**Python Mega Assignment**

## Assignment Part-1

Q1. Why do we call Python as a general purpose and high-level programming language?

**Ans:** Python is a general purpose and high-level programming language because:

* Easy to read
* It is not specialized for any specific domain
* Can be used to create different variety of programs

Q2. Why is Python called a dynamically typed language?

**Ans:** Python doesn’t know the type of variable until it’s run.

Q3. List some pros and cons of Python programming language?

|  |  |
| --- | --- |
| **Python - Pros** | **Python - Cons** |
| Beginner Friendly | Issues with Design |
| Flexible and Extensible | Slower than compiled languages |
| Extensive Libraries | Security |
| Embeddable | Work Environment |
| Highly Scalable | Dynamically Typed |
| IoT Opportunities | High memory consumption |

Q4. In what all domains can we use Python?

Ans:

* AI
* ML
* Deep Learning
* Data Analysis
* Data Visualization
* Task Automation
* Developing websites and softwares

Q5. What are variable and how can we declare them?

Ans: Python has no command for declaring a variable. A variable is created the moment you first assign a value to it.

Q6. How can we take an input from the user in Python?

Ans:

* Python provides a built-in function called Input() which takes the input from the user.
* When the input function is called it stops the program and waits for the user’s input.
* When the user presses enter, the program resumes and returns what the user typed.

Q7. What is the default datatype of the value that has been taken as an input using input() function?

Ans: String

Q8. What is type casting?

Ans: Converting one datatype into another is known as type casting.

Q9. Can we take more than one input from the user using single input() function? If yes, how? If no, why?

**Ans:** Yes, we can take multiple inputs in one single line by using the Input() function several times.

**For e.g.**

#multiple inputs in Python using input

x, y = input("Enter First Name: "), input("Enter Last Name: ")

print("First Name is: ", x)

print("Second Name is: ", y)

**Output:**

Enter First Name: FACE

Enter Last Name: Prep

First Name is: FACE

Second Name is: Prep

Q10. What are keywords?

**Ans**: Python has a set of keywords that are reserved words that cannot be used as variable names, function names, or any other identifiers.

For e.g.:

|  |  |
| --- | --- |
| Keyword | Purpose |
| and | A logical operator |
| Break | To break out of a loop |

Q11. Can we use keywords as a variable? Support your answer with reason.

**Ans:** No, we can not use a keyword as an variable. If we use them we will get syntax error.

Q12. What is indentation? What's the use of indentation in Python?

**Ans:** Indentation refers to the spaces at the beginning of a code line. Python uses indentation to indicate a block of code.

Q13. How can we throw some output in Python?

**Ans:** As a Python developer you can choose to throw an exception if a condition occurs.

To throw (or raise) an exception, use the raise keyword.

Q14. What are operators in Python?

**Ans:** Operators are used to perform operations on variables and values.

Q15. What is difference between / and // operators?

**Ans:** In Python programming, you can perform division in two ways. The first one is Float Division("/") and the second is Integer Division("//") or Floor Division.

Q16. Write a code that gives following as an output.

```

iNeuroniNeuroniNeuroniNeuron

```

**Ans:**

*a = "iNeuroniNeuroniNeuroniNeuron"*

*print(a)*

Q17. Write a code to take a number as an input from the user and check if the number is odd or even.

**Ans:**

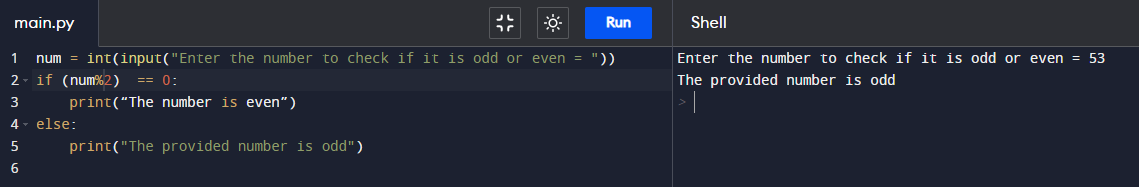
*num = int(Input(“Enter the number to check if it is odd or even = “))*

*if (num % 2) == 0:*

*print(“The number is even”)*

*else:*

*print(“The provided number is odd”)*



Q18. What are boolean operator?

**Ans:** Operators are special symbols in Python that is used to perform arithmetic or logical computations. The values on which operation is to be done are called operands. while the operation is denoted by operator.

e.g. (+, -, /, \*, %, etc.)

Q19. What will the output of the following?

**Ans:** 1 or 0 = True

0 and 0 = True

True and False and True = False

1 or 0 or 0 = True

```

Q20. What are conditional statements in Python?

**Ans:** Python supports the usual logical conditions from mathematics:

* Equals: a == b
* Not Equals: a != b
* Less than: a < b
* Less than or equal to: a <= b
* Greater than: a > b
* Greater than or equal to: a >= b

These conditions can be used in several ways, most commonly in "if statements" and loops. An "if statement" is written by using the if keyword.

Q21. What is use of 'if', 'elif' and 'else' keywords?

**Ans:** If-elif-else keywords are used in Python for decision-making i.e the program will evaluate test expression and will execute the remaining statements only if the given test expression turns out to be true. This allows validation for multiple expressions.

Q22. Write a code to take the age of person as an input and if age >= 18 display "I can vote". If age is < 18 display "I can't vote".

**Ans:**

*Age = int(Input(“Enter age = ”))*

*if Age >= 18:*

*Print(“I can Vote”)*

*else:*

*print(“I can’t vote”)*

Q23. Write a code that displays the sum of all the even numbers from the given list.

