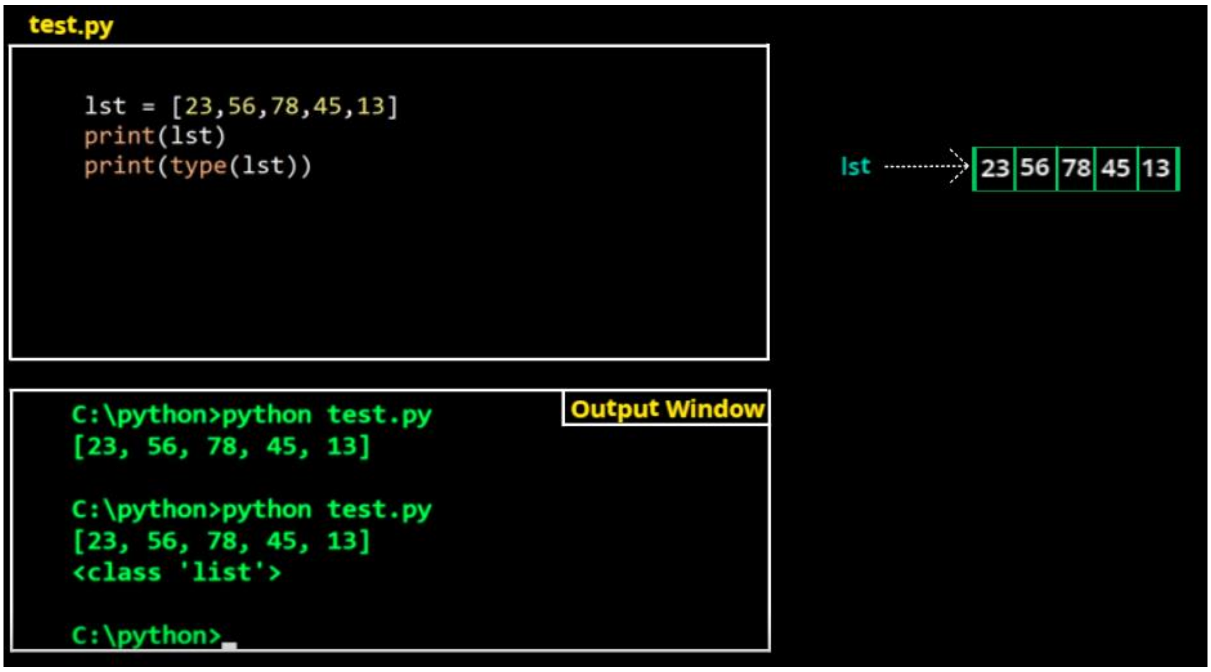
Python Fundamentals day 4

Today's Agenda

- List
- Tuple
- Set
- Dictionary

Lists

A list object is an ordered collection of one or more data items, not necessarily of the same type, put in square brackets.



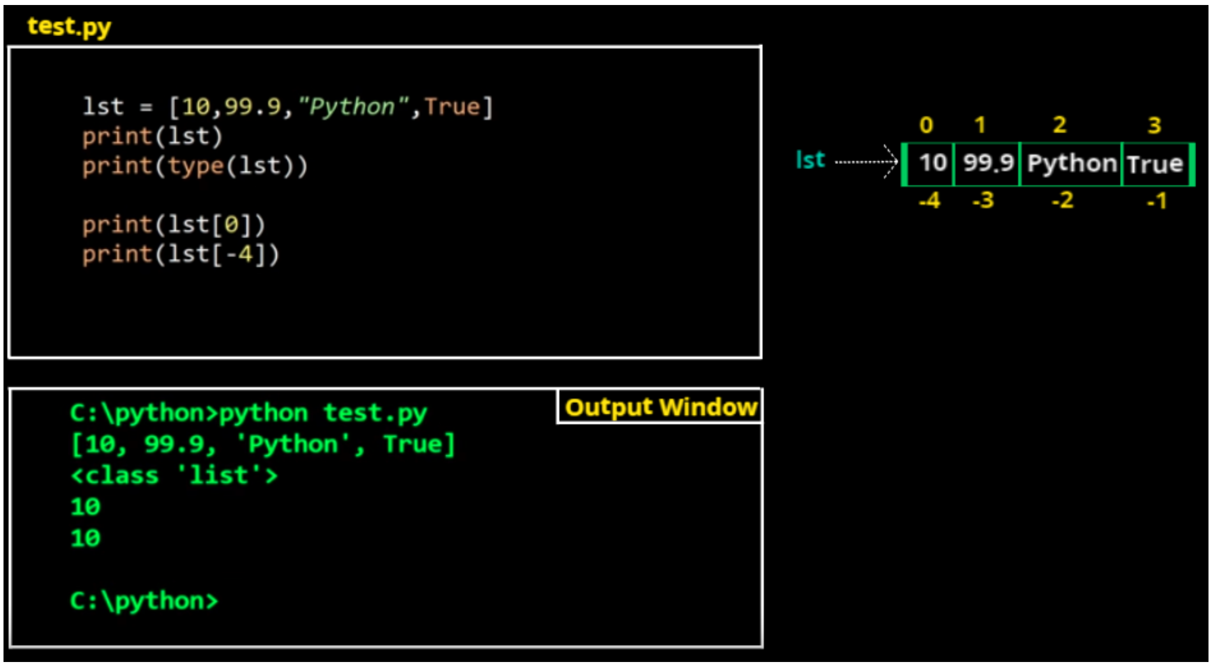
test.py

```
lst = [23,56,78,45,13]
print(lst)
print(type(lst))
```

