



## PRODDEC PYTHON DAY 3

### String Slicing

```
In [1]: # Slicing -> Cutting out portion of a string.  
  
animal='giraffe'  
print(animal[0]) # Print the first letter of the string.  
print(animal[1]) # Print the second letter of the string.  
print(animal[-1]) # Print last letter of the string.  
print(animal[-2]) # Print second last letter of the string.  
print(animal[1:3]) # Print portion of the string.  
print(animal[1:]) # Printing from the second letter onwards.  
print(animal[:4]) # Print upto 4th letter.  
  
# Exclusive offer  
print(animal[::-2]) # Print odd position character in string.  
print(animal[::-1]) # Reverse a string.  
print(animal[::-2]) # Reverse a string and print odd position character.
```

### Methods

```
In [2]: # It's a function in a class.  
  
In [2]: name="peter"  
print(name.upper()) # Convert to uppercase.  
print(name)  
  
PETER  
peter
```

```
In [2]: name="DAVID ABRAHAM"  
print(name.lower()) # Lowercase letter  
print(name.title()) # First Letter of each word->uppercase  
print(name.capitalize()) # Only first letter ->upper case
```

david abraham  
David Abraham  
David abraham

```
In [28]: name="Peter Davidson"  
print(name.swapcase()) # Change uppercase to lowercase and viceversa.  
print(name.startswith('P')) # Return True or False  
print(name.endswith('f')) # Return True or False.
```

pETER dAVIDSON  
True  
False

```
In [2]: name="Peter hai"  
print(name.strip(),"hai") # Cut off the spaces on right and Left.  
print(name.rstrip(),"hai") # Cut off the spaces on the right.  
print(name.lstrip(),"hai") # Cut off the spaces on the left.
```

Peter hai  
Peter hai  
Peter hai

```
In [1]: name="vargheese"  
print(name.center(30),"tea") # Centre aligned by adding space to left and right.  
print(name.rjust(15)) # Fill space to the left -> right aligned.  
print(name.ljust(20),"hai") # Fill space to right -> Left aligned.  
print(name.replace("v","b")) # Replace t with b.  
print(name.count("e")) # Count the e in the string.  
print(name.index("g")) # Return the location of the g in string.
```

vargheese tea  
vargheese hai  
bargheese  
3  
3

```
In [3]: # String formatting  
name="Hari"  
college="CEC"  
print("{} is studying in {}".format(name,college))  
print("%s is studying in %s" %(name,college))  
print(name,"is studying in ",college)  
print(name + " is studying in " + college)
```

Hari is studying in CEC  
Hari is studying in CEC  
Hari is studying in CEC  
Hari is studying in CEC

```
In [6]: name="alen"  
print(name.isnumeric()) # Check if string consist of numeric character.  
print(name.isalpha()) # Check if string is a alphabet.  
print(name.islower()) # Check string is lower case alphabets.  
print(name.zfill(30)) # Fill zero before string.
```

False  
True  
True  
00000000000000000000000000000000alen

### Logical operator

```
In [58]: # and operator -> both condition must be True.  
name="haris"  
age=20  
if name=="haris" and age==20:  
    print("You are haris and age is 20.")
```

You are haris and age is 20.

```
In [59]: # or operator -> one of the condition should be True  
name="haris"  
college="CEC"  
if name=="haris" or college=="CEC":  
    print("you are haris or your college is CEC.")
```

you are haris or your college is CEC.

```
In [7]: # not operator  
isMale=True  
  
if not(isMale):  
    print("You are Male.")  
else:  
    print("You are female.")
```

You are female.

### Membership operator

```
In [48]: # in operator  
animal="giraffe"  
if "raff" in animal:  
    print("The animal is giraffe")
```

The animal is giraffe

```
In [65]: # (not in) operator.  
vehicle="autorikshaw"  
if "rikshaw" not in vehicle:  
    print("You are using some other vehicle")  
else:  
    print("You are driving autorickshaw")
```

You are driving autorickshaw

### LIST

```
In [8]: # For storing multiple values in a single variable.  
  
shoppingList=["carrot","onion","cabbage","milk","egg"]  
print(shoppingList)  
  
print(len(shoppingList)) # Printing the Length of the list.
```

```
# LIST SLICING  
print(shoppingList[0]) # Getting the first element in the list.  
print(shoppingList[2]) # Getting the third element in the list.  
print(shoppingList[-1]) # Getting the last element in the list.  
print(shoppingList[0:-2]) # Printing upto second last element in the list.  
print(shoppingList[:-2]) # Printing upto second last element in the list.  
print(shoppingList[2:]) # Printing from second element in the list.  
print(shoppingList[::-1]) # Reverse the items in the list
```

['carrot', 'onion', 'cabbage', 'milk', 'egg']  
5  
carrot  
cabbage  
egg  
['carrot', 'onion', 'cabbage']  
['carrot', 'onion', 'cabbage']  
['cabbage', 'milk', 'egg']  
['egg', 'milk', 'cabbage', 'onion', 'carrot']

```
In [2]: # Slicing in tuple  
print(car[0]) # Getting the first element in the tuple  
print(car[-1]) # Printing the Last element in the tuple.
```

ferari  
jaguar

### Functions in python

```
In [11]: # Creating a basic function  
# Functions are used for avoiding repeated code.  
# Help to organise code.  
  
def sayHello():  
    print("Welcome to the show")  
    print("Please be seated")  
    print("Function will start in 5 minutes")
```

sayHello()  
sayHello()  
sayHello()

