



Chapter 1 and 2 Quiz:

- Write an expression that equals to 200. For example, `100 + 100`. Trying using more than one operator and submit only one line of code.

```
100+150-100/2
```

- My phone bill for the last three months has been \$35, \$40, and \$54. What is the average monthly electricity bill over the three-month period? Write an expression to calculate the mean and use `print ()` to view the result.

```
# store first bill in bill1,second bill in bill2 and third in bill3
```

```
bill1= 35
```

```
bill2= 40
```

```
bill3= 54
```

```
total = (bill1 +bill2 +bill3 )
```

```
#write equation to store the calulated avg
```

```
avg = total/3.0
```

```
print(avg)
```

- Which of these lines of Python code are well formatted? How would you improve the readability of the codes that don't use good formatting? (Choose all that apply)

```
print(((3+ 32))+ -15//2)
```

```
print((17 - 6)%(5 + 2))
```

```
print ((1 + 2 + 4) / 13)
```

```
print(4/2 - 7*7)
```

Ans:

line no. 3 and 4 are well formatted

Readability of line one and four can be improved as follows:

```
print (((3+32) + (-15))/2)
```

```
print ((4/2)-(7*7))
```

- Now it is your turn to work with variables. The comments in this quiz (the lines that begin with #) have instructions for creating and modifying variables. After each comment write a line of code that implements the instruction.

Note that this code uses scientific notations to define large numbers. `4.445e8` is equal to `4.445 * 10 ** 8` which is equal to `444500000.0`.

```
# The current volume of a water reservoir (in cubic metres)
```

```
>>>vol_res =4.445**(8)
```

```
or >>> math.pow((4.445** (8))
```

- `reservoir_volume = 4.445e8`

```
# The amount of rainfall from a storm (in cubic metres)
```

- `rainfall = 5e6`

```
rainfall=5**6
```

```
# decrease the rainfall variable by 10% to account for runoff
```

```
rainfall= float rainfall-(rainfall*(1/10))
```

```
# add the rainfall variable to the reservoir_volume variable
```

```
# increase reservoir_volume by 5% to account for stormwater that flows
```

```
# into the reservoir in the days following the storm
```

```
# decrease reservoir_volume by 5% to account for evaporation
```

```
# subtract 2.5e5 cubic metres from reservoir_volume to account for water
```

```
# that's piped to arid regions.
```

```
# print the new value of the reservoir_volume variable
```

- How does changing the value of a variable affect another variable that was defined in terms of it? Let's look at an example.

We're intentionally not providing a place to execute the code here, because we want to help you practice the important skill of walking through lines of code by hand.

Each line of code executes in order, one at a time, with control going from one line to the next.

```
>>> carrots = 24
>>> rabbits = 8
>>> crs_per_rab = carrots/rabbits
```

Now we add a new 4th line to this code, that assigns a new value to the `rabbits` variable:

```
>>> rabbits = 12
```

If we now add this new 5th line of code to the above, what will the output be?

```
>>> print(crs_per_rab)
```

- 0.5
- 2.0
- 3.0
- None of the above
- 1.5

- In Python 3 what is the output of $\frac{1}{2}$?

.5

- Guess the correct output of following code.

```
str1 = "PYnative"
```

```
print(str1[1:4], str1[:5], str1[4:], str1[0:-1], str1[: -1])
```

PYn PYnat ive PYnativ vitanYP

Yna PYnat tive PYnativ vitanYP

Yna PYnat tive PYnativ PYnativ

Correct Output:Yna PYnat tive PYnativ vitanYP

- Python does not support a character type; a single character is treated as strings of length one.

True

- List all the python string methods with their definition. You can refer the internet for this.

1)isupper() Returns True if all characters in the string areupper case

2)join(())joins the elements of an iterable to the end of the string

3)ljust() Returns a left justified version of string

4)lower()

5)capitalize()

6)casefold()

7) center()

8) count()

9)encode()

10)endswith()

11)expandtabs()

12)find()

13)format

14) format_map()

- Write one line of code giving an example of indexing. For example 'Hello World'[0] returns 'H'. Make any two examples of your own one with normal indexing and other one with reverse indexing.

```
word='name'
```

```
print(word[0])
```

```
print(word[-2])
```

- Reverse this 'This is an easy quiz' sentence without using string reverse method.

Method1

```
List2=['This','is','an','easy','quiz']
```

```
List2[::-1]
```

Method2

```
List2=['This','is','an','easy',quiz']
```

```
List2.index.[-4:0]
```

```
print( List2.index)(-4:0))
```

Are strings immutable?

Yes.