Output Window

```
C:\python>python test.py
[23, 56, 78, 45, 13]

C:\python>python test.py
[23, 56, 78, 45, 13]
<class 'list'>

C:\python>_
```

- We can access single elements from list by using positive indices or negative.



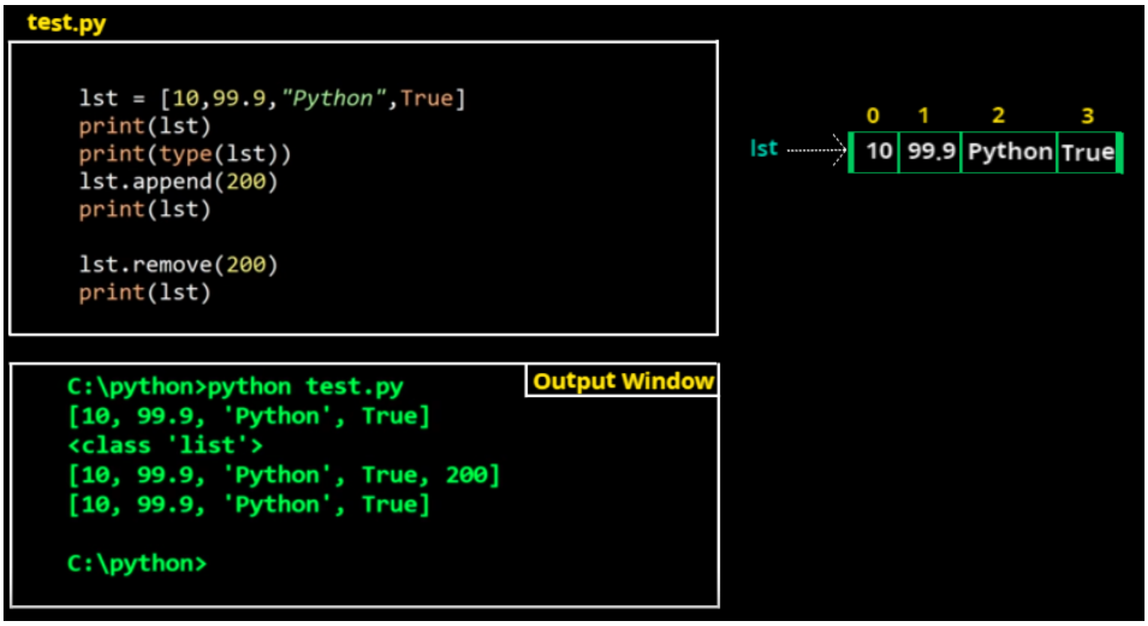
test.py

```
lst = [10,99.9,"Python",True]
print(lst)
print(type(lst))

print(lst[0])
print(lst[-4])
```

Output Window

```
C:\python>python test.py
[10, 99.9, 'Python', True]
<class 'list'>
10
10

C:\python>
```

- **Lists are mutable**, which means we can append elements and remove whenever needed. By using append() and remove() respectively.



test.py

```
lst = [10,99.9,"Python",True]
print(lst)
print(type(lst))
lst.append(200)
print(lst)

lst.remove(200)
print(lst)
```

