In [1]:
```
print('Hello World!')
```

```
Hello World!
```

In [2]:
```
a = 'Hello World!'  #taking variable
print(a)
```

```
Hello World!
```

In [3]:
```
print('indexing: ', a[2])     #indexing
print('negative indexing: ', a[-2])     #negative indexing
print('length of string is:', len(a))       #printing length
```

```
indexing:  l
negative indexing:  d
length of string is: 12
```

In [4]:
```
for i in a:     #for loop
    print(i, end='-')
print()
```

```
H-e-l-l-o- -W-o-r-l-d-!-
```

In [5]:
```
if 'e' in a:    #if-else
    print('True')
else:
    print('False')

if 'a' not in a:
    print('True')
else:
    print('False')
```

```
True
True
```

In [6]:
```
print('slicing from 0:9=', a[0:9])     #slicing
print('slicing from 0:-7=', a[:-7])
print('slicing from 0:10=', a[0:10])
print('slicing from 4:11=', a[4:11])
```

```
slicing from 0:9= Hello Wor
slicing from 0:-7= Hello
slicing from 0:10= Hello Worl
slicing from 4:11= o World
```

In [7]:
```
print('upper():', a.upper())
print('lower():', a.lower())
print('capitalize():', a.capitalize())
print('replace(): ', a.replace('H', 'M'))
print('split():', a.split())
```

```
upper(): HELLO WORLD!
lower(): hello world!
capitalize(): Hello world!
replace():  Mello World!
split(): ['Hello', 'World!']
```

In [8]:
```
my_list = ['apple', 'banana', 'orange', 'cherry']       #List
print(my_list)
print(my_list[0])   #List with indexing
print(my_list[2])
print(my_list[-1])     #List with negative indexing
print('length of my_list:', len(my_list))
print('slicing with my_list', my_list[0:-2])     #slicing with list
```

```
['apple', 'banana', 'orange', 'cherry']
apple
orange
cherry
length of my_list: 4
slicing with my_list ['apple', 'banana']
```

In [9]:
```
if 'orange' in my_list:
    print('Yes!')
else:
    print('No!')
```

```
Yes!
```

In [10]:
```
my_list.append('kiwi')
my_list.insert(1, 'watermelon')
print('append():', my_list)
```

```
print('insert():', my_list)
print('pop():', my_list.pop())
print('pop():', my_list.pop(1))
my_list.remove('banana')
print('remove() "banana":', my_list)
```

```
append(): ['apple', 'watermelon', 'banana', 'orange', 'cherry', 'kiwi']
insert(): ['apple', 'watermelon', 'banana', 'orange', 'cherry', 'kiwi']
pop(): kiwi
pop(): watermelon
remove() "banana": ['apple', 'orange', 'cherry']
```

In [11]:
```
#my_list.clear()
#print('empty list with clear():', my_list)
```

```
empty list with clear(): []
```

In [13]:
```
for i in my_list:
    print(i, end=' ')
print()
```

In [14]:
```
i=0
while i < len(my_list):
    print(my_list)
    i += 1
print()
```

In [15]:
```
my_list = ['apple', 'banana', 'orange', 'cherry']       #list
for i in my_list:
    print(i, end=' ')
print()
```

```
apple banana orange cherry
```

In [16]:
```
i=0
while i < len(my_list):
    print(my_list)
    i += 1
print()
```

```
['apple', 'banana', 'orange', 'cherry']
['apple', 'banana', 'orange', 'cherry']
['apple', 'banana', 'orange', 'cherry']
['apple', 'banana', 'orange', 'cherry']
```

In [17]:
```
print('list comprehension method:-')
[print(i) for i in my_list] #list comprehension method
[print(i, end=' ') for i in range(1, 11, 2)]
print()
```

```
list comprehension method:-
apple
banana
orange
cherry
1 3 5 7 9
```

In [18]:
```
print('list comprehension method using while loop:-')
i=1
while i<6:
    print(i, end=' ')
    i += 1
print()
```

```
list comprehension method using while loop:-
1 2 3 4 5
```

In [19]:
```
def call(x):        #callable function
    return x
print(callable(call))
```

```
True
```

In [20]:
```
def foo():          #type
    pass
print(type(foo))
print('foo name attribute: ', foo.__name__)
```

```
<class 'function'>
foo name attribute:  foo
```

In [21]:
```
sq = lambda x: x*x      #lambda function
res = sq(25)
print(res)
```

```
625
```

In [22]:
```
lst = [1,5,7,9,10,12,14,32,85]      #Lambda function using filter
```

```
newlst = list(filter(lambda a: a%2==0, lst))
print('even list:', newlst)
newlst2 = list(filter(lambda a: a%2!=0, lst))
print('odd list:', newlst2)
```

```
even list: [10, 12, 14, 32]
odd list: [1, 5, 7, 9, 85]
```

In [23]:
```
num1 = int(input('Enter num1:'))
num2 = int(input('Enter num2:'))

sum = num1 + num2

print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
```

```
Enter num1:42
Enter num2:99
The sum of 42 and 99 is 141
```

In [24]:
```
name = 'Shiva'
greet = 'How are you?'
print('Hello {0}! {1}'.format(name, greet))
```

```
Hello Shiva! How are you?
```

In [25]:
```
x = 5
y = 19

temp = x
x = y
y = temp

print('The value of x after swapping is {}'.format(x))
print('The value of y after swapping is {}'.format(y))
```

```
The value of x after swapping is 19
The value of y after swapping is 5
```

In [26]:
```
a = 25
b = 3
if(a>b):
    print(a,'is greater than', b)
elif(a==b):
    print(a, 'is equal to', b)
else:
    print(b, 'is greater than', a)
```

```
25 is greater than 3
```

In [27]:
```
fruits = ['apple', 'banana', 'cherry', 'orange', 'kiwi']
for i in fruits:
    if i == 'orange':
        continue
    print(i)
print()
```

```
apple
banana
cherry
kiwi
```

In [28]:
```
fruits = ['apple', 'banana', 'cherry', 'orange', 'kiwi']
for i in fruits:
    if i == 'orange':
        break
    print(i)
print()
```

```
apple
banana
cherry
```

In [29]:
```
cars = ["Mahindra", "TATA", "Maruti"]
for i in cars:
    print(i, end=" ")
print()
```

```
Mahindra TATA Maruti
```

In [30]:
```
cars = {"Mahindra":"XUV 700", "Tata":"Nexon EV"}
print(cars["Mahindra"])
print(cars["Tata"])
```

```
XUV 700
Nexon EV
```

In [31]:
```
cars = {
    "brand": "Mahindra",
    "model": "XUV700",
    "year": "2021",
}
```

```
print(cars["brand"])
```

```
Mahindra
```

In [32]:
```
import module as m      #custom module
result = m.details
print(result)
y = m.loads(result)
y = m.dumps(result)
print(y)
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
C:\Users\PRATIK~1\AppData\Local\Temp/ipykernel_17680/1954517631.py in <module>
----> 1 import module as m      #custom module
      2 result = m.details
      3 print(result)
      4 y = m.loads(result)
      5 y = m.dumps(result)

ModuleNotFoundError: No module named 'module'
```

In [33]:
```
import json     #parse json
details = """{
    "name": "John Watts",
    "age": "30",
    "gender": "Male",
    "role": ["SDE", "Data Engineer", "Associate Engineer"]
}"""
y = json.loads(details)
print(y["role"])
```

```
['SDE', 'Data Engineer', 'Associate Engineer']
```

In [ ]: