Automate The Boring Stuff With Python

This document will contain my answers to the questions in the book by Al Sweigart. Each chapter will have its own questions and answers section.

Chapter 1 <https://automatetheboringstuff.com/chapter1/>

Q1: What of the following are operators and which are values?

1. \* The Asterisk is an Operator
2. ‘hello’ this is a value, ‘hello’ Is in quotes so Is a string
3. -88.8 is a value, it is a negative floating-point number
4. -, /, + Are all operators, the – is to subtract, the / is to divide and the + is to add
5. 5 is a value of an integer that is 5

Q2: What of the following is a value and a string?

1. Spam is a variable, as it is not in quotes
2. ‘spam’ is a string as it is in quotes

Q3: Name three Data types:

1. Integer
2. String
3. Boolean

Q4: What is an expression made up of? What do they do?

An expression is a set of operators and values doing something together, for example you can concatenate ‘Alice’ and ‘Bob’

Q5: What is the difference between an expression and a statement?

An expression is a declaration of a variable, a statement tells the interpreter to do something.

Q6: What does the variable BACON contain after the code?

Bacon = 20

Bacon + 1

Therefore bacon = 21

Q7: what does the following evaluate to?

‘spam’ + ‘spamspam’ = ‘spamspamspam’

‘spam’ \* 3 = ‘spamspamspam’

Q8: Why is eggs a valid variable name and 100 isnt?

Python will interpret the eggs as a title to the variable or what its called.

100 will make the interpreter think that you are entering in an integer as it is a literal, literals cannot be changed.

Q9: What three functions can be used to get the integer, string or f-p number version of a value?

Int()

Str()

Float()

Q10: Why does this expression cause an error? What can you do to fix it?

‘I have eaten’ + 99 + ‘burritos’

It throws an error as you are trying to concatenate an integer with two strings, for this to work it would need to be written as:

‘I have eaten ’ + str(99) + ‘ burritos’

Chapter 2 <https://automatetheboringstuff.com/chapter2/> Flow Control

Q1: What are the two values of the Boolean data type?

True OR False

Q2: What are the three Boolean operators?

And, Or, Not

Q3: Write out the truth table for python:

True and True = True

True and False = False

False and True = True

False and False = False

|  |  |
| --- | --- |
| Q: | 4. What do the following expressions evaluate to? Answers after #  (5 > 4) and (3 == 5) # False  not (5 > 4) # False  (5 > 4) or (3 == 5) # True  not ((5 > 4) or (3 == 5)) # False  (True and True) and (True == False) #False  (not False) or (not True) # True |

Q5: What are the six comparison operators?

== Equal To

!= Not equal to

< Less Than

> Greater Than

<= LT or Equal To

>= GT or Equal To

|  |  |
| --- | --- |
| Q: | 6. What is the difference between the equal to operator and the assignment operator? |

The difference is that the equal to operator (==) Is DOUBLE equals signs, therefor this checks if the values are equal or not, the Assignment operator (=) is a SINGLE equals sign, this tells python that chicken = ‘egg’

|  |  |
| --- | --- |
| Q: | 7. Explain what a condition is and where you would use one. |

The most basic example is a password, you would need to check a password entered vs a stored one in the database, therefore an IF condition would be used for checking the PW VS the DB

8. Identify the three blocks in this code:

spam = 0

if spam == 10: # First Block

print('eggs')

if spam > 5:# Second Block

print('bacon')

else:# Third BLock

print('ham')

print('spam')

print('spam')

|  |  |
| --- | --- |
| Q: | 9. Write code that prints Hello if 1 is stored in spam, prints Howdy if 2 is stored in spam, and prints Greetings! if anything else is stored in spam. |

Find this code titled C2q9.py :

spam = 1

if spam == 1:

print('Hello')

elif spam == 2:

print('Howdy')

else:

print('Greetings!')

|  |  |
| --- | --- |
| Q: | 10. What can you press if your program is stuck in an infinite loop? |

CTRL + C

|  |  |
| --- | --- |
| Q: | 11. What is the difference between break and continue? |

Break will stop the code from running if the nested break is ever executed from a code block.

