

## Day5\_Datatypes(Cont)

s2 = "Transformation"

#sequencename[start:stop:step]

start : The start index(inclusive) (0 by default)

stop : The end index(exclusive) (len(string) by default)

step : The pattern (1 by default)

s2[2:8:2]

#'aso'

S2[3:9:2]

#'nfr'

s2[-10:-2:1]

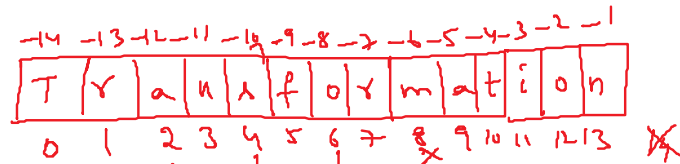
# 'sformati'

s2[::]

#'Transformation'

s2[:]

#"Transformation"



s2 [2:8:2] → a s o

s2 [3:9:2] → n f r

s2 [-10:-2:1] s f o r m a t i

list: 1. A list is a collection of heterogeneous elements(all types of elements)

2. A list is explicitly created by using [ ]

Syntax:

[el1,el2,el3....]

Ex:

lst = [45,69.32,False,None,6-5j,"Hello",45,False]

3. A list can even contain duplicate elements

4. A list will follow indexing

5. We can even create an empty list using list()

6. We can even convert other sequential type elements

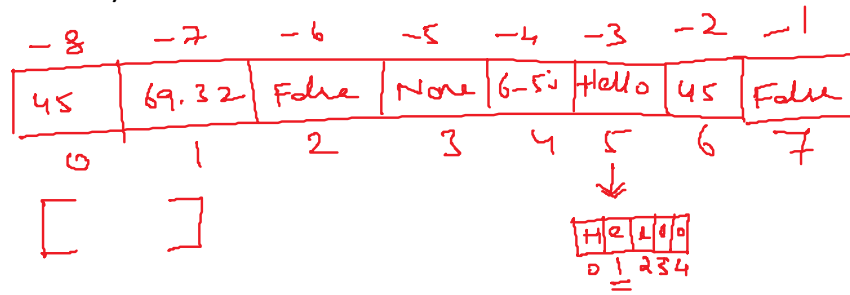
into a list using list(element)

list("Hello")

#[ 'H', 'e', 'l', 'l', 'o' ]

7. The list is mutable i.e., we can update, grow or shrink the list

8. A list can be nested i.e., we can have a collection of elements inside the list and they are accessed by sub-indexing



lst = [[12,18],

[16,21],

[19,17]]

lst

#[ [12, 18], [16, 21], [19, 17] ]

0 1 2

lst[0][0]

#12

lst[1][0]

#16

lst[2][1]

#17

tuple :

1. A tuple is a collection of heterogeneous elements(all types of elements)

2. A tuple is explicitly created by using ( )

Syntax:

(el1,el2,el3....)

Ex:

tpl=(45,69.32,False,None,6-5j,"Hello",45,False)

3. A tuple can even contain duplicate elements

4. A tuple will follow indexing

5. We can even create an empty list using tuple()

6. We can even convert other sequential type elements

into a list using tuple(element)

tuple("Hello")

```
#('H', 'e', 'l', 'l', 'o')
```

7. The tuple is immutable i.e., we can't update, grow or shrink the list

8. A tuple can be nested i.e., we can have a collection of elements inside the tuple and they are accessed by sub-indexing

```
tpl = (False, 45, 'Hello', (6-5j), None, True, 69.32, 45)
print(tpl)
```

```
#{False, 45, 'Hello', (6-5j), None, True, 69.32, 45)
type(tpl)
#tuple
```

```
tpl[2] = "Hi"
#TypeError: 'tuple' object does not support item assignment
```

```
tpl = ([12,18],
       [16,21],
       [19,17])
tpl
tpl[0]
#[12, 18]
```

```
tpl[0][1]
#18
```

range :1. This type will be used for providing a range of numbers(Integers) , it can be used in 3 ways

1. range(number) : This gives the numbers from 0 to (number-1)

2. range(startnumber, endnumber) : This gives the numbers from startnumber to (endnumber-1)

3. range(startnumber,endnumber,step) : This gives the numbers from startnumber to endnumber(exclusive) with a step

2. We cannot be able to see the range of numbers explicitly so we convert them into either a list or a tuple

```
list(range(5))
#[0, 1, 2, 3, 4]
```

```
list(range(5,15))
#[5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
```

```
list(range(5,15,2))
#[5, 7, 9, 11, 13]
```