python-day2-today

```
# -*- coding: utf-8 -*-
"""Untitled1.ipynb

Automatically generated by Colab.

Original file is located at
https://colab.research.google.com/drive/1I4x4I8R-vXqjj4E8ndFOhbyO_6YGXzzU

#METHODS -

methods are functions that belong to an object and can be used to perform operations or manipulate the data within that object. They are essentially functions defined inside a class and can be used to perform tasks related to the object they are associated with. When a method is called on an object, it can access and modify the internal state of the object.

#LIST

# 1- .append( )
"""

#example1
fruits = ["apple", "banana"]

fruits.append("kiwi")

print(fruits)

#example2

numbers = [4,5,6,4,8,8]

numbers.append(9)

numbers

"""#2. .extend( )  """

#example
```

```
fruits = ["apple", "banana"]

fruits.extend(["kiwi", "orange", "watermellon"])

fruits

#example2

numbers = [1,2,3,4,5,6]

numbers.extend([9,8,7])

numbers

"""#3. .insert ( )

Insert an element at a specified position in the list
"""

fruits = ["apple", "banana"]   #[0 , 1 , 2, .....] positions

fruits.insert(1, "kiwi")

fruits

fruits.insert(0, "cherry")

fruits

numbers = [1,2,3]

numbers.insert(0, 0)

numbers

"""#4. .remove()"""

fruits

fruits.remove("banana")
```

```
fruits

numbers =[1,2,3,4,5,6]

numbers.remove(2)

numbers

"""#5. .pop()

Removes and return the element at the specific position.
"""

fruits

fruits.append("banana")

fruits

fruits.pop(1)

fruits

"""#6.   .clear()

removes all the element from the list
"""

fruits

fruits.clear()

fruits

type(fruits)

"""#7.  .index()"""

fruits = ["banana", "apple", "kiwi", "cherry"]

fruits.index("apple")
```

```
fruits.index("kiwi")

index = fruits.index("kiwi")

index

"""#8. . count()

It returns the number of times a specified value appears in the list
"""

fruits

fruits.count("apple")

list1 = [1,2,3,4,5,6,4,1,2,3,1,2,4,5,6,4,1,2,3,1,1,4,5,6,3,2,1,1,1,2,2,3,2,3,4,5]

list1.count(1)

list1.count(2)

count_of_one = list1.count(1)

count_of_one

"""#9.  .sort()"""

fruits

fruits.sort()

fruits

list1

list1.sort()

list1

print(list1)

"""#10. reverse ()"""
```

```
fruits

fruits.reverse()

fruits

list1.reverse()

print(list1)

"""#11.  .copy()"""

fruits

fruits_copy = fruits.copy()

fruits_copy

"""#12. list() constructor

use to create a list
"""

data = ("apple", "banana", "kiwi")

type(data)

data

new_list = list(data)

new_list

type(new_list)

"""## Touple Methods

#1. count()
"""

fruits
```

```
fruits = ("apple", "banana", "kiwi", "apple")

type(fruits)

fruits.count("apple")

list = (4,5,6,4,1,2,3,4,1,2,3,1,2,4,1,2,3,4,1,1)

list.count(1)

"""#2. index()"""

fruits

fruits.index("kiwi")

"""#3. concatination"""

touple1 = (1,2,3)
touple2 = (4,5,6)

new_touple = touple1 + touple2

new_touple

"""##3. SET { }

#1. .add( )
"""

set1 = {"football", "cricket"}

dictonory1 = {"football": "a game play in ground with big ball", "cricket":" a game played with tennise ball"}

dictonory1["football"]

set1

set1.add("tennise")
```

```
set1

"""#2. .remove( )"""

set1

set1.remove("football")

set1

set1.remove("chess")

"""#3. discard ( )

removes the specified item without raising an error if the item is not fount
"""

set1

set1.discard("chess")

"""#4. pop( )"""

fruits = {"apple", "banana", "cherry", "orange"}

fruits.pop()

fruits.pop(1)

fruits

"""#5. clear ( )"""

fruits.clear()

fruits

"""#6. update ( )"""

set1 = {1,2,3}
set2 = {4,5,6}
```

```
set1.update(set2)

set1

set2

"""#7. copy ()"""

set1

copy_of_set1 = set1.copy()

copy_of_set1

"""## Dictinories {key :value}

#1. .clear()
"""

person = {"name":"harsh", "age":25, "city": "seloo"}

person2 = person.copy()

person.clear()

person

"""#2. .keys()  & values()"""

person2

person2.keys()

person2.values()

person2["age"] = 30

person2

person2.update({"gender":"male"})

person2
```

"""# **## SLICING**

#list slicing
"""

list1 = [10,20,30,40,50,60,70]

print(list1[0:4])

print(list1[:4])

print(list1[2:5])

print(list1[::-1])

list3 = list1[::-1]

list3

list3[0]