

Python For Everyone, Enhanced eText

Cay S. Horstmann; Rance D. Necaise

Expand | Collapse

Title Page

Getting the Most from Your eText

Quick Reference

1. Introduction

1.1 Computer Programs

1.2 The Anatomy of a Computer

1.3 The Python Programming Language

1.4 Becoming Familiar with Your Programming Environment

1.5 Analyzing Your First Program

1.6 Errors

1.7 Problem Solving: Algorithm Design

Chapter Summary

Interactive Review and Practice

End-of-Chapter Exercises

2. Programming with Numbers and Strings

3. Decisions

DECISIONS.

the **hardware**. The programs the computer executes are called the **software**.

Today's computer programs are so sophisticated that it is hard to believe that they are composed of extremely primitive instructions. A typical instruction may be one of the following:

- Put a red dot at a given screen position.
- Add up two numbers.
- If this value is negative, continue the program at a certain instruction.


The computer user has the illusion of smooth interaction because a program contains a huge number of such instructions, and because the computer can execute them at great speed.

Programming is the act of designing and implementing computer programs.

The act of designing and implementing computer programs is called **programming**. In this book, you will learn how to program a computer—that is, how to direct the computer to execute tasks.

To write a computer game with motion and sound effects or a word processor that supports fancy fonts and pictures is a complex task that requires a team of many highly-skilled programmers. Your first programming efforts will be more mundane. The concepts and skills you learn in this book form an important foundation, and you should not be disappointed if your first programs do not rival the sophisticated software that is familiar to you. Actually, you will find that there is an immense thrill even in simple programming tasks. It is an amazing experience to see the computer precisely and quickly carry out a task that would take you hours of drudgery, to make small changes in a program that lead to immediate improvements, and to see the computer become an extension of your mental powers.

SELF CHECK



- 1. The physical computer and peripheral devices are collectively called the _____.

☐ software

☐ programming system

☒ hardware

☐ programs

One correct, 0 errors, 100%
- 2. Writing the program for a computer game with motion, graphics, and sound effects usually requires _____.

☒ a team of highly skilled programmers writing a large number of simple instructions

☐ one programmer writing a large number of simple instructions

☐ a team of highly skilled programmers writing a small number of complex instructions

☐ one programmer writing a small number of complex instructions

One correct, 0 errors, 100%

Back to Page

3 / 554

1.1

1.1 Computer Programs (1. & 2)

<

Python For Everyone, Enhanced eText
Cay S. Horstmann; Rance D. Necaise

Expand | Collapse

Title Page i

Getting the Most from Your eText iii

Quick Reference xiii

1. Introduction 1

1.1 Computer Programs 2

1.2 The Anatomy of a Computer 3

1.3 The Python Programming Language 5

1.4 Becoming Familiar with Your Programming Environment 6

1.5 Analyzing Your First Program 11

1.6 Errors 13

1.7 Problem Solving: Algorithm Design 15

Chapter Summary 21

Interactive Review and Practice 22

End-of-Chapter Exercises EX1-1

2. Programming with Numbers and Strings 23

3. Decisions 73

Figure 3 Schematic Design of a Personal Computer

SELF CHECK

1. The part of the computer that executes instructions is the ____.


☐ primary storage
☐ secondary storage
☒ central processing unit
☐ hard disk

One correct, 0 errors, 100%

2. ____ storage, which persists without electricity, is slower but not very expensive.

☐ Primary
☒ Secondary
☐ Electronic
☐ Program

One correct, 0 errors, 100%



Python For Everyone, Enhanced eText
Cay S. Horstmann; Rance D. Necaise

Expand | Collapse

Title Pagei

Getting the Most from Your eTextiii

Quick Referencexiii

1. Introduction1

1.1 Computer Programs2

1.2 The Anatomy of a Computer3

1.3 The Python Programming Language5

1.4 Becoming Familiar with Your Programming Environment6

1.5 Analyzing Your First Program11

1.6 Errors13

1.7 Problem Solving: Algorithm Design15

Chapter Summary21

Interactive Review and Practice22

End-of-Chapter ExercisesEX1-1


2. Programming with Numbers and Strings23

3. Decisions73

important to him that he could author the programs quickly and update them quickly as his needs changed. Therefore, he designed a language that made it very easy to work with complex data. Python has evolved considerably since its beginnings. In this book, we use version 3 of the Python language. Van Rossum is still the principal author of the language, but the effort now includes many volunteers.

Python is portable and easy to learn and use. Python has become popular for business, scientific, and academic applications and is very suitable for the beginning programmer. There are many reasons for the success of Python. Python has a much simpler and cleaner syntax than other popular languages such as Java, C, and C++, which makes it easier to learn. Moreover, you can try out short Python programs in an interactive environment, which encourages experimentation and rapid turnaround. Python is also very portable between computer systems. The same Python program will run, without change, on Windows, UNIX, Linux, and Macintosh.

A package provides code for a particular problem domain. Nowadays, many programmers choose Python because of the availability of packages—bundles of code that solve a particular problem. You can find an astounding number of packages for a wide variety of domains, such as computational biology, machine learning, statistics, data visualization, and many others. Expert developers produce these packages and distribute them, often free of charge. By using a package, you can leverage that expertise in your own projects. For example, you can use a machine learning package to find patterns in your data and a visualization package to display the results. In this book, we introduce you to some of these packages in the optional Toolbox sections.

**SELF CHECK**

1. A _____ consists of a large number of simple CPU instructions that can be tedious and error-prone to specify one by one.

☐ programming language

☒ computer program

☐ high-level language

☐ Python program

One correct, 0 errors, 100%

2. Which of the following is NOT a characteristic of the Python programming language?


☐ It has a simple and clean syntax.

☐ It was designed to make it easy to work with complex data.


☐ It is suitable for the beginner programmer.

☒ Python programs are not portable between computers running different operating systems.


One correct, 0 errors, 100%



Guido van Rossum



6 / 554



Python For Everyone, Enhanced eText

Cay S. Horstmann; Rance D. Necaise

Expand | Collapse

Title Page	i
Getting the Most from Your eText	iii
Quick Reference	xiii
1. Introduction	1
1.1 Computer Programs	2
1.2 The Anatomy of a Computer	3
1.3 The Python Programming Language	5
1.4 Becoming Familiar with Your Programming Environment	6
1.5 Analyzing Your First Program	11
1.6 Errors	13
1.7 Problem Solving: Algorithm Design	15
Chapter Summary	21
Interactive Review and Practice	22
End-of-Chapter Exercises	EX1-1
2. Programming with Numbers and Strings	23
3. Decisions	73

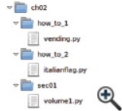



Figure 6 A Folder Hierarchy

Some programming environments place your programs into a default location if you don't specify a folder. In that case, you need to find out where those files are located. Be sure that you understand where your files are located in the folder hierarchy. This information is essential when you submit files for grading, and for making backup copies (see [Programming Tip 1.2](#)).



SELF CHECK

• 1. A(n) _____ is the software that allows a programmer to write, modify, and test programs.

- ☐ editor
- ☐ browser
- ☐ compiler
- ☒ integrated development environment

One correct, 0 errors, 100%

• 2. A Python program is executed using _____.

- ☐ an editor
- ☒ the Python interpreter
- ☐ a compiler
- ☐ the Python library

One correct, 0 errors, 100%

Python For Everyone, Enhanced eText

Cay S. Horstmann; Rance D. Necaise

Expand | Collapse

Title Page

Getting the Most from Your eText

Quick Reference

1. Introduction

1.1 Computer Programs

1.2 The Anatomy of a Computer

1.3 The Python Programming Language

1.4 Becoming Familiar with Your Programming Environment

1.5 Analyzing Your First Program

1.6 Errors

1.7 Problem Solving: Algorithm Design

Chapter Summary

Interactive Review and Practice

End-of-Chapter Exercises

2. Programming with Numbers and Strings

3. Decisions

SELF CHECK

1. A sequence of characters enclosed in quotation marks is called a _____.

☐ comment

☐ statement

☒ string

☐ parameter

One correct, 0 errors, 100%

2. What do the following program instructions print? If the instructions are not correct, type *error* in the Output column.

