

ASSIGNMENT-3

NAME: PRATHIKSHA HARISH BALERI

USN:4CB22CB044

DOMAIN:DATA SCIENCE

DICTIONARY:

#DICTIONARY IN PYTHON

```
d1={101:"pooja",102:"sanjay",103:"harshitha"}
```

#HERE WE CAN SEE THE KEY-VALUE PAIRS

#KEYS SHOULD BE UNIQUE

```
print(d1)
```

Output:

```
{101: 'pooja', 102: 'sanjay', 103: 'harshitha'}
```

```
s1={1:'tally',2:'oracle',3:'java',4:'python',5:'vba',6:'autocad'}
```

```
print(s1)
```

Output:

```
{1: 'tally', 2: 'oracle', 3: 'java', 4: 'python', 5: 'vba', 6: 'autocad'}
```

#Updating Dictionary

```
s1={1:'tally',2:'oracle',3:'java',4:'python',5:'vba',6:'autocad'}
```

```
print(s1)
```

```
s1[1]='networking'
```

```
print(s1)
```

```
s1[5]='MD'
```

```
print(s1)
```

```
{1: 'tally', 2: 'oracle', 3: 'java', 4: 'python', 5: 'vba', 6: 'autocad'}
```

```
{1: 'networking', 2: 'oracle', 3: 'java', 4: 'python', 5: 'vba', 6: 'autocad'}
```

```
{1: 'networking', 2: 'oracle', 3: 'java', 4: 'python', 5: 'MD', 6: 'autocad'}
```

#remove entry with key(Deleting the Data from the Dictionary)

```
s1={1:'networking',2:'oracle',3:'java',4:'python',5:'vba',6:'autocad'}
```

```
print(s1)
```

Output

```
{1: 'networking', 2: 'oracle', 3: 'java', 4: 'python', 5: 'vba', 6: 'autocad'}
```

```
{2: 'oracle', 3: 'java', 4: 'python', 5: 'vba', 6: 'autocad'}
```

```
del s1[1]
print(s1)
```

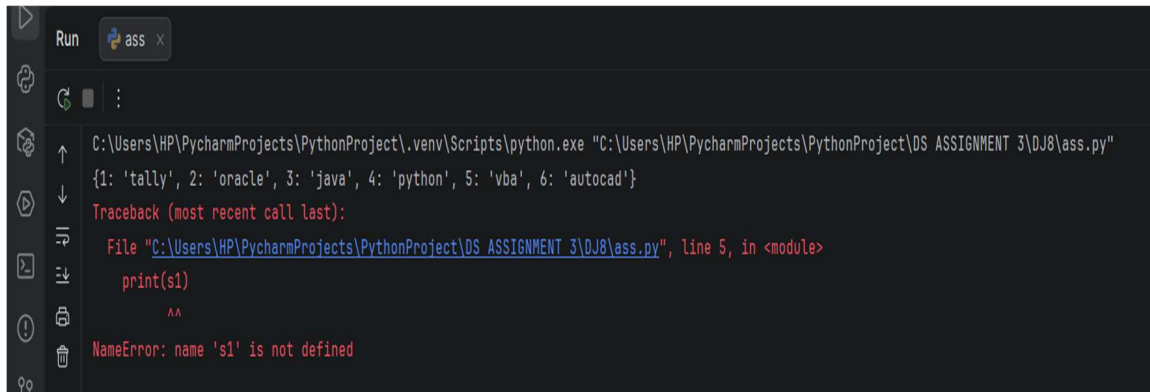
```
#remove all entries(deletes only the data, NOT the dictionary)
s1={1:'tally',2:'oracle',3:'java',4:'python',5:'vba',6:'autocad'}
print(s1)
s1.clear()
print(s1)
```

Output

```
{1: 'tally', 2: 'oracle', 3: 'java', 4: 'python', 5: 'vba', 6: 'autocad'}
{}

