface Tension}$	
	(dynes/cm), two Replicates	
0. 034	301	316
0. 093	430	422
0. 30	593	586

- Find the transformations of X and Y so that in the transformed scale the regression is linear.
- Assuming that X is transformed to $\ln(X)$, which choice of Y gives better results, Y or $\ln(Y)$? (Slove, 1972).
- In the case of Δ_1 ?
- In the case of Δ_2 ?

3.3 Examine the Longley data, Problem 3.3, for applicability of assumptions of the linear model.

3.4 In the case of Γ_1 ?

3.5 In the case of Γ_2 ?

3.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 4

HOME

4.1 Sample Section

Here is some sample text.

4.2 Example, Figure and Tables

EXAMPLE 4.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

4.3 Algorithm

This is a sample algorithm.

Survey Methodology, Second Edition.
By Robert M. Groves Copyright © 2017 John Wiley & Sons, Inc.

4.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 5

OVERVIEW

5.1 Sample Section

Here is some sample text.

5.2 Example, Figure and Tables

EXAMPLE 5.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

5.3 Algorithm

This is a sample algorithm.

5.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 6

ENVIRONMENT SETUP

6.1 Sample Section

Here is some sample text.

6.2 Example, Figure and Tables

EXAMPLE 6.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

6.3 Algorithm

This is a sample algorithm.

6.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 7

BASIC SYNTAX

7.1 Sample Section

Here is some sample text.

7.2 Example, Figure and Tables

EXAMPLE 7.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

7.3 Algorithm

This is a sample algorithm.

Survey Methodology, Second Edition.
By Robert M. Groves Copyright © 2017 John Wiley & Sons, Inc.

7.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 8

VARIABLE TYPE

8.1 Sample Section

Here is some sample text.

8.2 Example, Figure and Tables

EXAMPLE 8.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

8.3 Algorithm

This is a sample algorithm.

8.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 9

BASIC OPERATOR

9.1 Sample Section

Here is some sample text.

9.2 Example, Figure and Tables

EXAMPLE 9.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

9.3 Algorithm

This is a sample algorithm.

Survey Methodology, Second Edition.
By Robert M. Groves Copyright © 2017 John Wiley & Sons, Inc.

9.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 10

DESICION MAKING

10.1 Sample Section

Here is some sample text.

10.2 Example, Figure and Tables

EXAMPLE 10.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

10.3 Algorithm

This is a sample algorithm.

Survey Methodology, Second Edition.
By Robert M. Groves Copyright © 2017 John Wiley & Sons, Inc.

10.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 11

LOOP

11.1 Sample Section

Here is some sample text.

11.2 Example, Figure and Tables

EXAMPLE 11.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

11.3 Algorithm

This is a sample algorithm.

Survey Methodology, Second Edition.
By Robert M. Groves Copyright © 2017 John Wiley & Sons, Inc.

11.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 12

NUMBERS

12.1 Sample Section

Here is some sample text.

12.2 Example, Figure and Tables

EXAMPLE 12.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

12.3 Algorithm

This is a sample algorithm.

12.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 13

STRINGS

13.1 Sample Section

Here is some sample text.

13.2 Example, Figure and Tables

EXAMPLE 13.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

13.3 Algorithm

This is a sample algorithm.

Survey Methodology, Second Edition.
By Robert M. Groves Copyright © 2017 John Wiley & Sons, Inc.

13.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 14

LISTS

14.1 Sample Section

Here is some sample text.

14.2 Example, Figure and Tables

EXAMPLE 14.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

14.3 Algorithm

This is a sample algorithm.

Survey Methodology, Second Edition.
By Robert M. Groves Copyright © 2017 John Wiley & Sons, Inc.

14.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 15

TUPLES

15.1 Sample Section

Here is some sample text.

15.2 Example, Figure and Tables

EXAMPLE 15.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

15.3 Algorithm

This is a sample algorithm.

Survey Methodology, Second Edition.
By Robert M. Groves Copyright © 2017 John Wiley & Sons, Inc.

15.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 16

DICTIONARY

16.1 Sample Section

Here is some sample text.

16.2 Example, Figure and Tables

EXAMPLE 16.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

16.3 Algorithm

This is a sample algorithm.

Survey Methodology, Second Edition.
By Robert M. Groves Copyright © 2017 John Wiley & Sons, Inc.

16.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 17

DATE TIME

17.1 Sample Section

Here is some sample text.

17.2 Example, Figure and Tables

EXAMPLE 17.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

17.3 Algorithm

This is a sample algorithm.

Survey Methodology, Second Edition.
By Robert M. Groves Copyright © 2017 John Wiley & Sons, Inc.

17.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 18

FUNCTIONS

18.1 Sample Section

Here is some sample text.

18.2 Example, Figure and Tables

EXAMPLE 18.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

18.3 Algorithm

This is a sample algorithm.

Survey Methodology, Second Edition.
By Robert M. Groves Copyright © 2017 John Wiley & Sons, Inc.

18.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 19

MODULES

19.1 Sample Section

Here is some sample text.

19.2 Example, Figure and Tables

EXAMPLE 19.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

19.3 Algorithm

This is a sample algorithm.

Survey Methodology, Second Edition.
By Robert M. Groves Copyright © 2017 John Wiley & Sons, Inc.

19.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 20

FILES I/O

20.1 Sample Section

Here is some sample text.

20.2 Example, Figure and Tables

EXAMPLE 20.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

20.3 Algorithm

This is a sample algorithm.

Survey Methodology, Second Edition.
By Robert M. Groves Copyright © 2017 John Wiley & Sons, Inc.

20.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 21

EXCEPTIONS

21.1 Sample Section

Here is some sample text.

21.2 Example, Figure and Tables

EXAMPLE 21.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

21.3 Algorithm

This is a sample algorithm.

Survey Methodology, Second Edition.
By Robert M. Groves Copyright © 2017 John Wiley & Sons, Inc.

21.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 22

CLASESS/OBJECT

22.1 Sample Section

Here is some sample text.

22.2 Example, Figure and Tables

EXAMPLE 22.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

22.3 Algorithm

This is a sample algorithm.

Survey Methodology, Second Edition.
By Robert M. Groves Copyright © 2017 John Wiley & Sons, Inc.

22.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 23

REG EXPRESSION

23.1 Sample Section

Here is some sample text.

23.2 Example, Figure and Tables

EXAMPLE 23.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

23.3 Algorithm

This is a sample algorithm.

Survey Methodology, Second Edition.
By Robert M. Groves Copyright © 2017 John Wiley & Sons, Inc.

23.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 24

CGI PROGRAMMING

24.1 Sample Section

Here is some sample text.

24.2 Example, Figure and Tables

EXAMPLE 24.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

24.3 Algorithm

This is a sample algorithm.

Survey Methodology, Second Edition.
By Robert M. Groves Copyright © 2017 John Wiley & Sons, Inc.

24.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 25

DATABASES ACCESS

25.1 Sample Section

Here is some sample text.

25.2 Example, Figure and Tables

EXAMPLE 25.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

25.3 Algorithm

This is a sample algorithm.

Survey Methodology, Second Edition.
By Robert M. Groves Copyright © 2017 John Wiley & Sons, Inc.

25.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 26

NETWORKING

26.1 Sample Section

Here is some sample text.

26.2 Example, Figure and Tables

EXAMPLE 26.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

26.3 Algorithm

This is a sample algorithm.

Survey Methodology, Second Edition.
By Robert M. Groves Copyright © 2017 John Wiley & Sons, Inc.

26.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 27

SENDING EMAIL

27.1 Sample Section

Here is some sample text.

27.2 Example, Figure and Tables

EXAMPLE 27.1 Optional Example Name

Use Black's law [Equation (6.3)] to estimate the reduction in useful product life if a metal line is initially run at 55°C at a maximum line current density.

27.3 Algorithm

This is a sample algorithm.

Survey Methodology, Second Edition.
By Robert M. Groves Copyright © 2017 John Wiley & Sons, Inc.

27.4 Summary

This is a summary of this chapter. Here are some references: [?], [?].