Welcome to the show  
Please be seated  
Function will start in 5 minutes

Welcome to the show  
Please be seated  
Function will start in 5 minutes

```
In [9]: shoppingList=["carrot","onion","cabbage","milk","egg"]  
babyShopping=["watch","socks","toys","pen","milk"]  
  
print(shoppingList.index("cabbage")) # Getting the index of cabbage.
```

shoppingList.append("ghee") # Adding an item to list  
print(shoppingList)

shoppingList.pop() # Removing the last element from the list.  
print(shoppingList)

shoppingList.insert(3,"coffee") # insert coffee at 3rd index.  
print(shoppingList)

shoppingList.extend(babyShopping) # Adding a new list to old list.  
print(shoppingList)

print(shoppingList.count("milk")) # Count an item in the list.

shoppingList.reverse() # Reversing the list.  
print(shoppingList)

shoppingList.remove("cabbage") # Removing an item from the list  
print(shoppingList)

shoppingList.clear() # Clear the items in the list  
print(shoppingList)

2  
['carrot', 'onion', 'cabbage', 'milk', 'egg', 'ghee']  
['carrot', 'onion', 'cabbage', 'milk', 'egg']  
['carrot', 'onion', 'cabbage', 'coffee', 'milk', 'egg']  
['carrot', 'onion', 'cabbage', 'coffee', 'milk', 'egg', 'watch', 'socks', 'toys', 'pen', 'milk']  
2  
['milk', 'pen', 'toys', 'socks', 'watch', 'egg', 'milk', 'coffee', 'cabbage', 'onion', 'carrot']  
['milk', 'pen', 'toys', 'socks', 'watch', 'egg', 'milk', 'coffee', 'onion', 'cabbage', 'carrot']  
['milk', 'pen', 'toys', 'watch', 'egg', 'milk', 'coffee', 'onion', 'cabbage', 'carrot']  
['egg', 'milk', 'cabbage', 'onion', 'carrot']  
[]

### Membership operator in list

```
In [6]: car=["ferari","fiat","oddy","bmw","jaguar"]  
  
if "bmw" in car:  
    print("bmw is available here.")  
else:  
    print("Please choose another car.")
```

bmw is available here.

### TUPLE

```
In [3]: # Same as list  
# Tuple are immutable ->Once created we cant add anything or change.  
# Tuple is fast compared to List.
```

car=("ferari","fiat","oddy","bmw","jaguar")

print(type(car)) # print type of datatype

print(len(car)) # Getting the length of the tuple.

print(car.index("oddy")) # Getting the index of the oddy.

print(car.count("oddy")) # Getting the count of an item in tuple.

<class 'tuple'>

6

2

2

['carrot', 'onion', 'cabbage', 'milk', 'egg']

5

carrot

cabbage

egg

['carrot', 'onion', 'cabbage']

['carrot', 'onion', 'cabbage']

['cabbage', 'milk', 'egg']

['egg', 'milk', 'cabbage', 'onion', 'carrot']

['egg', 'milk', 'cabbage', 'onion', 'carrot']

[]

### Slicing in tuple

```
In [7]: # Slicing in tuple  
print(car[0]) # Getting the first element in the tuple  
print(car[-1]) # Printing the Last element in the tuple.
```

ferari

jaguar

### Functions in python

```
In [11]: # Creating a basic function  
# Functions are used for avoiding repeated code.  
# Help to organise code.
```

def sayHello():

print("Welcome to the show")

print("Please be seated")

print("Function will start in 5 minutes")

sayHello()

sayHello()

sayHello()

Welcome to the show

Please be seated

Function will start in 5 minutes

Welcome to the show

Please be seated

Function will start in 5 minutes

```
In [9]: shoppingList=["carrot","onion","cabbage","milk","egg"]  
babyShopping=["watch","socks","toys","pen","milk"]  
  
print(shoppingList.index("cabbage")) # Getting the index of cabbage.
```

shoppingList.append("ghee") # Adding an item to list  
print(shoppingList)

shoppingList.pop() # Removing the last element from the list.  
print(shoppingList)

shoppingList.insert(3,"coffee") # insert coffee at 3rd index.  
print(shoppingList)

shoppingList.extend(babyShopping) # Adding a new list to old list.  
print(shoppingList)

print(shoppingList.count("milk")) # Count an item in the list.

shoppingList.reverse() # Reversing the list.  
print(shoppingList)

shoppingList.remove("cabbage") # Removing an item from the list  
print(shoppingList)

shoppingList.clear() # Clear the items in the list  
print(shoppingList)

2  
['carrot', 'onion', 'cabbage', 'milk', 'egg', 'ghee']  
['carrot', 'onion', 'cabbage', 'milk', 'egg']  
['carrot', 'onion', 'cabbage', 'coffee', 'milk', 'egg']  
['carrot', 'onion', 'cabbage', 'coffee', 'milk', 'egg', 'watch', 'socks', 'toys', 'pen', 'milk']  
2  
['milk', 'pen', 'toys', 'socks', 'watch', 'egg', 'milk', 'coffee', 'cabbage', 'onion', 'carrot']  
['milk', 'pen', 'toys', 'socks', 'watch', 'egg', 'milk', 'coffee', 'onion', 'cabbage', 'carrot']  
['milk', 'pen', 'toys', 'watch', 'egg', 'milk', 'coffee', 'onion', 'cabbage', 'carrot']  
['egg', 'milk', 'cabbage', 'onion', 'carrot']  
[]

### Return keyword

```
In [5]: # The return is used to give back result.  
def power(num1,num2):  
    return num1**num2
```

print(power(2,100))

print(power(2,2))

126760600228229401496703205376

4

```
In [ ]:
```

```
In [ ]:
```