```

```
#delete entire dictionary(deletes both the data and the dictionary)
s1={1:'tally',2:'oracle',3:'java',4:'python',5:'vba',6:'autocad'}
print(s1)
del s1
#print(s1)
#THIS THROWS AN ERROR SAYING NOT AVAILABLE
```

A screenshot of a Python IDE's Run window. The top bar shows 'Run' and a file named 'ass.py'. The main area displays the command prompt output: the file path 'C:\Users\HP\PycharmProjects\PythonProject\.venv\Scripts\python.exe "C:\Users\HP\PycharmProjects\PythonProject\DS ASSIGNMENT 3\DJ8\ass.py"', followed by the dictionary '{1: 'tally', 2: 'oracle', 3: 'java', 4: 'python', 5: 'vba', 6: 'autocad'}'. Below this is a 'Traceback (most recent call last):' section showing the file path and line 5, and the error message 'NameError: name 's1' is not defined' with '^' markers pointing to the 'print(s1)' line in the code.

```
C:\Users\HP\PycharmProjects\PythonProject\.venv\Scripts\python.exe "C:\Users\HP\PycharmProjects\PythonProject\DS ASSIGNMENT 3\DJ8\ass.py"
{1: 'tally', 2: 'oracle', 3: 'java', 4: 'python', 5: 'vba', 6: 'autocad'}
Traceback (most recent call last):
  File "C:\Users\HP\PycharmProjects\PythonProject\DS ASSIGNMENT 3\DJ8\ass.py", line 5, in <module>
    print(s1)
    ^
NameError: name 's1' is not defined
```

```
#SLICING CANNOT BE DONE IN DICTIONARY
#BUT IT IS POSSIBLE ONLY IN LISTS
```

```
#FIND LENGTH
s1={1:'tally',2:'oracle',3:'java',4:'python',5:'VBA',6:'CN'}
print(s1)
print(len(s1))
```

Output

```
{1: 'tally', 2: 'oracle', 3: 'java', 4: 'python', 5: 'VBA', 6: 'CN'}
```

```
6
```

#one more way to find length

```
s1={1:'tally',2:'oracle',3:'java',4:'python',5:'vba',6:'CN'}
```

```
print(s1)
```

```
x=len(s1)
```

```
print(x)
```

Output

```
{1: 'tally', 2: 'oracle', 3: 'java', 4: 'python', 5: 'VBA', 6: 'CN'}
```

```
6
```

#copying data

```
s1={1:'tally',2:'oracle',3:'java',4:'python',5:'vba',6:'autocad'}
```

```
print(s1)
```

```
s2=s1.copy()
```

```
print(s2)
```

Output

```
{1: 'tally', 2: 'oracle', 3: 'java', 4: 'python', 5: 'vba', 6: 'autocad'}
```

```
{1: 'tally', 2: 'oracle', 3: 'java', 4: 'python', 5: 'vba', 6: 'autocad'}
```

#here 2 dictionaries are copied

#USING get()

```
s1={1:'tally',2:'oracle',3:'java',4:'python',5:'vba',6:'autocad'}
```

```
print(s1)
```

```
print(s1.get(3))
```

```
print(s1[3])
```

Output

```
{1: 'tally', 2: 'oracle', 3: 'java', 4: 'python', 5: 'vba', 6: 'autocad'}
```

```
java
```

```
java
```

```
#USING values() & items()
s1={1:'tally',2:'oracle',3:'java',4:'python',5:'vba',6:'autocad'}
print(s1)
print(s1.items())
```

Output

```
{1: 'tally', 2: 'oracle', 3: 'java', 4: 'python', 5: 'vba', 6: 'autocad'}
dict_items([(1, 'tally'), (2, 'oracle'), (3, 'java'), (4, 'python'), (5, 'vba'), (6, 'autocad')])
```

```
s1={1:'tally',2:'oracle',3:'java',4:'python',5:'vba',6:'autocad'}
print(s1)
print(s1.values())
# keys() to get the unique keys
print(s1.keys())
```

Output

```
{1: 'tally', 2: 'oracle', 3: 'java', 4: 'python', 5: 'vba', 6: 'autocad'}
dict_values(['tally', 'oracle', 'java', 'python', 'vba', 'autocad'])
dict_keys([1, 2, 3, 4, 5, 6])
```

```
s1={1:'tally',2:'oracle',3:'java',4:'python',5:'vba',6:'autocad'}
print(s1)
print(len(s1.keys()))
```

#Another method to display the data of the dictionary data --USING "for" loop

```
s1={1:'tally',2:'oracle',3:'java',4:'python',5:'vba',6:'autocad'}
print(s1)
for i in s1:
    print(s1.get(i))
```

Output:

```
{1: 'tally', 2: 'oracle', 3: 'java', 4: 'python', 5: 'vba', 6: 'autocad'}
```

6

```
{1: 'tally', 2: 'oracle', 3: 'java', 4: 'python', 5: 'vba', 6: 'autocad'}
```

tally

oracle

java

python

vba

autocad

#concatenate or merging 2 dictionaries

```
s1={1:'tally',2:'oracle',3:'java',4:'python',5:'vba',6:'autocad'}
```

```
print(s1)
```

```
s2={7:'networking',8:'cn',9:'cyber security'}
```

```
print(s2)
```

```
s3=s1.update(s2)
```

```
print(s1)
```

Output:

```
{1: 'tally', 2: 'oracle', 3: 'java', 4: 'python', 5: 'vba', 6: 'autocad'}
```

```
{7: 'networking', 8: 'cn', 9: 'cyber security'}
```

```
{1: 'tally', 2: 'oracle', 3: 'java', 4: 'python', 5: 'vba', 6: 'autocad', 7: 'networking', 8: 'cn', 9: 'cyber security'}
```

#converting all dictionary data to List

```
s1={1:'tally',2:'oracle',3:'java',4:'python',5:'vba',6:'autocad'}
```

```
print(s1)
```

```
x1=list(s1.values())
```

```
print(x1)
```

Output:

```
{1: 'tally', 2: 'oracle', 3: 'java', 4: 'python', 5: 'vba', 6: 'autocad'}
```

```
{7: 'networking', 8: 'cn', 9: 'cyber security'}
```

```
{1: 'tally', 2: 'oracle', 3: 'java', 4: 'python', 5: 'vba', 6: 'autocad', 7: 'networking', 8: 'cn', 9: 'cyber security'}
```

#converting all dictionary data to Tuple

```
s1={1:'tally',2:'oracle',3:'java',4:'python',5:'vba',6:'autocad'}
```

```
print(s1)
```

```
x1=tuple(s1.values())
```

```
print(x1)
```

Output:

```
{1: 'tally', 2: 'oracle', 3: 'java', 4: 'python', 5: 'vba', 6: 'autocad'}
```

```
('tally', 'oracle', 'java', 'python', 'vba', 'autocad')
```

#ASSIGNMENT

#convert the dictionary in reverse

```
s1={1:'tally',2:'oracle',3:'java',4:'python',5:'vba',6:'autocad'}
```

```
s1=list(s1.values())
```

```
print(s1[::-1])
```

#here I converted the dictionary to list and performed slicing

Output:

```
['autocad', 'vba', 'python', 'java', 'oracle', 'tally']
```

DJ9 FILE

SETS

```
s1={"tally","oracle","java","python","vba","autocad"}  
print(s1)
```

Output:

```
{'python', 'java', 'tally', 'autocad', 'vba', 'oracle'}
```

#printing using for loop

```
s1={"tally","oracle","java","python","vba","autocad"}  
for i in s1:  
    print(i)  
print(type(s1))
```

Output:

autocad

tally

oracle

python

vba

java

<class 'set'>

#converting the set data into list data

```
s1={"tally","oracle","java","python","vba","autocad"}  
print(s1)  
s2=([ "tally","oracle","java","python","vba","autocad"])  
print(s2)
```

```
s1={"tally","oracle","java","python","vba","autocad"}  
print(s1)  
s1.add("networking")  
print(s1)
```