CHAPTER 28

PYTHON MULTITHREAD PROGRAMMING

Menjalankan beberapa *thread* mirip dengan menjalankan beberapa program yang berbeda secara bersamaan, namun dengan manfaat berikut :

- Beberapa *thread* dalam proses berbagi ruang data yang sama dengan benang induk dan karena dapat saling berbagi informasi atau berkomunikasi satu sama lain dengan lebih mudah daripada jika prosesnya terpisah
- *thread* terkadang disebut proses ringan dan tidak membutuhkan banyak memori atas, mereka lebih murah daripada proses.

Sebuah *thread* memiliki permulaan, urutan eksekusi dan sebuah kesimpulan. Ini memiliki pointer perintah yang melacak dari mana dalam konteksnya saat ini berjalan.

- Hal ini dapat dilakukan sebelum *pre-empted (interrupted)*
- Untuk sementara dapat ditunda sementara *thread* lainnya yang sedang berjalan ini disebut unggul.

28.1 Memulai Thread Baru

Untuk melakukan *thread* lain, perlu memanggil metode berikut yang tersedia dimodul *thread* :

```
Thread.start_new_thread (function, args [, kwargs] )
```

Pemanggilan metode ini memungkinkan cara cepat dan tepat untuk membuat *thread* baru di linux dan window.

Pemanggilan metode segera kembali dan anak *thread* dimulai dan fungsi pemanggilan dengan daftar *args* telah berlalu. Saat fungsi kembali ujung *thread* akan berakhir.

Disini, *args* adalah tupel argumen. Gunakan tupel kosong untuk memanggil fungsi tanpa melewati argumen. *Kwargs* adalah kamus opsional argumen kata kunci. Contoh :

```
#!/usr/bin/python
```

```
Import thread
```

```
Import time
```

```
# Define a function for the thread
```

```
Def print_time (threadName, delay):
```

```
    Count = 0
```

```
    While count < 5:
```

```
        Time.sleep(delay)
```

```
        Count += 1
```

```
        Print " %s : %s " % (threadName, time.ctime(time.time()))
```

```
# Create two thread as follows
```

```
try:
```

```
    thread.start_new_thread(print_time, ( "Thread-1 ", 2, ))
```

```
    thread.start_new_thread(print_time, ( "Thread-2 ", 4, ))
```

```
except:
```

```
    print "Error: unable to start thread "
```

```
while 1:
```

```
    pass
```

Bila kode diatas dieksekusi, maka menghasilkan hasil sebagai berikut :

```
Thread-1 : Thu Jan 22 15:42:17 2009
```

```
Thread-1 : Thu Jan 22 15:42:19 2009
```

```
Thread-2 : Thu Jan 22 15:42:19 2009
```

Thread-1 : Thu Jan 22 15:42:21 2009

Thread-2 : Thu Jan 22 15:42:23 2009

Thread-1 : Thu Jan 22 15:42:23 2009

Thread-1 : Thu Jan 22 15:42:23 2009

Thread-1 : Thu Jan 22 15:42:25 2009

Thread-2 : Thu Jan 22 15:42:27 2009

Thread-2 : Thu Jan 22 15:42:31 2009

Thread-2 : Thu Jan 22 15:42:35 2009

Meskipun sangat efektif untuk benang tingkat rendah, namun modul *thread* sangat terbatas dibandingkan dengan modul yang baru.

28.2 Modul Threading

Modul threading yang lebih baru disertakan dengan Python 2.4 memberikan jauh lebih kuat, dukungan tingkat tinggi untuk *thread* dari modul *thread* dibahas pada bagian sebelumnya.

The *threading* modul mengekpos semua metode dari *thread* dan menyediakan beberapa metode tambahan :

- **threading.activeCount()**

Mengembalikan jumlah objek *thread* yang aktif

- **threading.currentThread()**

Mengembalikan jumlah objek *thread* dalam kontrol benang pemanggil

- **threading.enumerate()**

Mengembalikan daftar semua benda *thread* yang sedang aktif

Selain metode, modul *threading* memiliki *thread* kelas yang mengimplementasikan *threading*. Metode yang disediakan oleh *thread* kelas adalah sebagai berikut :

- **run()**

Metode adalah titik masuk untuk *thread*

- **start()**
Metode dimulai *thread* dengan memanggil metode `run`
- **join([time])**
Menunggu benang untuk mengakhiri
- **isAlive()**
Metode memeriksa apakah *thread* masih mengeksekusi
- **getName()**
Metode mengembalikan nama *thread*
- **setName()**
Metode menetapkan nama *thread*

28.3 Membuat Thread Menggunakan Threading Modul

Untuk melaksanakan *thread* baru menggunakan *threading* harus melakukan hal berikut :

Mendefinisikan subclass dari *thread* kelas

Menimpa `__init__` (self [args]) metode untuk menambahkan argumen tambahan

Menimpa `run(self[args])` metode untuk menerapkan apa *thread* harus dilakukan ketika mulai

Setelah membuat baru *thread* subclass, dapat membuahkan sebuah instance dari itu dan kemudian memulai *thread* baru dengan menerapkan `start()`, yang ada gilirannya panggilan `run()` metode.

Contoh :

```
#!/usr/bin/python
```

```
import threading
```

```
import time
```

```
exitFlag = 0
```

```
class myThread (threading.Thread):
    def __init__(self, threadID, name, counter) :
        threading.Thread.__init__(self)
        self.threadID = threadID
        self.name = name
```

```

        self.counter = counter
def run (self) :
    print "Starting " + self.name
    print _time(self.name, self.counter, 5)
    print "Exiting " + self.name

def print _time(threadName, delay, counter):
while counter:
    if exitFlag:
        threadName.exit()
    time.sleep(delay)
    print " %s: %s " % (threadName, time.ctime(time.time()))
counter -= 1

# Create new threads
thread1 = myThread(1, "Thread-1 ", 1)
thread2 = myThread(2, "Thread-2 ", 2)

# Start new threads
thread1.start()
thread2.start()
print "Exiting Main Thread "
```

Ketika kode diatas dijalankan, menghasilkan hasil sebagai berikut:

```

Starting Thread-1
Starting Thread-2
Exiting Main Thread
Thread-1 : Thu Mar 21 09:10:03 2013
Thread-1 : Thu Mar 21 09:10:04 2013
Thread-2 : Thu Mar 21 09:10:04 2013
Thread-1 : Thu Mar 21 09:10:05 2013
Thread-2 : Thu Mar 21 09:10:06 2013
Thread-1 : Thu Mar 21 09:10:07 2013
Exiting Thread-1
Thread-2 : Thu Mar 21 09:10:08 2013
Thread-2 : Thu Mar 21 09:10:10 2013
Thread-2 : Thu Mar 21 09:10:12 2013
Exiting Thread=2
```

28.4 Sinkronisasi Thread

Threading modul disediakan dengan Python termasuk sederhana untuk menerapkan mekanisme bahwa memungkinkan untuk menyinkronkan *thread* penguncian.

Sebuah kunci baru dibuat dengan memanggil *lock()* metode yang mengembalikan kunci baru.

The *acquire (blocking)* metode objek kunci baru digunakan untuk memaksa *thread* untuk menjalankan serempak. Opsional *blocking* parameter memungkinkan untuk mengontrol apakah *thread* menunggu untuk mendapatkan kunci.

Jika *blocking* diatur ke 0, *thread* segera kembali dengan nilai 0 jika kunci tidak dapat diperoleh dan dengan 1 jika kunci dikuasai. Jika pemblokiran diatur ke 1, blok dan menunggu kunci yang akan dirilis.

The *release()* metode objek kunci baru digunakan untuk melepaskan kunci ketika tidak lagi diperlukan.

