# List []

- A list is a data structure that holds an ordered collection of items i.e. you can store a sequence of items in a list.

The list of items should be enclosed in square brackets so that Python understands that you are specifying a list. Once you have created a list, you can add, remove or search for items in the list. Since we can add and remove items, we say that a list is a mutable data type i.e. this type can be altered.

```
## Runs
```

```
m1 = int(input())
m2 = int(input())
m3 = int(input())
m4 = int(input())
m5 = int(input())
```

```
 100
 20
 340
 345
 45
```

```
print(m1, m2, m3, m4, m5)
```

```
100 20 340 345 45
```

```
odi_scores = [100, 20, 340, 345, 45]
```

```
type(odi_scores)
```

```
list
```

```
for run in odi_scores:
    print(run)
```

```
100
20
340
345
45
```

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

```
0
1
2
3
```

```
4
5
6
7
8
9
```

```python
for i in 123:
    print(i)
```

```
---------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_3290/286694
8089.py in <module>
----> 1 for i in 123:
      2     print(i)

TypeError: 'int' object is not iterable
```

```python
# Is list iterable?
```

```python
# len function
```

```python
len(odi_scores)
```

```
5
```

```python
len(123)
```

```
---------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_3290/190637
1528.py in <module>
----> 1 len(123)

TypeError: object of type 'int' has no len()
```

## Indexing in list: Like we used to have in our attendance register, roll no.

```
odi_scores
```

```
[100, 20, 340, 345, 45]
```

```
odi_scores[0]
```

```
100
```

## last index = 4, len - 1

```
odi_scores[4]
```

```
45
```

```
odi_scores[len(odi_scores) - 1]
```

```
45
```

```
len(odi_scores) - 1
```

```
4
```


## Let Sachin play another match: append


```
odi_scores
```

```
[100, 20, 340, 345, 45]
```

```
# odi_scores.append(value)
```

```
odi_scores.append(101)
```

```
odi_scores
```

```
[100, 20, 340, 345, 45, 101]
```

```
odi_scores.append(200)
```

```
odi_scores
```

```
[100, 20, 340, 345, 45, 101, 200]
```

```
odi_scores.append(100, 230)
```

```
--------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_3290/194534
4280.py in <module>
----> 1 odi_scores.append(100, 230)

TypeError: list.append() takes exactly one argument (2 given)
```

## Negative indexing

```
odi_scores
```

```
[100, 20, 340, 345, 45, 101, 200]
```

```
odi_scores[-1]
```

```
200
```

```
l = [2, 3,4, 5, 6, 7]
print(l[-2])
```

```
6
```

```
len(l)
```

```
6
```

```
l[-6]
```

```
2
```

## input().split()

```
s = input().split()
```

```
 1 2 3 4 5
```

```
s
```

```
['1', '2', '3', '4', '5']
```

```
type(s[0])
```

```
s[0]
```

```
'1'
```

```
type(s)
```

```
str
```

```
l = s.split()
```

```
l
```

```
['1', '2', '3', '4', '5']
```

## Insert at index

```
odi_scores
```

```
[100, 20, 340, 345, 45, 101, 200]
```

```
odi_scores.insert?
```

```
Signature: odi_scores.insert(index, object, /)
Docstring: Insert object before index.
Type:      builtin_function_or_method
```

```
odi_scores.insert(1, 200)
```

```
odi_scores
```

```
[100, 200, 20, 340, 345, 45, 101, 200]
```

## pop element: It also returns the element

```
odi_scores
```

```
[100, 200, 20, 340, 345, 45, 101, 200]
```

```
odi_scores.pop?
```

```
Signature: odi_scores.pop(index=-1, /)
Docstring:
Remove and return item at index (default last).

Raises IndexError if list is empty or index is out of range.
Type:      builtin_function_or_method
```

```
odi_scores.pop()
```

```
200
```

```
odi_scores

[100, 200, 20, 340, 345, 45, 101]

odi_scores.pop(1)

200
```

```
## remove element: using an element
## Remove function removes the first occurance of that element
## remove function doesn't return the value
odi_scores

[100, 20, 340, 345, 45, 101]

odi_scores.remove(20)

odi_scores

[100, 340, 345, 45, 101]

l = [1, 3, 1, 2, 1]

l.remove(1)

l

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

```
## Sum of array
l = input().split()
l.pop(0)
result = []
for i in l:
    result.append(int(i))
sum = 0
for i in result:
    sum += i
print(sum)

 5 1 2 3 4 5

15
```

```python
l = input().split()
```

```
 5 1 2 3 4 5
```

```python
l
```

```
['5', '1', '2', '3', '4', '5']
```

```python
l.pop(0)
```

```
'5'
```

```python
l
```

```
['1', '2', '3', '4', '5']
```

```python
result = []
for i in l:
    result.append(int(i))
```

```python
result
```

```
[1, 2, 3, 4, 5]
```

```python
for i in result:
    print(i, type(i))
```

```
1 <class 'int'>
2 <class 'int'>
3 <class 'int'>
4 <class 'int'>
5 <class 'int'>
```

```python
sum = 0
for i in result:
    sum += i
```

```python
sum
```

```
15
```

```python
l = [1, 2, 3, 4, 5]
```

```python
len(l)
```

```
5
```

```python
for i in range(0, len(l)):
    print(i)
```

```
0
1
2
3
4
```

```python
for i in range(0, len(l)):
    print(l[i])
```

```
1
2
3
4
5
```

```python
l[0]
```

```
1
```

```python
for i in range(0, len(l)):
    print(l[i])
```

```python
# len, element, indexing
```

```python
l = [1, 2, 3, 4, 5]
```

```python
l[0]
```

```
1
```