

Impractical Python Projects

Playful Programming Activities to Make You Smarter

by Lee Vaughan

errata updated to print 7

Page	Error	Correction	Print corrected
33	dairy raid	welsh slew	Print 6
68	<pre>for i in range(len(list_of_lists)): print(list_of_lists[i])</pre>	<pre>for nested_list in list_of_lists: print(nested_list)</pre>	Print 5
79	Here's the output of the program, using the ciphertext from Figure 4-2:	Here's the output of the program, using the ciphertext from Figure 4-3:	Print 2
85	<pre>row1 = (message[:row_1_len]) row2 = (message[row_1_len:])</pre>	<pre>row1 = (message[:row_1_len]).lower() row2 = (message[row_1_len:]).lower()</pre>	Print 3
85	<pre>plaintext.append(r1.lower()) plaintext.append(r2.lower())</pre>	<pre>plaintext.append(r1) plaintext.append(r2)</pre>	Print 3
100	Panel at east end of chapel slides	Panelateastendofchapelslides	Print 3
103	The cold tea didn't please the old finicky woman	So, the cold tea didn't please the old finicky woman	Print 3
111	<p>A paragraph object has a variety of properties that specify its placement within a container—typically a page—and the way it divides its contents into separate lines. You can access the formatting properties of a paragraph with the ParagraphFormat object available through the ParagraphFormat property of the paragraph, and you can set all the paragraph properties using a <i>paragraph style grouping</i> or apply them directly to a paragraph.</p> <p>A run is an <i>inline-level</i> object that occurs within paragraphs or other block-level objects. A run object has a read-only font property providing access to a font object. A font object provides properties for getting and setting the character formatting for that run. You'll need this feature for setting your hidden message's text color to white.</p>	<p>A paragraph object has a variety of attributes that specify its placement within a container—typically a page—and the way it divides its contents into separate lines. You can access the formatting attributes of a paragraph with the ParagraphFormat object available through the ParagraphFormat attribute of the paragraph, and you can set all the paragraph attributes using a <i>paragraph style grouping</i> or apply them directly to a paragraph.</p> <p>A run is an <i>inline-level</i> object that occurs within paragraphs or other block-level objects. A run object has a read-only font attribute providing access to a font object. A font object provides attributes for getting and setting the character formatting for that run. You'll need this feature for setting your hidden message's text color to white.</p>	Print 5

Page	Error	Correction	Print corrected
116	Define a function that formats the spacing between paragraphs using python-docx's paragraph_format property ❶.	Define a function that formats the spacing between paragraphs using python-docx's paragraph_format attribute ❶.	Print 5
141	<pre># mutate lock_wheel = int(randrange(0, len(combo))) ❶ next_try[lock_wheel] = randint(0, len(combo)-1)</pre>	<pre># mutate lock_wheel = randrange(0, len(combo)) ❶ next_try[lock_wheel] = randint(0, 9)</pre>	Print 3
156	... and adding the key/value pair (at any location, since dictionaries are unordered).	... and adding the key/value pair at any location .	Print 5
164	Because of the very short training corpus, the moon is the only word pair with multiple keys .	Because of the very short training corpus, the moon is the only word pair with multiple values .	Print 4
171	This is a far better solution than manually finding and commenting out print() statements!	This is a far better solution than manually finding and commenting out calls to print() !	Print 6
182	Cool stars enter the Window this hot evening all Heaven and earth ache	A line flap-flapping Across the dark crimson sky On this winter pond	Print 5
205	The transformation to generate points over a <i>unit</i> disc is: $x = \sqrt{r} \cdot \cos\theta$ The equations yield (x, y) values between 0 and 1.	The transformation to generate points evenly over a <i>unit</i> disc is: $x = \sqrt{r} \cdot \cos\theta$ The equations yield (x, y) values between -1 and 1.	Print 3
218	<pre>>>> from random import randint >>> trials = 100000 >>> success = 0 >>> for trial in range(trials): faces = set() for rolls in range(6): roll = randint(1, 6) faces.add(roll) if len(faces) == 6: success += 1 >>> print("probability of success = {}".format(success/trials))</pre>	<pre>>>> from random import randint >>> trials = 100000 >>> success = 0 >>> for trial in range(trials): faces = set() for rolls in range(6): roll = randint(1, 6) faces.add(roll) if len(faces) == 6: success += 1 >>> print("probability of success = {}".format(success/trials))</pre>	Print 2
250	<pre>❶ prompt = '{} [{}]: '.format(prompt, default) ❶ response = input(prompt) ❶ if not response and default:</pre>	<pre>❶ prompt = '{} [{}]: '.format(prompt, default) ❶ response = input(prompt) ❶ if not response and default:</pre>	Print 2
252	Set the default to ' sbc_blend ', since this is theoretically the most stable mix of the four choices .	Set the default to ' bonds ', in order to see how this supposedly 'safe' choice performs .	Print 3