Contoh:

```
#!/usr/bin/python
```

```
import threading
import time
```

```
class myThread (threading.Thread):
    def _init_(self, threadID, name, counter):
        threading.Thread._init_(self)
        self.threadID = threadID
        self.name = name
        self.counter = counter
    def run(self)
        print "Starting " + self.name
        # Get lock to synchronize threads
        ThreadLock.acquire()
        print _time(self.name, self.counter, 3)
        # Free lock to release next thread
        ThreadLock.release()
```

```
Def print _time(threadName, delay, counter):
    while counter:
        time.sleep(delay)
        print " %s: %s " % (threadName, time.ctime(time.time()))
        counter -= 1
    threadLock = threading.Lock()
    threads = []
```

```
# Create new threads
thread1 = myThread(1, "Thread-1,1 ")
thread2 = myThread(2, "Thread-2,2 ")
```

```
# Start new Threads
thread1.start()
thread2.start()
```

```
# Add threads to thread list
threads.append(thread1)
thread2.append(thread2)

# Wait for all threads to complete
for t in threads:
    t.join()
print "Exiting Main thread "
```

Bila kode diatas dieksekusi, maka menghasilkan sebagai berikut :

```
Starting Thread-1
Starting Thread-2
Thread-1: Thu Mar 21 09:11:28 2013
Thread-1: Thu Mar 21 09:11:29 2013
Thread-1: Thu Mar 21 09:11:30 2013
Thread-2: Thu Mar 21 09:11:32 2013
Thread-2: Thu Mar 21 09:11:34 2013
Thread-2: Thu Mar 21 09:11:36 2013
Exiting Main Thread
```

28.5 Multithreaded Antrian Prioritas

The queue modul memungkinkan untuk membuat objek antrian baru yang dapat menampung jumlah tertentu item. Ada metode berikut untuk mengontrol antrian :

- **get()**
Menghapus dan mengembalikan item dari antrian
- **put()**
Menambahkan item ke antrian
- **qsize()**
Mengembalikan jumlah item yang saat ini dalam antrian
- **empty()**
Mengembalikan benar jika antrian kosong jika tidak, salah
- **full()**

Mengembalikan benar jika antrian penuh jika tidak, salah

Contoh:

```
#!/usr/bin/python
```

```
import Queue
import threading
import time
```

```

exitFlag = 0

class myThread (threading.Thread):
    def _init_(self, threadID, name, q):
        threading.Thread._init_(self)
        self.name = name
        self.q = q
    def run(self):
        print "Starting " + self.name
        process_data(self.name, self.q)
        print "Exiting " + self.name

def process_data(threadName, q):
    while not exitFlag:
        queueLock.acquire()
        if not workQueue.empty():
            data = q.get()
            queueLock.release()
            print " %s processing %s " % (threadName, data)
        else:
            queueLock.release()
            time.sleep(1)

threadList = [ "Thread-1 ", "Thread-2 ", "Thread-3 "]
nameList = [ "One ", "Two ", "Three ", "Four ", "Five "]
queueLock = threading.Lock()
workLock = Queue.Queue(10)
threads = []
threadID = 1

# Create new threads
For tName in threadList:
    thread = myThread(threadID, tName, workQueue)
    thread.start()
    thread.append(thread)
    threadID +=1

# Fill the queue
queueLock.acquire()
for word in nameList:
    workQueue.put(word)
queueLock.release()

# Wait for queue to empty
while not workQueue.empty():
    pass

```

```
# Notify threads its time to exit
exitFlag = 1

# Wait for all threads to complete
For t in threads:
    t.join()
print "Exiting Main Thread "
```

Bila kode diatas dieksekusi, maka menghasilkan hasil sebagai berikut:

```
Starting Thread-1
Starting Thread-2
Starting Thread-3
Thread-1 processing One
Thread-2 processing Two
Thread-3 processing Three
Thread-1 processing Four
Thread-2 processing Five
Exiting Thread-3
Exiting Thread-1
Exiting Thread-2
Exiting Main Thread
```


CHAPTER 29

XML PROCESSING

XML adalah bahasa open source portable yang memungkinkan pemrogram mengembangkan aplikasi yang dapat dibaca oleh aplikasi lain, terlepas dari sistem operasi dan bahasa pengembangnya.

Apa itu XML?

Extensible Markup Language (XML) adalah bahasa markup seperti HTML atau SGML. Ini direkomendasikan oleh World Wide Web Consortium dan tersedia sebagai standar terbuka.

XML sangat berguna untuk mencatat data berukuran kecil dan menengah tanpa memerlukan tulang punggung berbasis SQL.

29.1 Arsitektur Parsing XML dan API

Perpustakaan standar Python menyediakan seperangkat antarmuka minimal tapi berguna untuk bekerja dengan XML.

Dua API yang paling dasar dan umum digunakan untuk data XML adalah antarmuka SAX dan DOM.

API sederhana untuk XML (SAX): mendaftarkan panggilan kemali untuk acara yang diminati dan kemudian membiarkan parser berjalan melalui dokumen. Ini berguna bila dokumen berukuran besar atau memiliki keterbatasan memori, ini memarsing file tidak pernah tersimpan dalam memori.

API Document Objek Model (DOM): ini adalah rekomendasi World Wide Web Consortium dimana keseluruhan file dibaca ke memori dan disimpan dalam bentuk hierarkies (tree-based) untuk mewakili semua fitur dokumen XML.

SAX jelas tidak bisa memproses informasi secepat DOM saat bisa terjadi dengan file besar. Di sisi lain, menggunakan DOM secara eksklusif benar-benar dapat membunuh sumber daya, terutama jika digunakan pada banyak file kecil.

SAX hanya bisa dibaca sementara DOM mengizinkan perubahan pada file XML. Kedua API yang berbeda ini saling melengkapi satu sama lain, tidak ada alasan mengapa tidak dapat menggunakannya untuk proyek besar.

Contoh:

```

<collection shelf="New Arrivals">
  <movie title="Enemy Behind">
    <type>War, Thriller</type>
    <format>DVD</format>
    <year>2003</year>
    <rating>PG</rating>
    <stars>10</stars>
    <description>Talk about a US-Japan war</description>
  </movie>
  <movie title="Transformers">
    <type>Anime, Science Fiction</type>
    <format>DVD</format>
    <year>1989</year>
    <rating>R</rating>
    <stars>8</stars>
    <description>A schientific fiction</description>
  </movie>
  <movie title="Trigun">
    <type>Anime, Action</type>
    <format>DVD</format>
    <episodes>4</episodes>
    <rating>PG</rating>
    <stars>10</stars>
    <description>Vash the Stampede!</description>
  </movie>
  <movie title="Ishtar">
    <type>Comedy</type>

```

```

;format;VHS;/format;
;rating;PG;/rating;
;stars;2;/stars;
;description;Viewable boredom;/description;
;/movie;
;/collection;

```

29.2 Parsing XML dan API SAX

SAX adalah antarmuka standar untuk parsing XML berbasis event. Parsing XML dengan SAX umumnya mengharuskan untuk membuat *ControlHandler* dengan subclassing `xml.sax.controlhandler`.

ControlHandler menangani tag dan atribut tertentu dari XML. Objek *ControlHandler* menyediakan metode untuk menangani berbagai aktivitas parsing. Parsing memanggil metode *ControlHandler* saat memarsing file XML.

Metode *startDocument* dan *endDocument* disebut awal dan akhir setiap elemen. Jika parsing tidak dalam mode namespace, metode *startElement* (tag attribute) dan *endElement* (tag) dipanggil. Jika tidak, metode yang sesuai *startElementNS* dan *endElementNS* dipanggil. Disini, tag adalah tag elemen dan atribut adalah atribut.

Berikut ini metode penting untuk memahami sebelum melanjutkan ke materi berikutnya :

Metode *make_parser*

Metode berikut membuat objek parsing baru dan mengembalikannya. Objek parsing dibuat akan menjadi tipe parsing pertama yang ditemukan sistem.

```
xml.sax.make_parser([parser_list])
```

Berikut adalah detail parameternya :

Parser *_list* : pilihan argumen yang terdiri dari daftar parsing untuk digunakan yang semuanya harus menerapkan metode *make_parse*

Metode *parser*

Metode berikut membuat parsing SAX dan menggunakannya untuk mengurai dokumen

```
xml.sax.parser(xmlfile, contenthandler[, errorhandler])
```

Berikut adalah detail dari parameternya:

- *Xmlfile*

Ini adalah nama file XML yang bisa dibaca.