Output:

```
{'java', 'vba', 'autocad', 'tally', 'python', 'oracle'}  
['tally', 'oracle', 'java', 'python', 'vba', 'autocad']  
{'java', 'vba', 'autocad', 'tally', 'python', 'oracle'}  
{'java', 'vba', 'autocad', 'networking', 'tally', 'python', 'oracle'}
```

#add to set and using for loop

```
s1={"tally","oracle","java","python","vba","autocad"}  
print(s1)  
s1.add("networking")  
for i in s1:  
    print(i)
```

Output:

```
{'java', 'tally', 'vba', 'autocad', 'python', 'oracle'}  
java  
tally  
networking  
vba  
autocad  
python  
oracle
```



```
#To add more than one item in the set, Python provides the update() method.--USING SET()
s1=set(['tally','oracle','java','python','vba','autocad'])
print(s1)
s1.update(['networking','ccna','mcse','aws'])
print(s1)
```

Output:

```
{'python', 'java', 'vba', 'oracle', 'autocad', 'tally'}
```

```
{'python', 'java', 'mcse', 'aws', 'networking', 'vba', 'ccna', 'oracle', 'autocad', 'tally'}
```

#To add more than one item in the set, Python provides the update() method--DIRECTLY INTO THE SET

```
s1={'tally','oracle','java','python','vba','autocad'}
print(s1)
s1.update(['networking','ccna','mcse','aws'])
print(s1)
```

Output:

```
{'python', 'oracle', 'java', 'tally', 'vba', 'autocad'}
```

```
{'oracle', 'java', 'python', 'aws', 'vba', 'ccna', 'autocad', 'tally', 'mcse', 'networking'}
```

#Removing items from the set--ONLY ONE DATA AT TIME

```
#discard()
s1={'tally','oracle','java','python','vba','autocad'}
print(s1)
s1.discard('oracle')
print(s1)
```

Output:

```
{'java', 'python', 'oracle', 'tally', 'vba', 'autocad'}
```

```
{'java', 'python', 'tally', 'vba', 'autocad'}
```

```
print(s1)
```

```
s1.pop()
```

```
print(s1)
```

Output:

```
{'tally', 'vba', 'oracle', 'autocad', 'python', 'java'}
```

```
{'vba', 'oracle', 'autocad', 'python', 'java'}
```

```
#clear()
```

```
s1={'tally','oracle','java','python','vba','autocad'}
```

```
print(s1)
```

```
s1.clear()
```

```
print(s1)
```

Output:

```
{'oracle', 'python', 'vba', 'tally', 'autocad', 'java'}
```

```
set()
```

```
#use of union()
```

```
s1={'tally','oracle','java','python','vba','autocad'}
```

```
print(s1)
```

```
s2={'python','vba','autocad'}
```

```
print(s1.union(s2))
```

Output:

```
{'tally', 'python', 'autocad', 'oracle', 'vba', 'java'}
```

```
{'java', 'tally', 'python', 'oracle', 'vba', 'autocad'}
```

```
s1={'tally','oracle','java','python','vba','autocad'}
s2={'python','vba','autocad'}
print(s1|s2)
print(s1&s2)
```

Output:

```
{'tally', 'java', 'python', 'oracle', 'autocad', 'vba'}
{'python', 'autocad', 'vba'}
```

#use of intersection and operator

```
s1={'tally','oracle','java','python','vba','autocad'}
print(s1)
s2={'python','vba','autocad'}
print(s1.intersection(s2))
```

Output:

```
{'autocad', 'python', 'java', 'vba', 'tally', 'oracle'}
{'autocad', 'vba', 'python'}
```

#difference

```
s1={'tally','oracle','java','python','vba','autocad'}
s2={'python','vba','autocad'}
print(s1-s2)
print(s1.difference(s2))
```

Output:

```
{'java', 'oracle', 'tally'}
{'java', 'oracle', 'tally'}
```