

Starting Out with Python, 4th Edition, Global Edition

Answers to Review Questions

Chapter 1

Multiple Choice

1. b
2. a
3. d
4. b
5. c
6. a
7. c
8. b
9. a
10. a
11. d
12. b
13. c
14. b
15. c
16. a
17. b
18. d
19. b
20. b
21. c
22. a
23. d
24. a
25. b

True or False

1. False
2. True
3. True
4. False
5. True
6. False
7. True
8. False
9. False
10. False

Short Answer

1. A computer's main memory (random access memory) temporarily stores programs and data that are currently in use, while the computer's secondary storage retains programs and data in an ongoing manner, for long periods of time.

2. A bit that is turned on represents 1, and a bit that is turned off represents 0.
3. 128 different character codes. The Unicode character set addresses this limitation, as it is able to represent many more characters.
4. A statement
5. Mnemonics
6. A compiler is a program that translates a high-level language program into a separate machine language program. The machine language program can then be executed any time it is needed. An interpreter is a program that both translates and executes the instructions in a high-level language program. As the interpreter reads each individual instruction in the program, it converts it to a machine language instruction and then immediately executes it. Because interpreters combine translation and execution, they typically do not create separate machine language programs.
7. Operating system

Exercises

1. *No solution -- This is a hands-on exercise to help you learn how to work with the Python interpreter in interactive mode.*
2. *No solution -- This is a hands-on exercise to help you learn how to work with the IDLE programming environment.*
3.

Decimal	Binary
14	1110
87	01010111
128	10000000
170	10101010
4.

Binary	Decimal
101	5
1111	15
110010	50
5. The first printable character (a space) has a code of 32, the “A” character has a code of 65, and the “a” character has a code of 97.
6.
 - Guido van Rossum is the creator of the Python programming language, and his title of “Benevolent Dictator for Life” (BFDL) means that despite Python being a community-developed project, he has final say in any issues that the community cannot reach an agreement on.
 - A collection of aphorisms that summarise the core guiding principles and design goals of Python. One can print them in IDLE by typing `import this`.
 - The first version of Python 3 (version 3.0) was released in 2008, and the final version of Python 2 (version 2.7) was released in 2010.]

Chapter 2

Multiple Choice

1. c
2. b
3. d
4. b
5. a
6. c
7. a
8. b
9. d
10. a
11. b
12. d
13. b
14. a
15. a
16. c
17. a
18. b
19. a
20. b
21. b
22. b

True or False

1. False
2. True
3. False
4. True
5. False

Short Answer

- 1 Interview the customer
- 2 Pseudocode, an informal language with no syntax rules, is intended to establish and communicate the design of a program. Actual code must adhere to the syntax of the language it is written in, and is used to implement a program.
- 3 (1) Input is received.
(2) Some process is performed on the input.
(3) Output is produced.
- 4 Variable names must adhere to the naming rules of the language, such as not being a key word of the language, beginning with a letter, and not containing a space. Variable names should preferably reflect the nature of the data that they reference to make the variable's purpose clear.

5. Floating point division returns a floating point number that may include fractions. Integer division returns an integer and ignores any fractional part of the division result.
6. A magic number is an unexplained value that appears in a program's code. Magic numbers can be problematic, for a number of reasons. First, it can be difficult for someone reading the code to determine the purpose of the number. Second, if the magic number is used in multiple places in the program, it can take painstaking effort to change the number in each location, should the need arise. Third, you take the risk of making a typographical mistake each time you type the magic number in the program's code.
7. The named constant makes the program more self-explanatory. In a math statement, it is evident that `PI` represents the value of pi. Another advantage to using the named constant is that widespread changes can easily be made to the program. Let's say the value of pi appears in several different statements throughout the program. If you need to change the number of decimal places of precision used with the number, the initialization value in the declaration of the named constant is the only value that needs to be modified. For example, to use only two decimal places of precision, the declaration can be changed to:

```
PI = 3.14
```

The new value of 3.14 will then be used in each statement that includes the `PI` constant. Another advantage to using the named constant is that it helps to prevent the typographical errors that are common when using magic numbers. For example, if you accidentally type 31.4159 instead of 3.14159 in a math statement, the program will calculate the wrong value. However, if you misspell `PI`, the Python interpreter will display a message indicating that the name is not defined.

Algorithm Workbench

1. `age = int(input('Enter your age: '))`
2. `color = input('Enter your favorite color: ')`
3.
 - a. `b = a + 2`
 - b. `a = b * 4`
 - c. `b = a / 3.14`
 - d. `a = b - 8`
4.
 - a. 12
 - b. 4
 - c. 2
 - d. 6
 - e. 2
5. `product = 10 * 15`
6. `due = total - down_payment`
7. `total = subtotal * 0.15`

8. 11
9. 5
10. `print(format(sales, '.2f'))`
11. `print(format(number, ',.1f'))`
12.

\$X	O	X	
O	X	O	
X	O	X	\$
13. `turtle.circle(75)`
14.

```
turtle.fillcolor('blue')
turtle.begin_fill()
turtle.forward(100)
turtle.left(90)
turtle.forward(100)
turtle.left(90)
turtle.forward(100)
turtle.left(90)
turtle.forward(100)
turtle.end_fill()
```
15.

```
turtle.forward(100)
turtle.left(90)
turtle.forward(100)
turtle.left(90)
turtle.forward(100)
turtle.left(90)
turtle.forward(100)
turtle.penup()
turtle.left(90)
turtle.forward(50)
turtle.right(90)
turtle.forward(30)
turtle.setheading(0)
turtle.pendown()
turtle.fillcolor('red')
turtle.begin_fill()
turtle.circle(80)
turtle.end_fill()
```