Lab 5

August 22, 2023

[1]:
```
Days = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday",␣
 ↪"Sunday"}
print(Days)
print(type(Days))
print("looping through the set elements ... ")
for i in Days:
    print(i)
```

```
{'Monday', 'Tuesday', 'Saturday', 'Friday', 'Sunday', 'Wednesday', 'Thursday'}
<class 'set'>
looping through the set elements …
Monday
Tuesday
Saturday
Friday
Sunday
Wednesday
Thursday
```

[2]:
```
Days = set(["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday",␣
 ↪"Sunday"])
print(Days)
print(type(Days))
print("looping through the set elements ... ")
for i in Days:
    print(i)
```

```
{'Saturday', 'Wednesday', 'Tuesday', 'Thursday', 'Monday', 'Friday', 'Sunday'}
<class 'set'>
looping through the set elements …
Saturday
Wednesday
Tuesday
Thursday
Monday
Friday
Sunday
```

[3]:
```
set5 = {1,2,4,4,5,8,9,9,10}
print("Return set with unique elements:",set5)
```

```
Return set with unique elements: {1, 2, 4, 5, 8, 9, 10}
```

[5]:
```
Months = set(["January","February", "March", "April", "May", "June"])
print("\nprinting the original set ... ")
print(Months)
print("\nAdding other months to the set...");
Months.add("July");
Months.add ("August");
print("\nPrinting the modified set...");
print(Months)
print("\nlooping through the set elements ... ")
for i in Months:
    print(i)
```

```
printing the original set …
{'April', 'June', 'February', 'January', 'March', 'May'}

Adding other months to the set…

Printing the modified set…
{'April', 'June', 'July', 'February', 'January', 'August', 'March', 'May'}

looping through the set elements …
April
June
July
February
January
August
March
May
```

[6]:
```
Months = set(["January","February", "March", "April", "May", "June"])
print("\nprinting the original set ... ")
print(Months)
print("\nupdating the original set ... ")
Months.update(["July","August","September","October"]);
print("\nprinting the modified set ... ")
print(Months);
```

```
printing the original set …
{'April', 'June', 'February', 'January', 'March', 'May'}

updating the original set …
```

```
printing the modified set …
{'October', 'April', 'June', 'July', 'February', 'September', 'January',
'August', 'March', 'May'}
```

[7]:
```
months = set(["January","February", "March", "April", "May", "June"])
print("\nprinting the original set ... ")
print(months)
print("\nRemoving some months from the set...");
months.discard("January");
months.discard("May");
print("\nPrinting the modified set...");
print(months)
print("\nlooping through the set elements ... ")
for i in months:
    print(i)
```

```
printing the original set …
{'April', 'June', 'February', 'January', 'March', 'May'}

Removing some months from the set…

Printing the modified set…
{'April', 'June', 'February', 'March'}

looping through the set elements …
April
June
February
March
```

[8]:
```
months = set(["January","February", "March", "April", "May", "June"])
print("\nprinting the original set ... ")
print(months)
print("\nRemoving some months from the set...");
months.remove("January");
months.remove("May");
print("\nPrinting the modified set...");
print(months)
```

```
printing the original set …
{'April', 'June', 'February', 'January', 'March', 'May'}

Removing some months from the set…

Printing the modified set…
{'April', 'June', 'February', 'March'}
```

[9]:
```
Months = set(["January","February", "March", "April", "May", "June"])
print("\nprinting the original set ... ")
print(Months)
print("\nRemoving some months from the set...");
Months.pop();
Months.pop();
print("\nPrinting the modified set...");
print(Months)
```

```
printing the original set …
{'April', 'June', 'February', 'January', 'March', 'May'}

Removing some months from the set…

Printing the modified set…
{'February', 'January', 'March', 'May'}
```

[10]:
```
Months = set(["January","February", "March", "April", "May", "June"])
print("\nprinting the original set ... ")
print(Months)
print("\nRemoving all the items from the set...");
Months.clear()
print("\nPrinting the modified set...")
print(Months)
```

```
printing the original set …
{'April', 'June', 'February', 'January', 'March', 'May'}

Removing all the items from the set…

Printing the modified set…
set()
```

