

Python List

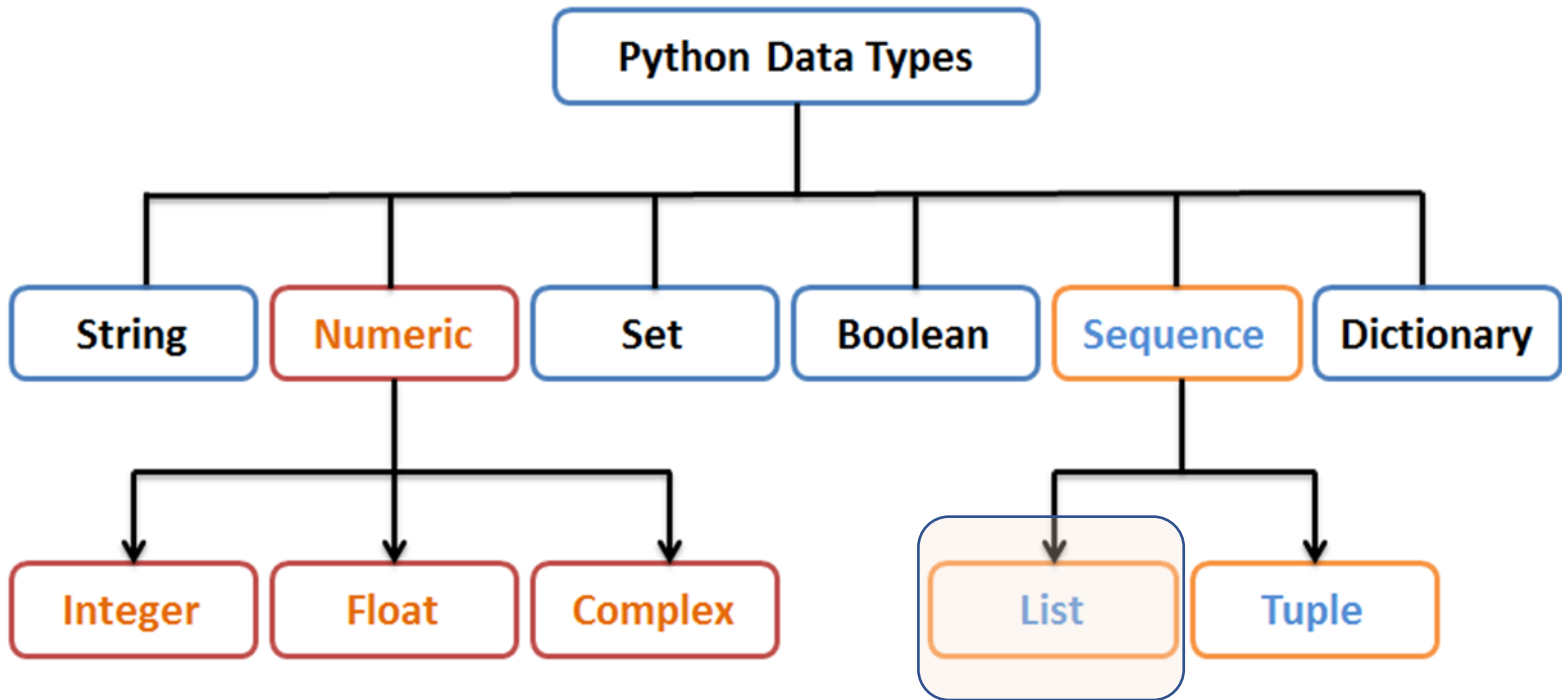
Anoop S Babu

Faculty Associate

Dept. of Computer Science & Engineering

bsanoop@am.amrita.edu

Python Data Types



List

- A sequence of comma separated values
- Enclosed in [] (square brackets)
- Can hold multiple data types
- Allow duplicate values
- Ordered
- Use index to access contents
- Mutable

Creating a List

```
>>> languages = ["Python", "Java", "C", "C++"]
```

```
>>> print(languages)
```

```
['Python', 'Java', 'C', 'C++']
```

```
>>> ageList = [12, 23, 34, 3]
```

```
>>> print(ageList)
```

```
[12, 23, 34, 3]
```

```
>>> myList = ["Python", 3, 3.8]
```

```
>>> print(myList)
```

```
['Python', 3, 3.8]
```

Creating an empty List

- Using empty square brackets

```
>>> languages = []
```

- Using the keyword **list**

```
>>> languages = list()
```

Accessing list values

```
>>> languages = ["Python", "Java", "C", "C++"]
```

- Indexing

```
>>> print(languages[0])
```

Python

- Slicing

```
>>> print(languages[1:3])
```

['Java', 'C']

Also allows negative indexing

Mutable

```
>>> languages = ["Python", "Java", "C", "C++"]
```

```
>>> print(languages)
```

```
['Python', 'Java', 'C', 'C++']
```

```
>>> languages[2] = "Perl"
```

```
>>> print(languages)
```

```
['Python', 'Java', 'Perl', 'C++']
```

List Traversal

```
>>> languages = ["Python", "Java", "C", "C++"]  
>>> for lang in languages:  
    print(lang)
```

Python

Java

C

C++

Built-in functions on List

```
languages = ["Python", "Java", "C", "C++"]
```

- **len**(*listobject*)

```
>>> len(languages)
4
```

- **max**(*listobject*)

```
>>> max(languages)
'Python'
```

- **min**(*listobject*)

```
>>> min(languages)
'C'
```

Built-in functions on List

- **list**(seq) – turns a sequence into a list

```
>>> list((1,2,3,4))  
[1, 2, 3, 4]
```

- **del**(*listobject*) - deletes the list object

- Returns nothing

```
>>> list1 = [1,2,3,4]  
>>> del(list1)  
>>> print(list1)
```

...

**NameError: name 'list1' is
not defined**

Operators allowed on List

- Concatenation
- Repetition
- Membership
- Identity
- Logical

Concatenation(+) and repetition(*) of list

```
>>> list1 = [1,2,3]
```

```
>>> list2 = [1,2,3]
```

```
>>> list1+list2
```

```
[1, 2, 3, 1, 2, 3]
```

```
>>> list1*3
```

```
[1, 2, 3, 1, 2, 3, 1, 2, 3]
```

```
>>> print(list1)
```

```
[1, 2, 3]
```

- Both operations returns a new list object.

Membership on List

```
>>> list1 = [1,2,3]
```

```
>>> 1 in list1
```

```
True
```

```
>>> 2 not in list1
```

```
False
```

Identity Operator on List

- Identity operator compares the objects w.r.t their identity.

```
>>> list1 = [1,2,3]
```

```
>>> list2 = [1,2,3]
```

```
>>> list1 is list2
```

```
False
```

- No two lists have the same identity in Python.

```
>>> id(list1)
```

```
3147374564608
```

```
>>> id(list2)
```

```
3147374566464
```

Logical Operators on List

- Can use all the logical operators on a list

```
>>> list1 = [1,2,3]
```

```
>>> list2 = [1,2,3]
```

```
>>> list1 == list2
```

```
True
```

```
>>> list1 != [4,5]
```

```
True
```