- *ContentHandler*

Ini harus menjadi objek *ContentHandler*

- *ErrorHandler*

Jika ditentukan, *errorhandler* harus menjadi objek *ErrorHandler* SAX

- Metode *parseString*

Membuat parsing SAX dan mengurai string XML yang ditentukan :

```
xml.sax.parsestring(xmlstring, contenthandler[, errorHandler])
```

Berikut ini adalah detail nama dan parameter :

XMLstring

Nama dari string yang bisa dibaca

ContentHandler

Menjadi objek ContentHandler

ErrorHandler

Menjadi objek ErrorHandler SAX

Contoh :

```
#!/usr/bin/python
```

```
import xml.sax
```

```
class MovieHandler( xml.sax.ContentHandler ):
```

```
    def __init__(self):
        self.CurrentData = ""
        self.type = ""
        self.format = ""
        self.year = ""
        self.rating = ""
        self.stars = ""
        self.description = ""
```

```
# Call when an element starts
```

```
def startElement(self, tag, attributes):
    self.CurrentData = tag
    if tag == "movie":
        print "*****Movie*****"
        title = attributes["title"]
        print "Title:", title
```

```
# Call when an element ends
```

```
def endElement(self, tag):
    if self.CurrentData == "type":
        print "Type:", self.type
```

```

elif self.CurrentData == "format":
    print "Format:", self.format
elif self.CurrentData == "year":
    print "Year:", self.year
elif self.CurrentData == "rating":
    print "Rating:", self.rating
elif self.CurrentData == "stars":
    print "Stars:", self.stars
elif self.CurrentData == "description":
    print "Description:", self.description
self.CurrentData = ""

# Call when a character is read
def characters(self, content):
    if self.CurrentData == "type":
        self.type = content
    elif self.CurrentData == "format":
        self.format = content
    elif self.CurrentData == "year":
        self.year = content
    elif self.CurrentData == "rating":
        self.rating = content
    elif self.CurrentData == "stars":
        self.stars = content
    elif self.CurrentData == "description":
        self.description = content

if ( __name__ == "__main__" ):

    # create an XMLReader
    parser = xml.sax.make_parser()
    # turn off namespaces
    parser.setFeature(xml.sax.handler.feature_namespaces, 0)

    # override the default ContextHandler
    Handler = MovieHandler()
    parser.setContentHandler( Handler )

    parser.parse("movies.xml")

```

Ini akan menghasilkan hasil sebagai berikut:

```
*****Movie*****
```

```
*****Movie*****
```

```
Title: Enemy Behind
```

Type: War, Thriller
 Format: DVD
 Year: 2003
 Rating: PG
 Stars: 10
 Description: Talk about a US-Japan war
 *****Movie*****
 Title: Transformers
 Type: Anime, Science Fiction
 Format: DVD
 Year: 1989
 Rating: R
 Stars: 8
 Description: A schientific fiction
 *****Movie*****
 Title: Trigun
 Type: Anime, Action
 Format: DVD
 Rating: PG
 Stars: 10
 Description: Vash the Stampede!
 *****Movie*****
 Title: Ishtar
 Type: Comedy
 Format: VHS
 Rating: PG
 Stars: 2
 Description: Viewable boredom

29.3 Parsing XML dan API DOM

Document Object Model (DOM) adalah API lintas bahasa dari World Wide Web Consortium (W3C) untuk mengakses dan memodifikasi dokumen XML.

DOM sangat berguna untuk aplikasi akses acak. SAX hanya memungkinkan melihat satu bit dokumen sekaligus. Jika melihat satu elemen SAX, tidak memiliki akses ke yang lain.

Berikut adalah cara termudah untuk memuat dokumen XML dengan cepat dan membuat objek minidom menggunakan modul xml.dom. Objek minidom menyediakan metode parsing sederhana yang dengan cepat memuat pohon DOM dari file XML.

Contoh frase memanggil fungsi parsing (file [,parsing]) dari objek minidokumen untuk mengurai file XML yang ditunjuk oleh file ke objek pohon DOM.

```
#!/usr/bin/python

from xml.dom.minidom import parse
import xml.dom.minidom
```

```
# Open XML document using minidom parser
DOMTree = xml.dom.minidom.parse("movies.xml")
collection = DOMTree.documentElement
if collection.hasAttribute("shelf"):
    print "Root element : %s" % collection.getAttribute("shelf")

# Get all the movies in the collection
movies = collection.getElementsByTagName("movie")

# Print detail of each movie.
for movie in movies:
    print "*****Movie*****"
    if movie.hasAttribute("title"):
        print "Title: %s" % movie.getAttribute("title")

    type = movie.getElementsByTagName('type')[0]
    print "Type: %s" % type.childNodes[0].data
    format = movie.getElementsByTagName('format')[0]
    print "Format: %s" % format.childNodes[0].data
    rating = movie.getElementsByTagName('rating')[0]
    print "Rating: %s" % rating.childNodes[0].data
    description = movie.getElementsByTagName('description')[0]
    print "Description: %s" % description.childNodes[0].data
```

Ini akan menghasilkan hasil sebagai berikut :

```
Root element : New Arrivals
*****Movie*****
Title: Enemy Behind
Type: War, Thriller
Format: DVD
Rating: PG
Description: Talk about a US-Japan war
*****Movie*****
Title: Transformers
Type: Anime, Science Fiction
Format: DVD
Rating: R
Description: A schientific fiction
*****Movie*****
Title: Trigun
Type: Anime, Action
Format: DVD
Rating: PG
Description: Vash the Stampede!
*****Movie*****
Title: Ishtar
Type: Comedy
Format: VHS
Rating: PG
Description: Viewable boredom
```

29.4 Membangun Parsing Document XML menggunakan Python

Python mendukung untuk bekerja dengan berbagai bentuk markup data terstruktur. Selain mengurai `xml.etree.ElementTree` mendukung pembuatan dokumen XML yang terbentuk dengan baik dari objek elemen yang dibangun dalam aplikasi. Kelas elemen digunakan saat sebuah dokumen diurai untuk mengetahui bagaimana menghasilkan bentuk serial dari isinya kemudian dapat ditulis ke sebuah file.

Untuk membuat instance elemen gunakan fungsi elemen constructor dan `SubElement()` pabrik.

Import `xml.etree.ElementTree` as `xml`

```
filename = "/home/abc/Desktop/test_xml.xml "
```

```
root = xml.Element( "Users "
```

```
userelement = xml.Element( "user "
```

```
root.append(userelement)
```

Bila menjalankan ini, akan menghasilkan sebagai berikut :

```
<Users>
  <user>
  <user>
</Users>
```

Tambahkan anak-anak pengguna

```
Uid = xml.SubElement(userelement, "uid ")
Uid.text = "1 "
```

```
FirstName = xml.SubElement(userelement, "FirstName ")
FirstName.text = "testuser "
```

```
LastName = xml.SubElement(userelement, "LastName ")
LastName.text = "testuser "
```

```
Email = xml.SubElement(userelement, "Email ")
Email.text = "mailto:testuser@test.comtestuser@test.com "
```

```
state = xml.SubElement(userelement, "state ")
state.text = "xyz "
```

```
location = xml.SubElement(userelement, "location")
location.text = "abc "
```

```
tree = xml.ElementTree(root)
with open(filename, "w ") as fh:
tree.write(fh)
```

Pertama buat elemen root dengan menggunakan fungsi `ElementTree`. Kemudian membuat elemen pengguna dan menambahkannya ke root. Selanjutnya membuat `SubElement` dengan melewati elemen pengguna (`userelement`) ke `SubElement` beserta namanya seperti `"FirstName "`. Kemudian untuk setiap `SubElement` tetapkan

properti teks untuk memberi nilai. Di akhir, membuat *ElementTree* dan menggunakannya untuk menulis XML ke file.