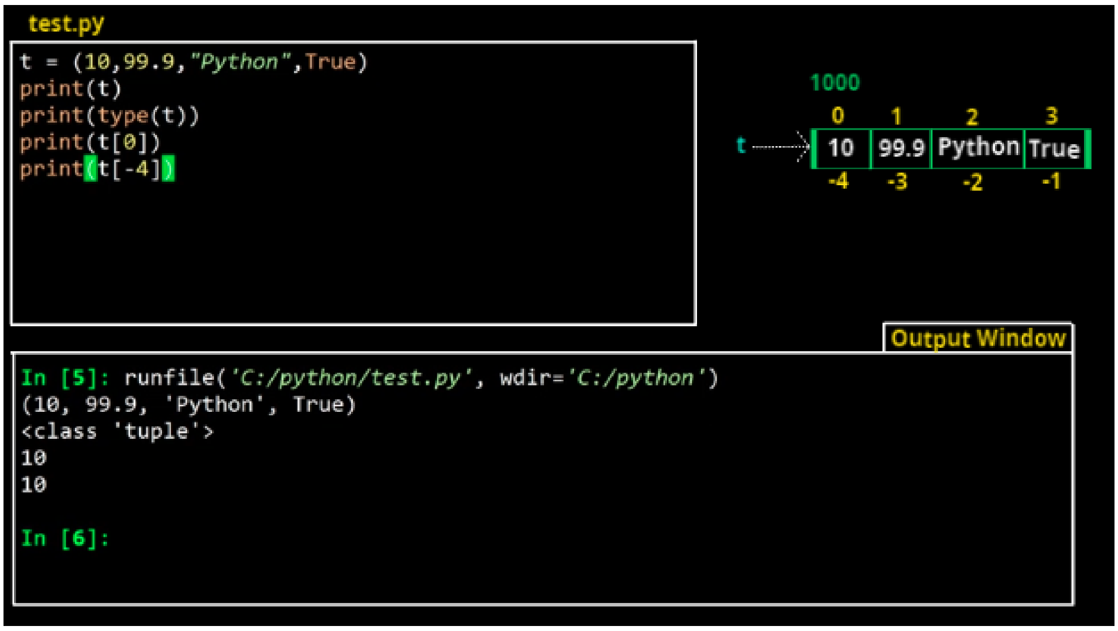
Output Window

```
C:\python>python test.py
[10, 99.9, 'Python', True]
<class 'list'>
[10, 99.9, 'Python', True, 200]
[10, 99.9, 'Python', True]

C:\python>
```

Tuple

A Tuple object is an ordered collection of one or more data items, not necessarily of the same type, put in parentheses.

- Like lists in tuple also we can access single elements with both positive and negative indices.



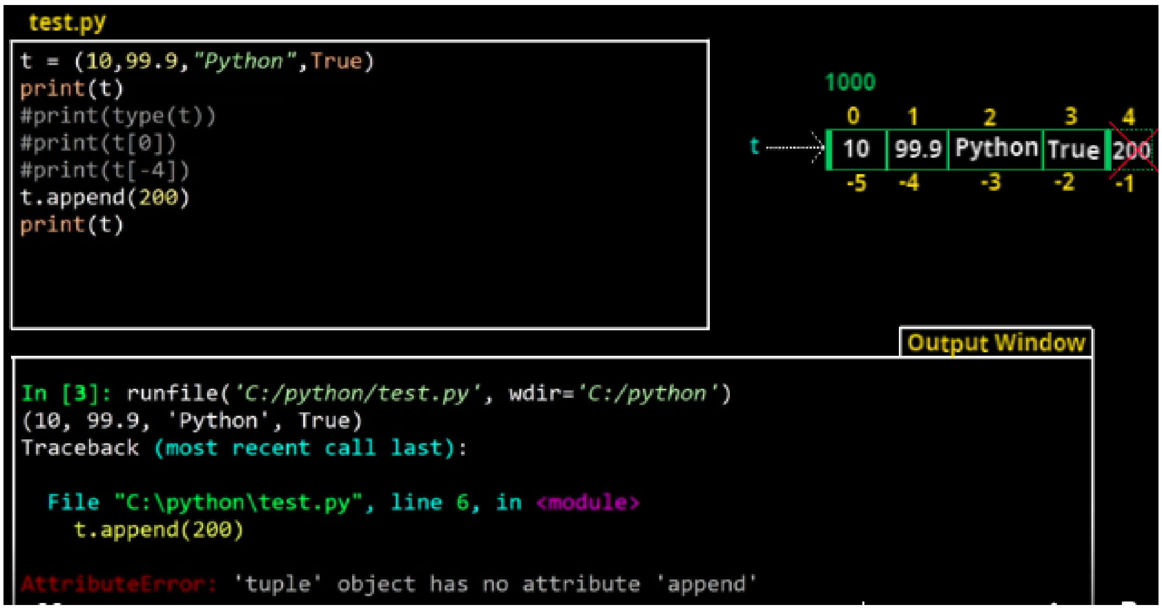
test.py

```
t = (10,99.9,"Python",True)
print(t)
print(type(t))
print(t[0])
print(t[-4])
```