```

numbers = [12, 75, 150, 180, 145, 525, 50]

```

**Ans:**

*numlist = [12, 75, 150, 180, 145, 525, 50]*

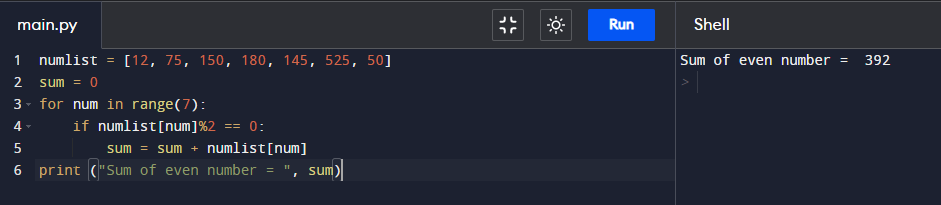
*sum = 0*

*for num in range(7):*

*if numlist[num]%2 == 0:*

*sum = sum + numlist[num]*

*print ("Sum of even number = ", sum)*



Q24. Write a code to take 3 numbers as an input from the user and display the greatest no as output.

**Answer:**

*num1 = float(input("Enter first number: "))*

*num2 = float(input("Enter second number: "))*

*num3 = float(input("Enter third number: "))*

*if (num1 >= num2) and (num1 >= num3):*

*greatest = num1*

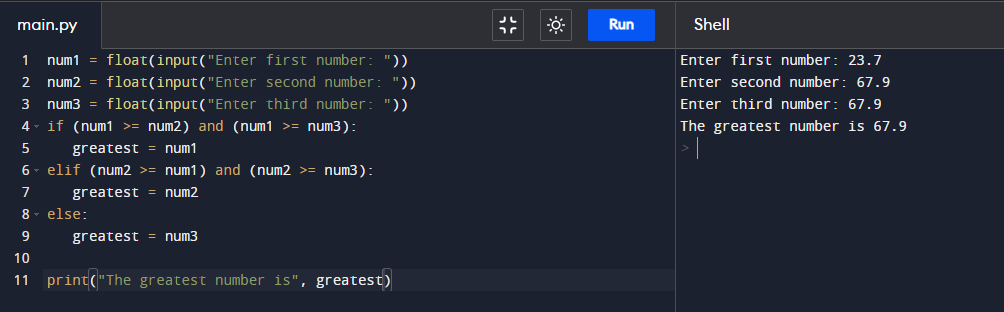
*elif (num2 >= num1) and (num2 >= num3):*

*greatest = num2*

*else:*

*greatest = num3*

*print("The greatest number is", greatest)*



Q26. What is a string? How can we declare string in Python?

**Ans:** Strings in Python are arrays of bytes representing unicode characters.

Strings in python are surrounded by either single quotation marks, or double quotation marks.

Q27. How can we access the string using its index?

**Ans:** Square brackets can be used to access elements of the string.

**e.g.**

*a = "Hello, World!"  
print(a[1])*

**Output:** e

Q28. Write a code to get the desired output of the following

```

string = "Big Data iNeuron"

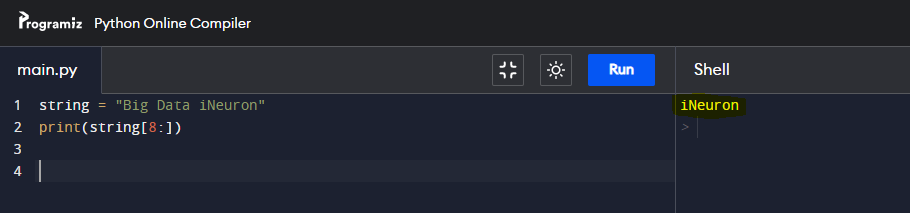
desired\_output = "iNeuron"

```

**Solution:**

*string = "Big Data iNeuron"*

*print(string[8:])*



Q29. Write a code to get the desired output of the following

```

string = "Big Data iNeuron"

desired\_output = "norueNi"

```

Q30. Resverse the string given in the above question.

Q31. How can you delete entire string at once?

**Ans:** We can remove the entire string variable using the del command.

Q32. What is escape sequence?

**Ans:** An escape sequence is a sequence of characters that, when used inside a character or string, does not represent itself but is converted into another character or series of characters.

Q33. How can you print the below string?

```

'iNeuron's Big Data Course'

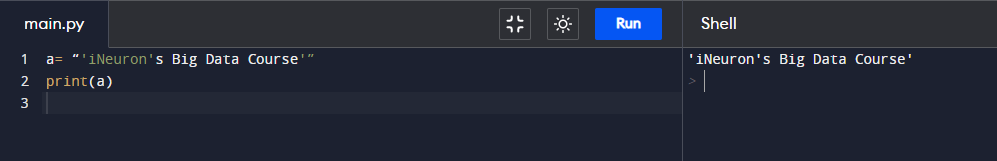
```

**Solution:**

*a= “'iNeuron's Big Data Course'”*

*print(a)*

**Output:**



Q34. What is a list in Python?

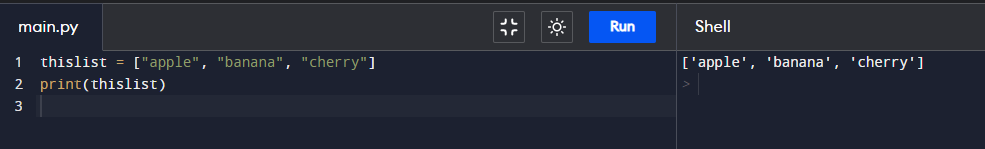
**Ans:** Lists are used to store multiple items in a single variable. Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are Tuple, Set, and Dictionary, all with different qualities and usage. Lists are created using square brackets.

Q35. How can you create a list in Python?

**Ans:** Lists are created using square brackets:

**Example**

*thislist = ["apple", "banana", "cherry"]  
print(thislist)*



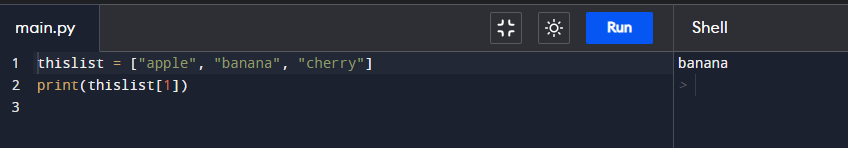
Q36. How can we access the elements in a list?

**Ans:** List items are indexed and you can access them by referring to the index number.

**Example**

Print the second item of the list:

*thislist = ["apple", "banana", "cherry"]  
print(thislist[1])*

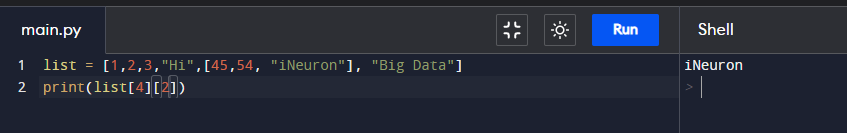


Q37. Write a code to access the word "iNeuron" from the given list.

```

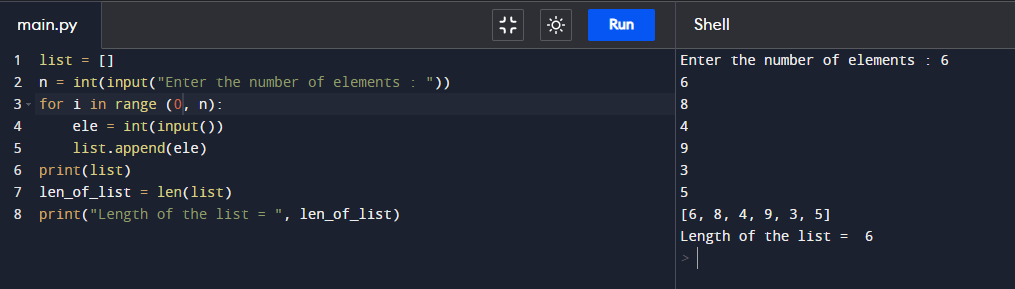
list = [1,2,3,"Hi",[45,54, "iNeuron"], "Big Data"]

```



Q38. Take a list as an input from the user and find the length of the list.

**Solution:**



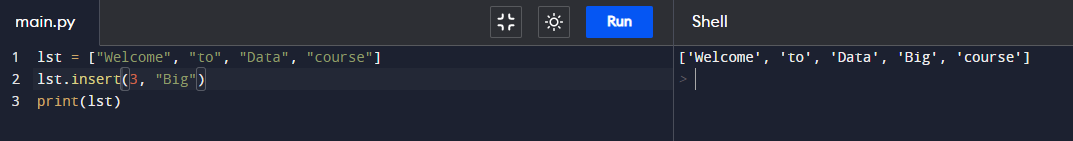
Q39. Add the word "Big" in the 3rd index of the given list.

```

lst = ["Welcome", "to", "Data", "course"]

```

**Solution:**



Q40. What is a tuple? How is it different from list?

**Ans:**

* Tuples are used to store multiple items in a single variable.
* A tuple is a collection which is ordered and unchangeable.
* Tuples are written with round brackets.
* **Example:** thistuple = ("apple", "banana", "cherry")

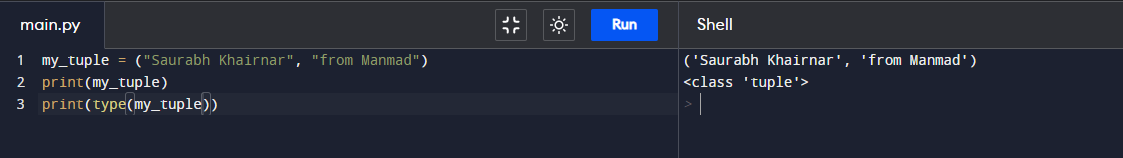
Q41. How can you create a tuple in Python?

**Ans:**

* Tuples are written with round brackets.
* **Example:** thistuple = ("apple", "banana", "cherry")

Q42. Create a tuple and try to add your name in the tuple. Are you able to do it? Support your answer with reason.

**Solution:**



Q43. Can two tuple be appended. If yes, write a code for it. If not, why?

**Ans:** You can't add elements to a tuple because of their immutable property. There's no append() or extend() method for tuples, You can't remove elements from a tuple, also because of their immutability.

Q44. Take a tuple as an input and print the count of elements in it.

Q45. What are sets in Python?

Ans:

* Sets are used to store multiple items in a single variable.
* A set is a collection which is unordered, unchangeable, and unindexed.
* Sets are written with curly brackets.
* **Note:** Set items are unchangeable, but you can remove items and add new items.
* **Example:** thisset = {"apple", "banana", "cherry"}

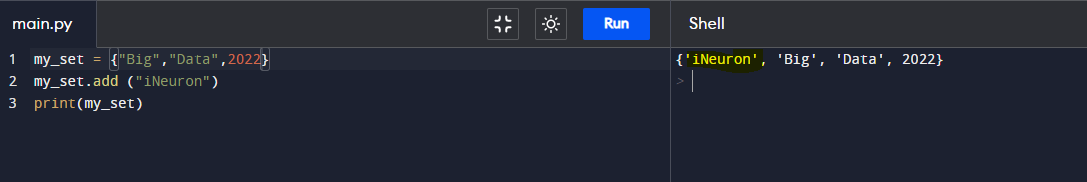
Q46. How can you create a set?

**Solution:**

**Example:** thisset = {"apple", "banana", "cherry"}

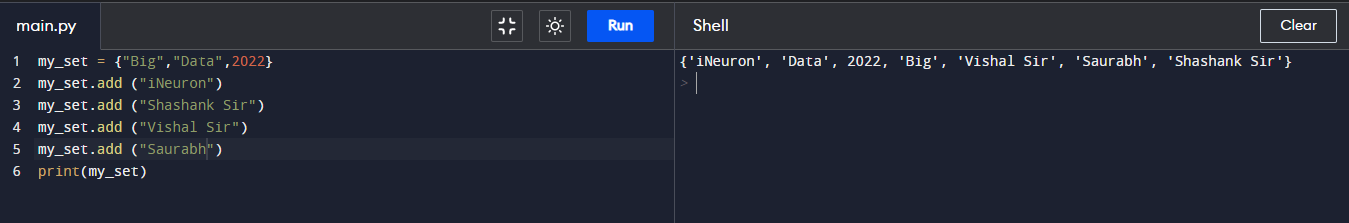
Q47. Create a set and add "iNeuron" in your set.

**Solution:**



Q48. Try to add multiple values using add() function.

**Solution:**



Q49. How is update() different from add()?

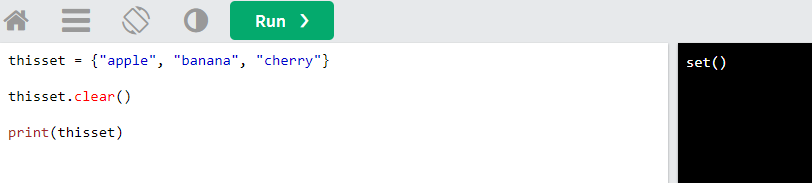
**Ans:**

**Use of update():** To add items from another set into the current set, use the update() method.

**Use of add():** To add one item to current set add() method is used.

Q50. What is clear() in sets?

**Ans:** The clear() method empties the set.



Q51. What is frozen set?

**Ans:**

The frozenset() function returns an unchangeable frozenset object (which is like a set object, only unchangeable).



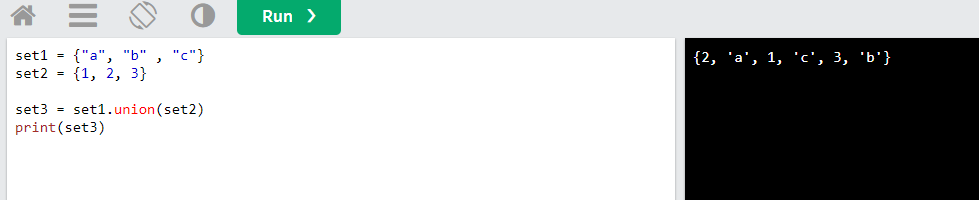
Q52. How is frozen set different from set?

Ans: The frozenset() function returns an unchangeable frozenset object (which is like a set object, only unchangeable).

Q53. What is union() in sets? Explain via code.

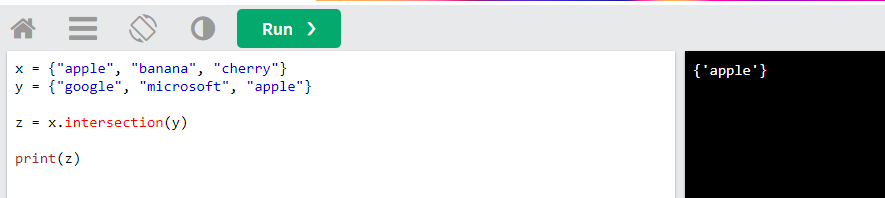
**Ans:**

union() in sets returns a new set containing all items from both sets.



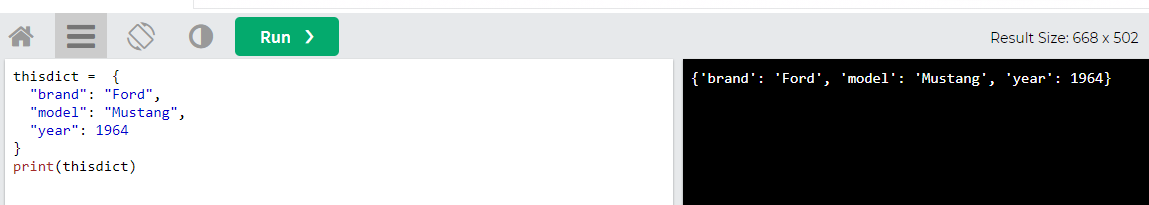
Q54. What is intersection() in sets? Explain via code.

Ans: The intersection() method will return a new set, that only contains the items that are present in both sets.



Q55. What is dictionary in Python?

**Ans:**

* Dictionaries are used to store data values in key:value pairs.
* A dictionary is a collection which is ordered\*, changeable and do not allow duplicates.
* Dictionaries are written with curly brackets, and have keys and values.
* 

Q56. How is dictionary different from all other data structures.

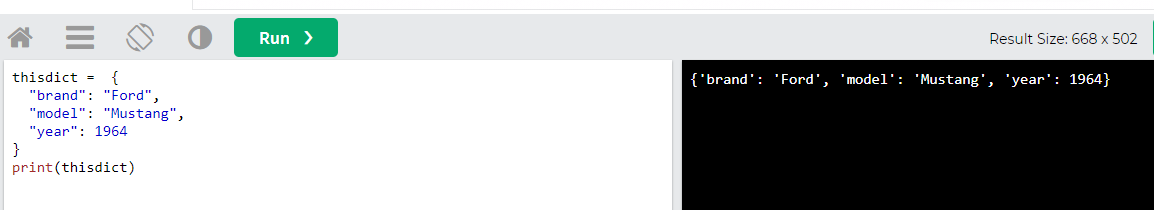
**Ans:**

* Dictionaries are used to store data values in key:value pairs.
* A dictionary is a collection which is ordered\*, changeable and do not allow duplicates.

Q57. How can we declare a dictionary in Python?

**Ans:**

Dictionaries are written with curly brackets, and have keys and values.



Q58. What will the output of the following?

