**Test your knowledge**

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

**Write (or say out loud to yourself) a brief description of all the following Object Types and Data Structures we've learned about. You can edit the cell below by double-clicking on it. This is to test if you know the difference between these, so feel free to consider it since your answers are self-graded.**

**Numbers:** *Numbers are the sequence of numeric values that are classified into two categories:*

1. *Integers: All numeric values that are integers are covered under this category.*
2. *Floating Point Numbers: All numeric values with a decimal point in it is considered under this category.*

**Strings:** *The ordered sequence of characters is defined under the category Strings.*

**Lists:** *Ordered sequence of objects or elements which are mutable.*

**Tuples:** *Ordered sequence of objects or elements which are immutable.*

**Dictionaries:** *Mappings of objects or elements which is unordered and can be called by keys given to each object.*

**Numbers**

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

*Ans:- ((5\*5-5)\*\*2+1)/4*

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

**Answer these three questions without typing code. Then type the code to check your answer.**

**What is the value of the expression 4 \* (6 + 5)** *:- 44*

**What is the value of the expression 4 \* 6 + 5***:- 29*

**What is the value of the expression 4 + 6 \* 5** *:- 34*

**What is the *type* of the result of the expression 3 + 1.5 + 4?**

*Ans:- Floating Point Number*

**What would you use to find a number’s square root and its square?**

**# Square root:***100\*\*(1/2)*

**# Square:***10\*\*2*

**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**

*Ans:- s[1]*

**Reverse the string 'hello' using slicing**

**s ='hello'**

**# Reverse the string using slicing**

*Ans:- s[::-1]*

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

**s ='hello'**

**# Print out the 'o'**

**# Method 1:** *s[4]*

**# Method 2:** *s[4:] or s[-1]*

**Lists**

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

**# Method 1:** *list1=[0,0,0] or [0]\*3*

**# Method 2:** *list1=[]*

*list1.append(0)*

*list1.append(0)*

*list1.append(0)*

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

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

*list3[2][2]=’goodbye’*

**Sort the list below:**

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

*“list4.sort()” or “sorted(list4)”*

**Dictionaries**

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

**# Grab 'hello.'**

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

*Ans: d[simple\_key]*

**# Grab 'hello.'**

**d = {'k1’: {'k2':'hello'}}**

*Ans: d[k1][k2]*

**# Grab 'hello.'**

**# Getting a little tricker**

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

*Ans: d[k1][0][nest\_key][1][0]*

**#Grab hello**

**# This will be hard and annoying!**

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

*Ans: d[k1][2][k2][1][tough][2][0]*

**Can you sort a dictionary? Why or why not?**

*Ans: No, because dictionaries are unordered way of storing elements which are called by keys assigned to each value. In other words, Dictionaries are mappings not a sequence.*

**Tuples**

**What is the significant difference between tuples and lists?**

*Ans: A tuples is very similar to the list except that the “tuple is immutable”.*

**How do you create a tuple?**

*Ans: A tuple is crated by parenthesis not square brackets as it will lead to a list. For example t=(1,2,33,55,76). Here, t is a tuple.*

**Sets**

**What is unique about a set?**

*Ans: A set is a collection of unordered unique elements that is stored in it.*

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

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

*Ans: set(list5)*

*Output that is going to come out is: {1,2,33,4,11,22,3}*

**Booleans**

**We will get a preview of comparison operators for the following quiz questions. 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 the values of two operands are not equal, then the condition becomes true.** | **(a != b) is true.** |
| **>** | **If the value of the left operand is greater than the value of the right, then the condition becomes true.** | **(a > b) is not true.** |
| **<** | **If the value of the left operand is less than that of the right, then the condition becomes true.** | **(a < b) is true.** |
| **>=** | **If the value of the left operand is greater than or equal to the value of the right, then the condition becomes true.** | **(a >= b) is not true.** |
| **<=** | **If the value of the left operand is less than or equal to the value of the right, then the condition becomes true.** | **(a <= b) is true.** |

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

**# Answer before running the cell**

**2 > 3:** *False*

**# Answer before running the cell**

**3 <= 2***: False*

**# Answer before running the cell**

**3 == 2.0:** *False*

**# Answer before running the cell**

**3.0 == 3** *:True*

**# Answer before running the cell**

**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']**

*Ans: False*