\*\* Answer the following questions \*\*

Write a brief description of all the following Object Types and Data Structures we've learned about:

Numbers: Which stores Numerical values like whole numbers and floating point values

​

Strings: Ordered sequence of characters

​

Lists: ordered sequence of objects (Mutable = Once assigned Elements can be changed)

​

Tuples: Ordered sequence of objects (Immutable = Once assigned Elements can not be changed)

​

Dictionaries: Key value pairing that is unordered.

​----------

## Numbers

​

Write an equation that uses multiplication, division, an exponent, addition, and subtraction that is equal to 100.25.

​

Hint: This is just to test your memory of the basic arithmetic commands, work backwards from 100.25

(60 + (10 \*\* 2)/4\*7) - 134.75

100.25

Answer these 3 questions without typing code. Then type code to check your answer.

​

What is the value of the expression 4 \* (6 + 5)

What is the value of the expression 4 \* 6 + 5

What is the value of the expression 4 + 6 \* 5

x = 4 \* (6 + 5)

​

y = 4 \* 6 + 5

​

z = 4 + 6 \* 5

​

print(x,y,z)

44 29 34

---------

What is the \*type\* of the result of the expression 3 + 1.5 + 4?<br><br>

​FLOAT

--------

What would you use to find a number’s square root, as well as its square?

# Square root:

36 \*\* 0.5

6.0

------

# Square:

10 \*\* 3

1000

---------

## Strings

Given the string 'hello' give an index command that returns 'e'. Enter your code in the cell below:

s = 'hello'

# Print out 'e' using indexing

​

s[1]

'e'

Reverse the string 'hello' using slicing:

s ='hello'

# Reverse the string using slicing

​s[::-1]

'olleh'

---------

Given the string hello, give two methods of producing the letter 'o' using indexing.

s ='hello'

# Print out the 'o'

​

# Method 1:

​

print(s[4])

​

o

--------

# Method 2:

​

print(s[-1])

o

------

## Lists

Build this list [0,0,0] two separate ways.

list

# Method 1:

list = [0,0,0]

list

[0, 0, 0]

3

# Method 2:

[0]\*3

[0, 0, 0]

Reassign 'hello' in this nested list to say 'goodbye' instead:

list3 = [1,2,[3,4,'hello']]

​

list3[::] #this prints the entire list

list3[2][2] = "goodbye"

list3

[1, 2, [3, 4, 'goodbye']]

----------

Sort the list below:

list4 = [5,3,4,6,1]

​

# list4[::] prints entire string

​

# sorted(list4) one way of sorting

​

list4.sort()

list4

[1, 3, 4, 5, 6]

--------

## Dictionaries

Using keys and indexing, grab the 'hello' from the following dictionaries:

simple\_key

d = {'simple\_key':'hello'}

# Grab 'hello'

d['simple\_key']

'hello'

'

d = {'k1':{'k2':'hello'}}

# Grab 'hello'

d['k1']['k2']

'hello'

0

# Getting a little tricker

d = {'k1':[{'nest\_key':['this is deep',['hello']]}]}

#Grab hello

d['k1'][0]['nest\_key'][1][0]

'hello'

0

# This will be hard and annoying!

d = {'k1':[1,2,{'k2':['this is tricky',{'tough':[1,2,['hello']]}]}]}

​

d['k1'][2]['k2'][1]['tough'][2][0]

'hello'

--------

Can you sort a dictionary? Why or why not?

No! Normal dictionaries are just for mapping not a sequence

----------

Tuples

What is the major difference between tuples and lists?<br><br>

​Tuples are Immutable means -> They can not be changed

#How do you create a tuple?<br><br>

​t = (1,2,3)

t

(1, 2, 3)

-----------

Sets

What is unique about a set?<br><br>

They don't allow duplicate values

------

Use a set to find the unique values of the list below:

list5 = [1,2,2,33,4,4,11,22,3,3,2]

set(list5)

{1, 2, 3, 4, 11, 22, 33}

------

Booleans

For the following quiz questions, we will get a preview of comparison operators. In the table below, a=3 and b=4.

Operator Description Example

== If the values of two operands are equal, then the condition becomes true. (a == b) is not true.

!= If values of two operands are not equal, then condition becomes true. (a != b) is true.

> If the value of left operand is greater than the value of right operand, then condition becomes true. (a > b) is not true.

< If the value of left operand is less than the value of right operand, then condition becomes true. (a < b) is true.

>= If the value of left operand is greater than or equal to the value of right operand, then condition becomes true. (a >= b) is not true.

<= If the value of left operand is less than or equal to the value of right operand, then condition becomes true. (a <= b) is true.

What will be the resulting Boolean of the following pieces of code (answer fist then check by typing it in!)

# Answer before running cell (False)

2 > 3

False

alse

# Answer before running cell (False)

3 <= 2

False

lse

# Answer before running cell (False)

3 == 2.0

False

True

# Answer before running cell (True)

3.0 == 3

True

False

# Answer before running cell (False)

4\*\*0.5 != 2

False

Final Question: What is the boolean output of the cell block below?

# two nested lists

l\_one = [1,2,[3,4]]

l\_two = [1,2,{'k1':4}]

​

# True or False?

l\_one[2][0] >= l\_two[2]['k1']

False

Great Job on your first assessment!