# Objectives for class 12

--- Chapter 7 ---

- 7.1 To describe why lists are useful in programming (§7.1).
- 7.2 To learn how to create lists (§7.2).
- 7.3 To use the len, min, max, sum, and random.shuffle functions with a list (§7.2.2).
- 7.4 To access list elements by using indexed variables (§7.2.3).
- 7.5 To obtain a sublist from a larger list by using the slicing operator [start : end : step] (§7.2.4).
- 7.6 To use the + (concatenation), * (repetition), and in/not in operator
- 7.7 To traverse elements in a list using a for loop (§7.2.6).

# Two Programs To Display 5 Numbers User Has Entered

```
num1 = int(input("Enter a number:"))
num2 = int(input("Enter a number:"))
num3 = int(input("Enter a number:"))
num4 = int(input("Enter a number:"))
num5 = int(input("Enter a number:"))

print('num1 is',num1)
print('num2 is',num2)
print('num3 is',num3)
print('num4 is',num4)
print('num5 is',num5)
```

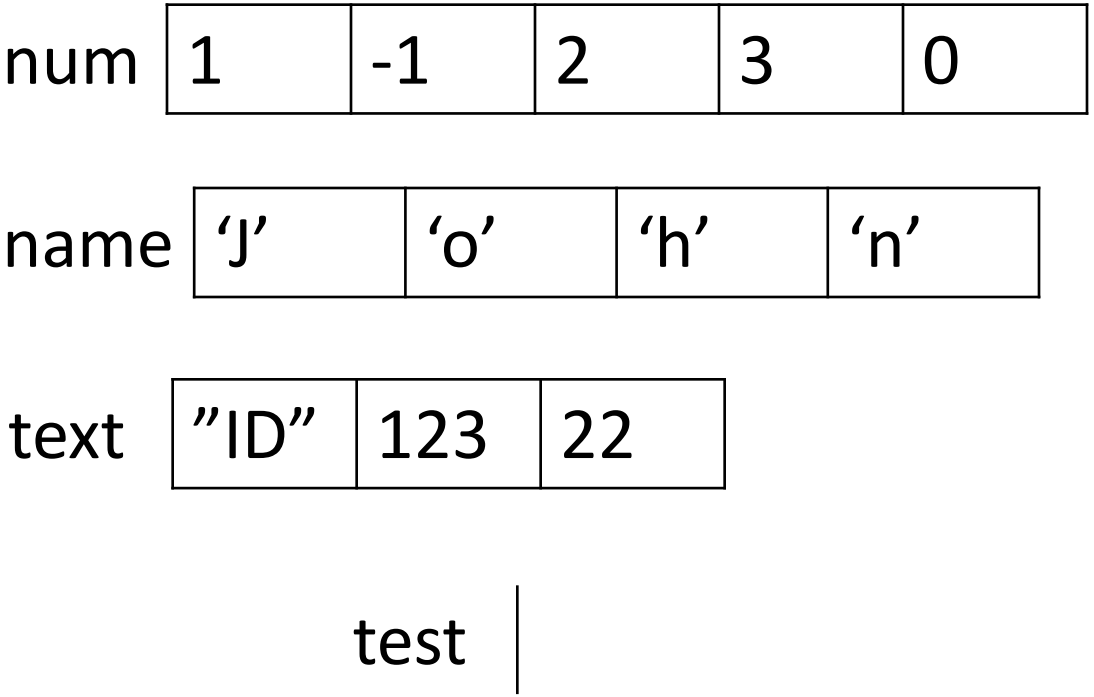
**Without List**

```
num=[0,0,0,0,0]
for i in range(0,5):
    num[i]=int(input("Enter a number:"))

sum=0

for i in range(0,5):
   print("num",i,"is",num[i])
```

**With List and Loop**

# A List is A Sequence Of Values

- Like string, a **list** is a sequence of values.
- In a string, values are characters; in a list, they can be **any type**.
- A list that contains no elements is called **empty list**
- Values are sharing the same name.
- Values in a list are called **elements** or **items**.

