

القوائم

القائمة تكتب بهذا الشكل ، وقد يكون فيها 0 أو أكثر من العناصر:
قائمة فارغة

```
a = []
```

قوائم غير فارغة

```
x = ['orange', 'apple']
```

```
y = [8,78,5]
```

```
z = [True,5,'Hello',30]
```

للتحقق من نوع أي متغير نستعمل الدالة type()

```
a = [1,2,3]
```

```
print type(a)
```

تعطينا

```
<type 'list'>
```

الفهرسة indexing

تستعمل لإيجاد عنصر معين في القائمة.

الفهرسة تبدأ من الصفر يعني أن أول عنصر في القائمة فهرسه 0

```
x = [3,5,6]
```

```
x[0] # 3
```

نستطيع إستعمال أرقام سلبية

الفهرس 1- هو آخر عنصر في القائمة 2- هو العنصر قبل الأخير وهكذا

```
x = [3,5,6]
```

```
x[-1] # 6
```

إذا كان الفهرس المطلوب غير موجود نتحصل على خطأ
مثلاً:

```
x = [3,5,6]
```

```
x[3]
```

تعطينا

```
x[3]
```

```
IndexError: list index out of range
```

التجزئة slicing

تستعمل للحصول على جزء من قائمة

الهيئة العامة للتجزئة `x[start:stop:step]`

العنصر `step` هو إختياري وغير إجباري إذا لم نحدده فهو سيكون مساوي ال 1

عنصر التوقف `stop` لن يتم ادراجه في النتيجة

النتيجة دائماً تكون قائمة حتى لو كانت فارغة أو فيها عنصر واحد

مثلاً

`x = [3,5,6]`

`x[1:2]`

تعطينا

`[5]`

لن نتحصل على خطأ إذا كانت الفهارس المحددة غير موجودة بل نتحصل على قائمة فارغة

`y = [3,5,6]`

`y[5:8]`

تعطينا

`[]`

`x = [3,4,5,7,8]`

`x[::]`

تعطينا

`[3,4,5,7,8]`

نستطيع تحديد الخطوة لتخطي بعض العناصر

`z = [3,4,5,7,8]`

`z[::2]`

تعطينا

`[3, 5, 8]`

nested lists القوائم المتفرعة

نستطيع إنشاء قائمة مكونة من قوائم فرعية

```
Matrix = [ [1,4,5] , [6,9,2] , [2,6,8] ]  
print Matrix[0][1]# 4
```

تحديث قيم داخل القائمة

تستطيع بعد إنشاء الدالة تغيير أي قيمة بداخلها متى شئنا:

```
mylist = [9,5,6,8]  
mylist[0] = 10  
print mylist
```

تعطينا

```
[10, 5, 6, 8]
```

جميع الأساليب المستعملة مع القوائم lists methods

الأسلوب	المعنى	الإستخدام
append()	تضيف عنصر إلى آخر القائمة	<pre>myList = [3,5,6,8] myList.append(2) print myList #gives [3,5,6,8,2]</pre>
extend()	تدمج قائمتين	<pre>myList = [3,5,6,8] myList.extend([2,4]) print myList # gives [3, 5, 6, 8, 2, 4]</pre>
count()	تحسب عنصر معين داخل القائمة	<pre>myList = ['a','c','e','a'] print myList.count('a') # gives 2 myList = ['a','c','e','a'] print myList.count('A') # gives 0</pre>

<pre>myList = ['a','c','e','a'] print myList.index('a') # gives 0 myList = ['a','c','e','a'] print myList.index('a',1) #gives 3</pre>	<p>تعطي الفهرس الأول لعنصر معين ، كما يمكن أن نحدد الفهرس الذي نبدأ البحث منه</p>	index()
<pre>myList = ['a','c','e','a'] myList.insert(2,'A') print myList # gives ['a', 'c', 'A', 'e', 'a']</pre>	<p>تضيف عنصراً في فهرس محدد</p>	insert()
<pre>myList = [3,5,6,8] myList.pop(0) print myList # gives [5,6,8]</pre>	<p>ت حذف عنصراً في فهرس محدد إذا لم نحدد الفهرس سوف ت حذف آخر عنصر</p>	pop()
<pre>myList = ['a','c','e','a'] myList.remove('a') print myList # gives ['c', 'e', 'a']</pre>	<p>ت حذف الضهور الأول لعنصر ما داخل القائمة</p>	remove()
<pre>myList = ['a','c','e','b'] myList.reverse() print myList # gives ['b', 'e', 'c', 'a']</pre>	<p>تعكس الترتيب</p>	reverse()
<pre>myList = ['a','c','e','b'] myList.sort() print myList #gives ['a', 'b', 'c', 'e']</pre>	<p>ترتب القائمة تصاعدياً أو تنازلياً</p>	sort()

