

# Full stack web development using python

dict



Saurabh Shukla (MySirG)

# Agenda

- ① dict introduction
- ② creating dict object
- ③ Accessing dict elements
- ④ built-in methods
- ⑤ Concatenation and Repetition
- ⑥ Comparison operator
- ⑦ dict object methods
- ⑧ dict comprehension
- ⑨ user input

## dict

dict is a class

dict is mutable

dict is not hashable

dict is iterable

dict is not a sequence

dict cannot have duplicate keys (not data values)

indexing is not applicable to dict object

slicing operators is not applicable

dict elements are pair of key-value and  
data-value

one dict  
element is  
(key, data)

## How to create dict object?

Rollno      Student name

102	"Rahul"
105	"Payal"
106	"Arjun"
107	"Prachi"

di = {102: 'Rahul', 105: 'Payal', 106: 'Arjun', 107: 'Prachi'}

di = {}      #empty object

di = dict (a=10, b=20, c=30)

key	data
'a'	10
'b'	20
'c'	30

## Accessing dict elements

$di = \{102: 'Rahul', 105: 'Payal', 106: 'Arjun', 107: 'Prachi'\}$

- ① `print(di)`
- ② `print(di[102], di[105], di[106], di[107])`
- ③ `for k in di:`  
 `print(k)` → only Keys
- ④ `for k in di:`  
 `print(k, di[k])`

## How to edit dict element ?

Editing dict element means you want to change data-value of the element and not the key-value

dict Object [key-value] = newdataValue

```
del di[102]
```

How to add new element in the dict?

dictObject [new-key-value] = data-value

## methods

items() → Collection dict elements

keys() → Collection of keys only of the elements

values() → Collection of data-values only of the dict elements

All these methods are dict class attributes

## built-in methods

len()

min()

max()

sum()

sorted()

## Concatenation and Repetition Operator

dict + dict

not supported

dict \* int

not supported

## Comparison Operator

$d1 > d2$   
 $d1 \geq d2$   
 $d1 < d2$   
 $d1 \leq d2$

} not supported

$d1 == d2$   
 $d1 != d2$

} supported

Two dict objects are equal if their items are equal. Elements can be stored in any order.

## dict object methods

pop(key)

popitem()

clear()

## dict Comprehension

di = { Key-expression : data-expression for v in Seq }

user input

10 9 8 7 6 5 4 3 2 1

i=10

while i>0:

    print(i, end=' ')

    i-=1

i=1

while i<=10:

    print(11-i, end=' ')

    i+=1

1 3 5 7 9 11 13 15 17 19

19 17 15 13 11 9 7 5 3 1

i=19

while i>=1:

    print(i)

    i-=2

1 4 9 16 25 36 49 64 81 100

i = 1

while i <= 10:

print(i\*\*2, end=' ')

i += 1

5 10 15 20 25 30 35 40 45 50

i = 1

while i <= 10:

print(i \* 5, end=' ')

i += 1

1 MySirG  
2 MySirG  
3 MySirG

$S \leftarrow "MyS\pi Gr"$

$S \leftarrow \begin{bmatrix} \cdot & \cdot & -1 \\ \cdot & \cdot & \end{bmatrix}$

$\leftarrow "Gris y M"$