Output Window

```
In [5]: runfile('C:/python/test.py', wdir='C:/python')
(10, 99.9, 'Python', True)
<class 'tuple'>
10
10

In [6]:
```

- **Tuple are immutable**, which means we cannot append or remove any elements from a tuple.



test.py

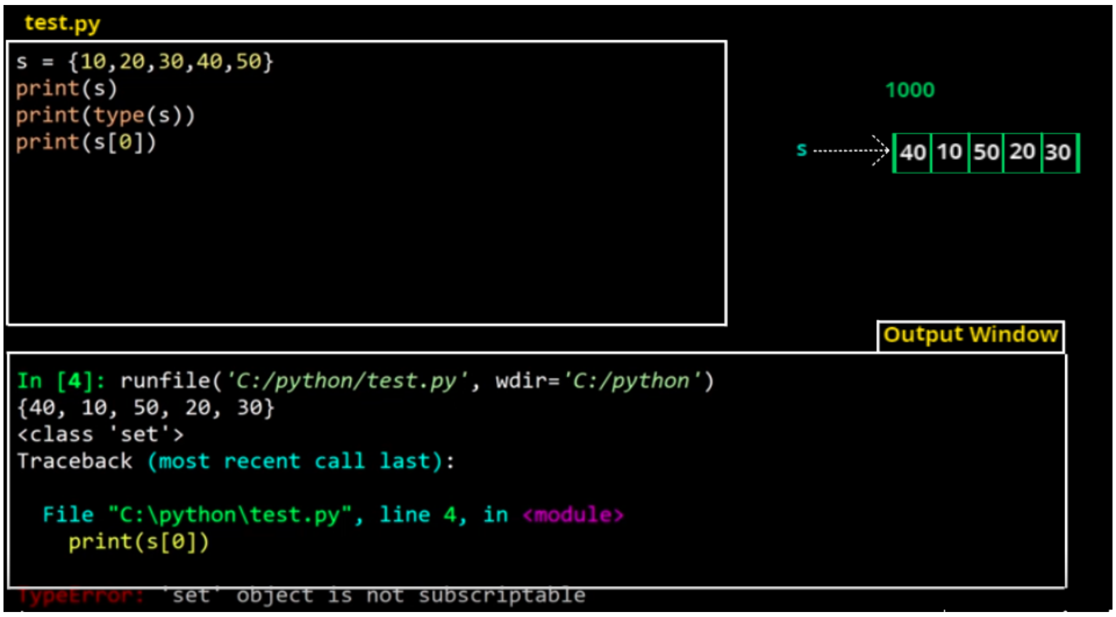
```
t = (10,99.9,"Python",True)
print(t)
#print(type(t))
#print(t[0])
#print(t[-4])
t.append(200)
print(t)
```

Output Window

```
In [3]: runfile('C:/python/test.py', wdir='C:/python')
(10, 99.9, 'Python', True)
Traceback (most recent call last):

  File "C:\python\test.py", line 6, in <module>
    t.append(200)

AttributeError: 'tuple' object has no attribute 'append'
```