```

var = {}

print(type(var))

```

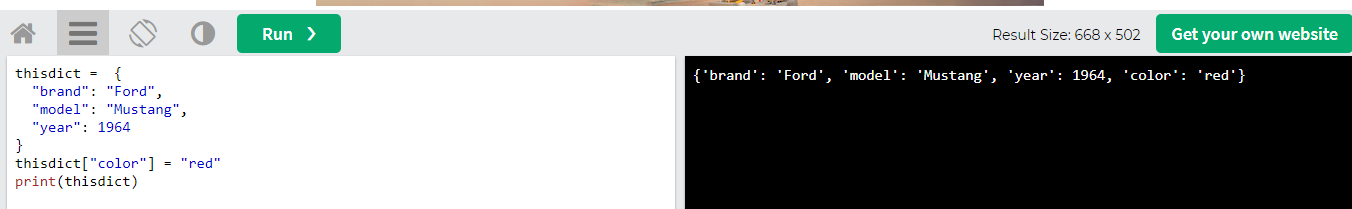
**Output:** <class 'dict'>

Q59. How can we add an element in a dictionary?

Ans: Adding an item to the dictionary is done by using a new index key and assigning a value to it.

**Example:**

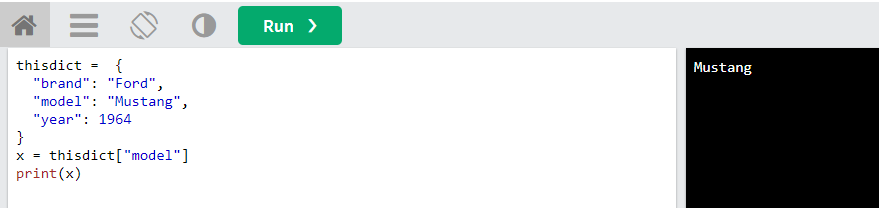
thisdict = {  
  "brand": "Ford",  
  "model": "Mustang",  
  "year": 1964  
}  
thisdict["color"] = "red"  
print(thisdict)



Q60. Create a dictionary and access all the values in that dictionary.

**Ans:** You can access the items of a dictionary by referring to its key name, inside square brackets.

thisdict = {  
  "brand": "Ford",  
  "model": "Mustang",  
  "year": 1964  
}  
x = thisdict["model"]



Q61. Create a nested dictionary and access all the element in the inner dictionary.

**Ans:**

child1 = {

"name" : "Emil",

"year" : 2004

}

child2 = {

"name" : "Tobias",

"year" : 2007

}

child3 = {

"name" : "Linus",

"year" : 2011

}

myfamily = {

"child1" : child1,

"child2" : child2,

"child3" : child3

}

print(myfamily)

print(myfamily["child1"]["name"])

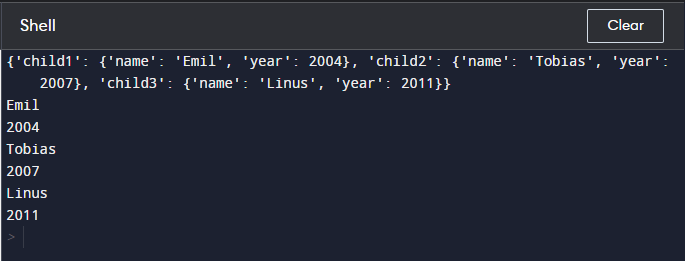
print(myfamily["child1"]["year"])

print(myfamily["child2"]["name"])

print(myfamily["child2"]["year"])

print(myfamily["child3"]["name"])

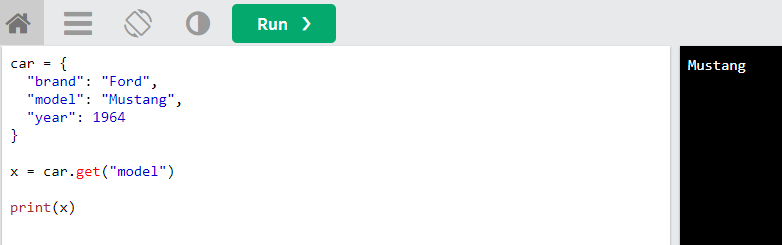
print(myfamily["child3"]["year"])



Q62. What is the use of get() function?

**Ans:** The get() function returns the value of the item with the specified key.

car = {  
  "brand": "Ford",  
  "model": "Mustang",  
  "year": 1964  
}  
  
x = car.get("model")  
print(x)



Q63. What is the use of items() function?

Ans:

* The items() method returns a view object. The view object contains the key-value pairs of the dictionary, as tuples in a list.
* The view object will reflect any changes done to the dictionary.
* **Example:**

car = {

"brand": "Ford",

"model": "Mustang",

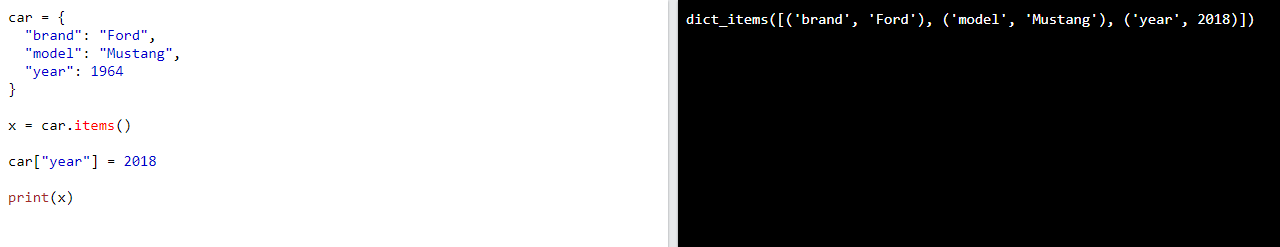
"year": 1964

}

x = car.items()

car["year"] = 2018

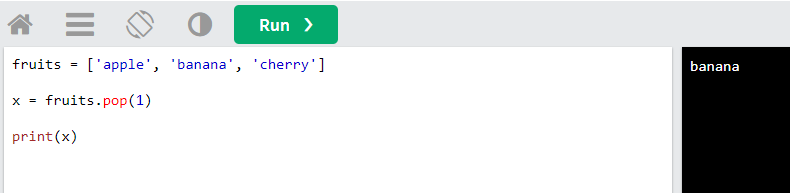
print(x)



Q64. What is the use of pop() function?

**Ans:**

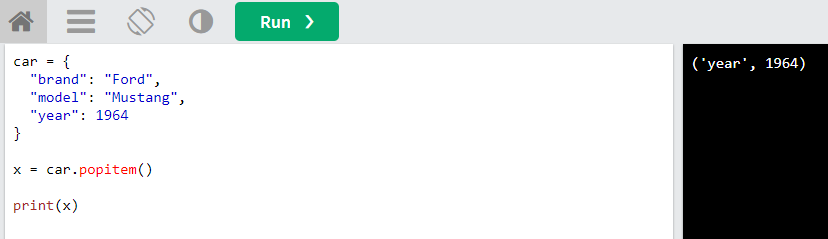
