Untitled

March 29, 2018

In [1]:
```
# Types
    # Number
        # float, int
    # String - immutable
    # Bool - True or False
    # List
    # ---
    # Tuple - immutable
    # Dict
    # Set - unique list (no duplicates)

# Casting
    x = 3
    y = float(x) # 3.0
    z = str(3) # "3"

# Variables

# Operations
    # each type

# Loops
    # For
    # While

# Control Flow
    # if
    # elif
    # else

# Imports
    # import math
        # math.sqrt()
        # math.pi = 3.141592653...
```

In [3]:
```
# Create 2 variables of each type
integer1 = 4
integer2 = -6
```

```
float1 = 0.54
float2 = 7.4576

string1 = "red"
string2 = "yellow car"

bool1 = True
bool2 = False

list1 = [banana, apply, tree]
list2 = [fire, water, car]
```

In [39]:
```
# Operations
    # Numbers
        # +, -, *, / (division as expected) , // (integer division - rounds down), % -

# avg = Average of integer1 and integer2
avg = (integer1+integer2)/2
avg

# Knowing if something is even or odd - (even_number % 2 == 0) - (odd_number % 2 == 1)
7 % 2
8 % 2

    # Strings
        # + (concatenation)
string1
string2
string3 = string1 + " " + string2
string3
        # Methods
            # .isupper(), .islower(), .isdigit(), len()
            # len() - length
len(string1)
string4 = "5"
string4.isdigit() #isdigit() returns a boolean

# indexing - string2[index]
    # Slice (slicing)
        # string2[start_index : end_index] -> gives you indexes from start_index to end
# string2 = "yellow car"
# string5 = "yellow"
print(len(string2))
string5 = string2[0:6]
print(string5)

# string1 = "red"
```

```
        # string2 = "yellow car"
        # string6 = "red car"
        string6 = string1[0:3] + " " + string2[7:10]
        # iter[:] == iter - gives you everything inside of iter
            # iter[:end_index] == iter[0:end_index]
            # iter[start_index:] == iter[start_index : -1] == iter[start_index : len(iter)]

        string7 = string1 + " " + string2[7:10]
        print(string6)
        string8 = 'hello'
        print(string8[-1])
        print(string8[-2])
        print(string8[-3:])
```

```
10
yellow
red car
o
l
llo
```

```
In [34]:
```

```
a = [1, 2, 3]
b = [1, 2, 3]
a = [4, 5, 6]
b = [1, 2, 3]
```

```
In [50]: list1 = ["banana", "apply", "tree"]
         list2 = ["fire", "water", "car"]

         # Adding to list
             # insert(index_to_insert_at, value_to_insert) - , append() - add to the end of list

         # Add "forest" to end of list 1
         list1.append("forest")
         #print(list1)
         # Add "earth" as 2nd element of list2
         list2.insert(1, "earth")
         #print(list2)
         # Add "air" as 3rd index of list2
         list2.insert(3,"air")
         #print(list2)

         # Removing from list
             # del - not actually a list method, its more general, to delete anything
             # del xyz
```

```
            # del list[0]
            # del thing_to_delete, remove(value_in_list)

            # 2 Ways to remove things from list
                # remove an index from a list
                # remove an element from a list

                # list.remove(element) - delete an element - list1.remove("banana")
                # list.pop(index) - delete index from list - list1.pop(0)

        list1 = ['banana', 'apply', 'tree', 'forest']
        list2 = ['fire', 'earth', 'water', 'air', 'car']
        # remove banana from list1 using remove()
        list1.remove("banana")
        print(list1)
        # remove water from list2 using pop()
        list2.pop(2)
        print(list2)

        x = 3.14 # x is a float
        string_x = str(x) # string_x is a string - "Cast x to a string"

        #print("descripting thing " + str(x)) # string + float -> BAD! ... string + string -> G

        def add_one_to_input():
            ui = input("Enter your favorite integer: ") # input() ALWAYS returns a string
            number = int(ui)
            print(number + 1)
        add_one_to_input()
```

```
['apply', 'tree', 'forest']
