

OBJECT ORIENTED PROGRAMMING WITH PYTHON

Second Class

1st Semester

For loop

Nested loop

Lists

2D Lists

List Methods

Several thin, parallel white lines are drawn diagonally across the bottom right corner of the slide, extending from the bottom edge towards the right edge.

FOR LOOP

- ▶ For loop used to iterate over items of the collection, for each character of the string.
- ▶ Here the for loop in each iteration will hold a variable of an item.

```
for item in 'Python':  
    print(item)
```

P
y
t
h
o
n

FOR LOOP

```
for item in ['Sara','John','Smith']:  
    print(item)
```

Sara
John
Smith

```
for item in rang(10):  
    print(item)
```

0
1
2
3
5
6
7
8
9

Note:

Rang(5,10)

5,6,7,8,9

Rang(5,10,2)

5,7,9

EXERCISE

Write a program to calculate the total cost of old items in the shopping card

Solution:

```
prices=[10,20,50,30,15,5]
```

```
total_cost=0
```

```
for price in Prices:
```

```
    total_cost +=price
```

```
print(f"The total cost of the old items in your casd is {Total_cost}!")
```



NESTED LOOP

Nested Loop means loop inside loop, if we have these coordinates (x,y)
this can easily be coded to:

Solution

```
for x_coordinate in range(4):  
    for y_coordinate in range(5):  
        print(f'{x_coordinate} ,{y_coordinate}')
```

(x,y)
(0,0)
(0,1)
(0,2)
(0,3)
(0,4)
(1,0)
(1,1)
(1,2)
(1,3)
(1,4)
(2,0)
(2,1)
(2,2)
(2,3)
(2,4)
(3,0)
(3,1)
(3,2)
(3,3)
(3,4)

CHALLENGE

Use nested loop to draw this F-shape

The numbers of the F shape X's will be stored in (**numbers**) list

The right way to is to use the nested loop :

```
XXXXX  
XX  
XXXXX  
XX  
XX
```

Solution

```
numbers=[5,2,5,2,2]  
for x_count in numbers:  
    output=""  
    for count in range(x_count):  
        output+='X'  
    print(output)
```

```
numbers=[5,2,5,2,2]  
for x_count in numbers:  
    print('X'* x_count )
```



LISTS

We will define a list of names:

```
Names=['John','Adam','Sara','Mary','Smith']
```

<pre>print(names)</pre>	displays all the list ['John','Adam','Sara','Mary','Smith']
<pre>print(names[0])</pre>	displays the first item in the list John
<pre>print(names[1])</pre>	displays the second item in the list Adam
<pre>print(names[-1])</pre>	displays the last item in the list Smith
<pre>print(names[-2])</pre>	displays before the last item in the list Mary
<pre>print(names[2:])</pre>	displays from index=2 to the end ['Sara','Mary','Smith']
<pre>print(names[2:4])</pre>	displays from index=2 to the index=3 ['Sara','Mary']
<pre>print(names[:])</pre>	displays all the items in the list ['John','Adam','Sara','Mary','Smith']

Note:

We can modify any element in the list easily by:

```
names[0]='Sali'  
print(names)
```

EXERCISE

Write a program to find the largest number in a list

Solution:

```
numbers=[3,6,2,8,4,10]
max_number=numbers[0]
for number in numbers:
    if number > max_number:
        max_number=number
print(Max_number)
```



Now can you modify the program to find the minimum number in the list? TRY IT!!

2D LISTS

Extremely powerful and have a lot of applications in Data Science and Machine Learning. Here we will use 2D lists to represent Matrix (rectangular array) in Mathematics.

Example:

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

We can model this in Python with 2Dimensional List where each item in that list is another list.

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

To access an item in a matrix use square brackets

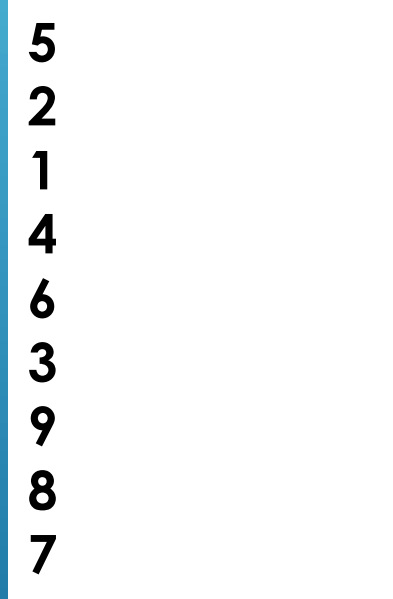
matrix[0]	returns	[1,2,3]
matix[0][1]	returns	2
matix[0][1]=20	changes	2 to 20

2D LISTS

In Python we can use nested loop to iterate all the items in this matrix

Example:

```
matrix=[  
    [5,2,1],  
    [4,6,3],  
    [9,8,7]  
]  
for row in matrix:  
    for item in row:  
        print(item)
```



5
2
1
4
6
3
9
8
7

LISTS METHODS

List's methods or list's functions, is the operations that performed on a list

numbers=[5,2,1,7,4]

- ▶ `numbers.append(20)` *# adds 20 to the end of the list*
`print(numbers)` **[5,2,1,7,4,20]**
- ▶ `numbers.insert(0, 6)` *# adds 6 at index position of 0*
`print(numbers)` **[6,5,2,1,7,4,20]**
- ▶ `numbers.remove(6)` *# removes the first 6*
`print(numbers)` **[5,2,1,7,4,20]**
- ▶ `numbers.pop()` *# removes the last item*
`print(numbers)` **[5,2,1,7,4]**
- ▶ `numbers.clear()` *# removes all the items*
`print(numbers)` **[]**

LISTS METHODS

numbers=[5,2,8,1,7,4,8]

- ▶ **numbers.index(8)** # returns the index of first occurrence of 8
 2
 - ▶ **numbers.sort()** # sorts the list
 - ▶ **numbers.reverse()** # reverses the list
 - ▶ **numbers.copy()** # returns a copy of the list
- 
- Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.