### **Chapter 1**

[1](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1250). The operators are +, -, \*, and /. The values are 'hello', -88.8, and 5.

[2](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1251). The variable is spam; the string is 'spam'. Strings always start and end with quotes.

[3](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1252). The three data types introduced in this chapter are integers, floating-point numbers, and strings.

[4](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1253). An expression is a combination of values and operators. All expressions evaluate (that is, reduce) to a single value.

[5](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1254). An expression evaluates to a single value. A statement does not.

[6](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1255). The bacon variable is set to 20. The bacon + 1 expression does not reassign the value in bacon (that would need an assignment statement: bacon = bacon + 1).

[7](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1256). Both expressions evaluate to the string 'spamspamspam'.

[8](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1257). Variable names cannot begin with a number.

[9](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1258). The int(), float(), and str() functions will evaluate to the integer, floating-point number, and string versions of the value passed to them.

[10](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1259). The expression causes an error because 99 is an integer, and only strings can be concatenated to other strings with the + operator. The correct way is I have eaten ' + str(99) + ' burritos.'.

### **Chapter 2**

[1](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1260). True and False, using capital T and F, with the rest of the word in lowercase

[2](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1261). and, or, and not

[3](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1262). True and True is True.

True and False is False.

False and True is False.

False and False is False.

True or True is True.

True or False is True.

False or True is True.

False or False is False.

not True is False.

not False is True.

[4](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1263). False

False

True

False

False

True

[5](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1264). ==, !=, <, >, <=, and >=

[6](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1265). == is the equal to operator that compares two values and evaluates to a Boolean, while = is the assignment operator that stores a value in a variable.

[7](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1266). A condition is an expression used in a flow control statement that evaluates to a Boolean value.

[8](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1267). The three blocks are everything inside the if statement and the lines print('bacon') and print('ham').

print('eggs')  
if spam > 5:  
    print('bacon')  
else:  
    print('ham')  
print('spam')

[9](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1268). The code:

if spam == 1:  
    print('Hello')  
elif spam == 2:  
    print('Howdy')  
else:  
    print('Greetings!')

[10](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1269). Press CTRL-C to stop a program stuck in an infinite loop.

[11](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1270). The break statement will move the execution outside and just after a loop. The continue statement will move the execution to the start of the loop.

[12](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1271). They all do the same thing. The range(10) call ranges from 0 up to (but not including) 10, range(0, 10) explicitly tells the loop to start at 0, and range(0, 10, 1) explicitly tells the loop to increase the variable by 1 on each iteration.

[13](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1272). The code:

for i in range(1, 11):  
    print(i)

and:

i = 1  
while i <= 10:  
    print(i)  
    i = i + 1

[14](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1273). This function can be called with spam.bacon().

### **Chapter 3**

[1](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1274). Functions reduce the need for duplicate code. This makes programs shorter, easier to read, and easier to update.

[2](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1275). The code in a function executes when the function is called, not when the function is defined.

[3](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1276). The def statement defines (that is, creates) a function.

[4](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1277). A function consists of the def statement and the code in its def clause. A function call is what moves the program execution into the function, and the function call evaluates to the function’s return value.

[5](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1278). There is one global scope, and a local scope is created whenever a function is called.

[6](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1279). When a function returns, the local scope is destroyed, and all the variables in it are forgotten.

[7](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1280). A return value is the value that a function call evaluates to. Like any value, a return value can be used as part of an expression.

[8](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1281). If there is no return statement for a function, its return value is None.

[9](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1282). A global statement will force a variable in a function to refer to the global variable.

[10](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1283). The data type of None is NoneType.

[11](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1284). That import statement imports a module named areallyourpetsnamederic. (This isn’t a real Python module, by the way.)

[12](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1285). This function could be called with spam.bacon().

[13](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1286). Place the line of code that might cause an error in a try clause.

[14](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1287). The code that could potentially cause an error goes in the try clause. The code that executes if an error happens goes in the except clause.