A number specifying the position of the element you want to remove, default value is -1, which returns the last item.



Q65. What is the use of popitems() function?

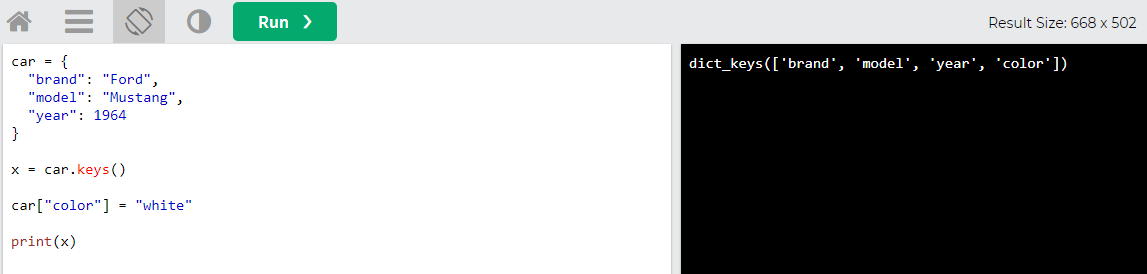
**Ans:**

The popitem() method removes the item that was last inserted into the dictionary. In versions before 3.7, the popitem() method removes a random item.



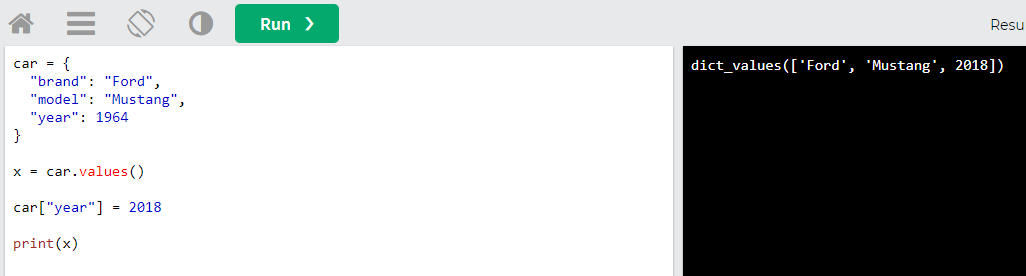
Q66. What is the use of keys() function?

**Ans:** The keys() method returns a view object. The view object contains the keys of the dictionary, as a list.



Q67. What is the use of values() function?

**Ans**: The values() method returns a view object. The view object contains the values of the dictionary, as a list.



Q68. What are loops in Python?

Ans: Looping means repeating something over and over until a particular condition is satisfied.

Q69. How many type of loop are there in Python?

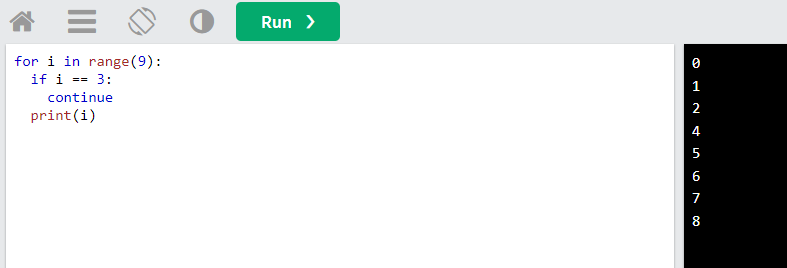
Ans: There are two types of loops in Python**, for** and **while**.

Q70. What is the difference between for and while loops?

Ans: The major difference between for loop and the while loop is that for loop is used when the number of iterations is known, whereas execution is done in the while loop until the statement in the program is proved wrong.

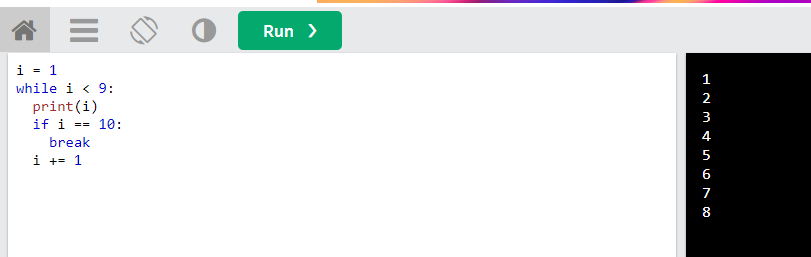
Q71. What is the use of continue statement?

Ans: The continue keyword is used to end the current iteration in a for loop (or a while loop), and continues to the next iteration.



Q72. What is the use of break statement?

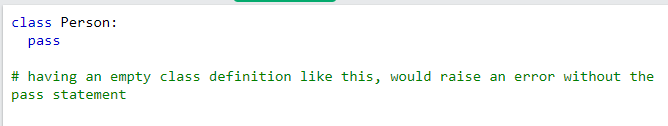
Ans: The break keyword is used to break out a for loop, or a while loop.

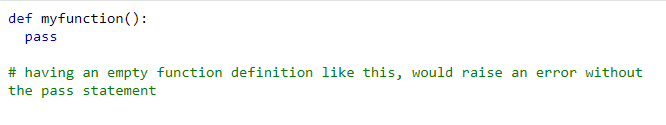


Q73. What is the use of pass statement?

Ans:

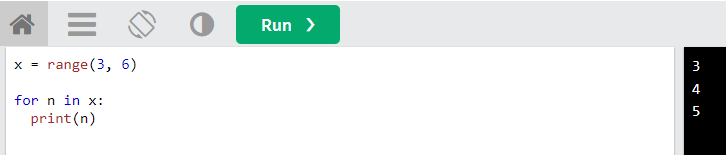
* The pass statement is used as a placeholder for future code.
* When the pass statement is executed, nothing happens, but you avoid getting an error when empty code is not allowed.
* Empty code is not allowed in loops, function definitions, class definitions, or in if statements.





Q74. What is the use of range() function?