<pre>myList = ['a','c','e','b'] myList.sort(reverse=True) print myList # gives ['e', 'c', 'b', 'a']</pre>		
--	--	--

التعليمة del

تستخدم لحذف عنصر معين من القائمة
مثلاً

```
myList = [3,5,6,8]
del myList[2]
print myList
```

تعطينا

```
[3, 5, 8]
```

الدالات المستعملة مع القوائم built-in functions

الدالة	المعنى	مثال
cmp(list1, list2)	تقارن بين قائمتين عنصراً عنصراً بالترتيب : إذا كان العنصر الأول المنتمي إلى القائمة الأولى أكبر تكون النتيجة 1 إذا كان أصغر تكون النتيجة -1 - إذا كنا متساويان نمر إلى العنصر الموالي إذا كانتا القائمتين متطابقتين تكون النتيجة 0.	<pre>list1 = [9,5,6,8] list2 = [9,6] print cmp(list1,list2) #gives -1</pre>
len(list)	تعطي طول القائمة	<pre>list1 = [9,5,6,8] print len(list1) # gives 4</pre>
max(list)	تعطي أكبر عنصر في القائمة	<pre>list1 = [9,5,6,8] print max(list1) # gives 9</pre>

list1 = [9,5,6,8] print min(list1) # gives 5	تعطي أصغر عنصر في القائمة	min(list)
list1 = (9,5,6,8) print list(list1) # gives [9, 5, 6, 8]	تحويل قاموس إلى قائمة	list(seq)

أدوات تستخدم مع القوائم

الأداة	مثال
الليصق	[1, 2, 3] + [4, 5, 6] # gives [1,2,3,4,5,6]
تكرار عناصر القائمة	['Python!'] * 3 # gives ['Python!', 'Python!', 'Python!']
التحقق	3 in [1, 2, 3] # gives True
تكرار لكل عنصر في القائمة	for x in [1, 2, 3]: print x

ملاحظات مهمة جداً

نسخ القائمة

```
lista = [2, 1, 4, 9, 8]
listb = lista
lista.pop()
print lista # [2, 1, 4, 9]
print listb # [2, 1, 4, 9]
```

لأن هذه الطريقة

```
listb = lista
```

تنشئ قائمة جديدة لكن كلا المتغيرين يشاران على نفس القيمة في الذاكرة ، لذلك أي تغيير يحدث لأحدهما يؤثر على الآخر

لنسخ قائمة بطريقة مستقلة نستعمل الطريقة التالية

```
lista = [2,1,4,3]
listb = lista[:]
lista.append(5)
print lista # [2,1,4,3,5]
print listb # [2,1,4,3]
```

حذف عنصر من القائمة

هناك طرق مختلفة لحذف عنصر من القائمة قد تتشابه ، لكنها مختلفة جداً

```
pop()
del
remove()
```

مثلاً سوف نجرب طريقة عمل كلاً منها مع هذه القائمة

```
myList = [0,4,3,5,7,6,4]
```

remove() تحذف أول عنصر مطابق للمدخل
مثلاً

```
myList.remove(4)
```

تعطينا

```
[0, 3, 5, 7, 6, 4]
```

هنا قد تم حذف أول عنصر مطابق ل 4 داخل القائمة
تعدل على القائمة مباشرة وليس لها أي إرجاع

del تحذف العنصر صاحب الفهرس المستخدم كمدخل
مثلاً

```
del myList[4]
```

تعطينا

```
[0, 4, 3, 5, 6, 4]
```

هنا قد تم حذف العنصر صاحب الفهرس رقم 4 وهو 7
تعدل على القائمة مباشرة وليس لها أي إرجاع

pop() تستخدم بطريقتين
بدون مدخل وفي هذه الحالة سوف تقوم بحذف آخر عنصر في القائمة
مثلاً

```
myList.pop()
```

تعطينا

```
[0, 4, 3, 5, 7, 6]
```

هنا قد تم حذف العنصر الأخير في القائمة وهو 4

مع مدخل هنا سوف تقوم بحذف العنصر صاحب الفهرس المحدد كمدخل
مثلاً

```
myList.pop(4)
```

تعطينا

```
[0, 4, 3, 5, 6, 4]
```

هنا قد تم حذف العنصر صاحب الفهرس رقم 4 وهو 7
لكن الاختلاف أن

pop لها مرجع وهو قيمة العنصر المحذوف

```
myList = [0,4,3,5,7,6,4]
```

```
x = myList.pop(4)
```

```
print x #7
```

```
print myList      #[0, 4, 3, 5, 6, 4]
```