### **Chapter 4**

[1](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1288). The empty list value, which is a list value that contains no items. This is similar to how '' is the empty string value.

[2](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1289). spam[2] = 'hello' (Notice that the third value in a list is at index 2 because the first index is 0.)

[3](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1290). 'd' (Note that '3' \* 2 is the string '33', which is passed to int() before being divided by 11. This eventually evaluates to 3. Expressions can be used wherever values are used.)

[4](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1291). 'd' (Negative indexes count from the end.)

[5](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1292). ['a', 'b']

[6](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1293). 1

[7](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1294). [3.14, 'cat', 11, 'cat', True, 99]

[8](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1295). [3.14, 11, 'cat', True]

[9](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1296). The operator for list concatenation is +, while the operator for replication is \*. (This is the same as for strings.)

[10](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1297). While append() will add values only to the end of a list, insert() can add them anywhere in the list.

[11](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1298). The del statement and the remove() list method are two ways to remove values from a list.

[12](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1299). Both lists and strings can be passed to len(), have indexes and slices, be used in for loops, be concatenated or replicated, and be used with the in and not in operators.

[13](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1300). Lists are mutable; they can have values added, removed, or changed. Tuples are immutable; they cannot be changed at all. Also, tuples are written using parentheses, ( and ), while lists use the square brackets, [ and ].

[14](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1301). (42,) (The trailing comma is mandatory.)

[15](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1302). The tuple() and list() functions, respectively

[16](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1303). They contain references to list values.

[17](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1304). The copy.copy() function will do a shallow copy of a list, while the copy.deepcopy() function will do a deep copy of a list. That is, only copy.deepcopy() will duplicate any lists inside the list.

### **Chapter 5**

[1](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1305). Two curly brackets: {}

[2](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1306). {'foo': 42}

[3](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1307). The items stored in a dictionary are unordered, while the items in a list are ordered.

[4](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1308). You get a KeyError error.

[5](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1309). There is no difference. The in operator checks whether a value exists as a key in the dictionary.

[6](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1310). The 'cat' in spam checks whether there is a 'cat' key in the dictionary, while 'cat' in spam.values() checks whether there is a value 'cat' for one of the keys in spam.

[7](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1311). spam.setdefault('color', 'black')

[8](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1312). pprint.pprint()

### **Chapter 6**

[1](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1313). Escape characters represent characters in string values that would otherwise be difficult or impossible to type into code.

[2](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1314). \n is a newline; \t is a tab.

[3](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1315). The \\ escape character will represent a backslash character.

[4](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1316). The single quote in Howl's is fine because you’ve used double quotes to mark the beginning and end of the string.

[5](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1317). Multiline strings allow you to use newlines in strings without the \n escape character.

[6](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1318). The expressions evaluate to the following:

* 'e'
* 'Hello'
* 'Hello'
* 'lo, world!

[7](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1319). The expressions evaluate to the following:

* 'HELLO'
* True
* 'hello'

[8](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1320). The expressions evaluate to the following:

* ['Remember,', 'remember,', 'the', 'fifth', 'of', 'November.']
* 'There-can-be-only-one.'

[9](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1321). The rjust(), ljust(), and center() string methods, respectively

[10](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1322). The lstrip() and rstrip() methods remove whitespace from the left and right ends of a string, respectively.

### **Chapter 7**

[1](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1323). The re.compile() function returns Regex objects.

[2](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1324). Raw strings are used so that backslashes do not have to be escaped.

[3](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1325). The search() method returns Match objects.

[4](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1326). The group() method returns strings of the matched text.

[5](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1327). Group 0 is the entire match, group 1 covers the first set of parentheses, and group 2 covers the second set of parentheses.

[6](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1328). Periods and parentheses can be escaped with a backslash: \., \(, and \).

[7](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1329). If the regex has no groups, a list of strings is returned. If the regex has groups, a list of tuples of strings is returned.

[8](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1330). The | character signifies matching “either, or” between two groups.