Continue will “Skip” over a piece of code from running and go to the next iteration.

|  |  |
| --- | --- |
| Q: | 12. What is the difference between range(10), range(0, 10), and range(0, 10, 1) in a for loop? |

Here is the syntax for range()

Range(start, stop step)

Only STOP is required, the start and step parameters are optional, step is defaulted to 1 If ignored.

|  |  |
| --- | --- |
| Q: | 13. Write a short program that prints the numbers 1 to 10 using a for loop. Then write an equivalent program that prints the numbers 1 to 10 using a while loop. |

for i in range(0, 10, 1):

print(i)

j = 1

while j < 11:

print(j)

j = j + 1

|  |  |
| --- | --- |
| Q: | 14. If you had a function named bacon() inside a module named spam, how would you call it after importing spam? |

Firstly you would need to import the module at the top of your program,

Import spam

Then you would need to call it:

spam.bacon()

Chapter 3 Functions: <https://automatetheboringstuff.com/chapter3/>

|  |
| --- |
|  |
| 1. Why are functions advantageous to have in your programs? |
|  |

They are a good way of compartmentalizing code, this way if there is ever a bug in your code you will be able to find and fix it without having to change all of your code if you’ve copied it over.

|  |
| --- |
|  |
| 2. When does the code in a function execute: when the function is defined or when the function is called? |
|  |

The code in a function is executed when it is called, as the defining of a function skips over the code and stores it for later usage, rather than running through it straight away.

|  |
| --- |
|  |
| 3. What statement creates a function? |
|  |

The DEF statement

|  |
| --- |
|  |
| 4. What is the difference between a function and a function call? |
|  |

A function call runs the function, a function in itself is a statement of operators and values to achieve something, a mini program.

|  |
| --- |
|  |
| 5. How many global scopes are there in a Python program? How many local scopes? |
|  |

Only as many as are defined, the programmer must name and change these global/local scopes when they are required to, the local scope will be deleted after it is run and the global will be remembered

|  |
| --- |
|  |
| 6. What happens to variables in a local scope when the function call returns? |
|  |

All the data in the variables is essentially deleted, this will remove all local variables from the local scope and it won’t be remembered the next time the program is run

|  |
| --- |
|  |
| 7. What is a return value? Can a return value be part of an expression? |
|  |

A return value is essentially a print statement, but returning an output based on the definition of an outcome in a function. Yes it can.

|  |
| --- |
|  |
| 8. If a function does not have a return statement, what is the return value of a call to that function? |
|  |

None

|  |
| --- |
|  |
| 9. How can you force a variable in a function to refer to the global variable? |
|  |

By using the global keyword before the variable

|  |
| --- |
|  |
| 10. What is the data type of None? |
|  |

None is a non-datatype it is essentially what It is called, nothing.

|  |
| --- |
|  |
| 11. What does the import areallyourpetsnamederic statement do? |
|  |

It imports the areallyourpetsnamederic module into the program

|  |
| --- |
|  |
| 12. If you had a function named bacon() in a module named spam, how would you call it after importing spam? |
|  |

I would write the following: spam.bacon()

|  |
| --- |
|  |
| 13. How can you prevent a program from crashing when it gets an error? |
|  |

You can use a try/except statement to catch out the bug, if the bug hits then it will run the return of your try/except, instead of breaking the code.

14. What goes in the try clause? What goes in the except clause?

The try clause will use a piece of code to try, the except clause will run an exception on a stated error message and not cause the code to break, rather stating the mentioned error message to the user.

See the collatz.py project in the root dir of this document, to see how recursion would work on a grander scale. For a collatz project.