Set

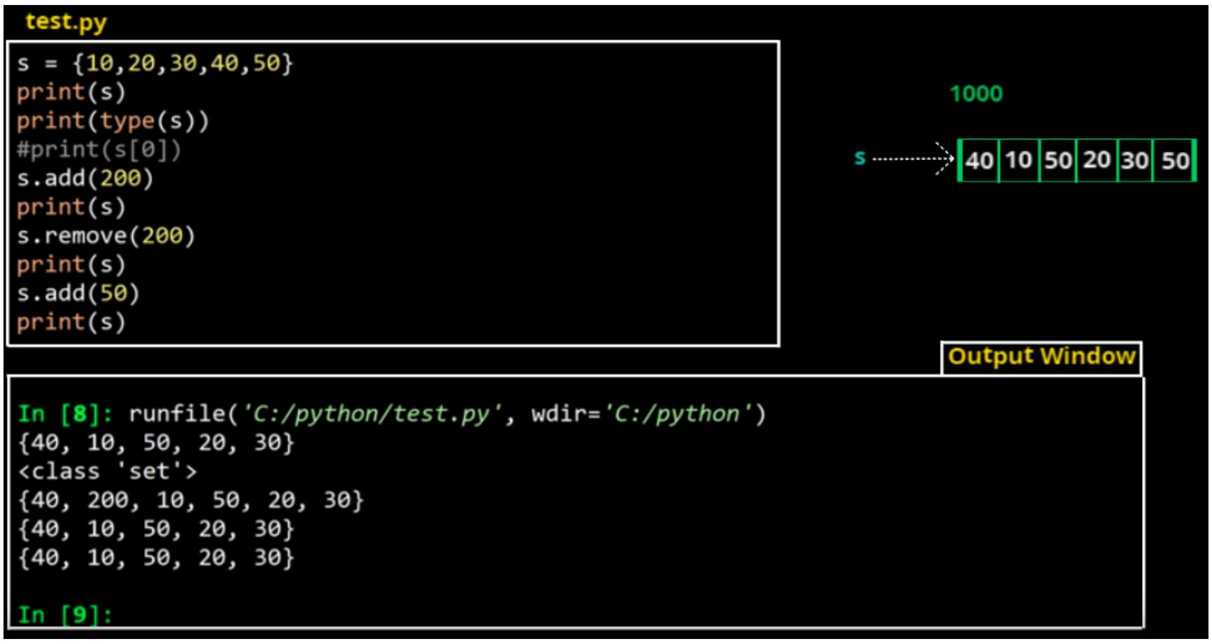
- In set we cannot access single elements because internally set does not store the data in the order we provide it. That is the reason it is called unordered collection of data. And as it is unordered data, certainly it doesn't have indices.



```
s = {10,20,30,40,50}
print(s)
print(type(s))
print(s[0])
```

```
In [4]: runfile('C:/python/test.py', wdir='C:/python')
{40, 10, 50, 20, 30}
<class 'set'>
Traceback (most recent call last):

  File "C:\python\test.py", line 4, in <module>
    print(s[0])

TypeError: 'set' object is not subscriptable
```

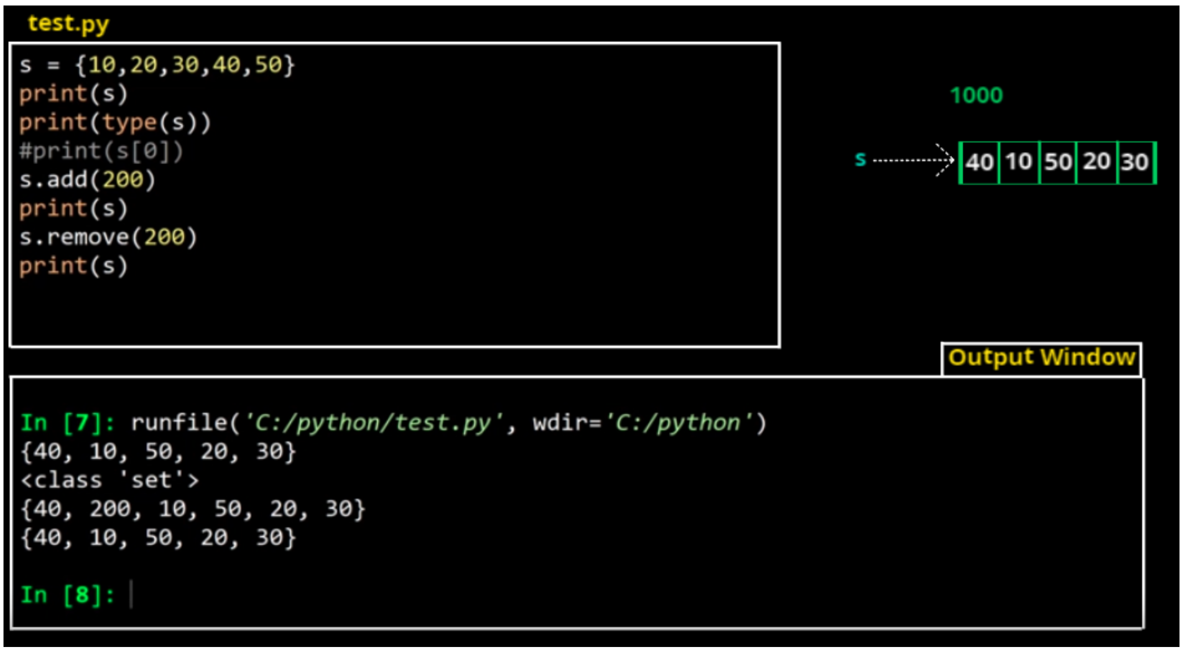
- **Set is mutable**, which means we can add or remove the elements using add() and remove() respectively. But remember that the order in which it adds an element is not as we enter it.



```
s = {10,20,30,40,50}
print(s)
print(type(s))
#print(s[0])
s.add(200)
print(s)
s.remove(200)
print(s)
s.add(50)
print(s)
```

```
In [8]: runfile('C:/python/test.py', wdir='C:/python')
{40, 10, 50, 20, 30}
<class 'set'>
{40, 200, 10, 50, 20, 30}
{40, 10, 50, 20, 30}
{40, 10, 50, 20, 30}

In [9]:
```