[9](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1331). The ? character can either mean “match zero or one of the preceding group” or be used to signify non-greedy matching.

[10](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1332). The + matches one or more. The \* matches zero or more.

[11](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1333). The {3} matches exactly three instances of the preceding group. The {3,5} matches between three and five instances.

[12](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1334). The \d, \w, and \s shorthand character classes match a single digit, word, or space character, respectively.

[13](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1335). The \D, \W, and \S shorthand character classes match a single character that is not a digit, word, or space character, respectively.

[14](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1336). The .\* performs a greedy match, and the .\*? performs a non-greedy match.

[15](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1337). Either [0-9a-z] or [a-z0-9]

[16](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1338). Passing re.I or re.IGNORECASE as the second argument to re.compile() will make the matching case insensitive.

[17](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1339). The . character normally matches any character except the newline character. If re.DOTALL is passed as the second argument to re.compile(), then the dot will also match newline characters.

[18](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1340). The sub() call will return the string 'X drummers, X pipers, five rings, X hens'.

[19](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1341). The re.VERBOSE argument allows you to add whitespace and comments to the string passed to re.compile().

[20](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1342). re.compile(r'^\d{1,3}(,\d{3})\*$') will create this regex, but other regex strings can produce a similar regular expression.

[21](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1343). re.compile(r'[A-Z][a-z]\*\sWatanabe')

[22](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1344). re.compile(r'(Alice|Bob|Carol)\s(eats|pets|throws)\s(apples|cats|baseballs)\.', re.IGNORECASE)

### **Chapter 8**

[1](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1345). No. PyInputPlus is a third-party module and doesn’t come with the Python Standard Library.

[2](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1346). This optionally makes your code shorter to type: you can type pyip.inputStr() instead of pyinputplus.inputStr().

[3](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1347). The inputInt() function returns an int value, while the inputFloat() function returns a float value. This is the difference between returning 4 and 4.0.

[4](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1348). Call pyip.inputint(min=0, max=99).

[5](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1349). A list of regex strings that are either explicitly allowed or denied

[6](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1350). The function will raise RetryLimitException.

[7](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1351). The function returns the value 'hello'.

### **Chapter 9**

[1](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1352). Relative paths are relative to the current working directory.

[2](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1353). Absolute paths start with the root folder, such as / or C:\.

[3](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1354). On Windows, it evaluates to WindowsPath('C:/Users/Al'). On other operating systems, it evaluates to a different kind of Path object but with the same path.

[4](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1355). The expression 'C:/Users' / 'Al' results in an error, since you can’t use the / operator to join two strings.

[5](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1356). The os.getcwd() function returns the current working directory. The os.chdir() function changes the current working directory.

[6](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1357). The . folder is the current folder, and .. is the parent folder.

[7](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1358). C:\bacon\eggs is the dir name, while spam.txt is the base name.

[8](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1359). The string 'r' for read mode, 'w' for write mode, and 'a' for append mode

[9](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1360). An existing file opened in write mode is erased and completely overwritten.

[10](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1361). The read() method returns the file’s entire contents as a single string value. The readlines() method returns a list of strings, where each string is a line from the file’s contents.

[11](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1362). A shelf value resembles a dictionary value; it has keys and values, along with keys() and values() methods that work similarly to the dictionary methods of the same names.

### **Chapter 10**

[1](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1363). The shutil.copy() function will copy a single file, while shutil.copytree() will copy an entire folder, along with all its contents.

[2](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1364). The shutil.move() function is used for renaming files as well as moving them.

[3](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1365). The send2trash functions will move a file or folder to the recycle bin, while shutil functions will permanently delete files and folders.

[4](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1366). The zipfile.ZipFile() function is equivalent to the open() function; the first argument is the filename, and the second argument is the mode to open the ZIP file in (read, write, or append).

### **Chapter 11**

[1](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1367). assert spam >= 10, 'The spam variable is less than 10.'

