

```
'''
Statring at 9:05 pm
Enjoy the song :)
'''
```

```
#Introduction to Dictionary
'''
- Store a key value pair
- Key is Unique , Values may or may not be unique
- Iterable
- Dictionary is Mutable
- Not ordered (not indexed)
- Heterogeneous
- But the key is immutable
'''
```

```
#Initialising a Dictionary
```

```
'''
- using {}
'''
d={} #empty dictionary
print(d)
print(type(d))
```

```
'''
- using dict()
'''
di=dict() #empty
print(di)
print(type(di))
```

```
{ }
<class 'dict'>
{ }
<class 'dict'>
```

```
d={'a':1,'b':1}

# dictionary d, keys -> a and b, values -> 1,1

print(d)

{'a': 1, 'b': 1}
```

```
d={'a':1,'a':10,'a':30}
#dictionary cant have duplicate keys, latest calue is updated
# dictionary d, keys -> a and b, values -> 1,1

print(d)

{'a': 30}
```

```
#Dictionary cannot have mutable keys
a={1,2,3}:'abc'
#key -> [1,2,3] -> list is mutable
#value -> 'abc'
```

```
-----
--
TypeError                                Traceback (most recent call
last)
/tmp/ipython-input-2310457895.py in <cell line: 0>()
      1 #Dictionary cannot have mutable keys
----> 2 a={1,2,3}:'abc'
      3 #key -> [1,2,3] -> list is mutable
      4 #value -> 'abc'
```

Next steps: [Explain error](#)

```
a={(1,2,3)}:'abc'
#key -> (1,2,3) -> tuple is immutable
#value -> 'abc'
print(a)
```

```
{(1, 2, 3): 'abc'}
```

```
#Dictionary is not ordered
nickname={
    "Akanksha":"Mickey",
    "Ajay":"AJ",
    "Mehul":"Monu",
    "Vignesh":"Vicky",
    "Ketan":"KD"
}
print(nickname)
print(nickname[1]) #dictionary does not have indexing
```

```
{'Akanksha': 'Mickey', 'Ajay': 'AJ', 'Mehul': 'Monu', 'Vignesh': 'Vicky',
```

```
-----
--
KeyError                                Traceback (most recent call
last)
/tmp/ipython-input-52480483.py in <cell line: 0>()
      8 }
      9 print(nickname)
----> 10 print(nickname[1])
```

```
KeyError: 1
```

Next steps: [Explain error](#)

```
#Accessing value from the dictionary using a key
print(nickname["Mehul"]) #dictionary[key]
print(nickname["Vignesh"])
```

```
print(nickname["ketan"]) #key is not present -> error
```

```
Monu
Vicky
```

```
--
KeyError                                Traceback (most recent call
last)
/tmp/ipython-input-96477889.py in <cell line: 0>()
      2 print(nickname["Mehul"]) #dictionary[key]
      3 print(nickname["Vignesh"])
----> 4 print(nickname["ketan"])
```

Next steps: [Explain error](#)

```
# get function, if value is not present it gives None instead of error
print(nickname.get("Akanksha"))
print(nickname.get("akanksha")) #prints None
print(nickname.get("akanksha","Name not found")) #dict_name.get(key,"Defa
print(nickname.get("Ketan","Name not found"))
```

```
Mickey
None
Name not found
KD
```

```
#Inserting values in Dictionary
```

```
a={True:'a','a':1,2:False,3.14:'pi'}
print(a)
#keys -> True, 'a',2,3.14
#values -> 'a',1,False,'pi'

#insert -> dict_name[key]=value
nickname["Rahul"]="sharma"
print(nickname)
print(nickname.get("Rahul","Not found"))
nickname["Anvesh"]="Anu"
print(nickname)
```

```
u', 'Vignesh': 'Vicky', 'Ketan': 'KD', 'Rahul': 'sharma'}
```

```
u', 'Vignesh': 'Vicky', 'Ketan': 'KD', 'Rahul': 'sharma', 'Anvesh': 'Anu'}
```

```
#Updating value from the dictionary using a key
nickname["Anvesh"]="anu"
print(nickname)
```

```
#only values can be updated
```

```
u', 'Vignesh': 'Vicky', 'Ketan': 'KD', 'Rahul': 'sharma', 'Anvesh': 'anu'}
```

```
# Length of a Dictionary -> number of keys
print(len(nickname))
```

```
7
```

```
#Delete a key from a Dictionary
'''
- using pop()
- using del
'''
print(nickname.pop("Saritha"))
```

```
-----
--
KeyError                                Traceback (most recent call
last)
/tmp/ipython-input-746746216.py in <cell line: 0>()
      4 - using del
      5 '''
----> 6 print(nickname.pop("Saritha"))
```

