A picture containing drawing

Description automatically generated

**Chapter 1 and 2 Quiz:**

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

A = 100, A + A

1. 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.

Print(127/3)

1. 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)

Simply the equation to equal the object just not as cluttered together

1. 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)

* reservoir\_volume = 4.445e8

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

* rainfall = 5e6

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

500,000

# add the rainfall variable to the reservoir\_volume variable

445000000

# increase reservoir\_volume by 5% to account for stormwater that flows

# into the reservoir in the days following the storm

4667725000

# decrease reservoir\_volume by 5% to account for evaporation

4.445e8 – 2.5e5

# subtract 2.5e5 cubic metres from reservoir\_volume to account for water

# that's piped to arid regions.

444250000

# print the new value of the reservoir\_volume variable

444250000

1. 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.o
* 3.0
* None of the above

1. In Python 3 what is the output of ½?

0.5

1. 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

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

 False

 True

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

capitalize() Converts the first character to upper case

casefold() Converts string into lower case

center() Returns a centered string

count() Returns the number of times a specified value occurs in a string

encode() Returns an encoded version of the string

endswith() Returns true if the string ends with the specified value

expandtabs() Sets the tab size of the string

find() Searches the string for a specified value and returns the position of where it was found

format() Formats specified values in a string

format\_map() Formats specified values in a string

index() Searches the string for a specified value and returns the position of where it was found

isalnum() Returns True if all characters in the string are alphanumeric

isalpha() Returns True if all characters in the string are in the alphabet

isdecimal() Returns True if all characters in the string are decimals

isdigit() Returns True if all characters in the string are digits

isidentifier() Returns True if the string is an identifier

islower() Returns True if all characters in the string are lower case

isnumeric() Returns True if all characters in the string are numeric

isprintable() Returns True if all characters in the string are printable

isspace() Returns True if all characters in the string are whitespaces

istitle() Returns True if the string follows the rules of a title

isupper() Returns True if all characters in the string are upper case

join() Joins the elements of an iterable to the end of the string

ljust() Returns a left justified version of the string

lower() Converts a string into lower case

lstrip() Returns a left trim version of the string

maketrans() Returns a translation table to be used in translations

partition() Returns a tuple where the string is parted into three parts

replace() Returns a string where a specified value is replaced with a specified value

rfind() Searches the string for a specified value and returns the last position of where it was found

rindex() Searches the string for a specified value and returns the last position of where it was found

rjust() Returns a right justified version of the string

rpartition() Returns a tuple where the string is parted into three parts

rsplit() Splits the string at the specified separator, and returns a list

rstrip() Returns a right trim version of the string

split() Splits the string at the specified separator, and returns a list

splitlines() Splits the string at line breaks and returns a list

startswith() Returns true if the string starts with the specified value

strip() Returns a trimmed version of the string

swapcase() Swaps cases, lower case becomes upper case and vice versa

title() Converts the first character of each word to upper case

translate() Returns a translated string

upper() Converts a string into upper case

zfill() Fills the string with a specified number of 0 values at the beginning

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

Welcome\_Aditya [1:]

New\_name = ‘B’ + new\_name

1. Reverse this ‘This is an easy quiz’ sentence without using string reverse method.

Str = “python”

Print(reversed)

1. Use slicing techniques with providing 4 different examples each one using different slicing techniques learned in the video.

1:, warn = arn, eat = at, oper = per, lotus = otus

:2, why = hy, good = od, qwerty = erty, bot = t

3:4, spoon = on, ball = ll, bottle = tl, light, = ht

::5, dictionary = iy, cartstonson = so, maplesyrup = s, elmerglue = g

1. Are strings immutable?

Yes

1. 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.

Name = Hero

New\_name = “Z” + new name

Name now =Zero

1. Write two expressions using format() method and f-string method.

Print (“we play lots {} games”. Format (‘of’))

Print (“I am a {} player”.format (‘good’))

1. 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.

True

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

1. Give examples of all the datatypes you learned.

Str, float

1. What function do you use to find length of a string?

len()

1. What type does this object have? "hippo" \*12

Lists

1. Create an example of list, with mixed data types. The answer should be just one line of code.

my\_list = ["green","yellow","blue",]

1. If list1 = [1,2,3,4,5]. What is the output of list1.pop ()? And what would be the result of list1[1:]? Output would be 1,2,3,4,5, and the result would be 2,3,4,5
2. Create a Dictionary where all the keys are strings and values are integers.

mydict={'key1':'2', 'key2':'3'}

1. Create a dictionary within a dictionary and write the code to find the values in the inside dictionary.

d = {'key1':123, 'weplaygames6':[0,1,2,3], 'wearefriends4' :{'insidekeyvalue':1000}}

1. Do dictionaries retain order and are they a sequence?

Yes, and they are a squenece

1. Given d = { ‘k1’:[1,2,3]}. What is the output of d[‘k1’][3]?
2. Are dictionaries immutable?

Yes

1. Do tuples have lots of methods associated with them?

Yes

1. When is the best time to use tuples over lists?

You you are typing with integers

1. Are tuples immutable?

Yes

1. Which of the following is a tuple?
2. [1,2,3]
3. (1,2,[1,2,3])
4. (1,2,4,)
5. None of the above?
6. Write an expression to turn the string ‘Mississippi’ into a set of unique characters.

Can’t figure it out

1. {1,2,3,4} Is this a set?

Yes

1. What method do you use to add an element to a string?

.add

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

1,2,3

1. An error
2. [1,2,3]
3. {1,2,2,3}
4. {1,2,3}
5. 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, 6

1. 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?

Yes

1. 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.

Population,

d ={'Shangai':'17.8','Istanbul':'13.3','Karachi':"13.0",'Mumbai':'12.5'}

d

# Key | Value

# Shanghai | 17.8

# Istanbul | 13.3

# Karachi | 13.0

# Mumbai | 12.5

**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---**