[2](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1368). Either assert eggs.lower() != bacon.lower() 'The eggs and bacon variables are the same!' or assert eggs.upper() != bacon.upper(), 'The eggs and bacon variables are the same!'

[3](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1369). assert False, 'This assertion always triggers.'

[4](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1370). To be able to call logging.debug(), you must have these two lines at the start of your program:

import logging  
logging.basicConfig(level=logging.DEBUG, format=' %(asctime)s -  
%(levelname)s -  %(message)s')

[5](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1371). To be able to send logging messages to a file named programLog.txt with logging.debug(), you must have these two lines at the start of your program:

import logging  
>>> logging.basicConfig(filename='programLog.txt', level=logging.DEBUG,  
format=' %(asctime)s -  %(levelname)s -  %(message)s')

[6](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1372). DEBUG, INFO, WARNING, ERROR, and CRITICAL

[7](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1373). logging.disable(logging.CRITICAL)

[8](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1374). You can disable logging messages without removing the logging function calls. You can selectively disable lower-level logging messages. You can create logging messages. Logging messages provides a timestamp.

[9](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1375). The Step In button will move the debugger into a function call. The Step Over button will quickly execute the function call without stepping into it. The Step Out button will quickly execute the rest of the code until it steps out of the function it currently is in.

[10](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1376). After you click Continue, the debugger will stop when it has reached the end of the program or a line with a breakpoint.

[11](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1377). A breakpoint is a setting on a line of code that causes the debugger to pause when the program execution reaches the line.

[12](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1378). To set a breakpoint in Mu, click the line number to make a red dot appear next to it.

### **Chapter 12**

[1](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1379). The webbrowser module has an open() method that will launch a web browser to a specific URL, and that’s it. The requests module can download files and pages from the web. The BeautifulSoup module parses HTML. Finally, the selenium module can launch and control a browser.

[2](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1380). The requests.get() function returns a Response object, which has a text attribute that contains the downloaded content as a string.

[3](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1381). The raise\_for\_status() method raises an exception if the download had problems and does nothing if the download succeeded.

[4](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1382). The status\_code attribute of the Response object contains the HTTP status code.

[5](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1383). After opening the new file on your computer in 'wb' “write binary” mode, use a for loop that iterates over the Response object’s iter\_content() method to write out chunks to the file. Here’s an example:

saveFile = open('filename.html', 'wb')  
for chunk in res.iter\_content(100000):  
    saveFile.write(chunk)

[6](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1384). F12 brings up the developer tools in Chrome. Pressing CTRL-SHIFT-C (on Windows and Linux) or -OPTION-C (on OS X) brings up the developer tools in Firefox.

[7](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1385). Right-click the element in the page and select **Inspect Element** from the menu.

[8](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1386). '#main'

[9](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1387). '.highlight'

[10](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1388). 'div div'

[11](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1389). 'button[value="favorite"]'

[12](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1390). spam.getText()

[13](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1391). linkElem.attrs

[14](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1392). The selenium module is imported with from selenium import webdriver.

[15](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1393). The find\_element\_\* methods return the first matching element as a WebElement object. The find\_elements\_\* methods return a list of all matching elements as WebElement objects.

[16](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1394). The click() and send\_keys() methods simulate mouse clicks and keyboard keys, respectively.

[17](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1395). Calling the submit() method on any element within a form submits the form.

[18](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1396). The forward(), back(), and refresh() WebDriver object methods simulate these browser buttons.

### **Chapter 13**

[1](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1397). The openpyxl.load\_workbook() function returns a Workbook object.

[2](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1398). The sheetnames attribute contains a Worksheet object.

[3](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1399). Run wb['Sheet1'].

[4](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1400). Use wb.active.

[5](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1401). sheet['C5'].value or sheet.cell(row=5, column=3).value

[6](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1402). sheet['C5'] = 'Hello' or sheet.cell(row=5, column=3).value = 'Hello'

[7](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1403). cell.row and cell.column

[8](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1404). They hold the highest column and row with values in the sheet, respectively, as integer values.

