PYTHON COURSE

Python course , from scratch to profissionalism ...

Made by : Taha Khaldoun Amro

دورة تعليم لغة بايثون

القوائم

تعتبر القوائم شكلا من أشكال المتغيرات التي تحتمل داخلها مجموعة من القيم النصية أو العددية او أي نوع بيانات اخر , و يمكن اعتبارها أنها من أهم أشكال حفظ البيانات الكثيرة أو التي تتصف بصفة معينة , أو يجمعها رابط مشترك , الان سنتعرف الى مفهوم القوائم و كيف يمكننا استعمالها و لما هي مهمة الى هذا الحد ؟

صنع القوائم :
يتم تعريف القوائم من خلال وضع مجموعة من البيانات داخل الأقواس المربعة [] , و يتم الفصل بين كل عنصر و اخر باستخدام الفاصلة (,) .

```
mylist = ['one' , 'two' , 'three']

print(mylist)
```

دورة تعليم لغة بايثون

القوائم

ترقيم العناصر :

في البداية يجب أن نتذكر بعض المعلومات المهمة في موضوع الترقيم (indexing) :

1. العد يبدأ من الصفر
2. كل عنصر داخل الكائن له رقم
3. يمكن الوصول للعنصر من خلال وضع اسم التغير و من ثم رقمه داخل أقواس مربعة [0]

و كذلك القوائم , يمكننا الدخول الى أي عنصر من خلال رقمه داخل القائمة .

```
mylist = ['one' , 'two' , 'three' , 1 , True]

print(mylist[1])

print(mylist[1:4]) # slicing ‘try adding some steps’
```

كما رأينا النتيجة , فاننا الان ندرك شيئا , أن التعامل مع القوائم ليس صعبا , و الان نعرف أنها تحتمل عدة أنواع من البيانات معا , و يمكننا بكل سهولة الوصول الى عناصرها .

القوائم

تغيير قيم العناصر :

يمكن تغيير قيمة العناصر التي داخل القائمة من خلال تعريف الموقع الخاص بالعنصر داخل القائمة بقيمة جديدة .

```
mylist = ['one' , 'two' , 'three' , 1 , True , [12 , 'sa3eed' , False ]]

print(mylist)

mylist[2] = 3

print(mylist)

print(mylist[-1][1]) # accessing the list in the list
```

دورة تعليم لغة بايثون

تقنيات القوائم

مثلها مثل اي نوع بيانات اخر في لغة بايثون , تمتلك القوائم مجموعة كبيرة من التقنيات التي تمكننا من التلاعب بالبيانات داخلها , و تسهل علينا اعمالا كثيرة .

1. append()

تعمل تقنية append أو الاضافة على اضافة عنصر جديد للقائمة في الموقع الأخير [-1] .

```
mylist = ['one' , 'two' , 'three' , 1 , True]
mylist.append(12) # change the value inside the brakets
print(mylist) # try to append a list in a list and see what happens
```

دورة تعليم لغة بايثون

القوائم

2. extend()

يعمل هذا الأمر على دمج قائمتين مع بعضهما البعض .

```
a = ['one' , 'two' , 'three' ]
b = [1 , 2 , 3 ]
print(a.extend(b))  # thefirst.concatinate( thesecond )
```

3. remove()

يمكن استنتاج عمل هذه التقنية من خلال اسمها , حيث أنها تعمل على ازالة عنصر من القائمة .

```
a = ['one' , 'two' , 'three' , 'one' , 'one'  ]
a.remove('one')
print(a) # removes one only
```

دورة تعليم لغة بايثون

تقنيات القوائم

4. sort()

تعمل هذه التقنية على ترتيب القوائم التي تحتوي على قيم عددية فقط , من الأكبر الى الأصغر و بالعكس .

```
a = [10 , 23 , 0 , -1 , 9 , 100 , -223]
a.sort() # try to put : reverse = -1 , in the brakets
print(a)
```

5. reverse()

أتذكرون عملية عكس النصوص في درس الslicing , حسنا فان هذه التقنية تعمل على عكس القائمة , بغض النظر عن أنواع و أشكال البيانات فيها .

```
a = ['one' , 'two' , 'three' , 12 , True , [1,2,3] ]
a.reverse()
print(a)
```

القوائم

6. clear()

يعمل هذا الأمر على محو كل العناصر داخل القائمة .

```
a = ['one' , 'two' , 'three' ]
a.clear()
print(a)
```

7. index()

تقوم بنفس المهمة التي تقوم بها مع النصوص , الا أنها حساسة , حيث أن كلمة "أحمد" ليست مثل "احمد" , و يتم التعامل معهما على أنهما نصان مختلفان .

```
a = ['one' , 'two' , 'three' ]

print(a.index('one'))
```

دورة تعليم لغة بايثون

تقنيات القوائم

8. copy()

في الواقع قد يشكل استعمال هذه التقنية , فارقا كبيرا بين المحترفين في اللغة و بين المبتدئين , حيث أنها تعمل على نسخ القائمة , نسخا سطحيا , مما يعني : مهما تغيرت القائمة في الكود , تبقى هذه النسخة محتفظة بالشكل الذي نسخت عليه , طبق الكود الاتي للمزيد من الفهم .

```
mylist = ['one' , 'two' , 'three' , 1 , True , [12 , 'sa3eed' , False ]]
mycopy = mylist.copy()
print(mylist)
print(mycopy)

mylist[2] = 3
mylist.append(1000)

print(mylist)
print(mycopy)
```

دورة تعليم لغة بايثون

القوائم

9. count()

في الواقع , لقد مر علينا هذا الأمر من قبل , و ذلك في عملية عد حرف ما في نص , و كذلك الأمر هنا , حيث يتم عد عنصر ما في القائمة .

```
mylist = [1 ,2 ,3, 2, 5, 22, 54,1, ,66, 2, 138]

print(mylist.count(2))
```

10. pop()

تقوم هذه التقنية بارجاع العنصر الذي يحمل الرقم المعطى داخلها .

```
mylist = ['one' , 'two' , 'three' , 1 , True ]

print(mylist.pop(4))
```

دورة تعليم لغة بايثون

تقنيات القوائم

11. insert()

يعمل الأمر insert على ادخال قيمة معينة قبل موقع معين , حيث يجب علينا اعطائه القمية التي نريد اضافتها و الموقع التي نريد أن تكون قبله .

```
mylist = ['one' , 'two' , 'three' , 1 , True ]
mylist.insert(2 , 'hi') # try index -1 (the last index)
print(mylist)
```

تعد دروس القوائم من أهم الدروس التي قد يتعلمها المبرمج في مسيرته , لذلك سنرفق لكم الروابط التالية التي تشرح القوائم بشكل مفصل و سهل الفهم :

https://youtu.be/EpZH9JozUzA?si=vTrnxLQoGlvmiVKd

https://youtu.be/b5cFjJ278Vk?si=lgSu0FHCZuelxXha

https://youtu.be/pP0QJbJalik?si=oz3GxVcCoy3FK95T