['fire', 'earth', 'air', 'car']
Enter your favorite integer: 7
8
```

```
In [57]: # Loops
            # For loops, and While loops
                # For - specified number of iterations
                # While - unknown number of iterations - dependent on some condition

            # For Loop has 2 main approaches
                # for index in range(start, end, step=1 by default): # from start up to end-1
                # for xyz in iterable (list, string, ... things that can be indexed):

        # Valid uses of range()
            # range(start, end, step)
            # range(start, end) -> assumes step=1
```

```python
    # range(end) -> assumes start=0
for i in range(10): # 0 1 2 ... 9
    print(i)

print()
for i in range(0, 10): # 0 1 2 ... 9
    print(i)

print()
for i in range(5, 10): # 5 6 7 ... 9
    print(i)

print()
for i in range(10, 2, -1): # assumes step=1 # specify step=-1 to go backwards
    print(i)

print()
for i in range(0, 11, 2): # 0 2 4 ... 10
    print(i)
```

```
0
1
2
3
4
5
6
7
8
9

0
1
2
3
4
5
6
7
8
9

5
6
7
8
9
```

```
10
9
8
7
6
5
4
3

0
2
4
6
8
10
```

```
In [58]: # Second way to do for loops
             # for element(can be any variable name) in iterable:
                 # do something with element

         list1 = [1, 2, 3]

         for number in list1: #for i in range(1, 4):
             print(number)
```

```
1
2
3
```

```
In [59]: list2 = ["hello world", [1,2,3], 7]

         for element in list2:
```

```
hello world
[1, 2, 3]
7
```

```
In [63]: # Nested for loops
             # for loop inside of a for loop

         nested_list = [[1,2], [4,5], [7,8]] # List of 3 lists - each list contains 3 numbers

         outer_iteration = 1
         for sublist in nested_list: # has 3 iterations
             print("Outer iteration: " + str(outer_iteration))
```

```
            inner_iteration = 1
            for number in sublist: # has 2 iterations for each outer iteration
                #print("List number: " + str(number))
                print("Inner iteration: " + str(inner_iteration)) # same as print("Inner iterat
                inner_iteration += 1
            outer_iteration += 1
```

```
Outer iteration: 1
Inner iteration: 1
Inner iteration: 2
Outer iteration: 2
Inner iteration: 1
Inner iteration: 2
Outer iteration: 3
Inner iteration: 1
Inner iteration: 2
```

```
In [70]: matrix = [['First Name', 'Last Name'], ['Ryan', 'McCormick'], ['Mr. Python', 'Rocks']]

         for line in matrix:
             for entry in line:
                 print(entry)
```

```
First Name
Last Name
Ryan
McCormick
Mr. Python
Rocks
```

```
In [74]: numbers = [[1,2,3], [4,5,6]]

         # Write some code to print out (1+4), (2+5), (3+6) - with indexing

         print(numbers)
         # print out first list
         print (numbers[0])
         # print out second list
         print (numbers[1])

         sublist1 = numbers[0]
         sublist2 = numbers[1]
         # print first number of sublist1
         print(sublist1[0])

         # print first number of sublist2
         print(sublist2[0])
```

```
[[1, 2, 3], [4, 5, 6]]
[1, 2, 3]
[4, 5, 6]
1
4
0
1
```

```
In [91]: # for loop to print all numbers in sublist1 using range()
         # range(start, end) -> goes over each number from start to end-1
         for position in range(0, len(sublist1)): # index = 0 1 2
             print(sublist1[position]) #sublist[0], sublist[1], sublist[2]

         print()
         # for loop to print all numbers in sublist1 without range()
         for element in sublist2:
             print(element)

         print()
         # Intermediate Python - enumerate() - ONLY FOR FUN
         for index, number in enumerate(sublist2):
             print(index, number)
```

```
1
2
3

4
5
6

0 4
1 5
2 6
```

```
In [92]: list1 = [1,2,3]
         list2 = [4,5,6]
         # Print out (1 + 4), (2 + 5), (3 + 6) with a for loop
         for index in range(0,len(sublist1)):
             print(sublist1[index] + sublist2[index])
```

```
5
7
9
```