Page	Error	Correction	Print corrected
259	... a 4 percent withdrawal rate (equal to \$80,000 per year), a 30-year retirement , and 50,000 cases.	... a 4 percent withdrawal rate (equal to \$80,000 per year), a 29-30-31 retirement range , and 50,000 cases.	Print 3
261	<pre> else: withdraw_infl_adj = withdraw_infl_adj_2 investments -= withdraw_infl_adj investments = int(investments * (1 + i)) </pre>	<pre> else: withdraw_infl_adj = withdraw_infl_adj_2 investments -= withdraw_infl_adj investments = int(investments * (1 + i)) </pre>	Print 3
305	You'll use the same <code>transform.rotate()</code> method you used to turn the satellite	You'll use the same <code>transform.rotate()</code> method you used to turn the satellite	Print 3
329	The shell utilities module, <code>shutil</code> , provides high-level functions for working with files and folders, such as copying, moving, renaming , and deleting.	The shell utilities module, <code>shutil</code> , provides high-level functions for working with files and folders, such as copying, moving, and deleting.	Print 6
356	<pre> ⑤ first_digits[sample[0]] += 1 ⑥ data_count = [v for (k, v) in sorted(first_digits.items())] </pre>	<pre> ⑤ first_digits[sample[0]] += 1 # check for missing digits keys = [str(digit) for digit in range(1, 10)] for key in keys: if key not in first_digits: first_digits[key] = 0 ⑥ data_count = [v for (k, v) in sorted(first_digits.items())] </pre>	Print 3
357	Deletion	Like all Python dictionaries, <code>first_digits</code> is unordered.	Print 5
360	This will work with no arguments, but set its size property to 15 and turn off the frame around the legend for an arguably more attractive result.	This will work with no arguments, but set its size attribute to 15 and turn off the frame around the legend for an arguably more attractive result.	Print 5

Page	Error	Correction	Print corrected
368	<pre>"""Remove single-letter words from list if not 'a' or 'i'.""" word_list = ['a', 'nurses', 'i', 'stack', 'b', 'cats', 'c'] permissible = ('a', 'i') for word in word_list: if len(word) == 1 and word not in permissible: word_list.remove(word) print("{}".format(word_list_clean))</pre>	<pre>"""Remove single-letter words from list if not 'a' or 'i'.""" word_list = ['a', 'nurses', 'i', 'stack', 'b', 'c', 'cat'] word_list_clean = [] permissible = ('a', 'i') for word in word_list: if len(word) > 1: word_list_clean.append(word) elif len(word) == 1 and word in permissible: word_list_clean.append(word) else: continue print("{}".format(word_list_clean))</pre>	Print 3
369	<pre>print(*digrams, sep='\n') --snip-- for k in mapped:</pre>	<pre>print(*sorted(digrams), sep='\n') --snip-- for k in sorted(mapped):</pre>	Print 5