

## المصفوفات والقوائم

- يجب التصريح عن المصفوفات. القوائم لا تفعل ذلك ، لأنها مدمجة في بايثون. في الأمثلة أعلاه ، رأيت أن القوائم يتم إنشاؤها ببساطة من خلال تضمين سلسلة من العناصر في أقواس مربعة. من ناحية أخرى ، يتطلب إنشاء مصفوفة وظيفة محددة من وحدة المصفوفة (على سبيل المثال ، `array.array()` أو حزمة NumPy (على سبيل المثال ، `numpy.array()`). لهذا السبب ، يتم استخدام القوائم أكثر من المصفوفات.
- يمكن للمصفوفات تخزين البيانات بشكل مضغوط للغاية وتكون أكثر كفاءة لتخزين كميات كبيرة من البيانات.
- المصفوفات رائعة للعمليات العددية ؛ لا يمكن للقوائم معالجة العمليات الحسابية بشكل مباشر. على سبيل المثال، يمكنك قسمة كل عنصر من عناصر المصفوفة على نفس الرقم بسطر واحد فقط من التعليمات البرمجية.  
إذا حاولت الشيء نفسه مع قائمة ، فستتلقى خطأ.
- وإذا كنت بحاجة إلى تخزين سلسلة قصيرة نسبيًا من العناصر ولا تخطط لإجراء أي عمليات حسابية بها ، فإن القائمة هي الخيار المفضل.
- إذا كان لديك تسلسل طويل جدًا من العناصر، ففكر في استخدام مصفوفة.

# التعامل مع المصفوفات في بايثون

نظم معلومات ادارية - المرحلة الثالثة  
م. زينب صبيح جمعة

## القوائم (lists)

القوائم - هذه هي بنية البيانات الأكثر تنوعًا في Python وتتم كتابتها كقائمة من العناصر المفصولة بفواصل داخل أقواس مربعة. يمكن أن تتكون القائمة من عناصر غير متجانسة ومتجانسة. بعض الطرق المطبقة في القائمة هي:

`index()`, `append()`, `extend()`, `insert()`, `remove()`, `pop`

القوائم قابلة للتغيير. أي أنه يمكن تغيير محتواها مع الحفاظ على الهوية سليمة.

```
fruit=['apple', 'orange', 'banana', 'banana', 100, 200, 300]
print(fruit)
a= fruit.count('banana')
print(a)
b= fruit.index('banana')
print(b)
fruit.reverse()
print(fruit)
x= fruit.pop()
print(x)
lesson=['math', 'science', 'art', 'lang', 'chimstry']
lesn= lesson.pop()
print(lesn)
lesson.append('space')
print(lesson)
```



## انشاء مصفوفة

```
1 from array import   
2   
3 arr = array('i', [1,2,3,4,5])  
4 print(arr)
```

```
array('i', [1, 2, 3, 4, 5])
```

طباعة عنصر في مصفوفة باستخدام الدليل

```
1 print(arr[2])
```

```
3
```

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject2 - main.py

pythonProject2 > main.py

Project
  pythonProject2 C:\Users\abr alsharq\PycharmProjects\pythonPr
    > venv library root
    main.py
  External Libraries
  Scratches and Consoles

1 from array import *
2 arr= array('i',[3,6,2,1])
3 print(arr)
4

Run: main x
  "C:\Users\abr alsharq\PycharmProjects\pythonProject2\venv\Scripts\python.exe" "C:/Users/abr alsharq/PycharmProjects/pythonProject2/main.py"
  array('i', [3, 6, 2, 1])
  Process finished with exit code 0
```

# append

## الماتريks في نهاية المصفوفة

The screenshot shows the PyCharm IDE interface. The main editor window displays the following Python code in `main.py`:

```
1 from array import *
2 arr = array('i', [3, 6, 2, 1])
3 print(arr)
4 arr.append(8)
5 print(arr)
6
```

The Run console at the bottom shows the execution output:

```
Run: main x
"C:\Users\abr alsharq\PycharmProjects\pythonProject2\venv\Scripts\python.exe" "C:/Users/abr alsharq/PycharmPro
array('i', [3, 6, 2, 1])
array('i', [3, 6, 2, 1, 8])
Process finished with exit code 0
```

# insert

اضافة عنصر بين عناصر المصفوفة حيث تتم الاضافة قبل العنصر المحدد

The screenshot displays the PyCharm IDE interface. The top toolbar shows icons for Project, Run, Debug, and Settings. The left sidebar shows the Project view with the following structure:

- pythonProject2
- venv library root
- main.py
- External Libraries
- Scratches and Consoles

The main editor window shows the code in `main.py`:

```
2 arr = array('i', [3, 6, 2, 1])
3 print(arr)
4 arr.append(8)
5 print(arr)
6 arr.insert(1, 10)
7 print(arr)
8
```

The Run window at the bottom shows the execution output:

```
Run: main x
"C:\Users\abr alsharq\PycharmProjects\pythonProject2\venv\Scripts\python.exe" "C:/Users/abr alsharq/Pycha
array('i', [3, 6, 2, 1])
array('i', [3, 6, 2, 1, 8])
array('i', [3, 10, 6, 2, 1, 8])
Process finished with exit code 0
```

The code in the editor includes a call to `print(arr.index(6))` on line 8, which is not visible in the output window.

# count

لمعرفة عدد مرات تكرار عنصر في مصفوفة

The screenshot shows the PyCharm IDE interface. The top-left pane displays the project structure for 'pythonProject2', including a 'venv' directory and a 'main.py' file. The top-right pane shows the code in 'main.py' with line numbers 11 through 18. The code defines an array and performs several operations: printing the count of '6', appending '6', printing the array, and printing the count of '6' again. The bottom pane shows the execution output, which includes the path to the Python executable, the array definition, and the results of the operations.

```
pythonProject2 > main.py  
Project  
Project  
pythonProject2 C:\Users\abr alsharq\PycharmProjects\pythonPr  
venv library root  
main.py  
External Libraries  
Scratches and Consoles  
11 print(arr.count(6))  
12 arr.append(6)  
13 print(arr)  
14 print(arr.count(6))  
15  
16  
17  
18  
Run: main  
"C:\Users\abr alsharq\PycharmProjects\pythonProject2\venv\Scripts\python.exe"  
array('i', [3, 6, 2, 1])  
array('i', [3, 6, 2, 1, 8])  
array('i', [3, 10, 6, 2, 1, 8])  
2  
2  
1  
array('i', [3, 10, 6, 2, 1, 8, 6])  
2
```



## reverse