- **No duplication**, which means the data inside set are unique. Although duplicated entries will not give an error, the entries will not get stored and be seen on screen.



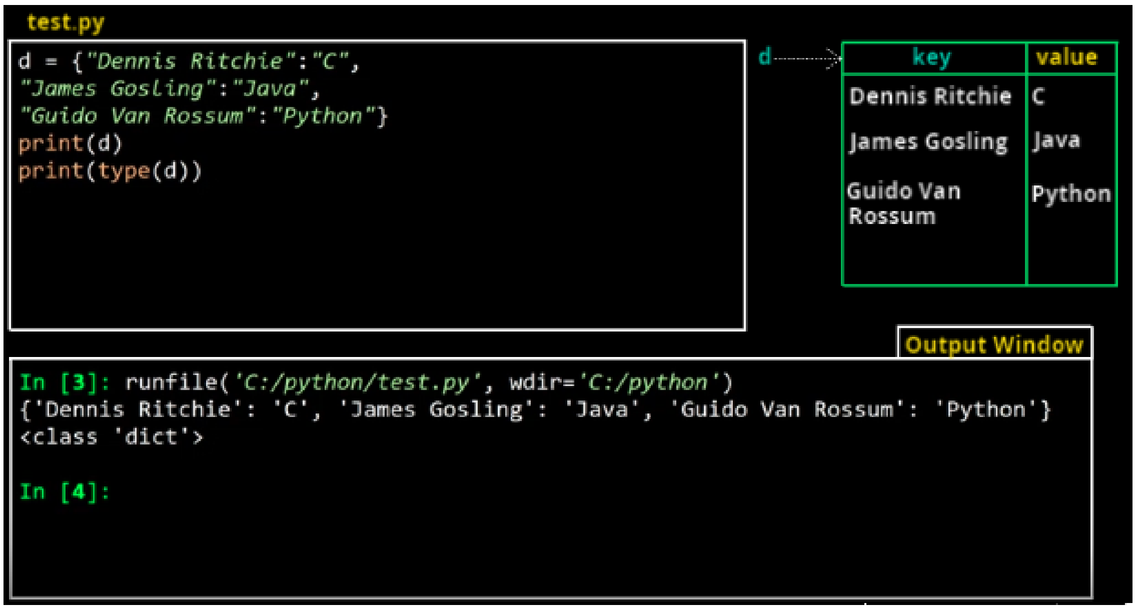
test.py

```
s = {10,20,30,40,50}
print(s)
print(type(s))
#print(s[0])
s.add(200)
print(s)
s.remove(200)
print(s)
```

```
In [7]: runfile('C:/python/test.py', wdir='C:/python')
{40, 10, 50, 20, 30}
<class 'set'>
{40, 200, 10, 50, 20, 30}
{40, 10, 50, 20, 30}

In [8]:
```

