Appending values within list

Append is nothing adding values at the end of the list.

This appending is done in two ways

1. append(value)
2. slicing operator

append(value) method is allows to add one value.

Syntax: list-name.append(value)

Example

```
>>> list1=[]
>>> print(list1)
[]
>>> list1.append(10)
>>> print(list1)
[10]
>>> list1.append(20)
>>> print(list1)
[10, 20]
>>> list1.append(30)
>>> print(list1)
[10, 20, 30]
>>> list1.append(40)
>>> print(list1)
[10, 20, 30, 40]
```

 # Write a program to append 5 values inside list

```
list1=[]

for i in range(5):
    value=int(input("Enter any value "))
    list1.append(value)

print(list1)
```

Output

```
Enter any value 10
Enter any value 20
Enter any value 30
Enter any value 40
Enter any value 50
[10, 20, 30, 40, 50]
```

Example

```
# Write a program read name, n subject marks and calculate
total,avg marks

# Input
name=input("Enter Name ")
n=int(input("Enter How Many Subject Marks?"))
marks=[]

for i in range(n):
    s=int(input("Enter Marks "))
    marks.append(s)

# Process
total=0
for s in marks:
    total+=s

avg=total/n

#Output
print(f'Name {name}')
print(f'Marks {marks}')
print(f'Total {total}')
print(f'Avg {avg:.2f}')
```

Output

```
Enter Name naresh
Enter How Many Subject Marks?3
Enter Marks 60
```

```
Enter Marks 90
Enter Marks 89
Name naresh
Marks [60, 90, 89]
Total 239
Avg 79.67
```

Example

```
# Write a program to read scores of n players and calculate total score,max score,min score

# Input
n=int(input("Enter How Many Players ?"))

players=[]

for i in range(n):
    s=int(input("Enter Score "))
    players.append(s)
# Process
# Total Score
total=0
for s in players:
    total+=s

# Max Score
max_score=0
for s in players:
    if s>max_score:
        max_score=s

# Min Score
min_score=players[0]
for s in players:
    if s<min_score:
        min_score=s

# Output
```

```
print(f'Scores {players}')
print(f'Total {total}')
print(f'Max Score {max_score}')
print(f'Min Score {min_score}')
```

Output

```
Enter How Many Players ?4
Enter Score 10
Enter Score 60
Enter Score 30
Enter Score 5
Scores [10, 60, 30, 5]
Total 105
Max Score 60
Min Score 5
```

Appending more than one value using slice operator

Syntax: list-name[len(list):len(list)]=iterable

Example:

```
>>> list1=[10,20,30]
>>> list1.append(40,50)
Traceback (most recent call last):
  File "<pyshell#11>", line 1, in <module>
    list1.append(40,50)
TypeError: list.append() takes exactly one argument (2 given)
>>> list1[3:3]=[40,50]
>>> print(list1)
[10, 20, 30, 40, 50]
>>> list1[len(list1):]=range(60,110,10)
>>> print(list1)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
>>> list1[len(list1):]="ABC"
>>> print(list1)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 'A', 'B', 'C']
```

extend(iterable)

this method append more than one value

Syntax: list-name.extend(iterable)

```
>>> list1=[10,20]
>>> print(list1)
[10, 20]
>>> list1.extend([30,40,50])
>>> print(list1)
[10, 20, 30, 40, 50]
>>> list1.extend("ABC")
>>> print(list1)
[10, 20, 30, 40, 50, 'A', 'B', 'C']
>>> list1.extend(range(1,6))
>>> print(list1)
[10, 20, 30, 40, 50, 'A', 'B', 'C', 1, 2, 3, 4, 5]
```

What difference is between append and extend methods of list?

Append add one value at the end of the list

Extend add more than one value at the end of list

Example

```
>>> list1=[10,20]
>>> print(list1)
[10, 20]
>>> list1.append([30,40,50])
>>> print(list1)
[10, 20, [30, 40, 50]]
```

Example

```
# Check if element exists in list in Python

num_list=[10,50,23,76,34,98,34,12,69,34,54,78,56]

print(f'List is {num_list}')

# Method-1
```

```
search=int(input("Search Value :"))
found=False
for value in num_list:
    if value==search:
        found=True
        break

if found:
    print(f'{search} found')
else:
    print(f'{search} not found')

# Method-2 : using membership operator (in)
if search in num_list:
    print(f'{search} found')
else:
    print(f'{search} not found')
```

Example

```
# Write a program to input n values in list
# if value is even, append to even_list
# if value if odd, append to odd_list


n=int(input("Enter how many integer values ?"))
num_list=[]

for i in range(n):
    value=int(input("Enter any integer value "))
    num_list.append(value)

even_list=[]
odd_list=[]
for value in num_list:
    if value%2==0:
        even_list.append(value)
    else:
        odd_list.append(value)
```

```
print(f'Number List {num_list}')
print(f'Even List {even_list}')
print(f'Odd List {odd_list}')
```

Removing values from list

There are different ways from removing/deleting values from list

1. using del keyword
2. using remove method
3. using clear method
4. using pop method

using "del" keyword

"del" keyword is used to delete one or more than one value
"del" keyword required index or slicing

Syntax-1: del list-name[index]
Syntax-2: del list-name[start:stop:step]

Example

```
>>> list1=list(range(10,110,10))
print(list1)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
>>> del list1[0]
>>> print(list1)
[20, 30, 40, 50, 60, 70, 80, 90, 100]
>>> del list1[-2]
>>> print(list1)
[20, 30, 40, 50, 60, 70, 80, 100]
>>> del list1[10]
Traceback (most recent call last):
  File "<pyshell#36>", line 1, in <module>
    del list1[10]
IndexError: list assignment index out of range
```

Example of deleting more than one value using slicing

```
>>> list1=list(range(10,110,10))
>>> print(list1)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
>>> del list1[0:3]
>>> print(list1)
[40, 50, 60, 70, 80, 90, 100]
>>> del list1[-3:]
>>> print(list1)
[40, 50, 60, 70]
>>> list2=list(range(10,110,10))
>>> print(list2)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
>>> del list2[::2]
>>> print(list2)
[20, 40, 60, 80, 100]
>>> list3=list(range(10,110,10))
>>> print(list3)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
>>> del list3[2:-2]
>>> print(list3)
[10, 20, 90, 100]
```

remove(value)

remove method remove/delete values from list using value. it will remove only first occurrence of value.

Syntax: list-name.remove(value)

Example

```
>>> list1=[10,10,10,10,10]
>>> print(list1)
[10, 10, 10, 10, 10]
>>> list1.remove(10)
>>> print(list1)
[10, 10, 10, 10]
```

```
>>> list2=[10,20,30,20,20,30,40,50,40]
>>> print(list2)
[10, 20, 30, 20, 20, 30, 40, 50, 40]
>>> list2.remove(30)
>>> print(list2)
[10, 20, 20, 20, 30, 40, 50, 40]
>>> list2.remove(100)
Traceback (most recent call last):
  File "<pyshell#59>", line 1, in <module>
    list2.remove(100)
ValueError: list.remove(x): x not in list
```