Jika menjalankan ini akan menjadi sebagai berikut :

```
<?xml version="1.0" encoding="UTF-8" ?>
<Users>
  <user>
    <uid>1</uid>
    <FirstName>testuser</FirstName>
    <LastName>testuser</LastName>
    <Email>mailto:testuser@test.com %3c/Emailtestuser@test.com</Email>
  </user>
  <state>xyz</state>
  <location>abc</location>
</Users>
```

Parsing XML Documen :

```
import xml.etree.ElementTree as ET
tree = ET.parse(Your_XML_file_path)
root = tree.getroot()
```

Disini *getroot()* akan mengembalikan elemen dari dokumen XML

```
<?xml version="1.0" language="SPA" ?>
<Users>
  <user>
    <uid>1</uid>
    <FirstName>testuser</FirstName>
    <LastName>testuser</LastName>
    <Email>testuser@tes.com</Email>
    <state>xyz</state>
    <location>abc</location>
  </user>
</Users>
```


CHAPTER 30

GUI PROGRAMMING

Python menyediakan berbagai pilihan untuk mengembangkan antarmuka pengguna grafis (GUIs). Yang paling tercantum dibawah ini :

- Tkinter
Antarmuka Python ke toolkit Tk GUI dikirimkan dengan Python.
- wxPython
antarmuka Python open-source untuk wxWindows
- Jpython
Port Python untuk java yang memberikan Python script akses tanpa batas ke perpustakaan kelas java pada mesin lokal

30.1 Tkinter Pemrograman

Tkinter adalah perpustakaan GUI standar untuk Python. Python bila dikombinasikan dengan Tkinter menyediakan cara yang mudah dan cepat untuk mem-

buat aplikasi GUI. Tkinter menyediakan antarmuka berorientasi objek yang kuat untuk toolkit Tk GUI.

Membuat aplikasi GUI menggunakan Tkinter adalah tugas yang mudah. Yang diperlukan adalah melakukan langkah-langkah sebagai berikut :

- Mengimpor Tkinter modul
- Buat jendela utama aplikasi GUI
- Tambahkan satu atau lebih dari widget tersebut diatas ke aplikasi GUI
- Masukkan acara loop utama untuk mengambil tindakan terhadap setiap peristiwa dipicu oleh pengguna

Contoh :

```
#!/usr/bin/python

import Tkinter
top = Tkinter.Tk()
# Code to add widgets will go here...
top.mainloop()
```

30.2 Tkinter Widget

Tkinter menyediakan berbagai kontrol seperti tombol, label dan kotak teks yang digunakan dalam aplikasi GUI. Kontrol ini biasanya disebut widget.

Saat ini ada 15 jenis widget di Tkinter. Menyajikan widget serta penjelasan singkat pada tabel berikut ini :

Beberapa atribut umum sebagai ukuran, warna dan font ditentukan. Berikut adalah beberapa atribut standar :

Ukuran

Berbagai panjang, lebar, dan dimensi lain dari widget digambarkan dalam banyak unit yang berbeda seperti :

- Jika menetapkan dimensi ke integer diasumsikan dalam piksel
- Menentukan unit dengan menentukan dimensi untuk string yang berisi sejumlah diikuti oleh :

Tkinter mengungkapkan panjang sebagai integer jumlah piksel.

Berikut ini adalah daftar pilihan panjang umum:

- `borderwidth`
Lebar batas yang memberikan tampilan tiga dimensi untuk widget
- `highlightthickness`
Lebar puncak persegi panjang ketika widget memiliki fokus

- `padX padY`
Ruang tambahan widget dari manajer tata letak luar minimum widget perlu menampilkan isinya di x dan y arah
- `selectborderwidth`
Lebar perbatasan tiga dimensi disekitar dipilih item widget
- `wraplength`
Panjang garis maksimum untuk widget yang melakukan kata membungkus
- `height`
Tinggi diinginkan widget
- `underline`
Indeks karakter untuk menggarisawahi dalam teks widget
- `width`
Lebar diinginkan widget

Warna

Tkinter memiliki warna dengan string. Ada dua cara umum untuk menentukan warna di Tkinter, yaitu :

- Menggunakan string menentukan proporsi merah, hijau dan biru didigit heksadesimal. Misalnya " #ffff " putih, " #000000 " hitam dan " #000fff00 " hijau.
- Menggunakan lokal standar nama warna . warna-warna "white ", "black ", "green " dan "magenta " akan selalu tersedia.

Pilihan warna umum :

- `activebackground`
Warna latar belakang untuk widget ketika widget aktif
- `activeforeground`
Warna depan untuk widget ketika widget aktif
- `background`
Merepresentasikan sebagai *bg*
- `disableforeground`
Warna depan untuk widget ketika widget dinonaktifkan
- `foreground`
Merepresentasikan *fg*
- `highlightbackground`
Warna latar belakang dari daerah puncak ketika widget memiliki fokus
- `highlightcolor`

Warna depan dari wilayah puncak ketika widget memiliki fokus

- `selectbackground`

Warna latar belakang untuk item yang dipilih dari widget

- `selectforeground`

Warna depan untuk item yang dipilih dari widget

- **Font**

Sebagai tupel yang elemen pertama adalah keluarga font diikuti dengan string yang berisi satu atau lebih gaya pengubah tebal, miring, garis bawah dan overstrike.

Contoh :

- ("Helvetica", "16"-point Helvetica biasa
- ("Times", "24", "beranimiring") untuk 24-point kali miring tebal

Dapat membuat "font object" dengan mengimpor modul `tkFont` dan menggunakan kelas konstruktor font nya :

```
Import tkFont
Font = tkFont.Font (option, ....)
```

Berikut adalah daftar pilihan :

- **Family**

Font nama keluarga sebagai string

- **Size**

Font tinggi sebagai integer dalam poin

- **Weight**

Bold untuk tebal, normal untuk berat badan secara teratur

- **Slant**

Italic untuk miring, roman untuk unstlanted

- **Underline**

1 untuk teks yang digarisbawahi, 0 untuk normal

- **Overstrike**

1 untuk teks telak, 0 untuk normal

Jika berjalan di bawah X window system, dapat menggunakan salah satu nama font X. Sebagai contoh, font bernama "lucidatypewriter-medium-r-*-*-*-*140-*-*-*" adalah favorit fixed-width font penulis untuk digunakan pada layar.

- **Jangkar**

Jangkar digunakan untuk mendefinisikan mana teks diposisikan relatif terhadap titik acuan. Berikut adalah daftar kemungkinan konstanta yang dapat digunakan :

- NW
- N
- NE
- W
- TENGAH
- E
- SW
- S
- SE

Jika menggunakan tengah sebagai jangkar tek, tek akan ditengahkan horizontal dan vertikal disekitar titik referensi.

Jangkar NW akan posisi teks sehingga titik referensi bertepatan dengan laut sudut kotak berisi teks

Jangkar W akan pusat teks secara vertikal disekitar titik referensi dengan tepi kiri kotak teks yang melewati titik itu dan sebagainya.

Jika membuat widget kecil didalam bingkai besar dan menggunakan jangkar = SE pilihan, widget akan ditempatkan disudut kanan bawah gambar. Jika menggunakan anchor = N sebaliknya widget akan dipusatkan disepanjang tepi atas.

