Wing Lau

Professor M. Waugh

Data Bootcamp

January 30th 2018

Code Practice 1

Question #1: What do these expressions do...

```
In [85]:
```

```
2+5 # This one adds
```

```
Out[85]:

7
```

```
In [86]:
```

```
2 + 5 # This one adds too, just more white space.
```

```
Out[86]:

7
```

```
In [87]:
```

```
2*5 # This is multiplication
```

```
Out[87]:

10
```

```
In [88]:
```

```
2**5 # This is taking 2 to the power of 5.
```

```
Out[88]:

32
```

Question #2: What do these expressions do...

```
In [89]:
```

```
x = 7
x = x + 3
x
```

```
Out[89]:

10
```

The value x should be 10. Why? The first line assigns the value 7 to the variable x. The second line then, starting on the right adds the value x (which is 7) to 3 and then reassigns the variable x the value 10.

Question #3: What is the value of y after running these statements in order? Of x? Why?

In [90]:

```
x = 3
y = x
x = 10
print(x)
print(y)
```

```
10
3
```

The value of y is three. Why? The logic of the program is to work line by line (not to automatically re-update). So the first line says, set x to three. Then the next line says set y equal to x, i.e. three. Then the next line redefines x, but the statement y = x is not executed again. Thus y stays as it was, then x equals ten.

Question #4: Does this code run without error? If so, what does it produce? If not, explain why.

In [91]:

```
x = 3
x = x/2
y = 'abc'
z = y + y
print(x,z)
```

```
1.5 abcabc
```

It runs without error. x is simply computing 3/2. The tricky part is z = y + y which is adding two strings. This is ok, since y is a string and will produce abcabc. The we print both, the comma separates the value.

Question #5: Does this code run without error? If so, what does it produce? If not, explain why.

In [92]:

```
x = 3
x = x/2
y = 'abc'
z = x + y
print(x,z)
```

```
---------------------------------------------------------------------------
-------
TypeError                                 Traceback (most recent cal
l last)
<ipython-input-92-c3b06c306749> in <module>()
      2 x = x/2
      3 y = 'abc'
----> 4 z = x + y
      5 print(x,z)

TypeError: unsupported operand type(s) for +: 'float' and 'str'
```

It runs with error. The issue is the command z = x + y, there we are trying to add two different types of variables. x is an float, y is a string. The plus function is only ok with two of the same types.

Question #6: Does this code run without error? If so, what does it produce? If not, explain why.

```
In [94]:
```

```
x = 3
y = 24
z = y / x
print(x, y, z, sep=' | ')
```

```
3 | 24 | 8.0
```

It runs without error. x,y,z are clearly defined, and their values are printed and separated by "|".

Question #7: Does this code run without error? If so, what does it produce? If not, explain why.

```
In [95]:
```

```
x = 3
y = '24'
z = y / x
print(x, z)
```

```
---------------------------------------------------------------------
-------
TypeError                                 Traceback (most recent cal
l last)
<ipython-input-95-41580f170617> in <module>()
      1 x = 3
      2 y = '24'
----> 3 z = y / x
      4 print(x, z)

TypeError: unsupported operand type(s) for /: 'str' and 'int'
```

It runs with error. The issue is with z = y / x. There we are tyring to add two different types of variables. x is an float, y is a string. The divide function only works with two of the same types.

Question #8: Does this code run without error? If so, what does it produce? If not, explain why.

```
In [97]:
```

```
x = "I am a #string" # Whoa, a string!
x
```

```
Out[97]:

'I am a #string'
```

It runs without error, because all characters in a string are classified as a single entity.

Question #9: Does this code run without error? If so, what does it produce? If not, explain why.

In [98]:

```
x = [1, 2, 3]
y = [42, 43]
z = x + y
print(z)
```

```
[1, 2, 3, 42, 43]
```

It runs without error. This is because x and y are defined as lists of numbers, and z is a function of x and y. The last line of code prints function z.

Questions #10: Does this code run without error? If so, what does it produce? If not, explain why.

In [99]:

```
x = [1, 2, 3]
y = 42
z = x + y
```

```
---------------------------------------------------------------------
-------
TypeError                                 Traceback (most recent cal
l last)
<ipython-input-99-6ce2df3c1c18> in <module>()
      1 x = [1, 2, 3]
      2 y = 42
----> 3 z = x + y

TypeError: can only concatenate list (not "int") to list
```

It runs with error, because x is a list, and y is a integer. The issue is with z = x + y, because it cannot add two different types of input.