[11]:
```
Months = set(["January","February", "March", "April", "May", "June"])
print("\nprinting the original set ... ")
print(Months)
print("\nRemoving items through discard() method...");
Months.discard("Feb"); #will not give an error although the key feb is not␣
 ↪available in the set
print("\nprinting the modified set...")
print(Months)
print("\nRemoving items through remove() method...");
Months.remove("Jan") #will give an error as the key jan is not available in the␣
 ↪set.
print("\nPrinting the modified set...")
print(Months)
```

```
printing the original set …
{'April', 'June', 'February', 'January', 'March', 'May'}

Removing items through discard() method…

printing the modified set…
{'April', 'June', 'February', 'January', 'March', 'May'}

Removing items through remove() method…
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Input In [11], in <cell line: 9>()
      7 print(Months)
      8 print("\nRemoving items through remove() method…");
----> 9 Months.remove("Jan") #will give an error as the key jan is not available
     ↪in the set.
     10 print("\nPrinting the modified set…")
     11 print(Months)

KeyError: 'Jan'
```

[12]:
```
Days1 = {"Monday",  "Tuesday", "Wednesday", "Thursday"}
Days2 = {"Monday", "Tuesday"}
Days3 = {"Monday", "Tuesday", "Friday"}

#Days1 is the superset of Days2 hence it will print true.
print (Days1>Days2)

#prints false since Days1 is not the subset of Days2
print (Days1<Days2)

#prints false since Days2 and Days3 are not equivalent
print (Days2 == Days3)
```

```
True
False
False
```

[13]:
```
Frozenset = frozenset([1,2,3,4,5])
print(type(Frozenset))
print("\nprinting the content of frozen set...")
for i in Frozenset:
    print(i);
Frozenset.add(6) #gives an error since we cannot change the content of
 ↪Frozenset after creation
```

```
<class 'frozenset'>

printing the content of frozen set…
1
2
3
4
5
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Input In [13], in <cell line: 6>()
      4 for i in Frozenset:
      5     print(i);
----> 6 Frozenset.add(6)

AttributeError: 'frozenset' object has no attribute 'add'
```

[14]:
```
Days1 = {"Monday","Tuesday","Wednesday","Thursday"}
Days2 = {"Friday","Saturday","Sunday"}
print(Days1.union(Days2)) #printing the union of the sets
# Create three sets
set1 = {1, 2, 3}
set2 = {2, 3, 4}
set3 = {3, 4, 5}

# Find the common elements between the three sets
common_elements = set1.union(set2, set3)

# Print the common elements
print(common_elements)
set1 = {"Devansh","John", "David", "Martin"}
set2 = {"Steve", "Milan", "David", "Martin"}
print(set1.intersection(set2)) #prints the intersection of the two sets
a = {"Devansh", "bob", "castle"}
b = {"castle", "dude", "emyway"}
c = {"fuson", "gaurav", "castle"}

a.intersection_update(b, c)

print(a)
Days1 = {"Monday",  "Tuesday", "Wednesday", "Thursday"}
Days2 = {"Monday", "Tuesday", "Sunday"}
print(Days1.difference(Days2)) # prints the difference of the two sets Days1␣
 ↪and Days2
a = {1,2,3,4,5,6}
```

```
b = {1,2,9,8,10}
c = a.symmetric_difference(b)
print(c)
Days1 = {"Monday",  "Tuesday", "Wednesday", "Thursday"}
Days2 = {"Monday", "Tuesday"}
Days3 = {"Monday", "Tuesday", "Friday"}

#Days1 is the superset of Days2 hence it will print true.
print (Days1>Days2)

#prints false since Days1 is not the subset of Days2
print (Days1<Days2)

#prints false since Days2 and Days3 are not equivalent
print (Days2 == Days3)
```

```
{'Monday', 'Tuesday', 'Saturday', 'Friday', 'Sunday', 'Wednesday', 'Thursday'}
{1, 2, 3, 4, 5}
{'Martin', 'David'}
{'castle'}
{'Wednesday', 'Thursday'}
{3, 4, 5, 6, 8, 9, 10}
True
False
False
```

[15]:
```
Employee = {"Name": "Johnny", "Age": 32, "salary":26000,"Company":"^TCS"}
print(type(Employee))
print("printing Employee data .... ")
print(Employee)
```

```
<class 'dict'>
printing Employee data …
{'Name': 'Johnny', 'Age': 32, 'salary': 26000, 'Company': '^TCS'}
```