Gaya relief

Widget mengacu pada efek 3-D simulasi terbaru disekitar bagian luar widget. Berikut adalah daftar konstanta yang mungkin dapat digunakan untuk atribut:

- Datar
- Dibesarkan
- Cekung
- Alur
- Punggung bukit

Contoh :

```
From Tkinter import *
Import Tkinter
```

```
top = Tkinter.Tk()
B1 = Tkinter.Button(top, text= "FLAT ", relief=FLAT)
B2 = Tkinter.Button(top, text= "RAISED ", relief=RAISED)
B3 =Tkinter.Button(top, text= "SUNKEN ", relief=SUNKEN)
B4=Tkinter.Button(top, text= "GROOVE ", relief=GROOVE)
B5=Tkinter.Button(top, text= "RIDGE ", relief=RIDGE)
```

```
B1.pack()
B2.pack()
```

```

B3.pack()
B4.pack()
B5.pack()
top.mainloop()

```

Bitmaps

Ada beberapa jenis bitmap yang tersedia, diantaranya:

- Kesalahan
- Gray75
- Gray50
- Gray12
- Jam Pasir
- Info
- Questhead
- Perantanyaan
- Peringatan

Contoh:

```

From Tkinter import *
Import Tkinter

```

```

Top = Tkinter.Tk()

```

```

B1 = Tkinter.Button(top, text = "error ", relief=RAISED, n bitmap= "error ")
B2 = Tkinter.Button(top, text = "hourglass ", relief=RAISED, n bitmap= "hourglass
")
B3 = Tkinter.Button(top, text = "info ", relief=RAISED, n bitmap= "info ")
B4 = Tkinter.Button(top, text = "question ", relief=RAISED, n bitmap= "question
")
B5 = Tkinter.Button(top, text = "warning ", relief=RAISED, n bitmap= "warning
")

```

```

B1.pack()
B2.pack()
B3.pack()
B4.pack()
B5.pack()
top.mainloop()

```

Kursor

Berikut daftar menarik :

- Panah
- Lingkaran
- Jam

- Menyebrang
- Dotbox
- Bertukar
- Fluer
- Jantung
- Manusia
- Tikus
- Bajak laut
- Tamah
- Antar jemput
- Perekat
- Laba-laba
- Kaleng semprot
- Bintang
- Target
- Tcross
- Melakukan perjalanan
- Menonton

Contoh :

```
From Tkinter import *
Import Tkinter
```

```
Top = Tkinter.Tk()
```

```
B1 = Tkinter.Button(top, text = "circle ", relief=RAISED, n bitmap= "circle ")
B2 = Tkinter.Button(top, text = "plus ", relief=RAISED, n bitmap= "plus ")
```

```
B1.pack()
B2.pack()
top.mainloop()
```


30.3 Manajemen Geometri

Semua widget tkinter memiliki akses ke metode manajemen geometri tertentu, yang memiliki tujuan mengorganisir widget diseluruh wilayah widget induk. Tkinter mengekspos kelas manager geometri berikut :

- Metode the *pack()*
Manajer geometri ini mengatur widget diblok sebelum menempatkan mereka di widget induk
- Metode the *grid()*
Manajer geometri ini mengatur widget dalam struktur tabel seperti di widget induk
- Metode the *place()*

Manajer geometri ini mengatur widget dengan menempatkan dalam posisi tertentu dalam widget induk

30.4 Manfaat Tkinter

Tkinter sangat sederhana. Berikut manfaat Tkinter dibandingkan GUI toolkit :

- Tkinter mudah diakses oleh siapa saja. (Accessibilty) Tkinter merupakan toolkit yang ringan dan satu-satunya solusi GUI yang paling sederhana untuk Python sampai saat ini. Cukup menuliskan beberapa baris kode Python untuk membuat aplikasi GUI sederhana dengan Tkinter. Untuk menambahkan komponen baru pada Tkinter, dapat membuatnya dalam kode Python atau menambahkan paket ekstensi seperti Pmw, Tix, atau ttk.
- Tkinter mudah digunakan di semua platform (Portability) Sebuah program Python yang dibangun menggunakan Tkinter dapat berjalan dengan baik di semua platform sistem operasi seperti Microsoft Windows, Linux, dan Macintosh. Dan dari segi tampilan window, akan terlihat sama dengan standar platform yang digunakan.
- Tkinter selalu tersedia di Python (Availability) Tkinter merupakan modul standar pada pustaka Python. Sebagian besar paket instalasi Python sudah langsung berisi Tkinter. Khusus untuk beberapa distro Linux, perlu menambahkan paket Tkinter secara terpisah. Pada Windows, bisa langsung menggunakan Tkinter sesaat setelah menginstal paket instalasi Python.
- Dokumentasi Tkinter sangat LUAR BIASA (Documentation) Python (plus Tkinter) ini bersifat open-source, maka banyak sekali komunitas-komunitas yang membahas Python dan Tkinter dan bisa belajar dan bertanya langsung dengan para ahli.

Operator	width= Penjelasan
– Button	Menampilkan tombol dalam aplikasi
– Canvas	Menggambar bentuk seperti garis, oval, poligon dan persegi panjang dalam aplikasi
– Checkbutton	Menampilkan sejumlah pilihan sebagai kotak centang. Pengguna dapat memilih beberapa pilihan pada suatu waktu
– Entry	Menampilka bidang garis teks tunggal untuk menerima nilai-nilai dari pengguna
– Frame	Wadah untuk mengatur widget lainnya
– Label	Memberikan keterangan garis single untuk widget lainnya. Hal ini berisi gambar
– Listbox	Menyediakan daftar pilihan kepada pengguna
– Menubutton	Menampilkan menu dalam aplikasi
– Menu	Memberikan berbagai perintah untuk pengguna. Perintah-perintah ini terkandung di dalam MenuButton
– Message	Menampilkan bidang teks multiline untuk menerima nilai-nilai dari pengguna
– RadioButton	Menampilkan sejumlah pilihan sebagai tombol radio. Pengguna dapat memilih hanya satu pilihan pada suatu waktu
– Scale	Menyediakan widget slide
– Scrollbar	Menambah kemampuan bergulir ke berbagai widget seperti kotak daftar
– Text	Menampilka teks dalam beberapa garis
– Toplevel	Menyediakan wajah jendela terpisah
– PanedWindow	Wadah yang mengandung sejumlah panel disusun horizontal atau vertikal
– LabelFrame	Wadah widget sederhana. Bertindak sebagai spacer atau wajah untuk layout jendela kompleks
– TkMessageBox	Menampilkan kotak pesan dalam aplikasi
– Spinbox	Memilih sejumlah tetap nilai-nilai

<div>—</div> <div>Karakter</div>	<div>Penjelasan</div> <div>width=</div>
<div>—</div> <div>c</div>	<div>Sentimeter</div>
<div>—</div> <div>i</div>	<div>Inci</div>
<div>—</div> <div>m</div>	<div>Milimeter</div>
<div>—</div> <div>p</div>	<div>Poin printer (1/72 ")</div>

CHAPTER 31

FUTHER EXPRESSION

Stiap kode yang dituliskan menggunakan bahasa yang dikompilasi seperti C, C++ atau Java dapat diintegrasikan ke skrip Python lainnya. Kode ini dianggap sebagai ekstensi.

Modul ekstensi Python tidak lebih dari sekedar perpustakaan C biasa. Pada mesin Unix, perpustakaan ini biasanya diakhiri dengan `.so` (untuk objek bersama). Pada mesin windows, biasanya melihat `.dll` (untuk perpustakaan yang terhubung secara dinamis).

31.1 Pra-Persyaratan untuk Menulis Ekstensi

Untuk memulai ekstensi, memerlukan file header Python. Pada mesin Unix, biasanya memerlukan instalasi paket khusus pengembang seperti python 2-5.

Pengguna window mendapatkan header ini sebagai bagian dari paket saat menggunakan pemasang Python biner.

Harus memiliki pengetahuan yang baik tentang C atau C++ untuk menulis ekstensi Python menggunakan pemrograman C.

Untuk melihat modul ekstensi Python, perlu mengelompokkan kode menjadi empat bagian :

- File header Python.h
- Fungsi C yang ingin ditampilkan sebagai antarmuka dari modul
- Sebuah tabel memetakan nama-nama fungsi saat pengembang Python melihat ke fungsi C didalam modul ekstensi
- Fungsi iniliasi

Perlu menyertakan file header Python.h di file sumber C memberi akses ke API Python internal digunakan untuk menghitung modul ke penerjemah.

Menyertakan header Python.h sebelum header lain yang mungkin dibutuhkan. Mengikuti termasuk dengan fungsi yang ingin dipanggil dari Python.

Tanda tangan penerapan C fungsi selalu mengambil salah satu dari tiga bentuk berikut :

```
static PyObject *MyFunction( PyObject *self, PyObject *args );
static PyObject *MyFunctionWithKeywords(PyObject *self,
                                     PyObject *args,
                                     PyObject *kw);
static PyObject *MyFunctionWithNoArgs( PyObject *self );
```

Masing-masing deklarasi seelumnya mengembalikan objek Python. Tidak ada yang namanya fungsi void dengan Python seperti ada di C. Jika ingin fungsi mengembalikan nilai, Python. Header Python mendefinisikan makro. Py_RETURN_NONE yang melakukan ini.