```
Scratches and Consoles
```

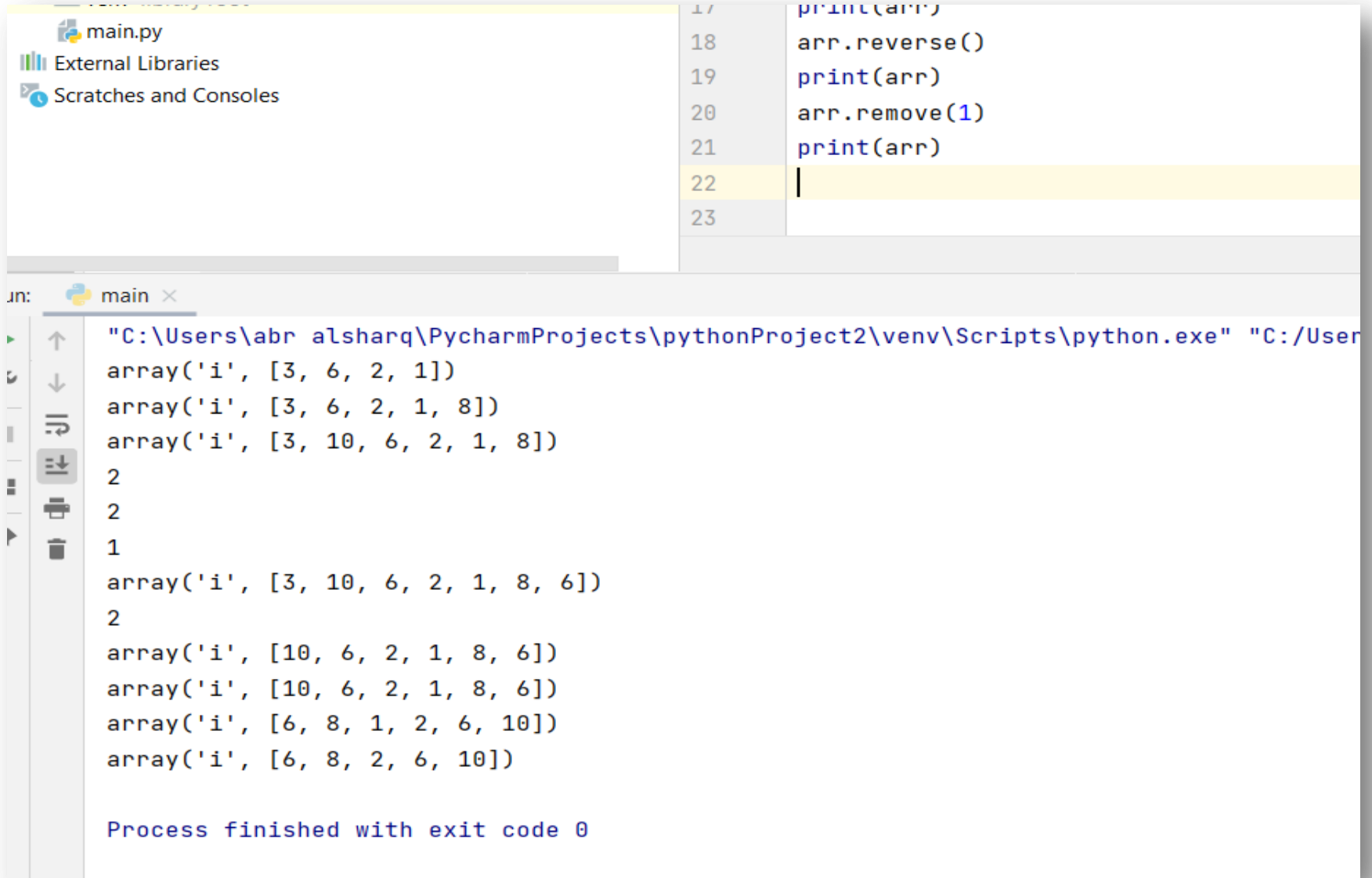
18	<code>arr.reverse()</code>
19	<code>print(arr)</code>
20	
21	

```
main x
"C:\Users\abr alsharq\PycharmProjects\pythonProject2\venv\Scripts\python.exe
array('i', [3, 6, 2, 1])
array('i', [3, 6, 2, 1, 8])
array('i', [3, 10, 6, 2, 1, 8])
2
2
1
array('i', [3, 10, 6, 2, 1, 8, 6])
2
array('i', [10, 6, 2, 1, 8, 6])
array('i', [10, 6, 2, 1, 8, 6])
array('i', [6, 8, 1, 2, 6, 10])

Process finished with exit code 0
```

# remove

حذف عنصر من مصفوفة



The screenshot shows a Python IDE with a file named 'main.py' open. The code in the editor is as follows:

```
17 print(arr)
18 arr.reverse()
19 print(arr)
20 arr.remove(1)
21 print(arr)
22 |
23
```

The output console shows the following execution results:

```
"C:\Users\abr alsharq\PycharmProjects\pythonProject2\venv\Scripts\python.exe" "C:/User
array('i', [3, 6, 2, 1])
array('i', [3, 6, 2, 1, 8])
array('i', [3, 10, 6, 2, 1, 8])
2
2
1
array('i', [3, 10, 6, 2, 1, 8, 6])
2
array('i', [10, 6, 2, 1, 8, 6])
array('i', [10, 6, 2, 1, 8, 6])
array('i', [6, 8, 1, 2, 6, 10])
array('i', [6, 8, 2, 6, 10])

Process finished with exit code 0
```

sorted

الترتيب

```
34 arr=sorted(arr,reverse=True)
35 print(arr)
36 arr=sorted(arr,reverse=False)
37 print(arr)
```

تنازلي

تصاعدي

```
main X
[99, 12, 10, 8, 6, 2, 2]
[2, 2, 6, 8, 10, 12, 99]
```