[18]:
```
# Creating an empty Dictionary
Dict = {}
print("Empty Dictionary: ")
print(Dict)

# Creating a Dictionary
# with dict() method
Dict = dict({1: 'Hcl', 2: 'WIPRO', 3:'Facebook'})
print("\nCreate Dictionary by using  dict(): ")
print(Dict)

# Creating a Dictionary
# with each item as a Pair
```

```
Dict = dict([(4, 'Rinku'), (2, "Singh")])
print("\nDictionary with each item as a pair: ")
print(Dict)
```

```
Empty Dictionary:
{}

Create Dictionary by using  dict():
{1: 'Hcl', 2: 'WIPRO', 3: 'Facebook'}

Dictionary with each item as a pair:
{4: 'Rinku', 2: 'Singh'}
```

[19]:
```
Employee = {"Name": "Dev", "Age": 20, "salary":45000,"Company":"WIPRO"}
print(type(Employee))
print("printing Employee data .... ")
print("Name : %s" %Employee["Name"])
print("Age : %d" %Employee["Age"])
print("Salary : %d" %Employee["salary"])
print("Company : %s" %Employee["Company"])
```

```
<class 'dict'>
printing Employee data …
Name : Dev
Age : 20
Salary : 45000
Company : WIPRO
```

[20]:
```
# Creating an empty Dictionary
Dict = {}
print("Empty Dictionary: ")
print(Dict)

# Adding elements to dictionary one at a time
Dict[0] = 'Peter'
Dict[2] = 'Joseph'
Dict[3] = 'Ricky'
print("\nDictionary after adding 3 elements: ")
print(Dict)

# Adding set of values
# with a single Key
# The Emp_ages doesn't exist to dictionary
Dict['Emp_ages'] = 20, 33, 24
print("\nDictionary after adding 3 elements: ")
print(Dict)

# Updating existing Key's Value
```

```
Dict[3] = 'JavaTpoint'
print("\nUpdated key value: ")
print(Dict)
```

```
Empty Dictionary:
{}

Dictionary after adding 3 elements:
{0: 'Peter', 2: 'Joseph', 3: 'Ricky'}

Dictionary after adding 3 elements:
{0: 'Peter', 2: 'Joseph', 3: 'Ricky', 'Emp_ages': (20, 33, 24)}

Updated key value:
{0: 'Peter', 2: 'Joseph', 3: 'JavaTpoint', 'Emp_ages': (20, 33, 24)}
```

[21]:
```
Employee={"Name":"John","Age":29,"Salary":25000,"Company":"WIPRO","Name":
"John"}
for x,y in Employee.items():
        print(x,y)
```

```
Name John
Age 29
Salary 25000
Company WIPRO
```

[22]:
```
Employee = {"Name": "John", "Age": 29, "salary":26000,"Company":
 ↪"WIPRO",[100,201,301]:"Department ID"}
for x,y in Employee.items():
    print(x,y)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [22], in <cell line: 1>()
----> 1 Employee = {"Name": "John", "Age": 29, "salary":26000,"Company":
 ↪"WIPRO",[100,201,301]:"Department ID"}
      2 for x,y in Employee.items():
      3     print(x,y)

TypeError: unhashable type: 'list'
```

[23]:
```
d=dict()
for x in range(1,16):
    d[x]=x**2
print(d)
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121,
12: 144, 13: 169, 14: 196, 15: 225}
```

[24]:
```
keys = ['red', 'green', 'blue']
values = ['#FF0000','#008000', '#0000FF']
color_dictionary = dict(zip(keys, values))
print(color_dictionary)
```

```
{'red': '#FF0000', 'green': '#008000', 'blue': '#0000FF'}
```

[25]:
```
L = [{"V":"S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"}, {"VII":
 ↪"S005"}, {"V":"S009"},{"VIII":"S007"}]
print("Original List: ",L)
u_value = set( val for dic in L for val in dic.values())
print("Unique Values: ",u_value)
```

```
Original List:  [{'V': 'S001'}, {'V': 'S002'}, {'VI': 'S001'}, {'VI': 'S005'},
{'VII': 'S005'}, {'V': 'S009'}, {'VIII': 'S007'}]
Unique Values:  {'S009', 'S002', 'S005', 'S007', 'S001'}
```

[ ]:
```

```