[9](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1405). openpyxl.cell.column\_index\_from\_string('M')

[10](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1406). openpyxl.cell.get\_column\_letter(14)

[11](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1407). sheet['A1':'F1']

[12](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1408). wb.save('example.xlsx')

[13](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1409). A formula is set the same way as any value. Set the cell’s value attribute to a string of the formula text. Remember that formulas begin with the = sign.

[14](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1410). When calling load\_workbook(), pass True for the data\_only keyword argument.

[15](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1411). sheet.row\_dimensions[5].height = 100

[16](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1412). sheet.column\_dimensions['C'].hidden = True

[17](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1413). Freeze panes are rows and columns that will always appear on the screen. They are useful for headers.

[18](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1414). openpyxl.chart.Reference(), openpyxl.chart.Series(), openpyxl.chart.BarChart(), chartObj.append(seriesObj), and add\_chart()

### **Chapter 14**

[1](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1415). To access Google Sheets, you need a credentials file, a token file for Google Sheets, and a token file for Google Drive.

[2](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1416). EZSheets has ezsheets.Spreadsheet and ezsheets.Sheet objects.

[3](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1417). Call the downloadAsExcel() Spreadsheet method.

[4](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1418). Call the ezsheets.upload() function and pass the filename of the Excel file.

[5](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1419). Access ss['Students']['B2'].

[6](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1420). Call ezsheets.getColumnLetterOf(999).

[7](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1421). Access the rowCount and columnCount properties of the Sheet object.

[8](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1422). Call the delete() Sheet method. This is only permanent if you pass the permanent=True keyword argument.

[9](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1423). The createSpreadsheet() function and createSheet() Spreadsheet method will create Spreadsheet and Sheet objects, respectively.

[10](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1424). EZSheets will throttle your method calls.

### **Chapter 15**

[1](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1425). A File object returned from open()

[2](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1426). Read-binary ('rb') for PdfFileReader() and write-binary ('wb') for PdfFileWriter()

[3](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1427). Calling getPage(4) will return a Page object for [page 5](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-18), since page 0 is the first page.

[4](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1428). The numPages variable stores an integer of the number of pages in the PdfFileReader object.

[5](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1429). Call decrypt('swordfish').

[6](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1430). The rotateClockwise() and rotateCounterClockwise() methods. The degrees to rotate is passed as an integer argument.

[7](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1431). docx.Document('demo.docx')

[8](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1432). A document contains multiple paragraphs. A paragraph begins on a new line and contains multiple runs. Runs are contiguous groups of characters within a paragraph.

[9](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1433). Use doc.paragraphs.

[10](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1434). A Run object has these variables (not a Paragraph).

[11](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1435). True always makes the Run object bolded and False makes it always not bolded, no matter what the style’s bold setting is. None will make the Run object just use the style’s bold setting.

[12](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1436). Call the docx.Document() function.

[13](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1437). doc.add\_paragraph('Hello there!')

[14](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1438). The integers 0, 1, 2, 3, and 4

### **Chapter 16**

[1](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1439). In Excel, spreadsheets can have values of data types other than strings; cells can have different fonts, sizes, or color settings; cells can have varying widths and heights; adjacent cells can be merged; and you can embed images and charts.

[2](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1440). You pass a File object, obtained from a call to open().

[3](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1441). File objects need to be opened in read-binary ('rb') for reader objects and write-binary ('wb') for writer objects.

[4](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1442). The writerow() method

[5](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1443). The delimiter argument changes the string used to separate cells in a row. The lineterminator argument changes the string used to separate rows.

[6](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1444). json.loads()

[7](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1445). json.dumps()

### **Chapter 17**

[1](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1446). A reference moment that many date and time programs use. The moment is January 1, 1970, UTC.

[2](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1447). time.time()

[3](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1448). time.sleep(5)

[4](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1449). It returns the closest integer to the argument passed. For example, round(2.4) returns 2.