**Chapter 4 Lists & Tuples** <https://automatetheboringstuff.com/chapter4/>

|  |
| --- |
| 1. What is []? This is an empty list, before having anything added to it |
| 2. How would you assign the value 'hello' as the third value in a list stored in a variable named spam? (Assume spam contains [2, 4, 6, 8, 10].) Answer:   spam.insert(2, ‘hello’)  For the following three questions, let’s say spam contains the list ['a', 'b', 'c', 'd']. |
| 3. What does spam[int(int('3' \* 2) // 11)] evaluate to? spam[int(int('3' \* 2) // 11)] = ‘d’ |
| 4. What does spam[-1] evaluate to? ‘d’ as well |
| 5. What does spam[:2] evaluate to? ‘a’, ‘b’  For the following three questions, let’s say bacon contains the list [3.14, 'cat', 11, 'cat', True]. |
| 6. What does bacon.index('cat') evaluate to? This will result in the second place in the index being referenced, as it is the first in the sequence. |
| 7. What does bacon.append(99) make the list value in bacon look like? It adds the integer 99 to the end of the list, after the True Boolean so [3.14, ‘cat’, ‘11’, ‘cat’, True, 99] |
| 8. What does bacon.remove('cat') make the list value in bacon look like? [3.14, 11, ‘cat’, True, 99] |
| 9. What are the operators for list concatenation and list replication? The operator for list concatenation is +, while the operator for replication is \*. (This is the same as for strings.) |
| 10. What is the difference between the append() and insert() list methods? .append() adds items at the end of the list, whereas insert() adds items at a specific point with kwargs. |
| 11. What are two ways to remove values from a list? Del statement and remove() list method are two ways to do this |
| 12. Name a few ways that list values are similar to string values. 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. What is the difference between lists and tuples? 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. How do you type the tuple value that has just the integer value 42 in it? Tuple((42,)) The trailing comma is mandatory for this. |
| 15. How can you get the tuple form of a list value? How can you get the list form of a tuple value? The tuple() and list() functions, respectively |
| 16. Variables that “contain” list values don’t actually contain lists directly. What do they contain instead? They contain references tot he instance oft he lists values. |
| 17. What is the difference between copy.copy() and copy.deepcopy()? 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 Dictionaries and Structuring Data** <https://automatetheboringstuff.com/chapter5/>

|  |  |
| --- | --- |
| Q: | 1. What does the code for an empty dictionary look like?  Dict = {} |
| Q: | 2. What does a dictionary value with a key 'foo' and a value 42 look like? {‘foo’: 42} |
| Q: | 3. What is the main difference between a dictionary and a list? A **list is** an ordered sequence **of** objects, whereas **dictionaries** are unordered sets. But the main **difference is** that items in **dictionaries** are accessed via keys and not via their position. ... The values **of** a **dictionary** can be any Python data type. |
| Q: | 4. What happens if you try to access spam['foo'] if spam is {'bar': 100}? You receive a KeyError as it does not exist in the dictionary. |
| Q: | 5. If a dictionary is stored in spam, what is the difference between the expressions 'cat' in spam and 'cat' in spam.keys()? nothing, they both evaluate to True, as |
| Q: | 6. If a dictionary is stored in spam, what is the difference between the expressions 'cat' in spam and 'cat' in spam.values()? True and False, respectively. As the ‘cat’ is a key. Not a value. |
| Q: | 7. What is a shortcut for the following code?  if 'color' not in spam:  spam['color'] = 'black'  spam.setdefault(‘color’, ‘black’) |
| Q: | 8. What module and function can be used to “pretty print” dictionary values?  You would need to import pprint at the top of the program then you would need to call it on a print function like so: pprint.pprint(spam) |

**Chapter 6 STRINGS** <https://automatetheboringstuff.com/chapter6/>

Lesson 20 Recap

- upper() & lower() return an upper/lower case string.

-isupper(), islower(), isaplha(), isalnum(), isdecimal(), isspace(), istitle() returns booleans if the string is that respective kind of string

-startswith() and endswith() also return bools

','.join() will return a string that combines the strings in a list ['cat', 'dog']

-'Hello world.'.split() will return a list of strings split from the string its calledon.

rjust(), ljust(), and center will just ify the content with padded spaces

strip(), rstrip() and lstrip() will return a string with the whitespace removed.

replace() will replace all occurences of the first string argument with the second string argument.

pyperclip has copy() and paste() functions for using the clipboard.