- Use slicing techniques with providing 4 different examples each one using different slicing techniques learned in the video.
- Change the name of the two strings from 'Hero' to 'Zero' using the slicing and string concatenation method. Use least lines of code as you can.
- Write two expressions using format() method and f-string method.
- Write code to compare these densities. Is the population of San Francisco more dense than that of Rio de Janeiro? Print `True` if it is and `False` if not.

```
sf_population, sf_area = 864816, 231.89  
rio_population, rio_area = 6453682, 486.5
```

```
san_francisco_pop_density = sf_population/sf_area  
rio_de_janeiro_pop_density = rio_population/rio_area
```

```
# Write code that prints True if San Francisco is denser than Rio, and False otherwise
```

```
->
```

- Give examples of all the datatypes you learned.

Numeric type

Int 5

float 5.89

Complex number $x+yj$ x and y may be float j may be a

Boolean Type

True,False

Sequence Type

String

'name' "name2"

List

```
[ 2,3,5,'world']
```

Tuple

```
my_tuple = (1, "Hello", 3.4)
```

Dictionary

```
city_popln = {  
'Shanghai' : 17.8,  
'Istanbul' : 13.3,  
'Karachi' : 13.0,  
'Mumbai' : 12.5  
}
```

- What function do you use to find length of a string?
`len()`
- What type does this object have? `"hippo" * 12`
`str`
- Create an example of list, with mixed data types. The answer should be just one line of code.
- `list_mixed = [1,'myname',3.5]`
- If `list1 = [1,2,3,4,5]`. What is the output of `list1.pop ()`? And what would be the result of `list1[1:]`?
`5`
`[2, 3, 4, 5]`
- Create a Dictionary where all the keys are strings and values are integers.

```
city_popln = {  
'Shanghai' : 17.8,  
'Istanbul' : 13.3,  
'Karachi' : 13.0,  
'Mumbai' : 12.5  
}
```
- Create a dictionary within a dictionary and write the code to find the values in the inside dictionary.

```
people = {1: {'name': 'John', 'age': '27', 'sex': 'Male'},  
          2: {'name': 'Marie', 'age': '22', 'sex': 'Female'}}
```
- Do dictionaries retain order and are they a sequence?

Python Dictionaries are Now Ordered. Keep Using **OrderedDict**. ... If you wanted a dictionary that preserved order, you could use the **OrderedDict class** in the standard library module `collections`. Dictionaries are not sequence

- Given `d = { 'k1':[1,2,3]}`. What is the output of `d['k1'][3]`?
list index out of range
- Are dictionaries immutable?
No
- Do tuples have lots of methods associated with them?
Yes
- When is the best time to use tuples over lists?
When iteration is required Use Tuple. Iteration in a tuple is faster as compared to lists since tuples in Python are immutable.

Tuples are generally used for different Python Data Types; whereas, lists are used for similar data types.

- Are tuples immutable?
Tuples are immutable, but their values may change. This may happen when a Tuple holds a reference to any mutable object, such as a List. This means values contained in the Tuple can not change, while Data contained in the List object can change.
- Which of the following is a tuple?
- [1,2,3]
- (1,2,[1,2,3]) is a Tuple
- {1,2,4,}
- None of the above?
- Write an expression to turn the string 'Mississippi' into a set of unique characters.
`my_set {'M','i','s','p'}`
- {1,2,3,4} Is this a set?
No, It is a Tuple
- What method do you use to add an element to a string?

`.join()`

Example:

`strings = ['do', 're', 'mi']`

`','.join(strings)`

- What is the result of: `set ([1,2,2,3])`

- An error
- `[1,2,3]`
- `{1,2,2,3}`
- `{1,2,3}`

- What is the output of following lines of code? What would the length be?

```
a = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4]
b = set(a)
print(len(a) - len(b))
```

6

- Consider:

```
a = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4]
b = set(a)
b.add(5)
b.pop()
```

After executing this code, will the number 5 be a part of the set `b`?

No

- Define a Dictionary, population,

```
# that provides information
# on the world's largest cities.
# The key is the name of a city
# (a string), and the associated
# value is its population in
# millions of people.
```

```
# Key | Value
# Shanghai | 17.8
# Istanbul | 13.3
# Karachi | 13.0
# Mumbai | 12.5
city_popln = {
'Shanghai' : 17.8,
'Istanbul' : 13.3,
'Karachi' : 13.0,
'Mumbai' : 12.5
}
print(city_popln)
```

You can always find more questions online and try to attempt those too. I tried keeping it basic and less questions. But try finding more quizzes online and try to solve those!

---X---