

RANGE

```
In [ ]: '''Range data type represents an immutable sequence of numbers.
it is most commonly used for looping a specific number of times in for loop.
instead of storing all the numbers in memory like list, a range object generates
the numbers on the fly as you need them, which makes it incredibly memory efficient
# syntax : range() fun can take up to three arguments
range(start,stop,step)
start : optional the starting value by default 0.
stop : required..> the number to stop at.(the range goes up to ,but does not include
step : optional the increment value(how much to skip)default is 1.'''
```

```
In [ ]: #memory efficient
'''a range object takes the same amount of memory regardless of the size
of the range it represents.
whether you have range(10) or range(100),
python only stores start,stop and step values.

immutability ----you can not change range object, once you created

indexing and slicing
even though it is not a list you can access specific elements or slice it.

range can also convert in to list.'''
```

```
In [1]: # basic range
range(5)
```

```
Out[1]: range(0, 5)
```

```
In [3]: #converting range to list
list(range(5))
```

```
Out[3]: [0, 1, 2, 3, 4]
```

```
In [4]: #only stop(n)
# prints (n-1)
for i in range(5):
    print(i)
```

```
0
1
2
3
4
```

```
In [5]: # start and stop

for i in range(0,10):
    print(i)
```

0
1
2
3
4
5
6
7
8
9

In [6]: *# start, stop and step*

```
for i in range(0,20,2):  
    print(i)
```

0
2
4
6
8
10
12
14
16
18

In [9]: *# reverse range(negative indexing):to print range in reverse order start should*

```
for i in range(10,0,-1):  
    print(i)
```

10
9
8
7
6
5
4
3
2
1

In [10]:

```
for i in range(10,0,-2):  
    print(i)
```

10
8
6
4
2

In [21]: *# even is immutalable we can access elements using indexing and slicing*

```
r = range(10)  
print(list(r))  
print(r[8])
```

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
8

In [18]:

```
r[0]
```

Out[18]: 0

In [22]: `r[4]`

Out[22]: 4

In [23]: `r1 = range(2,10)`

In [24]: `r1[0]`

Out[24]: 2

In [26]: `r1[4]`

Out[26]: 6

In [27]: `r1[6]`

Out[27]: 8

In [28]: `r1[2]`

Out[28]: 4

In [29]: `r = range(2,10)`
`print(r[2:8])`

range(4, 10)

In [30]: `r = range(2,100,5)`
`print(r[2:8])`

range(12, 42, 5)

In [42]: `list(r)`

Out[42]: [2, 7, 12, 17, 22, 27, 32, 37, 42, 47, 52, 57, 62, 67, 72, 77, 82, 87, 92, 97]

In [33]: `r = range(2,100,5)`
`print(r[2:8])`

range(12, 42, 5)

In [34]: `list(r[2:8])`

Out[34]: [12, 17, 22, 27, 32, 37]

In [35]: `list(r[2:10])`

Out[35]: [12, 17, 22, 27, 32, 37, 42, 47]

In [36]: `list(r[0:10])`

Out[36]: [2, 7, 12, 17, 22, 27, 32, 37, 42, 47]

In [41]: `list(r[20:0:-2])`

Out[41]: [97, 87, 77, 67, 57, 47, 37, 27, 17, 7]

In [44]: `list(r[::-1])`

Out[44]: [97, 92, 87, 82, 77, 72, 67, 62, 57, 52, 47, 42, 37, 32, 27, 22, 17, 12, 7, 2]

In [45]: `list(r[::-2])`

Out[45]: [97, 87, 77, 67, 57, 47, 37, 27, 17, 7]

In [46]: `list(r[:])`

Out[46]: [2, 7, 12, 17, 22, 27, 32, 37, 42, 47, 52, 57, 62, 67, 72, 77, 82, 87, 92, 97]

In []: