Tutorial 2

Activity IV

Int(Float_Number) - rounds down a float to the nearest integer.

Float(int_number) – adds a .0 to the end of an integer to make a float.

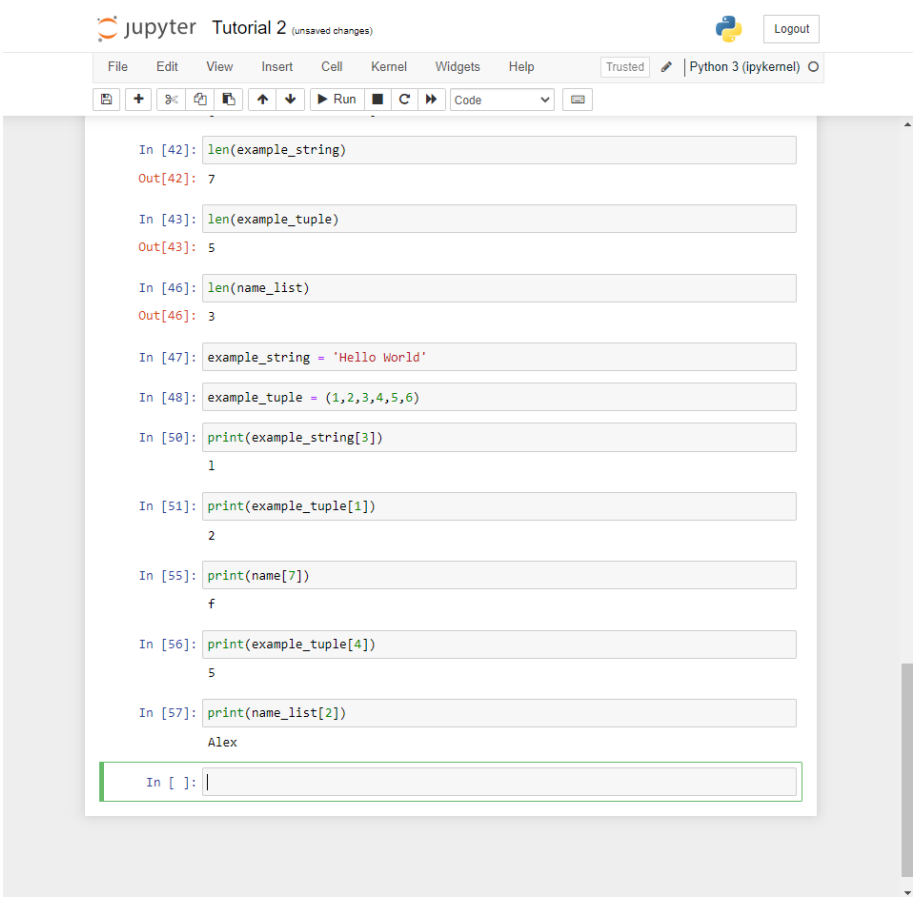
Int('2.0') – spits an error since the int function can only do one task at a time, it can't convert the string to a float then turn that into an int (do int(float('2.0)) to convert the string to a float, then turn the float to an int).

Int('2') – turns the string into an int.

Float('2.0') - turns the string into a float.

Str(1) – turns an int or float into a string.

Activity VII



TASK 1:

Variable Name	Data Type	Length	Description
Variable 1	String	13	A simple string
Variable 2	List (String)	3	A string list with 3 items
Variable3	Tuple (Int)	4	A Tuple with 4 ints
Variable4	List(Float)	5	A List of 5 floats
Variable5	Dictionary(String)	3	A Dictionary with 3 string keys and 3 int outputs
Variable6	Dictionary(String)	3	A Dictionary with 3 string keys and 3 float outputs
Variable7	Dictionary(Int)	3	A Dictionary with 3 int keys and 3 string outputs
Variable8	Dictionary(Int)	3	A Dictionary with 3 int keys and 3 string lists as outputs(lists have 3 strings)

Activity XII

Variable 1	Variable 2	Operation	Result	Comment
3 (int)	4 (int)	+	7 (int)	Add two ints together to get an int
4 (int)	3 (int)	-	1 (int)	subtracts two ints together to get an int
2 (int)	3 (int)	*	6 (int)	multiple two ints together to get an int
8 (int)	2 (int)	/	2 (int)	Divides two ints together to get an int
2 (int)	7 (int)	%	1 (int)	Finds the remainder of two ints and returns an int
2 (int)	2 (int)	**	4 (int)	Takes one int and puts it to

				the power of another int and returns an int

Activity XI

I: When you add two strings it combines them into one string.

II: Adding an integer to string normally would just add that integer onto the string like adding two strings. The proper way would be to use str() on the integer.

III: When you add two list, there merge together like adding two strings would.