Nama-nama fungsi C bisa menjadi apapun yang disukai karena tidak pernah diluar modul ekstensi mendefinisikan sebagai statis.

Fungsi C biasanya diberi nama dengan menggabungkan modul dan fungsi Python bersama-sama yang ditunjukkan disini :

```
static PyObject *module_func(PyObject *self, PyObject *args) {
    /* Do your stuff here. */
    Py_RETURN_NONE;
}
```

Ini adalah fungsi Python yang disebut func didalam modul-modul. Memasukkan petunjuk ke fungsi C ke dalam tabel metode untuk modul yang biasanya muncul selanjutnya dikode sumber tael pemetaan metode.

Tabel metode ini adalah susunan sederhana dari struktur PyMethodDef. Struktur itu terlihat seperti ini :

```
struct PyMethodDef {
    char *ml_name;
    PyCFunction ml_meth;
    int ml_flags;
    char *ml_doc;
};
```

Inilai uraian anggota struktur ini :

- `MI _name`
Nama fungsi yang digunakan penafsir Python saat digunakan dalam program Python
- `MI _meth`
Menjadi alamat ke fungsi yang memiliki salah satu tanda tangan yang dijelaskan dalam penelusuran sebelumnya
- `MI _flags`
Memberitahu penafsir yang mana dari tiga tanda tangan yang digunakan `ml _meth`. Bendera ini biasanya memiliki nilai `meth _varargs`. Bendera ini dapat digandakan dengan `ored` dengan `meth _keywords` jika ingin memiarkan argumen kata kunci masuk ke fungsi. Ini juga bisa memiliki nilai `meth _noargs` yang menunjukkan bahwa tidak ingin menerima argumen apa pun.
- `MI _doc`
Ini adalah docstring untuk fungsi yang bisa jadi NULL jika tidak ingin menulisnya.

Tabel ini perlu diakhiri dengan sentinel yang terdiri dari NULL dan 0 untuk anggota yang sesuai.

Contoh:

```
static PyMethodDef module _methods[] = {
    { "func", (PyCFunction)module _func, METH_NOARGS, NULL },
    { NULL, NULL, 0, NULL }
};
```

Bagian terakhir dari modul ekstensi adalah fungsi inialisasi. Fungsi ini dipanggil oleh juru bahasa Python saat modul diisikan. Hal ini diperlukan agar fungsi diberi nama `IntiModule` dimana modul adalah nama modul.

Fungsi inialisasi perlu diekspor dari perpustakaan yang akan dibangun. Header Python mendefinisikan `PyMODINIT_FUNC` untuk memasukkan mantra yang sesuai agar terjadi pada lingkungan tertentu tempat menyusun. Yang harus dilakukan adalah menggunakan saat menentukan fungsinya.

Fungsi inialisasi C umumnya memiliki strktur keseluruhan berikut :

```
PyMODINIT_FUNC initModule() {
    Py _InitModule3(func, module _methods, "docstring...");
}
```

Berikut adalah penjelasan fungsi `Py _IntiModule` :

- `Func`
Ini adalah fungsi yang akan diekspor
- `Module`

Ini adalah nama tabel pemetaan yang didefinisikan diatas

- Docstring

Ini adalah komentar yang ingin diberikan diekstensi

Menempatkan ini semua bersama-sama terlihat sebagai berikut :

```
#include <Python.h>
```

```
static PyObject *module_func(PyObject *self, PyObject *args) {
    /* Do your stuff here. */
    Py_RETURN_NONE;
}

static PyMethodDef module_methods[] = {
    { "func", (PyCFunction)module_func, METH_NOARGS, NULL },
    { NULL, NULL, 0, NULL }
};
```

```
PyMODINIT_FUNC inithellworld() {
    Py_InitModule3("hellworld", module_methods, "docstring...");
}
```

Contoh :

```
#include <Python.h>
```

```
static PyObject* helloworld(PyObject* self)
{
    return Py_BuildValue("s", "Hello, Python extensions!!");
}
```

```
static char helloworld_docs[] =
    "helloworld( ): Any message you want to put here!! \n";
```

```
static PyMethodDef helloworld_funcs[] = {
    { "helloworld", (PyCFunction)helloworld,
      METH_NOARGS, helloworld_docs },
    { NULL }
};
```

```
void inithellworld(void)
{
    Py_InitModule3("hellworld", helloworld_funcs,
        "Extension module example!");
}
```

Disini fungsi Py_BuildValue digunakan untuk membangun nilai Python.

31.2 Membangun dan Menginstal Ekstensi

Distutils paket membuatnya sangat mudah mendistribusikan modul Python, baik Python murni dan modul ekstensi dengan cara standar. Modul didistribusikan dalam bentuk sumber dan dibangun dan diinstal melalui skrip setup yang biasa disebut `setup.py` sebagai berikut :

```
from distutils.core import setup, Extension
setup(name='helloworld', version='1.0',
      ext_modules=[Extension('helloworld', ['hello.c'])]).
```

Sekarang gunakan perintah berikut yang akan melakukan semua kompilasi dan langkah penghubung yang diperlukan dengan perintah dan bendera penyusunan dan penghubung yang benar dan menyalin perpustakaan dinamis yang dihasilkan ke dalam direktori yang sesuai .

Contoh :

```
$ python setup.py install
```

Pada sistem berbasis Unix kemungkinan besar perlu menjalankan perintah ini sebagai root agar meminta izin untuk menulis ke direktori paket situs. Ini biasanya tidak menjadi masalah pada window.

Setelah menginstal ekstensi, akan dapat mengimpor dan memanggil ekstensi tersebut di skrip Python sebagai berikut :

```
#!/usr/bin/python
import helloworld
```

```
print helloworld.helloworld()
```

Ini akan menghasilkan hasil sebagai berikut :

```
Hello, Python extensions!!
```

Seperti kemungkinan besar ingin mendefinisikan fungsi yang menerima argumen, dapat menggunakan salah satu tanda tangan lain untuk fungsi C. Sebagai contoh, fungsi berikut yang menerima beberapa parameter akan didefinisikan seperti ini :

```
static PyObject *module_func(PyObject *self, PyObject *args) {
    /* Parse args and do something interesting here. */
    Py_RETURN_NONE;
}
```

Tabel metode yang berisi entri untuk fungsi baru akan terlihat seperti ini :

```
static PyMethodDef module_methods[] = {
    { "func", (PyCFunction)module_func, METH_NOARGS, NULL },
    { "func", module_func, METH_VARARGS, NULL },
    { NULL, NULL, 0, NULL }
};
```

Menggunakan fungsi API `PyArg_ParseTuple` untuk mengekstrak argumen dari satu pointer `PyObject` yang dikirimkan ke fungsi C. Argumen pertama untuk `PyArg_ParseTuple` adalah `args` argumen. Ini adalah objek yang akan parsing. Argumen

kedua adalah string format yang menggambarkan argumen saat mengharapkannya muncul. Setiap argumen diwakili oleh satu atau lebih karakter dalam format string sebagai berikut :

```
static PyObject *module_func(PyObject *self, PyObject *args) {
    int i;
    double d;
    char *s;

    if (!PyArg_ParseTuple(args, "ids", &i, &d, &s)) {
        return NULL;
    }

    /* Do something interesting here. */
    Py_RETURN_NONE;
}
```

Mengkompilasi versi baru dari modul dan mengimpornya memungkinkan untuk memanggil fungsi baru dengan sejumlah argumen dari jenis apa pun :

```
module.func(1, s="three", d=2.0)
module.func(i=1, d=2.0, s="three")
module.func(s="three", d=2.0, i=1)
```

Berikut adalah tanda tangan standar untuk fungsi PyArg_ParseTuple:

```
int PyArg_ParseTuple(PyObject* tuple, char* format,...)
```