GOOD JOB! ✓

Program instructions	Output	Explanation
<code>print("Hello", "World")</code>	Hello World	Note that there is a blank space between the outputs. Python automatically inserts a blank space between each value passed to the <code>print</code> function.
<code>print(2 * 2)</code>	4	The result of <code>2 * 2</code> is computed, then printed.
<code>print(Goodbye, World!)</code>	error	You need to enclose the string in quotation marks: <code>print("Goodbye, World!")</code>
How many lines of output does this code produce? <code>print("Hello")</code> <code>print("World!")</code>	2	The output is Hello World! Remember, each use of the <code>print</code> function produces a single line of output.

4 correct, 0 errors, 100%, 104 seconds

11 / 554

1.5 (p.1)

1.5 (p.1) Analyzing Your First Program (1. & 2)

Python For Everyone, Enhanced eText

Cay S. Horstmann; Rance D. Necaise

Expand | Collapse

Title Page

Getting the Most from Your eText

Quick Reference

1. Introduction

1.1 Computer Programs

1.2 The Anatomy of a Computer

1.3 The Python Programming Language

1.4 Becoming Familiar with Your Programming Environment

1.5 Analyzing Your First Program

1.6 Errors

1.7 Problem Solving: Algorithm Design

Chapter Summary

Interactive Review and Practice

End-of-Chapter Exercises

2. Programming with Numbers and Strings

3. Decisions

3. Observe this sequence of statements and update the Output column with what the program will print as each statement executes.

GOOD JOB! ✓

```
print("Hi")
print("there")
print(39 + 3)
print("39+3")
print("x =", 7 + 2)
```

Output
Hi
there
42
39+3
x = 9

5 correct, 0 errors, 100%, 59 seconds

Start over

4. Rearrange the following lines to produce a program that prints an image of a triangle constructed using asterisks.

GOOD JOB! ✓

```
print("  ")
print(" * ")
print(" * ")
print("*****")
```

4 correct, 0 errors, 100%, 42 seconds

11 / 554

1.5 (p.2)

1.5 (p.2) Analyzing Your First Program (3. & 4.)

Python For Everyone, Enhanced eText

Cay S. Horstmann; Rance D. Necaise

Expand | Collapse

Title Page

Getting the Most from Your eText

Quick Reference

1. Introduction

1.1 Computer Programs

1.2 The Anatomy of a Computer

1.3 The Python Programming Language

1.4 Becoming Familiar with Your Programming Environment

1.5 Analyzing Your First Program

1.6 Errors

1.7 Problem Solving: Algorithm Design

Chapter Summary

Interactive Review and Practice

End-of-Chapter Exercises

2. Programming with Numbers and Strings

3. Decisions

GOOD JOB! ✓

```
print("  ")
print(" * ")
print(" * ")
print("*****")
```

4 correct, 0 errors, 100%, 42 seconds

Start over

... 5. Complete the program below so that it displays a second line showing "Hello, All!" below the first line of "Hello, World!". In other words, the output of this program should match the following:

```
Hello, World!
Hello, All!
```

helloall.py

```
1 ##
2 # A program to print two lines
3 #
4
5 print("Hello, World!")
6 print("Hello, All!")
7
8
```

