

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

```
dummy = ["Rahul", 38, 98.8]
```

type

```
type(dummy)
```

```
list
```

```
runs = [100, 99, 0, 50, 99, 200, 99, 30]
```

```
type(runs)
```

```
list
```

len function

length is number of elements

```
len(runs)
```

```
8
```

List from string

```
s = input()
```

```
1 2 3 4 5 6
```

```
print(s)
```

```
1 2 3 4 5 6
```

```
type(s)
```

```
str
```

split is a string method which cuts the data around spaces and return list

```
new = s.split()
```

```
new
```

```
['1', '2', '3', '4', '5', '6']
```

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

quiz 2

```
runs
```

```
[100, 99, 0, 50, 99, 200, 99, 30]
```

```
runs[0]
```

```
100
```

```
runs[1]
```

```
99
```

```
n = len(runs)
```

```
n
```

```
8
```

runs[8]

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

```
30
```

```
runs = ["99", "100"]
```

```
runs
```

```
['99', '100']
```

```
runs = [100, 99, 0, 50, 99, 200, 99, 30]
```

```
runs[0]
```

```
100
```

```
type(runs[0])
```

```
int
```

```
l = [5, 1, -2, 2, 3, 4]
print(l[2])
```

-2

Negative indexing: Topper from bottom :)

quiz

runs

[100, 99, 0, 50, 99, 200, 99, 30]

len(runs)

8

runs[-8]

100

runs[0]

100

runs[len(runs) - 1]

30

runs[-1]

30

Quiz

runs = [10, 55, 4, 67, 17]

print(runs[0] + runs[-1] + runs[-3])

31

Updating a list

Let Sachin play another match: append

quiz

```
runs = [100, 99, 0, 50, 99, 200, 99, 30]
```

```
last = 150
```

```
# Append adds a data at last position in a list
```

```
runs.append(last)
```

```
runs
```

```
[100, 99, 0, 50, 99, 200, 99, 30, 150]
```

```
len(runs)
```

```
9
```

```
runs[-1]
```

```
150
```

```
runs[-2]
```

```
30
```

```
# Mutability: Modifying
```

```
li = [2, 3]
```

```
id(li)
```

```
140341573274496
```

```
li.append(23)
```

```
li
```

```
[2, 3, 23]
```

```
id(li)
```

```
140341573274496
```

```
# print(help(runs))
```

```
print(dir(runs))
```

```
['__add__', '__class__', '__class_getitem__', '__contains__',  
 '__delattr__', '__delitem__', '__dir__', '__doc__', '__eq__',  
 '__format__', '__ge__', '__getattribute__', '__getitem__', '__gt__',  
 '__hash__', '__iadd__', '__imul__', '__init__', '__init_subclass__',  
 '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__',  
 '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__',  
 '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__',  
 '__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend',  
 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
```

```
## Insert at index: runs.insert(index, value)
```

```
# Quiz
```

```
runs
```

```
[100, 99, 0, 50, 99, 200, 99, 30, 150]
```

```
missing_data = 99
```

```
runs.insert(5, missing_data)
```

```
runs
```

```
[100, 99, 0, 50, 99, 99, 200, 99, 30, 150]
```

Iterating a list

```
# Lists are iterable as well
```

```
print(dir(runs))
```

```
['__add__', '__class__', '__class_getitem__', '__contains__',  
 '__delattr__', '__delitem__', '__dir__', '__doc__', '__eq__',  
 '__format__', '__ge__', '__getattribute__', '__getitem__', '__gt__',  
 '__hash__', '__iadd__', '__imul__', '__init__', '__init_subclass__',  
 '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__',  
 '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__',  
 '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__',  
 '__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend',  
 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
```

```
runs
```

```
[100, 99, 0, 50, 99, 99, 200, 99, 30, 150]
```

```
for i in runs:  
    print(i)
```

```
100  
99  
0  
50  
99  
99  
200  
99  
30  
150
```

```
for i in runs:  
    print(i, end=" ")
```

```
100 99 0 50 99 99 200 99 30 150
```

```
# quiz
```

```
runs = [10, 55, 4, 67, 17]  
runs.append(6)  
runs.append(200)  
print(runs[-1] + runs[2])
```

```
204
```

```
l = [1, 2, 3, 3, 5, 6, 7, 5]  
l.insert(5, 10)  
print(l)
```

```
[1, 2, 3, 3, 5, 10, 6, 7, 5]
```

```
# Total runs scored by Sachin in his career
```

```
# quiz
```