Fungsi ini mengembalikan 0 untuk kesalahan, dan nilai tidak sama dengan 0 untuk kesuksesan. Tuple adalah PyObject * yang merupakan argumen kedua dari fungsi C. Format berikut adalah string C yang menggambarkan argumen wajib dan opsional. Berikut adalah daftar kode format untuk fungsi PyArg_ParseTuple:

Py_BuildValue mengambil format string seperti PyArg_ParseTuple. Alih-alih menyampaikan alamat nilai yang sedang bangun, melewati nilai sebenarnya. Berikut adalah contoh yang menunjukkan bagaimana menerapkan fungsi tambah :

```
static PyObject *foo_add(PyObject *self, PyObject *args) {
    int a;
    int b;

    if (!PyArg_ParseTuple(args, "ii", &a, &b)) {
        return NULL;
    }
    return Py_BuildValue("i", a + b);
}
```

Ini adalah apa yang akan terlihat seperti jika diimplementasikan dengan Python :

```
def add(a, b):
    return (a + b)
```

Mengembalikan dua nilai dari fungsi sebagai berikut, ini akan dipicu menggunakan daftar dengan Python :

```
static PyObject *foo _add _subtract(PyObject *self, PyObject *args) {
    int a;
    int b;

    if (!PyArg_ParseTuple(args, "ii", &a, &b)) {
        return NULL;
    }
    return Py_BuildValue("ii", a + b, a - b);
}
```

Ini adalah apa yang akan terlihat seperti jika diimplementasikan dengan Python :

```
def add _subtract(a, b):
    return (a + b, a - b)
```

Berikut adalah tanda tangan standar untuk fungsi Py_BuildValue :

```
PyObject* Py_BuildValue(char* format,...)
```

Format berikut adalah string C yang menggambarkan objek Python untuk dibangun. Argumentasi berikut Py_BuildValue adalah nilai C dari mana hasilnya dibuat. Hasil PyObject * adalah referensi baru.

Berikut daftar tabel string kode yang umum digunakan, yang nol atau lebihnya digabungkan ke dalam format string :

Kode {... } membangun kamus dari sejumlah nilai C, kunci dan nilai bergantian. Misalnya, Py_BuildValue (" {issi }", 23, "zig", "zag", 42) mengembalikan kamus seperti {23: 'zig', 'zag': 42 } Python.

Setiap blok memori yang dialokasikan dengan malloc () pada akhirnya harus dikembalikan ke genangan memori yang tersedia dengan satu panggilan untuk membebaskan (). Penting untuk menelepon gratis () pada waktu yang tepat. Jika alamat blok dilupakan tapi gratis () tidak dipanggil untuk itu, memori yang ditempatinya tidak dapat digunakan kembali sampai program berakhir. Ini disebut kebocoran memori. Di sisi lain, jika sebuah program memanggil gratis () untuk satu blok dan kemudian terus menggunakan blok tersebut, itu menciptakan konflik dengan penggunaan ulang blok melalui panggilan malloc () yang lain. Ini disebut dengan menggunakan memori yang dibebaskan. Ini memiliki konsekuensi buruk yang sama seperti merujuk pada data yang tidak diinisiasi - dump inti, hasil yang salah, crash misterius.

Karena Python membuat penggunaan malloc () dan gratis (), dibutuhkan strategi untuk menghindari kebocoran memori dan juga penggunaan memori yang bebas. Metode yang dipilih disebut penghitungan referensi. Prinsipnya sederhana: setiap objek berisi sebuah counter, yang bertambah saat referensi ke objek disimpan di suatu tempat, dan yang dikurangi saat referensi itu dihapus. Saat counter mencapai nol, referensi terakhir ke objek telah dihapus dan objeknya dibebaskan.

Code	C type	Meaning
c	char	String Python dengan panjang 1 menjadi huruf C.
d	double	Pelampung Python menjadi C ganda.
f	float	Pelampung Python menjadi pelampung C.
i	int	Int Python menjadi int int
l	long	Sebuah int Python menjadi panjang C.
L	long long	Sebuah int Python menjadi C panjang panjang
O	PyObject*	Gets non-NULL meminjam referensi ke argumen Python.
s	char*	Python string tanpa nulls tertanam ke C char*.
s #	char*+int	Setiap string Python ke alamat dan panjang C.
t #	char*+int	Read-only penyangga segmen tunggal ke alamat C dan panjangnya.
u	Py_UNICODE*	Python Unicode tanpa nulls tertanam ke C.
u #	Py_UNICODE*+int	Setiap alamat dan panjang Python Unicode C.
w #	char*+int	Membaca / menulis penyangga segmen tunggal ke alamat dan panjang C.
z	char*	Seperti s, juga menerima None (set C char * ke NULL).
z #	char*+int	Seperti s #, juga menerima None (set C char * ke NULL).
(...)	as per ...	Urutan Python diperlakukan sebagai satu argumen per item.
		Argumen berikut bersifat opsional.
:		Format akhir, diikuti dengan nama fungsi untuk pesan error.
;		Format akhir, diikuti oleh seluruh pesan kesalahan teks.

Code	Type C	Meaning
— c	char	Sebuah char C menjadi string Python dengan panjang 1.
— d	double	C ganda menjadi float Python.
— f	float	Pelampung C menjadi float Python.
— i	int	C int menjadi int Python.
— l	long	Sebuah C panjang menjadi int Python.
— N	PyObject*	Melewati objek Python dan mencuri referensi.
— O	PyObject*	Melewati objek Python dan MENINGKATKANNYA seperti biasa.
— O &	convert+void*	Konversi sewenang-wenang
— s	char*	C 0-diakhiri char * ke string Python, atau NULL to None.
— s #	char*+int	C char * dan panjang ke string Python, atau NULL to None.
— u	Py_UNICODE*	String C-wide, null-terminated menjadi Python Unicode, atau NULL to None.
— u #	Py_UNICODE*+int	String dan panjang lebar C ke Unicode Python, atau NULL to None.
— w #	char*+int	Membaca / menulis penyangga segmen tunggal ke alamat dan panjang C.
— z	char*	Seperti s, juga menerima None (set C char * ke NULL).
— z #	char*+int	Seperti s #, juga menerima None (set C char * ke NULL).
— (...)	as per ...	Membangun tuple Python dari nilai C.
— [...]	as per ...	Membangun daftar Python dari nilai C.
— {... }	as per ...	Bangun kamus Python dari nilai C, kunci dan nilai bergantian.

APPENDIX A

ALTERNATE REFERENCE STYLES

REFERENCES

1. J. S. Kilby, "Invention of the Integrated Circuit," *IEEE Trans. Electron Devices*, **ED-23**, 648 (1976).
2. R. W. Hamming, *Numerical Methods for Scientists and Engineers*, Chapter N-1, McGraw-Hill, New York, 1962.
3. J. Lee, K. Mayaram, and C. Hu, "A Theoretical Study of Gate/Drain Offset in LDD MOSFETs" *IEEE Electron Device Lett.*, **EDL-7**(3). 152 (1986).
4. A. Berenbaum, B. W. Colbry, D.R. Ditzel, R. D Freeman, and K.J. O'Connor, "A Pipelined 32b Microprocessor with 13 kb of Cache Memory," in Int. Solid State Circuit Conf., Dig. Tech. Pap., p. 34 (1987).

REFERENCES

- [Kil76] J. S. Kilby, "Invention of the Integrated Circuit," *IEEE Trans. Electron Devices*, **ED-23**, 648 (1976).
- [Ham62] R. W. Hamming, *Numerical Methods for Scientists and Engineers*, Chapter N-1, McGraw-Hill, New York, 1962.
- [Hu86] J. Lee, K. Mayaram, and C. Hu, "A Theoretical Study of Gate/Drain Offset in LDD MOSFETs" *IEEE Electron Device Lett.*, **EDL-7**(3). 152 (1986).
- [Ber87] A. Berenbaum, B. W. Colbry, D.R. Ditzel, R. D Freeman, and K.J. O'Connor, "A Pipelined 32b Microprocessor with 13 kb of Cache Memory," in *Int. Solid State Circuit Conf.*, Dig. Tech. Pap., p. 34 (1987).

