

# Python's List

- Store multiple types of data ex string, number etc. (even list also), like C-structure
- Syntax
  - []
  - `mylist = [1, 2, 3]`
  - `mylist = ["python", 2.7, 1991]`
  - Forward and Reverse Indexing Applicable
  - `>> mylist[1]`  
2.7
  - `>> mylist[-1]`  
1991
- Advantages over C-Array
  - Multiple types of values can be stored
  - Dynamic (can be update at runtime, no fixed length)
- Common list methods
  - **Append**
    - work: **Append a value at the last of the list**
    - syntax: `mylist.append(value)`
    - ex: `mylist = ["python", 2.7, 1991]`
    - run: `mylist.append("mynewvalue")`
    - output: `["python", 2.7, 1991, "mynewvalue"]`
  - **Insert**
    - work: **Append a value at a specified index in list**
    - syntax: `mylist.insert(index-where-to-insert, value)`
    - ex: `mylist = ["python", 2.7, 1991]`
    - run: `mylist.insert(1, "newvalue")`
    - output: `["python", "mynewvalue", 2.7, 1991]`
  - **Remove**
    - work: **Remove an element from list (by value)**
    - syntax: `mylist.remove(value-of-element)`
    - ex: `mylist = ["python", 2.7, 1991]`
    - run: `mylist.remove(2.7)`
    - output: `["python", "1991" ]`
  - **Pop**
    - work: **Remove an element from list (by index)**
    - syntax: `mylist.pop(index-of-element)`
    - ex: `mylist = ["python", 2.7, 1991]`
    - run: `mylist.pop(1)`
    - output: `["python", "1991" ]`
    - default: default index -1 or `pop()` i.e. `pop(-1)`

- **Extend**

- work: **Concate two lists**
- syntax: `mylist.extend(anotherlist)`
- ex: `mylist = ["python", 2.7, 1991]`  
`anotherlist = ["is", "awsome"]`
- run: `mylist.extend(anotherlist)`
- output: value of mylist is now `["python", 2.7, 1991, "is", "awsome"]`

Written By:

Vikas Kumar Sharma

[vikas.pypy@gmail.com](mailto:vikas.pypy@gmail.com)

defpy.com

*Feedback, Queries and Suggestions are invited on above mail id.*