

Beyond the Basic Stuff with Python

Best Practices for Writing Clean Code

by Al Sweigart

Errata updated to print 5

Page	Error	Correction	Print corrected
xii	Part III : BEST PRACTICES, TOOLS, AND TECHNIQUES	Part II : BEST PRACTICES, TOOLS, AND TECHNIQUES	Print 2
xiii	Deletion (from TOC)	Using the timeit Module to Measure Performance92	Print 2
92–93	Deletion (of section)	Using the timeit Module to Measure Performance	Print 2
100	<pre>>>> # Pythonic Example >>> numberOfPets = {'dogs': 2} >>> numberOfPets.setdefault('cats', 0) # Does nothing if 'cats' exists. 0 >>> workDetails['cats'] += 10 >>> workDetails['cats'] 10</pre>	<pre>>>> # Pythonic Example >>> numberOfPets = {'dogs': 2} >>> numberOfPets.setdefault('cats', 0) # Does nothing if 'cats' exists. 0 >>> numberOfPets['cats'] += 10 >>> numberOfPets['cats'] 10</pre>	Print 2
105	Deletion	You can use Python's timeit module to quickly profile code runtime, which is always better than simply assuming some code runs faster.	Print 2
123	For example, thecode that follows an if or for statement is called the statement's block.	For example, the code that follows an if or for statement is called the statement's block.	Print 2
128	<pre>❶ >>> def greeting(name, species): ... print(name + ' is a ' + description) ... ❷ >>> greeting('Zophie', 'cat') Zophie is a cat</pre>	<pre>❶ >>> def greeting(name, species): ... print(name + ' is a ' + species) ... ❷ >>> greeting('Zophie', 'cat') Zophie is a cat</pre>	Print 2

Page	Error	Correction	Print corrected
163, 164, 249, 251, 255	AB to moves a disk	AB to move a disk	Print 3
234	Six books require 2^6 or 32 photos, but 32 books would include 2^{32} or more than 4.2 billion photos. $O(2^n)$, $O(3^n)$, $O(4^n)$, and so on are different orders ut all have exponential time complexity.	Six books require 2^6 or 32 photos, but 32 books would include 2^{32} or more than 4.2 billion photos. $O(2^n)$, $O(3^n)$, $O(4^n)$, and so on are different orders but all have exponential time complexity.	Print 3
240	<pre>def findDuplicateBooks(books): for i in range(books): # n steps for j in range(i + 1, books): # n steps if books[i] == books[j]: # 1 step print('Duplicate:', books[i]) # 1 step</pre>	<pre>def findDuplicateBooks(books): for i in range(len(books)): # n steps for j in range(i + 1, len(books)): # n steps if books[i] == books[j]: # 1 step print('Duplicate:', books[i]) # 1 step</pre>	Print 5
253	<pre>FUNCTIONS displayDisk(width) Display a single disk of the given width. --snip--</pre>	<pre>FUNCTIONS displayDisk(width) Display a single disk of the given width. A width of 0 means no disk. --snip--</pre>	Print 5
265	<pre>BOARD_EDGE = " +" + ("-" * BOARD_WIDTH) + "+" BOARD_ROW = " " + ("{" * BOARD_WIDTH) + " \n" BOARD_TEMPLATE = "\n " + "".join(COLUMN_LABELS) + "\n" + BOARD_EDGE + "\n" + (BOARD_ROW * BOARD_WIDTH) + BOARD_EDGE</pre>	<pre>BOARD_EDGE = " +" + ("-" * BOARD_WIDTH) + "+" BOARD_ROW = " " + ("{" * BOARD_WIDTH) + " \n" BOARD_TEMPLATE = "\n " + "".join(COLUMN_LABELS) + "\n" + BOARD_EDGE + "\n" + (BOARD_ROW * BOARD_HEIGHT) + BOARD_EDGE</pre>	Print 3
286	<pre>def isValidSpace(board, space): """Returns True if the space on the board is a valid space number and the space is blank.""" return space in ALL_SPACES or board[space] == BLANK</pre>	<pre>def isValidSpace(board, space): """Returns True if the space on the board is a valid space number and the space is blank.""" return space in ALL_SPACES and board[space] == BLANK</pre>	Print 4
312	For example, a Customer object could have a birthdate method that is assigned a Date object rather than the Customer class subclassing Date.	For example, a Customer object could have a birthdate attribute that is assigned a Date object rather than the Customer class subclassing Date.	Print 3
317	<pre>class ClassWithProperties: def __init__(self): self.someAttribute = 'some initial value'</pre>	<pre>class ClassWithProperties: def __init__(self): self._someAttribute = 'some initial value'</pre>	Print 5