يكتبها – الشريف أحمد حسن خلف

مقدمة

حياة بلا قيود ، يقول مالك بن نبي " إن المجتمع الذي يعمل فيه كل فرد ما يحلو له ليس مجتمعاً ولكنه إما مجتمع في بداية تكونه وإما مجتمع بدأ حركة الانسحاب من التاريخ فهو بقية مجتمع " فليست الحضارة بالحرية المطلقة التى ليس لها قيود . إن الدولة التي ليست لها قوانين صارمة تحيى حياة متدهورة , ما بين قانون لا يردع المجرم و لا قانون يحمى المواطن . فالحياة لا بد لها من قوانين تحميها كى تستمر على الطريق المستقيم . فلابد لكل شئ من شروط وقيود وهذه هي سنن وقوانين أودعها الله سبحانه و تعالى فى هذا الكون ولا أفضل و لا أعظم من قوانين لهذه الدنيا إلا قوانين خالقها سبحانه وتعالى ألا وهي إتباع السراج المنير صلى الله عليه و سلم .

الشرط والمنطق

البرمجة أيضا لابد لها من وجود عمليات شرطية ومنطقية كي تصبح سليمة , ففي مثال حساب حجم الإسطوانة في المقال السابق قد يكون فيه خطأ , تخيل لو أن المدخل كان بإشارة سالبة ؟ فسيعطيك الحجم بالإشارة السالبة وهذا خطأ , و لنفترض أن بالمسألة حساب للوغاريتمات وكان المدخل سالب القيمة ؟ فسيعطيك ناتج خطأ . فكيف نحل هذه المشكلة ؟ بالطبع نحتاج إلى عمل شرط يرفض المدخل السالب و يقبل المدخلات الموجبة و لهذا نستخدم الجمل الشرطية .

الجمل الشرطية

الجمل الشرطية يقوم فيها المفسر بعمل مقارنة مثلا أو يكون مقيد بشرط منطقي معين و سنضرب أمثلة عديدة يتضح منها المعنى بكل وضوح ويسر .

نستخدم للجمل الشرطية فى لغة البايثون كلمة if وتستخدم هكذا :

```
if x > 0:
  print "x is positive"
```

نبدأ السطر بالكلمة المفتاحية if وتضع العملية المنطقية التي سنتطرق إليها الآن وبعد العملية تضع النقطتين ثم نبدأ السطر الجديد بعد أربع مسافات كالدوال ثم الجملة أو البيان .

العمليات المنطقية

العملية الأولى هى عملية مقارنة المساواة وهى عبارة عن علامة يساوى مكررة مرتان. جرب تكتب فى المفسر التفاعلى للغة البايثون 2 ==2 و أنظر الناتج:

```
>>> 2 == 2
True
>>> 2 == 3
False
```

الناتج يكون إما Flase أو True وهذه نواتج العمليات المنطقية إما أن يخبرك بأن العملية حقيقية أو مزيفة .

```
>>> type(True)
<type 'bool'>
```

ولاحظ أن True تبدأ بحرف كبير وليست true ، و باقي العمليات المنطقية كالآتي:

```
x != y        # x لا تساوى y
x > y         # x أكبر من y
x < y         # x أقل من y
x >= y        # x أكبر من أو يساوى y
x <= y        # x أصغر من أو يساوى y
```

باقي القسمة

عملية باقي القسمة هي الحصول على باقي قسمة عددين غير قابلين للقسمة كقسمة 10 على 3 و ناتجها أنها تساوي 3 والباقي واحد . وهي العلامة % وهي تعطيك الباقي فقط وليس ناتج عملية القسمة جرب في المفسر التفاعلي :

```
>>> 10 % 9
1
>>> 10 % 3
1
لاحظ أن الباقي واحد في نفس الحالتين لإنه يحسب باقي القسمة فقط
>>> 10 % 4
2
```