Next steps: [Explain error](#)

```
#dict.pop(key,default_value)
print(nickname.pop("Saritha","Not found, cant delete"))
print(nickname.pop("Akanksha","Not found, cant delete"))
print(nickname)
```

```
Not found, cant delete
Mickey
{'Ajay': 'AJ', 'Mehul': 'Monu', 'Vignesh': 'Vicky', 'Ketan': 'KD', 'Rahul':
```

```
del nickname["Ketan"]
print(nickname)
```

```
{'Ajay': 'AJ', 'Mehul': 'Monu', 'Vignesh': 'Vicky', 'Rahul': 'sharma', 'An
```

```
#Get keys and values of Dictionary
# keys() -> gives all keys of the dictionary
print(nickname.keys())
# values() -> gives all values of the dictionary
print(nickname.values())
print(nickname)
```

```
dict_keys(['Ajay', 'Mehul', 'Vignesh', 'Rahul', 'Anvesh'])
dict_values(['AJ', 'Monu', 'Vicky', 'sharma', 'anu'])
{'Ajay': 'AJ', 'Mehul': 'Monu', 'Vignesh': 'Vicky', 'Rahul': 'sharma', 'An
```

```
#Iterating a Dictionary
#Way 1 -> dict_name.keys -> all keys
for key in nickname.keys():
    print(key,"'s nickname is ",nickname[key])

#value -> dict_name[key]
```

```
Ajay 's nickname is  AJ
Mehul 's nickname is  Monu
Vignesh 's nickname is  Vicky
Rahul 's nickname is  sharma
Anvesh 's nickname is  anu
```

```
#Way 2 -> dict_name.items -> all key-value pairs
for key,value in nickname.items():
    print(key,"'s nickname is ",value)
```

```
Ajay 's nickname is  AJ
Mehul 's nickname is  Monu
Vignesh 's nickname is  Vicky
Rahul 's nickname is  sharma
Anvesh 's nickname is  anu
```

```
#in keyword to check presence of a key
# key in dict_name
"Guna" in nickname
```

```
False
```

```
"Vignesh" in nickname
```

```
True
```

```
"Vicky" in nickname #value cant be searched using in
```

```
False
```

```
# Quiz 1
a = {'a': 'A'}
print(type(a))
```

```
<class 'dict'>
```

```
# Quiz 2
d = {'a': 1, 'b': 2, 'c': 3}
print(d.get(1)) #None
print(d[1]) # 1 is not in the keys -> KeyError
# keys -> a,b,c
```

None

```

-----
--
KeyError                                Traceback (most recent call
last)
/tmp/ipython-input-365604623.py in <cell line: 0>()
      2 d = {'a': 1, 'b': 2, 'c': 3}
      3 print(d.get(1))
----> 4 print(d[1]) # 1 is not in the keys
      5 # keys -> a,b,c

```

```

# Quiz 3
a = {'Scaler': 1}
# {'Scaler': 1} -> len -> 1
a.pop('Scaler') # {} -> empty dictionary -> 0
print(len(a))

```

0

#Count frequency of characters

'''

Take a string as input, and print the frequencies
of every unique character of the string.

input - "Akanksha Gaur"

A - 1

k - 2

a - 3

n - 1

s - 1

h - 1

- 1

G - 1

u - 1

r - 1

'''

name=input() # Akanksha Gaur

dict_freq={} #{}

for char in name:

dict_freq[char]=dict_freq.get(char,0)+1 # 0+1=1, 1+1=2, 2+1=3

#dict_name[key]=value

print(dict_freq)