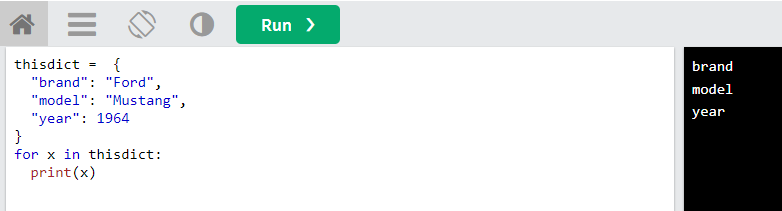
**Ans:** The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and stops before a specified number.

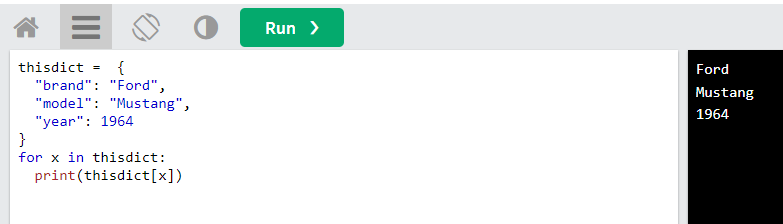


Q75. How can you loop over a dictionary?

**Ans:**

* You can loop through a dictionary by using a for loop.
* When looping through a dictionary, the return value are the keys of the dictionary, but there are methods to return the values as well.

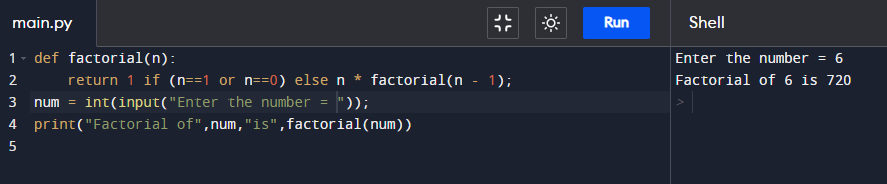




**### Coding Problems**

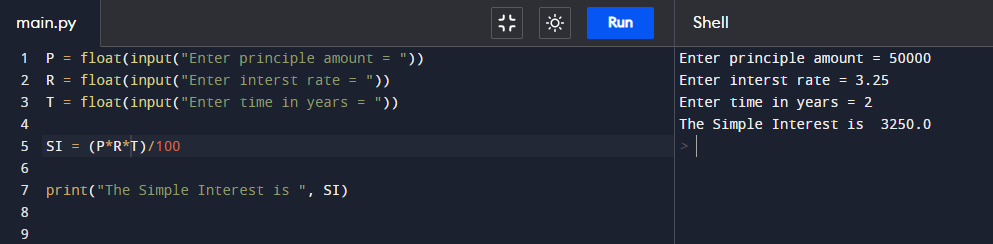
Q76. Write a Python program to find the factorial of a given number.

**Solution:**



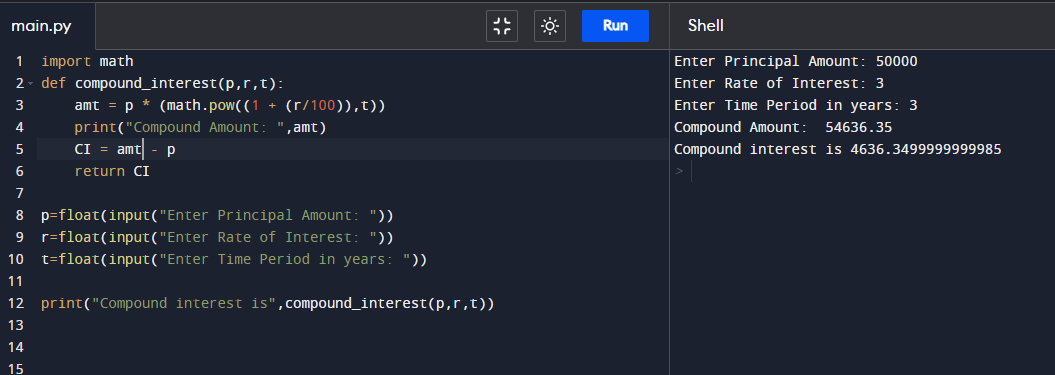
Q77. Write a Python program to calculate the simple interest. Formula to calculate simple interest is SI = (P\*R\*T)/100

**Solution:**



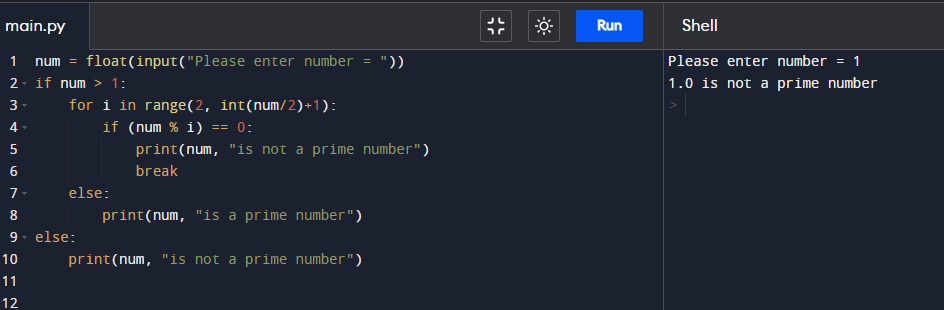
Q78. Write a Python program to calculate the compound interest. Formula of compound interest is A = P(1+ R/100)^t.

**Solution:**



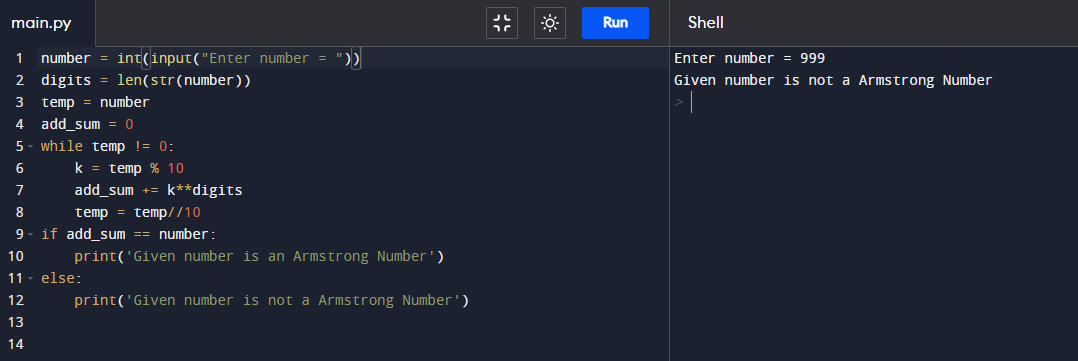
Q79. Write a Python program to check if a number is prime or not.

**Solution:**



Q80. Write a Python program to check Armstrong Number.

**Solution:**



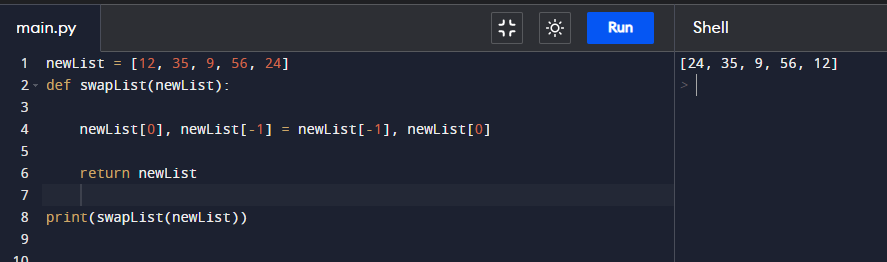
Q81. Write a Python program to find the n-th Fibonacci Number.

**Solution:**



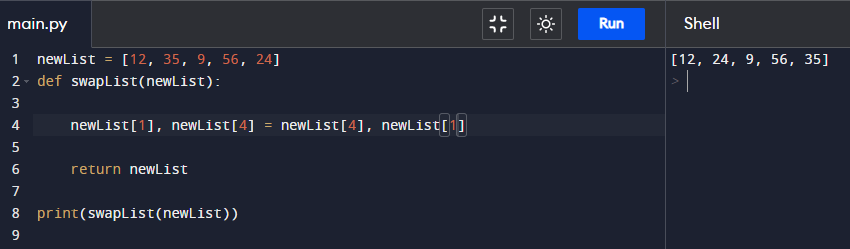
Q82. Write a Python program to interchange the first and last element in a list.

**Solution:**



Q83. Write a Python program to swap two elements in a list.

**Solution:**



Q84. Write a Python program to find N largest element from a list.

Q85. Write a Python program to find cumulative sum of a list.

Q86. Write a Python program to check if a string is palindrome or not.

Q87. Write a Python program to remove i'th element from a string.

Q88. Write a Python program to check if a substring is present in a given string.

Q89. Write a Python program to find words which are greater than given length k.

Q90. Write a Python program to extract unquire dictionary values.

Q91. Write a Python program to merge two dictionary.

Q92. Write a Python program to convert a list of tuples into dictionary.