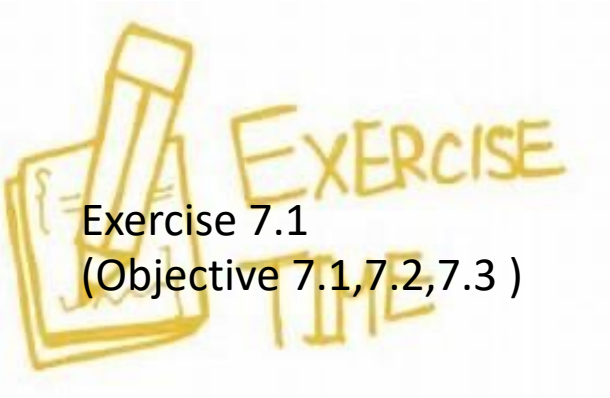


# How To Create A List?

- Enclose the elements in **square brackets**
- You can create empty list with **empty brackets**

```
>>> num=[1,-1,2,3,0]
>>> num
[1, -1, 2, 3, 0]
>>> name=['J','o','h','n']
>>> name
['J', 'o', 'h', 'n']
>>> text=["ID",123,22]
>>> text
['ID', 123, 22]
>>> test=[]
>>> test
[]
```

# Use Built-in Functions For Lists




Exercise 7.1
(Objective 7.1,7.2,7.3 )

```
>>> list1 = [2, 3, 4, 1, 32]
>>> len(list1)
5
>>> max(list1)
32
>>> min(list1)
1
>>> sum(list1)
42
```

```
>>> import random
>>> random.shuffle(list1)
>>> list1
[4, 1, 2, 32, 3]
```

Shuffle the items in the list

# Practice

- Should we use list in following cases?
- Stock prices in last month, display the highest stock price.
- Display the number of votes.

# Access List Elements Using **Index Operator []**

```
>>> num[0]
1
>>> num[-1]
0
>>> name[2]
'h'
>>> text[2]
2
>>> num[5]
Traceback (most recent call
last):
  File "<stdin>", line 1, in
<module>
IndexError: list index out of
range
```

**num**

| 1 | -1 | 2 | 3 | 0 |
|---|---|---|---|---|
| num[0] | num[1] | num[2] | num[3] | num[4] |

**name**

| 'J' | 'o' | 'h' | 'n' |
|---|---|---|---|
| name[0] | name[1] | name[2] | name[3] |

**text**

| "ID" | 123 | 22 |
|---|---|---|
| text[0] | text[1] | text[2] |

Index starts from **0** to **length-1**

# Access List Elements Using **Slice Operator**

| **num** | 1 | -1 | 2 | 3 | 0 |
|---|---|---|---|---|---|
| | num[0] | num[1] | num[2] | num[3] | num[4] |

```
>>> num=[1,-1,2,3,0]
>>> num[1:3]
[-1, 2]
>>> num[:3]
[1, -1, 2]
>>> num[1:3:1]
[-1, 2]
>>> num[1:4:2]
[-1, 3]
```

[start : end]
[end]
[start : end : step]

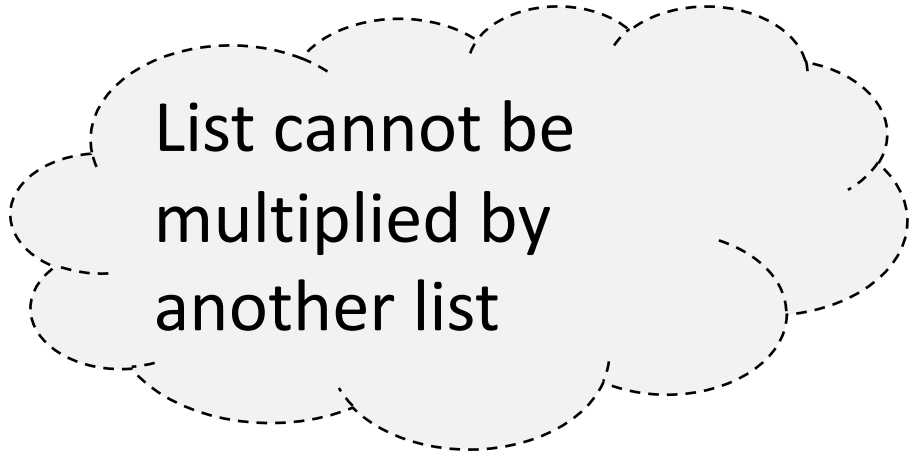
# Use + To Concatenate Lists

```
>>> list1=[2,3]
>>> list2=[1,9]
>>> list3=list1+list2
>>> list3
[2, 3, 1, 9]
>>> list4=list2+list1
>>> list4
[1, 9, 2, 3]
>>> list1+4
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can only concatenate list
(not "int") to list
```