'''

dict_freq = {'A':1, 'k':2, 'a':3, 'n':1, 's':1, 'h':1, ' ':1, 'G':1, 'u':1, 'r':1}

sha Gaur

1, 'k': 2, 'a': 3, 'n': 1, 's': 1, 'h': 1, ' ': 1, 'G': 1, 'u': 1, 'r': 1}

```

Break : 10:15 pm - 10:25 pm

```

```

#Sets
'''
- unique only values
- unordered
- heterogeneous
- immutable data
'''

s={1,2,3,4,5,6}
print(s)
print(type(s))

se=set()
print(type(se))

```

```

{1, 2, 3, 4, 5, 6}
<class 'set'>
<class 'set'>

```

```

colors = {'red', 'green', 'yellow'}
print(colors)
print(colors[1]) #sets are unordered

```

```

{'green', 'yellow', 'red'}

```

```

--
TypeError                                Traceback (most recent call
last)
/tmp/ipython-input-255355831.py in <cell line: 0>()
      1 colors = {'red', 'green', 'yellow'}
      2 print(colors)
----> 3 print(colors[1]) #sets are unordered

```

Next steps: [Explain error](#)

```

# Insert, Update, Delete in Sets
#add(value) -> insert values in teh set (one value at a time)
colors.add('black')
print(colors)

#update(list) - add multiple values
li=['white','blue','black']
colors.update(li)
print(colors)

#delete
#remove(value)
colors.remove('blue')
print(colors)

```

```
{'green', 'black', 'yellow', 'blue', 'white', 'red'}
{'green', 'black', 'yellow', 'blue', 'white', 'red'}
{'green', 'black', 'yellow', 'white', 'red'}
```

```
# Length and Iterating over a Set
print(len(colors))
```

```
for c in colors:
    print(c)
```

```
#in operator
"pink" in colors
```

```
5
green
black
yellow
white
red
False
```

```
name_set={'a','f','d','s','a','f','d','s'}
print(name_set)
print(len(name_set))
```

```
{'f', 'd', 'a', 's'}
4
```

```
# Intersection Union and Difference
```

```
food_I_like={"chicken","gol gappe","biryani","noodles","sushi","pizza","a
food_is_expensive={"pizza","sushi","caviar"}
```

```
#Intersection -> common
print(food_I_like & food_is_expensive)
```

```
#Union -> all
print(food_I_like | food_is_expensive)
```

```
#Difference -> A-B
print(food_I_like - food_is_expensive)
```

```
{'sushi', 'pizza'}
{'gol gappe', 'avocado', 'caviar', 'biryani', 'sushi', 'chicken', 'noodles'
{'avocado', 'gol gappe', 'biryani', 'chicken', 'noodles'}
```

```
# Quiz 4
```

```
a = {1, 2, 3}
b = {3, 4, 5}
print(a - b) # {1, 2}
print(a.union(b)) # {1, 2, 3, 4, 5}
print(a.intersection(b)) #{3}
```



```
{1, 2}
{1, 2, 3, 4, 5}
{3}
```

```
#Count unique words
"This is a sentence. This is not a paragraph."
{"This", "is", "a", "sentence.", "paragraph.", "not"}
# 6

sentence=input() #This is a sentence. This is not a paragraph.
words=sentence.split(" ") #[This, is, a, sentence., This, is, not, a, par
word_set=set(words) #{'paragraph.', 'a', 'sentence.', 'not', 'This', 'is'}
print(word_set)
print(len(word_set))
```

```
This is a sentence. This is not a paragraph.
{'paragraph.', 'a', 'sentence.', 'not', 'This', 'is'}
6
```

```
#Design a game FLAMES

F - Friends
L - Love
A - Affair
M - Marriage
E - Enemy
S - Siblings

#take input
boy = "aakar sharma"
girl = "disha patani"

#take only unique values in both the names
boy_unique={a,k,r,s,h,m}
girl_unique={d,i,s,h,a,p,t,n}

#find common -> intersection
common={a,s,h}

#after removing common
#boy-common, girl - common
boy={k,r,m} -> 3
girl={d,i,t,p,n} -> 5

#len(boy)+len(girl)
score= 8 -> Love
mod operator -> 8 % 6 = 2 #get the relationship

# Code
```

'Frequency of Characters'

