List

List

```
a = [1,2,3,4]
b = ["Seoul", "Daejeon", "Busan"]
c = []
d = [13.5, 35, "Seoul"]
e = [23, [1,2,3], "Daejeon"]
```

integers

strings

Empty list : no element

List can consist of elements whose data types are different.

List itself can be an element of a list.

- Any kind of data can be assigned to List.
- [] (squared bracket) is used for the list.

Index and Slicing

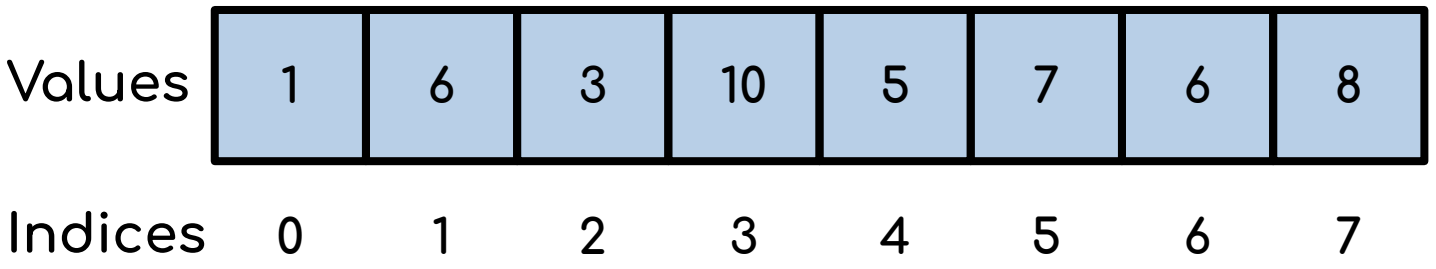
```
a = ["Seoul", "Daejeon", "Busan"]
print(a[1])
```

```
Daejeon
```

```
a = [1,6,3,10,5,7,6,8]

print(a[2:7])
print(a[3:])
print(a[:6])
print(a[:])
```

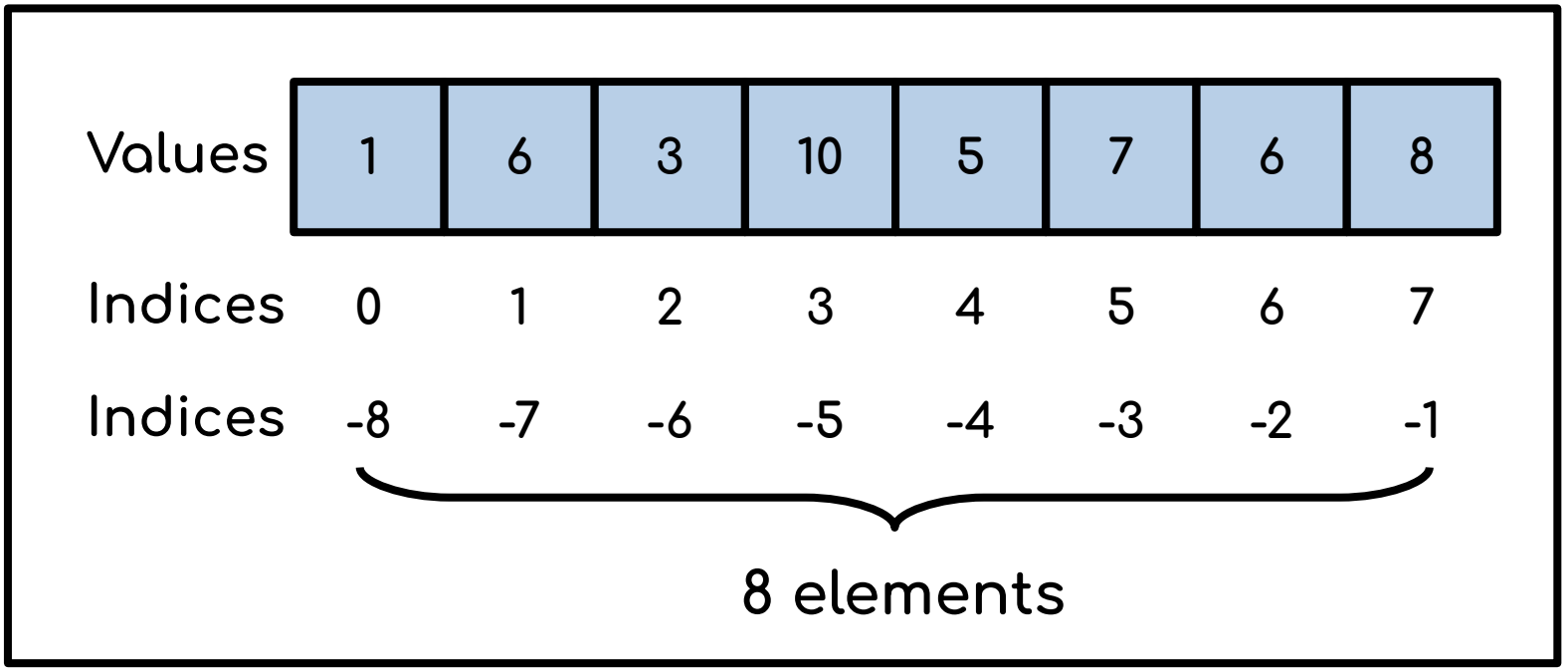
```
[3, 10, 5, 7, 6]
[10, 5, 7, 6, 8]
[1, 6, 3, 10, 5, 7]
[1, 6, 3, 10, 5, 7, 6, 8]
```



len(List) function

```
a = [1,6,3,10,5,7,6,8]
print(len(a))
```

```
8
```



* len() is a function that can be applied to not only strings and lists, but also any other container data types.

Tuple

Tuple

```
a = (1,2,3,4)
```
integers

```
b = ("Seoul", "Daejeon", "Busan")
```
strings

```
c = ()
```
empty tuple : no element

```
d = (13.5, 35, "Seoul")
```
Tuple can have elements whose data types are different.

```
e = (23, (1,2,3), "Daejeon")
```
Tuple itself can be an element of a tuple.

```
f = (23, [1,2,3], "Daejeon")
```
List can be an element of a tuple.

```
print(type(a), type(b), type(c))
print(type(d), type(e), type(f))
```

```
<class 'tuple'> <class 'tuple'> <class 'tuple'>
<class 'tuple'> <class 'tuple'> <class 'tuple'>
```

- ( ) (parethesis) indicates the tuple.

Indexing and Slicing of Tuples

```
a = ("Seoul", "Daejeon", "Busan")
print(a[1])
```

```
Daejeon
```

```
a = (1,6,3,10,5,7,6,8)

print(a[2:7])
print(a[3:])
print(a[:6])
print(a[:])
```

```
(3, 10, 5, 7, 6)
(10, 5, 7, 6, 8)
(1, 6, 3, 10, 5, 7)
(1, 6, 3, 10, 5, 7, 6, 8)
```

Tuple: Immutable Sequence Data Types

```
a = ("Seoul", "Daejeon", "Busan")
a[1] = "Incheon"
print(a)
```

```
---------------------------------------------------------------------
---------
TypeError                                 Traceback (most recent c
all last)
<ipython-input-7-83ba700e339c> in <module>()
      1 a = ("Seoul", "Daejeon", "Busan")
----> 2 a[1] = "Incheon"
      3 print(a)

TypeError: 'tuple' object does not support item assignment
```