الوحدة .... Module

الوحدة هـــى ملف يحتوى على مجموعـــة دوال لغرض معيـــن والوحدة يجب عمل إستيراد لها ، مثلا يوجد فى البايثون وحدة الرياضيات بها مجموعة دوال للعمليات الرياضية ،مثل حساب الزوايا المثلثية " جا و جتا و ظا و .. " و كيفية إستيراد الوحدة بالأمر import و لابد من عمل إستيراد قبل تنفيذ أى دالة منها .

```
>>> import math
```

لإستخدام دوال الوحدة تكتب إسم الوحدة متبوعا بنقطة و بعدها إسم الدالة " و القوسين لإدخال المدخلات "

```
>>> x=math.log10(30.0)
>>> print x
1.47712125472
```

هذاالدالة لحساب لوغاريتم 30.0 للأساس 10 و لو لم نكتب الرقم 10 تصبح دالة اللوغاريتم للأساس الطبيعى e .

أفتح محرر النصوص المفضل لديك و أكتب هذا البرنامج البسيط:

```
import math
angle=input(" Enter your angle .." )
print "the result sin(your angle is " , math.sin(angle)
print "the result cos(your angle is " , math.cos(angle)
print "the result tan(your angle is " , math.tan(angle)
```

السطر الأول هو إستيراد للوحدة التى نريدها و السطر الثانى هو متغير إسمه angle بدالة الإدخال input و هى دالة إدخال لا تقبل إلا الأرقام فقط و السطر الثانى هو الأمر print و به دالة جيب الزاوية sin و هكذا إستخدمنا cos و tan . كما يمكن وضع مصطلح أو تعبير (expression) كمدخل(argument) للدوال و الدالة ما هى إلا تعبير فمثلا :

```
x = math.sqrt(math.log(10.0))
```

قمـــت بإدخال دالة آخرى داخـــل الدالة الأصـــلية فالدالة الأصـــلية math.sqrt () هى لحساب الجذر التربيعى للمدخل و المدخل عبارة عن دالة تحسب قيمة اللوغاريتم.

كيفية تعريف دالة جديدة

تســـتطيع بلغـــة البايثون أن تقوم بتعريـــف دالة "وظيفـــة " جديدة لإستخدامها و الفائدة من عمل دالة هى توفير للخطوات فتقوم بعمل دالة تنجز وظيفة معينة فيما بعد فهى تقلل أسطر البرنامج و تسهل عمل مراجعة و تطوير البرنامج و تتفوق بايثون فى هذا المجال بشكل ملحوظ فهـــى لغـــة مفضلة عنـــد علماء الرياضيات و الباحثيـــن و المهندسين و مصممى الرسوم ثلاثية الأبعاد و برمجة الألعاب فهى أشهر اللغات فى برمجة الألعاب فهى قوية و سهلة .. تعال نتعرف عن كيفيـــة تعريف دالة جديدة داخـــل البايثون . لماذا تسمى بتعريف ؟ لإنك بكل بساطة تقوم بتعريف المفسر أن هذه دالة سيستخدمها و

تعريف الدالة يبدأ بكلمة def ثم إسم الدالة ثم قوسين ثم تنهيها بـ : و هذا يسمى رأس الدالة والجزء الثانى يبدأ فى السطر التالى للرأس بعد أربعة مسافات من أول السطر و تكون فيه الأوامر و البيانات المختلفـــة (STATEMENTS) فلنضرب مثال لتعريـــف دالة ..

```
def NAME( LIST OF PARAMETERS ):
   STATEMENTS
```

أسماء الدوال تتبع نفس قوانين تسمية المتغيرات التى ذكرتها فى المقال السابق ، ولا تسمى دالة بإسم متغير قمت بتسميته أو دالة آخرى بنفس الاسم وبين القوسين يكون المدخل و فى السطر الثانى بعد أربع مسافات يبدأ جسم الدالة و يمكن وضع عدد غير محدود من البيانات .

```
def your_name():
   print   " Your name"
```

هذه دالة تقوم بطباعة سلسلة نصية Your name إسم الدالة your_name و الأقواس الفارغة () تشير إلى أن هذه الدالة لا تأخذ مدخلات و السطر الثانى يحوى بيان (STATEMENT) للطباعة .

و كيفية استخدامها فى ملف نصى (script) هكذا " قد سبق و بينت فى المقال الأول الفرق بين كتابة البرنامج فى ملف نصى script و إستخدام المفسر التفاعلى للغة البايثون بكتابة كلمة python فى سطر الأوامر "

```
def your_name():
   print " Your name "
your_name()
```

وناتج هذا البرنامج سيكون تنفيذ أمر الطباعة ، فاستخدام الدالة هو بكتابة اسمها متبوعا بقوسين ، ولو كانت تحتاج مدخلات فستكتب داخل القوسين كما سنوضح واذ لم تكن تحتاج فتكون كما هى فارغة .

يمكن إستدعاء دالة داخل دالة آخرى ، وسوف نرى فى هذا المثال ولنفترض إن البرنامج يحتوى على الدالة السابقة أيضا ونضيف دالة جديدة.

```
def my_name():
   print " My name is sherif "
```

و الدالة القادمة ستحوى الدالتين السابقتين your_name و my_name .

```
def my_your_name():
   my_name()
   your_name()
```

ثم تقوم بتنفيذها فى سطر جديد my_your_name ()