```

Input : [('Sachin', 10), ('MSD', 7), ('Kohli', 18), ('Rohit', 45)]

Output : {'Sachin': 10, 'MSD': 7, 'Kohli': 18, 'Rohit': 45}

```

Q93. Write a Python program to create a list of tuples from given list having number and its cube in each tuple.

```

Input: list = [9, 5, 6]

Output: [(9, 729), (5, 125), (6, 216)]

```

Q94. Write a Python program to get all combinations of 2 tuples.

```

Input : test\_tuple1 = (7, 2), test\_tuple2 = (7, 8)

Output : [(7, 7), (7, 8), (2, 7), (2, 8), (7, 7), (7, 2), (8, 7), (8, 2)]

```

Q95. Write a Python program to sort a list of tuples by second item.

```

Input : [('for', 24), ('Geeks', 8), ('Geeks', 30)]

Output : [('Geeks', 8), ('for', 24), ('Geeks', 30)]

```

Q96. Write a python program to print below pattern.

```

\*

\* \*

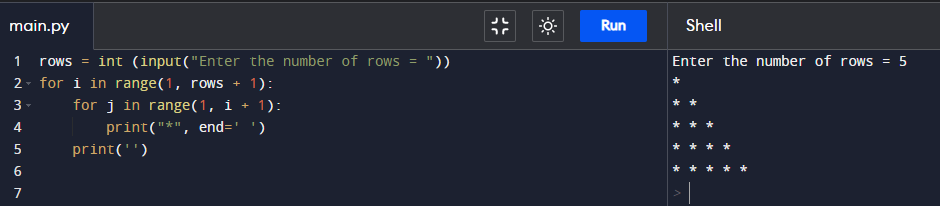
\* \* \*

\* \* \* \*

\* \* \* \* \*

```

**Solution:**



Q97. Write a python program to print below pattern.

```

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

```

Q98. Write a python program to print below pattern.

```

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

```

Q99. Write a python program to print below pattern.

```

1

1 2

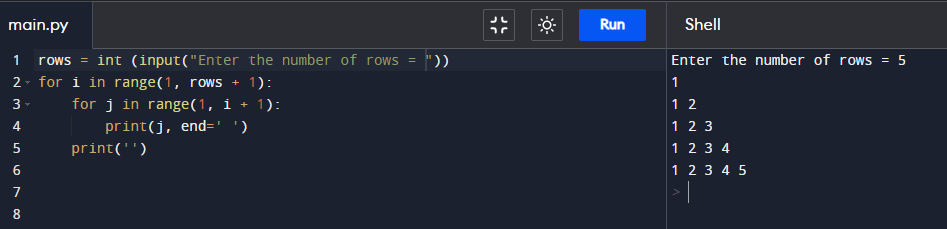
1 2 3

1 2 3 4

1 2 3 4 5

```

**Solution:**



Q100. Write a python program to print below pattern.

```

A

B B

C C C

D D D D

E E E E E

```