Dictionary

A dictionary is a collection which is unordered, changeable and indexed. In Python dictionaries are written with curly brackets, and they have keys and values. Each key will be mapped to a value.



test.py

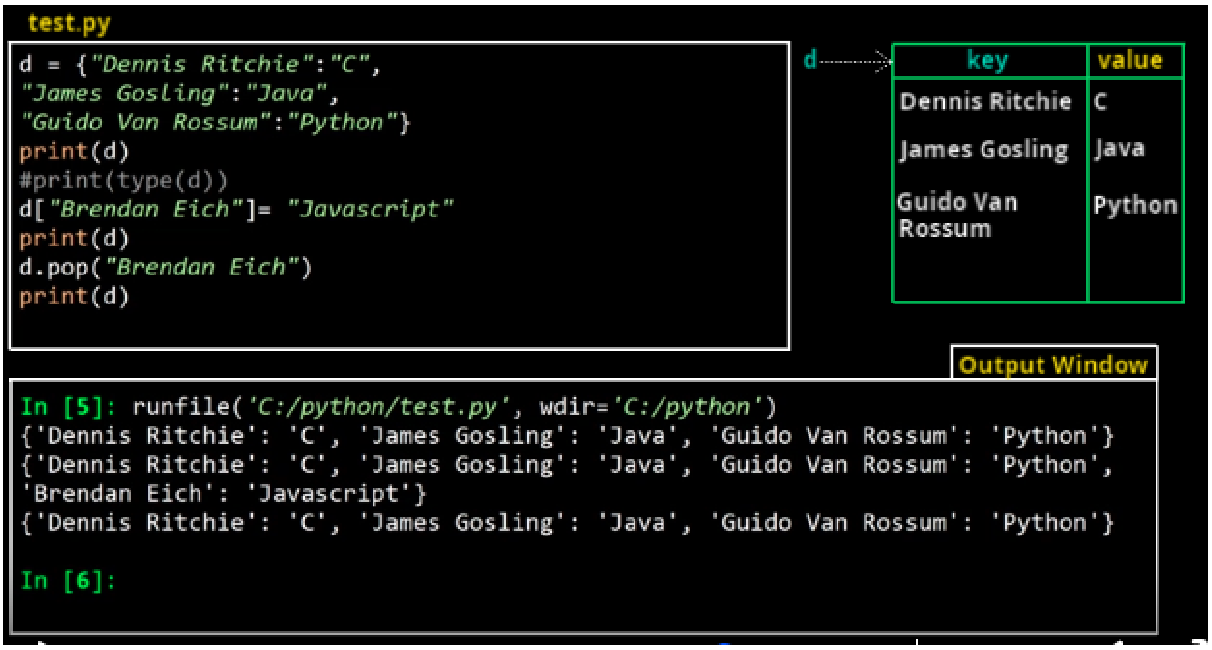
```
d = {"Dennis Ritchie":"C",
"James Gosling":"Java",
"Guido Van Rossum":"Python"}
print(d)
print(type(d))
```

d

key	value
Dennis Ritchie	C
James Gosling	Java
Guido Van Rossum	Python

```
In [3]: runfile('C:/python/test.py', wdir='C:/python')
{'Dennis Ritchie': 'C', 'James Gosling': 'Java', 'Guido Van Rossum': 'Python'}
<class 'dict'>

In [4]:
```

Dictionaries are mutable - which means we can add and remove keys. A key can be removed by using pop( ). Many other operations can be done as well, which we shall learn in future sessions.



```
d = {"Dennis Ritchie":"C",
"James Gosling":"Java",
"Guido Van Rossum":"Python"}
print(d)
#print(type(d))
d["Brendan Eich"]= "Javascript"
print(d)
d.pop("Brendan Eich")
print(d)
```

key	value
Dennis Ritchie	C
James Gosling	Java
Guido Van Rossum	Python

```
In [5]: runfile('C:/python/test.py', wdir='C:/python')
{'Dennis Ritchie': 'C', 'James Gosling': 'Java', 'Guido Van Rossum': 'Python'}
{'Dennis Ritchie': 'C', 'James Gosling': 'Java', 'Guido Van Rossum': 'Python',
'Brendan Eich': 'Javascript'}
{'Dennis Ritchie': 'C', 'James Gosling': 'Java', 'Guido Van Rossum': 'Python'}

In [6]:
```