List cannot be concatenated to a non-list data type

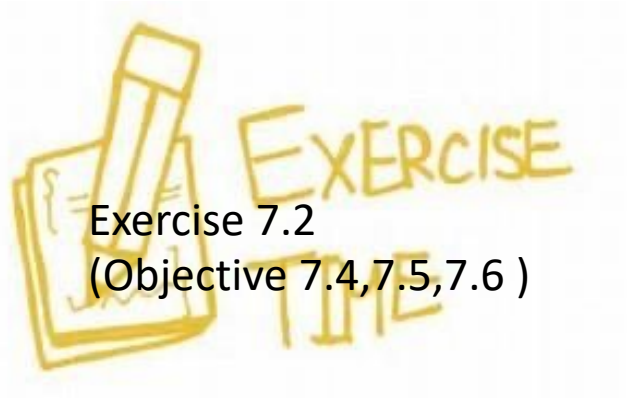
# Use * To Repeat Lists

```
>>> list1=[2,3]
>>> list2=[1,9]
>>> list1*2
[2, 3, 2, 3]
>>> list1*list2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can't multiply sequence by
non-int of type 'list'
>>> 2*list1
[2, 3, 2, 3]
```



# Use **in** or **not in** to search for elements in lists

```
>>> list1=[2,3]
>>> 2 in list1
True
>>> 2 in [1,2,3]
True
>>> 2 not in list1
False
```




Exercise 7.2
(Objective 7.4,7.5,7.6 )

# Traverse elements in a list with for loop

- Read list elements only

```
cheeses = ['Cheddar', 'Edam', 'Gouda']
for cheese in cheeses:
    print(cheese)
```

- Read and update list elements

```
cheeses = ['Cheddar', 'Edam', 'Gouda']
for i in range(0,len(cheeses)):
    # Add a postfix for each cheese
    cheeses[i]=cheeses[i]+'Cheese'

print(cheeses)
```

`len` returns the number of elements in the list

`range` returns a list of indices from 0 to *len*−1

`i` gets the index of the next element in each loop

# Case study: count zip code

- Create a list of postal codes, name it as `zipcode_list`
- Put following elements into `zipcode_list`

  `30003,30300,30329,30318,30472,30300`
- Traverse elements in `zipcode_list`
- Count the number of elements `30300` in `zipcode_list`
- Display count

# Case study: count zip code --- Implementation

```
zipcode_list=[30003,30300,30329,30318,30472,30300]

count=0
for zipcode in zipcode_list:
    if zipcode == 30300:
        count=count+1
print(count)
```

- Read list elements only

# Case study: keep last three digits of zip code

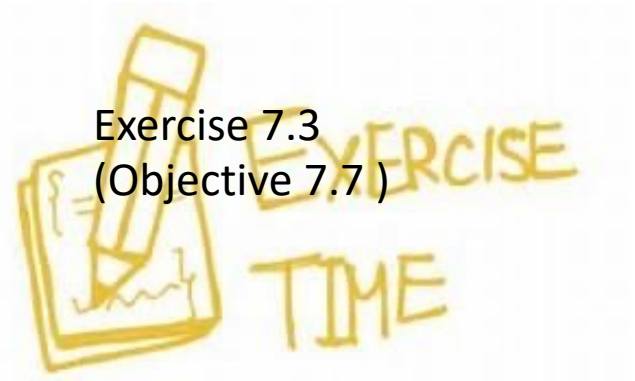
- Create a list of postal codes, name it as `zipcode_list`
- Put following elements into `zipcode_list`

  `30003,30300,30329,30318,30472,30300`
- Traverse elements in `zipcode_list`
- Update each element and keep only last three digits of zip code
- Display the updated `zipcode_list`

# Case study: keep last three digits of zip code --- Implementation

```
zipcode_list=[30003,30300,30329,30318,30472,30300]

count=0
for i in range(0,len(zipcode_list)):
    zipcode_list[i]=zipcode_list[i]-30000
print(zipcode_list)
```

- Read and update list elements




Exercise 7.3
(Objective 7.7 )