|  |  |
| --- | --- |
| Q: | 1. What are escape characters? \’ \” \t \r \\ \n are escape characters, they are used to stop the program from reading them as code but rather implement them in a string. |
| Q: | 2. What do the \n and \t escape characters represent? \n represents a new line and \t represents a Tab |
| Q: | 3. How can you put a \ backslash character in a string? by using the “\” twice: \\ |
| Q: | 4. The string value "Howl's Moving Castle" is a valid string. Why isn’t it a problem that the single quote character in the word Howl's isn’t escaped? Because we started the string with double quotes, instead of singles |
| Q: | 5. If you don’t want to put \n in your string, how can you write a string with newlines in it? By using the triple quotes, on both ends of a string. “”” SOMETHING “”” |
| Q: | 6. What do the following expressions evaluate to?   * 'Hello world!'[1]: ‘E’ * 'Hello world!'[0:5] ‘Hello’ * 'Hello world!'[:5]. ‘Hello’ * 'Hello world!'[3:]. ‘lo world!’ |
| Q: | 7. What do the following expressions evaluate to?   * 'Hello'.upper()== ‘HELLO’ * 'Hello'.upper().isupper()True * 'Hello'.upper().lower()’hello’ |
| Q: | 8. What do the following expressions evaluate to?   * 'Remember, remember, the fifth of November.'.split()It creates a LIST of the words in the string. * '-'.join('There can be only one.'.split()) ‘There-can-only-be-one’ |
| Q: | 9. What string methods can you use to right-justify, left-justify, and center a string? .rjust(), .ljust() .center() |
| Q: | 10. How can you trim whitespace characters from the beginning or end of a string? .rstrip() .strip() .lstrip() |

**Chapter 7: Regex** <https://automatetheboringstuff.com/2e/chapter7/>

Basics

Recap: RegExp are mini language for specifiying text patterns. Writing code to do pattern matching without regex is a pain.

Regex strings use backslashes \d so they are often raw strings r'\d'  
Need to import RE module first

Call the re.compile() function to create a regex object.

Call the regex object's search() method to create a match object (Save this to a var)

call the match object's group() method to get the matched string.

\d is the regex for a numerical digit character.

**A screenshot of a cell phone

Description automatically generated**

Recap on Regex Groups and Pipe Char:  
  
Groups are created in regex strings with parentheses.

the first set of parentheses is group 1, the second is 2, and so on.

Calling group() or group(0) returns the full matching string, group(1) returns group 1's matching string and so on.  
  
Use \( and \) to match literal parentheses in the regex string.

The | pipe can match one of many possible groups ( basically an OR operand)

Greedy matching is where a Regular Expression will try to match the maximum longest possible string. If we want to stop the greedy match, use the “?” after a curly brace.

So like this: digi = re.compile(r'(\d){3,5}?') will give you a specific 3 digit string.

Recap on RegEx converters

The ? Says the group matches zero or one times?  
The \* says the group matches zero or more times.  
The + says the group matches one or more times.  
The {} can match a specific number of times.  
the {X,Y} can match two numbers with a min and max value. X = Min y + Max  
Leaving out the first or second number in the curly braces says there is no minimum or maximum.  
Greedy matching match the longest string possible, nongreedy matching match the shortest string possible.  
Putting a question mark after the curly braces makes it do a non greedy match.

Recap:  
The regex method findall() is passed a string, and returns all matches in it, Not just the first match.  
If the regex has 0 or 1 group, findall returns a list of strings.  
If the regex has 2 or more groups, findall() returns a list of tuples of strings.  
\d is a shorthand character class that matches digits \w matches word chars, \s matches whitespaces  
The uppercase shorthand character classes \D,\W, \S match characters that are not digists words or spaces.  
You can make your own char classes with the square brackets []  
A ^ Caret makes it a negative char class matching anything not in the brackets [aeiou]

Further Regex Recap:  
^ Means the string must start with the pattern, $ means the string must end with the pattern. Both means the entire string must match the pattern.  
The . dot is a wildcard; it matches anything except newlines.  
Pass re.DOTALL as the second argument to re.compile() to make the dot match newlines as well.  
Pass re.I as the second argument to re.compile() to make the matching case-insensitive.

* The ? matches zero or one of the preceding group.
* The \* matches zero or more of the preceding group.
* The + matches one or more of the preceding group.
* The {n} matches exactly n of the preceding group.
* The {n,} matches n or more of the preceding group.
* The {,m} matches 0 to m of the preceding group.
* The {n,m} matches at least n and at most m of the preceding group.
* {n,m}? or \*? or +? performs a non-greedy match of the preceding group.
* ^spam means the string must begin with spam.
* spam$ means the string must end with spam.
* The . matches any character, except newline characters.
* \d, \w, and \s match a digit, word, or space character, respectively.
* \D, \W, and \S match anything except a digit, word, or space character, respectively.
* [abc] matches any character between the brackets (such as a, b, or c).
* [^abc] matches any character that isn’t between the brackets.