Akilesh K

k.akilesh123@gmail.com

Data engineering - Batch 1

Date: 31-01-24

DAY 9-PYTHON-LIST, UNIQUE, JSON

PYTHON LIST

Add

In [1]:
```
PYT = set()
PYT.add('s')
print("Letters are:", PYT)


PYT.add('e')
print("Letters are:", PYT)

PYT.add('s')
print("Letters are:", PYT)
```

```
Letters are: {'s'}
Letters are: {'e', 's'}
Letters are: {'e', 's'}
```

In [2]:
```
myset = set(["a", "b", "c"])
print(myset)


myset.add("d")
print(myset)
```

```
{'a', 'c', 'b'}
{'a', 'c', 'b', 'd'}
```

Clear

In [3]:
```
set1 = {1,2,3,4,5,6}

print("Initial set")
print(set1)

set1.clear()
print("\nSet after using clear() function")
print(set1)
```

```
Initial set
{1, 2, 3, 4, 5, 6}

Set after using clear() function
set()
```

Update

In [4]:
```


test_set = {6, 4, 2, 7, 9}


print("The original set is : " + str(test_set))


up_ele = [1, 5, 10]


test_set.update(up_ele)

print("Set after adding elements : " + str(test_set))
```

```
The original set is : {2, 4, 6, 7, 9}
Set after adding elements : {1, 2, 4, 5, 6, 7, 9, 10}
```

Discard

In [5]:
```
my_set = set([12, 10, 13, 15, 8, 9])


while my_set:
    my_set.discard(max(my_set))
    print(my_set)
```

```
{8, 9, 10, 12, 13}
{8, 9, 10, 12}
{8, 9, 10}
{8, 9}
{8}
set()
```

Difference

In [6]:
```
set1 = set()
set2 = set()

for i in range(5):
    set1.add(i)

for i in range(3,9):
    set2.add(i)

set3 = set1.difference(set2)

print(" Difference of two sets using difference() function")
print(set3)


set3 = set1 - set2

print("\nDifference of two sets using '-' operator")
print(set3)
```

```
 Difference of two sets using difference() function
{0, 1, 2}

Difference of two sets using '-' operator
{0, 1, 2}
```

Union

In [7]:
```
people = {"Jay", "Idrish", "Archil"}
vampires = {"Karan", "Arjun"}
dracula = {"Deepanshu", "Raju"}


population = people.union(vampires)

print("Union using union() function")
print(population)


population = people|dracula

print("\nUnion using '|' operator")
print(population)
```

```
Union using union() function
{'Karan', 'Jay', 'Arjun', 'Idrish', 'Archil'}

Union using '|' operator
{'Deepanshu', 'Jay', 'Idrish', 'Archil', 'Raju'}
```

UNIQUE

In [10]:
```
def unique(list1):

    # initialize a null list
    unique_list = []

    # traverse for all elements
    for x in list1:
        # check if exists in unique_list or not
        if x not in unique_list:
            unique_list.append(x)
    # print list
    for x in unique_list:
        print (x),


# driver code
list1 = [10, 20, 10, 30, 40, 40]
print("the unique values from 1st list is")
unique(list1)


list2 = [1, 2, 1, 1, 3, 4, 3, 3, 5]
print("\nthe unique values from 2nd list is")
unique(list2)
```

```
the unique values from 1st list is
10
20
30
40

the unique values from 2nd list is
1
2
3
4
```

In [11]:
```
import numpy as np

def unique(list1):
    x = np.array(list1)
    print(np.unique(x))

# driver code
list1 = [10, 20, 10, 30, 40, 40]
print("the unique values from 1st list is")
unique(list1)


list2 = [1, 2, 1, 1, 3, 4, 3, 3, 5]
print("\nthe unique values from 2nd list is")
unique(list2)
```

```
the unique values from 1st list is
[10 20 30 40]

the unique values from 2nd list is
[1 2 3 4 5]
```

In [12]:
```
list1 = [10, 20, 10, 30, 40, 40]

list2 = [1, 2, 1, 1, 3, 4, 3, 3, 5]


unique_list_1 = list(dict.fromkeys(list1))

unique_list_2 = list(dict.fromkeys(list2))

print(unique_list_1,unique_list_2,sep="\n")
```

```
[10, 20, 30, 40]
[1, 2, 3, 4, 5]
```

In [16]:
```
def myFunc(e):
  return e['year']

cars = [
  {'car': 'Ford', 'year': 2005},
  {'car': 'Mitsubishi', 'year': 2000},
  {'car': 'BMW', 'year': 2019},
  {'car': 'VW', 'year': 2011}
]

cars.sort(key=myFunc)
print(cars)
```

```
[{'car': 'Mitsubishi', 'year': 2000}, {'car': 'Ford', 'year': 2005}, {'car': 'VW', 'year': 2011}, {'car': 'BMW', 'year': 2019}]
```

JSON

In [1]:
```
import json

# Define JSON string
jsonString = '{ "id": 121, "name": "Naveen", "course": "MERN Stack"}'

# Convert JSON String to Python
student_details = json.loads(jsonString)

# Print Dictionary
print(student_details)

# Print values using keys
print(student_details['name'])
print(student_details['course'])
```

```
{'id': 121, 'name': 'Naveen', 'course': 'MERN Stack'}
Naveen
MERN Stack
```

LOAD

In [3]:
```
import json

# JSON string
json_string = '{"Name": "Suezen", "age": 23, "Course": "DSA"}'

# Convert JSON string to dictionary
json_dict = json.loads(json_string)

print(json_dict)
```

```
{'Name': 'Suezen', 'age': 23, 'Course': 'DSA'}
```

In [4]:
```
import json

# JSON string
employee ='{"id":"09", "name": "Nitin", "department":"Finance"}'

# Convert string to Python dict
employee_dict = json.loads(employee)
print(employee_dict)

print(employee_dict['name'])
```

```
{'id': '09', 'name': 'Nitin', 'department': 'Finance'}
Nitin
```

DUMPS

In [5]:
```
import json

# Data to be written
dictionary = {
  "id": "04",
  "name": "sunil",
  "department": "HR"
}

# Serializing json
json_object = json.dumps(dictionary, indent = 4)
print(json_object)
```

```
{
    "id": "04",
    "name": "sunil",
    "department": "HR"
}
```

In [6]:
```
import json

# Data to be written
dictionary ={
    "name" : "sathiyajith",
    "rollno" : 56,
    "cgpa" : 8.6,
    "phonenumber" : "9976770500"
}

with open("sample.json", "w") as outfile:
    json.dump(dictionary, outfile)
```

PRETTY PRINT

In [7]:
```
import json

# JSON string
employee ='{"id":"09", "name": "Nitin", "department":"Finance"}'

# Convert string to Python dict
employee_dict = json.loads(employee)

# Pretty Printing JSON string back
print(json.dumps(employee_dict, indent = 4, sort_keys= True))
```

```
{
    "department": "Finance",
    "id": "09",
    "name": "Nitin"
}
```