CodeCheck Reset

Running helloall.py

```
Hello, World!
Hello, All!
```

pass

Score

1/1

11 / 554

Python For Everyone, Enhanced eText

Cay S. Horstmann; Rance D. Necaise

Expand | Collapse

Title Page

Getting the Most from Your eText

Quick Reference

1. Introduction

1.1 Computer Programs

1.2 The Anatomy of a Computer

1.3 The Python Programming Language

1.4 Becoming Familiar with Your Programming Environment

1.5 Analyzing Your First Program

1.6 Errors

1.7 Problem Solving: Algorithm Design

Chapter Summary

Interactive Review and Practice

End-of-Chapter Exercises

2. Programming with Numbers and Strings

3. Decisions

SELF CHECK

1. For each of the code fragments below, correct the error, or type OK if there is no error.

GOOD JOB! ✓

Code	Correction or OK	Explanation
<code>print("Hi")</code>	OK	There is nothing wrong with this statement.
<code>print("Hi)</code>	<code>print("Hi")</code>	Each string has opening and closing quotation marks.
<code>pint("Hi")</code>	<code>print("Hi")</code>	If you misspell the name of a function, the compiler will not know what you mean.
<code>Print("Hi")</code>	<code>print("Hi")</code>	The names <code>print</code> and <code>Print</code> are different. Python is "case-sensitive".
<code>print(30 / 2)</code>	OK	There is nothing wrong with this statement. It prints 15.
<code>print("30 / 0")</code>	OK	There is nothing wrong with this statement. It prints <code>30 / 0</code> . Without the quotation marks, there would have been a run-time error because you cannot divide by zero.
<code>print("Result =" 17)</code>	<code>print("Result =", 17)</code>	Multiple items can be printed with a single <code>print</code> function, but the items must be separated by a comma.

7 correct, 0 errors, 100%, 51 seconds

Start over

2. Which of the following statements would generate a compile-time error?

☐ `print("Python")`

☒ `print(Python)`

☐ `print(4 + 7)`

☐ `print("4 + 7")`

One correct, 0 errors, 100%

Back to Page

13 / 554

Python For Everyone, Enhanced eText

Cay S. Horstmann; Rance D. Necaise

Expand | Collapse

Title Page

Getting the Most from Your eText

Quick Reference

1. Introduction

1.1 Computer Programs

1.2 The Anatomy of a Computer

1.3 The Python Programming Language

1.4 Becoming Familiar with Your Programming Environment

1.5 Analyzing Your First Program

1.6 Errors

1.7 Problem Solving: Algorithm Design

Chapter Summary

Interactive Review and Practice

End-of-Chapter Exercises

2. Programming with Numbers and Strings

3. Decisions

One correct, 0 errors, 100%

• 3. Which of the following statements are syntactically correct, but logically inconsistent?

☒

 print("The sum of 5 and 6 is 10")

☐

 print(The sum of 5 and 6 is 10)

☐

 print(The sum of 5 and 6 is 10")

☐

 print("The sum of 5 and 6 is 11")

One correct, 0 errors, 100%

•• 4. For each of the following, indicate whether the given action would result in a compile-time or a run-time error.

☒

compile-time

run-time

Omit the "" characters around Hello, World! from the hello.py program.

Text within a program that is not enclosed within quotes is treated as a command. Hello, World! would be an unknown command that is detected at compile time.

☒

compile-time

run-time

Change the parentheses used with the print function to curly braces.

The syntax requires a pair of parentheses following the print function. Invalid syntax is detected at compile time.

☒

compile-time

run-time

When a program "crashes" (quits spontaneously) or "hangs" (fails to respond to your input).

The program compiles correctly, but a logic error occurs when the program is executed, resulting in a run-time error.

☒

compile-time

run-time

Omit the hash symbol (#) that indicates a comment from the first line of the hello.py program.

As with omitting the quotes around a string, the words of the comment would be treated as a comment. Because they are not valid commands, a syntax error would be detected at compile time.

4 correct, 0 errors, 100%

Back to Page

13 / 554

Python For Everyone, Enhanced eText

Cay S. Horstmann; Rance D. Necaise

Expand | Collapse

Title Page

Getting the Most from Your eText

Quick Reference

1. Introduction

1.1 Computer Programs

1.2 The Anatomy of a Computer

1.3 The Python Programming Language

1.4 Becoming Familiar with Your Programming Environment

1.5 Analyzing Your First Program

1.6 Errors

1.7 Problem Solving: Algorithm Design

Chapter Summary

Interactive Review and Practice

End-of-Chapter Exercises

2. Programming with Numbers and Strings

3. Decisions

Figure 8 The Software Development Process

SELF CHECK

1. Fred's Finest Furniture has a "we pay the sales tax" sale. But Fred doesn't plan to actually pay the sales tax himself. So he has devised the following algorithm to adjust the prices of the items that he sells. Follow the algorithm and update the values in the columns as indicated. You'll probably need a calculator to follow along.

GOOD JOB! ✓

```
originalPrice = 995
taxRate = 6
tax = originalPrice * taxRate / 100
price = originalPrice + tax
Round price to nearest dollar
lastTwo = last two digits of price
If lastTwo > 95:
    Add 100 to price
```

Change last two digits of price to 95

8 correct, 0 errors, 100%, 68 seconds

Start over

15 / 554

Python For Everyone, Enhanced eText

Cay S. Horstmann; Rance D. Necaise

Expand | Collapse

Title Page

Getting the Most from Your eText

Quick Reference

1. Introduction

1.1 Computer Programs

1.2 The Anatomy of a Computer

1.3 The Python Programming Language

1.4 Becoming Familiar with Your Programming Environment

1.5 Analyzing Your First Program

1.6 Errors

1.7 Problem Solving: Algorithm Design

Chapter Summary

Interactive Review and Practice

End-of-Chapter Exercises

2. Programming with Numbers and Strings

3. Decisions

Round price to nearest dollar

$lastTwo = last\ two\ digits\ of\ price$

If $lastTwo > 95$:

Add 100 to price

Change last two digits of price to 95

8 correct, 0 errors, 100%, 68 seconds

Start over

2. Fred hates his bank. His checking account earns only 0.1% per month, and he is charged a monthly "maintenance fee". Plus, his bank always orders these transactions to maximize their profit. Fred is leaving the country for a three-month vacation, and he wonders how much money will be left in his account when he returns. The current balance in his account is \$1000. Rearrange the following lines to produce an algorithm that answers his question.

GOOD JOB! ✓

$vacationMonths = 3$

$interestRate = 0.1$

$balance = 1000$

$fee = 5$

Repeat $vacationMonths$ times:

$balance = balance - fee$

$interest = balance * interestRate / 100$

$balance = balance + interest$

8 correct, 0 errors, 100%, 386 seconds

Start over

15 / 554

1.7 (p.2)

1.7 (p.2) Problem Solving: Algorithm Design (2.)

Python For Everyone, Enhanced eText

Cay S. Horstmann; Rance D. Necaise

Expand | Collapse

Title Page

Getting the Most from Your eText

Quick Reference

1. Introduction

1.1 Computer Programs

1.2 The Anatomy of a Computer

1.3 The Python Programming Language

1.4 Becoming Familiar with Your Programming Environment

1.5 Analyzing Your First Program

1.6 Errors

1.7 Problem Solving: Algorithm Design

Chapter Summary

Interactive Review and Practice

End-of-Chapter Exercises

2. Programming with Numbers and Strings

3. Decisions

balance = balance + interest

8 correct, 0 errors, 100%, 386 seconds

Start over

3. What is the problem with the following algorithm?

Repeat a number of times
Add sales amount to total sales.

☐ The algorithm has a Repeat instruction, but statements in a program cannot be repeated.

☐ The Add statement in the algorithm is not executable.

☐ The Add statement in the algorithm does not clearly specify what is to be added.

☒ The algorithm is ambiguous because it does not specify how many times to repeat the Add statement.

One correct, 0 errors, 100%

4. Select a pseudocode statement to complete the following algorithm, which computes the cost of a shipment. Shipping costs depend on the weight of the item being shipped. The cost is \$10 if the item weighs up to five pounds. For heavier items, the cost is \$10 plus \$2 for each pound in excess of five.

cost = \$10
If weight is greater than five pounds

extra charge = \$2 x number of excess pounds
Add extra charge to cost.

☐ number of excess pounds = weight

☐ number of excess pounds = weight + 5

☒ number of excess pounds = weight - 5

☐ number of excess pounds = weight - 10

One correct, 0 errors, 100%

15 / 554