Question #11: What "types" are...

In [100]:

```
x1 = 12
print(type(x1)) # returns 'integer'
```

```
<class 'int'>
```

In [101]:

```
x2 = 12.0
print(type(x2)) # returns 'float'
```

```
<class 'float'>
```

In [102]:

```
x3 = '12.0'
print(type(x3)) # returns 'string'
```

```
<class 'str'>
```

In [103]:

```
x4 = [12]
print(type(x4)) # returns 'list'
```

```
<class 'list'>
```

In [104]:

```
x5 = [12, 12.0, '12.0']
print(type(x5))# returns 'list'
```

```
<class 'list'>
```

Question #12: Explain the result of each line.

In [105]:

```
print(type(42)) # integer
print(type(42.0)) # float
print(type('42.0')) # string
print(type("42.0")) # string
print(type("""42.0""")) # string
print(type([1, 2])) # list
print(type([1] + [2])) # list
print(type(1 + 2)) # integer
print(type(print)) # built in function
```

```
<class 'int'>
<class 'float'>
<class 'str'>
<class 'str'>
<class 'str'>
<class 'list'>
<class 'list'>
<class 'int'>
<class 'builtin_function_or_method'>
```

The type function displays the data type of any object.

Question #13: Describe and explain the result of this statement:

In [106]:

```
type(float(str(int('1234'))))
```

Out[106]:

```
float
```

Because the float function transforms all functions before it into a float, so the final type that is displayed is a float.

Question #14: Describe and explain the result of this statement:

In [107]:

```
type(int(float('12.34')))
```

Out[107]:

```
int
```

The float value is transformed into an integer, and is rounded to the nearest whole number. But since we are determining the type of the function, it becomes an integer.

Question #15: Explain each line:

In [108]:

```
print(len([1234]))
```

```
1
```

As a list, it is printed as 1 object of a list, or 1 number.

In [109]:

```
print(len("1234"))
```

```
4
```

As a string, it can be counted as number of characters.

In [110]:

```
print(len(1234))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent cal
l last)
<ipython-input-110-c851b49e2b0b> in <module>()
----> 1 print(len(1234))

TypeError: object of type 'int' has no len()
```

Integers have no length, and so it cannot be counted.

Question #16: What are the type and length of x = []?

In [111]:

```
x = []
len(x)
```

Out[111]:

```
0
```

```
In [112]:
```

```
type(x)
```

```
Out[112]:
```

```
list
```

Question #17: Convert the string x = 'abcde' to a list. What does it look like?

```
In [113]:
```

```
x = ['abcde']
print(x)
```

```
['abcde']
```

Question #18: Consider the integer x = 1234.

a. Convert x to a floating point number.

```
In [114]:
```

```
x = 1234
x = float(x)
print(type(x))
```

```
<class 'float'>
```

```
In [115]:
```

```
x = 1234
x = str(x)
print(type(x))
```

```
<class 'str'>
```

```
In [116]:
```

```
x = [1 , 2 , 3 , 4]
x = list(x)
print(type(x))
```

```
<class 'list'>
```

Questions #19: How would you convert x to \title case" (rst letter of each word capitalized)? Hint: Use tab completion to find an appropriate method.

```
In [117]:
```

```
x = "luke, i am your father"
x = x.title()
x
```

```
Out[117]:
```

```
'Luke, I Am Your Father'
```

Questions #20: Consider the string

```
In [118]:
```

```
x = "How many characters and words are in this string?"
```

a. How many characters does x contain?

```
In [119]:
```

```
print(len(x))
```

```
49
```

b. Convert x to a list of individual characters.

```
In [120]:
```

```
print(list(x))
```

```
['H', 'o', 'w', ' ', 'm', 'a', 'n', 'y', ' ', 'c', 'h', 'a', 'r',
'a', 'c', 't', 'e', 'r', 's', ' ', 'a', 'n', 'd', ' ', 'w', 'o',
'r', 'd', 's', ' ', 'a', 'r', 'e', ' ', 'i', 'n', ' ', 't', 'h',
'i', 's', ' ', 's', 't', 'r', 'i', 'n', 'g', '?']
```

c. Convert x to a list of individual words. Hint: Use tab completion to find a method that splits x into pieces.

```
In [121]:
```

```
print(x.split())
```

```
['How', 'many', 'characters', 'and', 'words', 'are', 'in', 'this',
'string?']
```

d. How many words does x contain?

```
In [122]:
```

```
print(len(x.split()))
```

```
9
```