PYTHON COURSE

Python course , from scratch to profissionalism ...

Made by : Taha Khaldoun Amro

Boolean values

تعتبر القيم المنطقية من أهم المفاهيم التي يجب أن يتعلمها أي مبرمج , حيث أنها تشكل اللبنة الأساسية لبناء الأكواد الكبيرة التي تحتوي على أي جملة شرطية .
ففي لغة بايثون , كل نوع من أنواع البيانات له قيمة منطقية , و هذه القيم اما ان تكون True أو أن تكون False .

في وقت سابق , كنا قد تعاملنا مع القيم المنطقية في بعض تقنيات الStrings , حيث يوجد مجموعة من التقنيات التي تعيد لنا قيما منطقية , نستذكر منها المثال الاتي :

```
name = 'salmon'

print(name.islower())
```

يقوم الكود السابق بفحص ما اذا كان النص يتكون من حروف صغيرة فقط , و يعيد لنا قيمة منطقية , تفيد ما اذا كان الشرط True أو False .

Boolean values

أنواع البيانات ذات القيم الTrue :

```
print(bool('Hi')) # string
print(bool(123)) # int
print(bool(10283.13)) # float
print(bool([12 , 34, 54])) # list
print(bool(True)) # True
print(bool((1,2,3,4))) # tuple
print(bool({13 , 4, 8,2, 1,6})) # set
```

بكل بساطة , فان أنواع البيانات الاتية , تعتبر قيمتها True .
و لكن و في المقابل , فان نفس أنواع البيانات قد تكون False في بعض الحالات الخاصة التي سنتعرف عليها الان .

Boolean values

أنواع البيانات ذات القيم False :

```
print(bool('')) # string --> empty
print(bool(0)) # int --> zero
print(bool([])) # list --> empty
print(bool(False)) # False
print(bool(())) # tuple --> empty
print(bool({})) # set --> empty
print(bool(None)) # nothing --> empty
```

كما رأينا الانٍ , فان كل البيانات الفارغة تعتبر قيمتها False , ذلك بالاضافة الى مجموعة أخرى من البيانات التي تعد منطقياً فارغةً مما يعني أنها تعيد القيمة false و ذلك مثل القيمة None و التي تعني لا شيء .

Boolean values

يمكننا استخدام القيم المنطقية لفحص ما اذا كانت العبارات الرياضية صحيحة :

```
print(2 > 4)
print(2 == 4)
print(100 < 2012)
print(123.12 > -4)
```

العبارات الاتية هي عبارات رياضية , حيث نفحص ما اذا كان عدد اكبر من الاخر , أو اصغر منه , او مساوي له , و تتم طباعة النتيجة على شكل قيمة منطقية .

Boolean operators

تعتبر العمليات المنطقية واحدة من أهم الدروس التي تساعدنا في الكثير من الأمور داخل عالم البرمجة .

1. and operator

تعمل العملية and على فحص شرطين اثنين , و حتى تكون القيمة True يجب ان يتحقق الشرطان معاً , فاذا لم يتحقق أأدهما على الأقل تكون الحالة False .

```
age = 20
country = 'palestine'
print(age > 15)
print(country == 'palestine')
print(country == 'palestine' and age > 18) # try making one false and the other true
```

فكما رأينا في المثال السابق , فحتى نحصل على القيمة True يجب أن يكون كلا الشرطين True و الا فسنحصل على القيمة False .

Boolean values

2. or operator

يتم استخدام عملية الorمن أجل الحصول على قيمة True في حال تحقق أحد الشروط على الأقل , فمثلاً لو تحققت جميع الشروط , ستكون النتيجة True , و اذا لم يتحقق أي شرط ستكون النتيجة False .

```
age = 20
country = 'palestine'
print(age > 15)
print(country == 'palestine')
print(country == 'palestine' or age > 18) # try making one false and the other true
```

Membership operators

سنتكلم في هذا الجزء من درس الجمل القيم المنطقية عن جمل العضوية , و هي ببساطة شروط يمكن استخدامها , لمعرفة ما اذا كان كائن iterable يحتوي على قيمة معينة .

حيث تمكننا هذه الجمل من فحص ما اذا كانت قيمة موجودة داخل list أو tuple أو string, أو أنها غير موجودة .

```
name = 'Jamal'

print('J' in name)
print('j' in name)
print('m' in name)
```

في الكود السابق , قمنا بفحص ما اذا كانت مجموعة من النصوص موجودة داخل نص اخر , حيث تقوم العملية الاتية بارجاع قيمة منطقية .

Membership operators

كما تمكننا من فحص عضوية نص في نص , فاننا يمكننا فحص عضوية عنصر في list .

```
mylist = ['Hi' , 1 , 2 ,'wow!']

print('Hi' in mylist)
print([] in mylist)
print(3 in mylist)
```

حاول الان ان تبدل الlist الى tuple و جرب فحص عضوية أي عنصر فيها .

Membership operators

تعلمنا الان التعامل مع الجملة in و سنتعلم الان التعامل مع ما هو عكسها , الا و هي not in و التي تقوم بفحص عدم وجود عنصر ما في iterable , حيث تعيد قيمة True في حال لم تجد العصر , و تعيد قيمة False في حال وجدته .

جرب ان تقوم باستخدام الاكواد السابقة مع هذه الجملة .

للمزيد من المعلومات :

https://youtu.be/FGnMK1y9TkE?si=fo70Xg4RZNXvrIQZ