[5](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1450). A datetime object represents a specific moment in time. A timedelta object represents a duration of time.

[6](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1451). Run datetime.datetime(2019, 1, 7).weekday(), which returns 0. This means Monday, as the datetime module uses 0 for Monday, 1 for Tuesday, and so on up to 6 for Sunday.

[7](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1452). threadObj = threading.Thread(target=spam) threadObj.start()

[8](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1453). Make sure that code running in one thread does not read or write the same variables as code running in another thread.

### **Chapter 18**

[1](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1454). SMTP and IMAP, respectively

[2](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1455). smtplib.SMTP(), smtpObj.ehlo(), smptObj.starttls(), and smtpObj.login()

[3](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1456). imapclient.IMAPClient() and imapObj.login()

[4](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1457). A list of strings of IMAP keywords, such as 'BEFORE <date>', 'FROM <string>', or 'SEEN'

[5](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1458). Assign the variable imaplib.\_MAXLINE a large integer value, such as 10000000.

[6](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1459). The pyzmail module reads downloaded emails.

[7](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1460). The credentials.json and token.json files tell the EZGmail module which Google account to use when accessing Gmail.

[8](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1461). A message represents a single email, while a back-and-forth conversation involving multiple emails is a thread.

[9](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1462). Include the 'has:attachment' text in the string you pass to search().

[10](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1463). You will need the Twilio account SID number, the authentication token number, and your Twilio phone number.

### **Chapter 19**

[1](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1464). An RGBA value is a tuple of 4 integers, each ranging from 0 to 255. The four integers correspond to the amount of red, green, blue, and alpha (transparency) in the color.

[2](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1465). A function call to ImageColor.getcolor('CornflowerBlue', 'RGBA') will return (100, 149, 237, 255), the RGBA value for that color.

[3](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1466). A box tuple is a tuple value of four integers: the left-edge x-coordinate, the top-edge y-coordinate, the width, and the height, respectively.

[4](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1467). Image.open('zophie.png')

[5](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1468). imageObj.size is a tuple of two integers, the width and the height.

[6](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1469). imageObj.crop((0, 50, 50, 50)). Notice that you are passing a box tuple to crop(), not four separate integer arguments.

[7](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1470). Call the imageObj.save('new\_filename.png') method of the Image object.

[8](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1471). The ImageDraw module contains code to draw on images.

[9](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1472). ImageDraw objects have shape-drawing methods such as point(), line(), or rectangle(). They are returned by passing the Image object to the ImageDraw.Draw() function.

### **Chapter 20**

[1](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1473). Move the mouse to the upper-left corner of the screen, that is, the (0, 0) coordinates.

[2](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1474). pyautogui.size() returns a tuple with two integers, for the width and height of the screen.

[3](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1475). pyautogui.position() returns a tuple with two integers, for the x- and y-coordinates of the mouse cursor.

[4](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1476). The moveTo() function moves the mouse to absolute coordinates on the screen, while the move() function moves the mouse relative to the mouse’s current position.

[5](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1477). pyautogui.dragTo() and pyautogui.drag()

[6](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1478). pyautogui.typewrite('Hello, world!')

[7](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1479). Either pass a list of keyboard key strings to pyautogui.write() (such as 'left') or pass a single keyboard key string to pyautogui.press().

[8](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1480). pyautogui.screenshot('screenshot.png')

[9](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1481). pyautogui.PAUSE = 2

[10](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1482). You should use Selenium for controlling a web browser instead of PyAutoGUI.

[11](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1483). PyAutoGUI clicks and types blindly and cannot easily find out if it’s clicking and typing into the correct windows. Unexpected pop-up windows or errors can throw the script off track and require you to shut it down.

[12](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1484). Call the pyautogui.getWindowsWithTitle('Notepad') function.

[13](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-1485). Run w = pyatuogui.getWindowsWithTitle('Firefox'), and then run w.activate().