Activity XII

Variable 1	Variable 2	Operation	Result	Comment
2 (int)	3 (int)	<	True (Bool)	2 is less than 3, so it returns true
2 (int)	3 (int)	>	False (Bool)	2 is not greater than 3, so it returns false
4 (int)	5 (int)	<=	True (Bool)	4 is less or equal than 5, so it returns true
4 (int)	4 (int)	>=	True (Bool)	4 is greater than or equal than 4, so it returns true
5 (int)	4 (int)	==	False (Bool)	5 does not equal 4, so it returns false
5 (int)	4 (int)	!=	True (Bool)	5 does not equal 4, so it returns true

Activity XIV

```
In [81]: if name == "Alex Hoffman":
             print("Name is Correct")

         Name is Correct
```

Activity XV

```
In [86]: if name == "Joey Bob":
             print("Name is Joey")
         elif name == "Bob the Builder":
             print("Name is Bob")
         else:
             print("Name Not Found")

         Name Not Found
```

Activity XVI

```
In [88]: for each_iteration in [0,1,2,3]:
             print(name[each_iteration])
             if (name[each_iteration] == ""):
                 break

         A
         l
         e
         x
```

Activity XVII

```
In [89]: c = 0
         while (c < 12):
             print(name[c])
             c = c + 1

         A
         l
         e
         x

         H
         o
         f
         f
         m
         a
         n
```

TASK II

Example 1:

```
In [90]: def print_name():
             print('My name is ' + name)
             return
```

```
In [91]: print_name()

         My name is Alex Hoffman
```

Example 2

```
In [102]: def find_hypotenuse(side_a, side_b):
              import numpy as np
              hypotenuse = np.sqrt((side_a**2) + (side_b**2))
              return hypotenuse

In [99]: triangle_side_a = 3.0

In [100]: triangle_side_b = 4.0

In [104]: hypotenuse = find_hypotenuse(triangle_side_a, triangle_side_b)

In [105]: print(hypotenuse)

          5.0
```

Example 3:

```
In [107]: for n in range(10):
              print_name()

My name is Alex Hoffman
My name is Alex Hoffman
My name is Alex Hoffman
My name is Alex Hoffman
My name is Alex Hoffman
My name is Alex Hoffman
My name is Alex Hoffman
My name is Alex Hoffman
My name is Alex Hoffman
My name is Alex Hoffman
```

```
def find_area_of_triangle(base, height):
    area = (base * height) / 2.0
    return area
```

```
def compute_triangle_properties(sA, sB):
    hyp = find_hypotenuse(sA, sB)
    A = find_area_of_triangle(sA, sB)
    return hyp, A
```

```
hypotenuse, tri_area = compute_triangle_properties(triangle_a, triangle_b)
```

```
print(hypotenuse)
```

```
5.0
```

```
print(tri_area)
```

```
6.0
```

TASK 3

A

```
In [129]: def find_eucildean_distance(x, y):
              e_distance = 0
              base1 = 0
              base2= 0
              if (x[0] < y[0]):
                  base1 = y[0] - x[0]
              else:
                  base1 = x[0] - y[0]

              if (x[1] < y[1]):
                  base1 = y[1] - x[1]
              else:
                  base2 = x[1] - y[1]

              e_distance = find_hypotenuse(base1, base2)

              return e_distance
```

```
In [116]: point_a = (1,2)
```

```
In [117]: point_b = (2,4)
```

```
In [124]: distance = find_eucildean_distance(point_a, point_b)
```

```
In [130]: print(distance)
```

```
2.0
```

B

```
In [136]: def find_circle_area(r):
              import numpy as np
              c_area = 0.0
              c_area = (r**2) * np.pi
              return c_area
```

```
In [132]: radius = 5.0
```

```
In [137]: circleArea = find_circle_area(radius)
```

```
In [138]: print(circleArea)
```

```
78.53981633974483
```

C

In [186]:
```
def find_square_of_dict_values (myDict):
    myDict_square = {}
    l = len(myDict)
    c = 0
    my_list = ('one', 'two', 'three','four','five')
    for i in range(1, l+1):
        myDict[my_list[c]] = i**2
        c = c+1
        print(myDict)
    myDict_square.update(myDict)
    return myDict_square
```

In [189]:
```
myDict = {'one' : 1, 'two' : 2, 'three' : 3, 'four' : 4, 'five' : 5}
```

In [190]:
```
myDict_square = find_square_of_dict_values(myDict)
```

```
{'one': 1, 'two': 2, 'three': 3, 'four': 4, 'five': 5}
{'one': 1, 'two': 4, 'three': 3, 'four': 4, 'five': 5}
{'one': 1, 'two': 4, 'three': 9, 'four': 4, 'five': 5}
{'one': 1, 'two': 4, 'three': 9, 'four': 16, 'five': 5}
{'one': 1, 'two': 4, 'three': 9, 'four': 16, 'five': 25}
```

In [191]:
```
print(myDict_square)
```

```
{'one': 1, 'two': 4, 'three': 9, 'four': 16, 'five': 25}
```