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

```
# iterate on the list to get all the elements
```

```
runs = [100, 99, 0, 50, 99, 99, 200, 99, 30, 150 ]
```

```
for i in runs:  
    print(i)
```

```
100
99
0
50
99
99
200
99
30
150
```

```
total = 0
```

```
for i in runs:
    # update the totals with the runs after every iteration
    total += i
```

```
total
```

```
926
```

```
# sum
```

```
sum(runs)
```

```
926
```

```
Iterating a list using indexes
```

```
runs
```

```
[100, 99, 0, 50, 99, 99, 200, 99, 30, 150]
```

```
len(runs)
```

```
10
```

```
# last index = 9
```

```
# first = 0
```

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

```
0
1
2
3
4
5
6
```

7
8
9

runs[0]

100

runs[1]

99

runs[2]

0

runs[3]

50

Final code

runs

[100, 99, 0, 50, 99, 99, 200, 99, 30, 150]

```
for i in range(len(runs)):
    # getting all the indexes here
    # use these indexes to get the runs
```

```
    print(runs[i], end=" ")
```

100 99 0 50 99 99 200 99 30 150

Find the runs scored by Sachin in even index matches

Here it is printing only even runs

runs

[100, 99, 0, 50, 99, 99, 200, 99, 30, 150]

```
for i in runs:
    if i % 2 == 0:
        print(i)
```

100

0

50

200

30
150

```
for i in range(len(runs)):
    # find even indexes
    if i % 2 == 0:
        print(i)
```

0
2
4
6
8

```
for i in range(len(runs)):
    # find even indexes
    if i % 2 == 0:
        print(runs[i])
```

100
0
99
200
30

final code

total = 0

```
for i in range(len(runs)):
    # find even indexes
    if i % 2 == 0:
        total += runs[i]
```

print(total)

429

total = 0

```
for i in range(0, len(runs), 2):
    # find even indexes
    total += runs[i]
```

```
print(total)
```

```
429
```

Iteration protocols

runs is iterable

```
runs
```

```
[100, 99, 0, 50, 99, 99, 200, 99, 30, 150]
```

dir(runs)

```
i = iter(runs)
```

```
next(i)
```

```
-----  
-----
```

```
StopIteration
```

```
Traceback (most recent call
```

```
last)
```

```
/var/folders/zn/hkv6562d6_d30glfs8yc76900000gn/T/ipykernel_1007/193974  
8483.py in <module>
```

```
----> 1 next(i)
```

```
StopIteration:
```

Taking List input from user

```
s = input()
```

```
1 2 3 4 5
```

int(s)

no

Getting a list from string with elements space separated

```
s
```

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

```
li = s.split()
```

```
li
['1', '2', '3', '4', '5']
li[0]
'1'
type(li[0])
str
li[0]
'1'
int(li[0])
1
int(li[1])
2
int(li[2])
3
for i in li:
    print(i, type(i))
1 <class 'str'>
2 <class 'str'>
3 <class 'str'>
4 <class 'str'>
5 <class 'str'>

for i in li:
    print(int(i))
1
2
3
4
5

# append these elements into a new list
new = []
```

```
for i in range(10):
    new.append(i)

print(new)

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
li

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

final code

```
new = []

for i in li:
    new.append(int(i))

print(new)

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

```
li

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

new = []

for i in range(len(li)):
    # print(li[i])
    new.append(int(li[i]))

print(new)

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

One more possibility

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

1 2 3 4 5

s

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

# int(s)
```

Doubts