لغة البرمجة بايثون
Python

يكتبه – الشريف أحمد حسن خلف

قبل البدء

تكلمنا فى المقال السابق عن المتغيرات و الثوابت و أدوات الإدخال و العمليات الحسابية ، وفى هذا المقال سنتكلم عن الدوال و كيفية استخدامها و الاستفادة منها و أهميتها مع ضرب الأمثلة المتعددة فبالمثال يتضح المقال .

ترجمة كلمة Functions تعنى وظيفة أو دالة ؛ الدالة فى الرياضيات هى علاقة بين متغير تابع و متغير مستقل مثلا دالة تدرس العلاقة بين س المتغير المستقل و ص المتغير التابع .

```
ص = س – 1
# نكتبها كدالة هكذا
ص = د(س) = س – ١
```

يعنى عند تغيير قيمة س تتغير قيمة ص تبعا لهذا التغير فنقول ؛ د(5)= 5 – 1 إذا د(5) = 4 يعنى قيمة ص عندما نضع قيمة س بخمسة . الرقم الذى بين القوسين يسمى مدخل argument و النتيجة تسمى القيمة المرجوعة أو العائدة return value إذا الدالة تأخذ قيمة مدخلة و تعطيك نتيجة وهناك دالة تأخذ قيمتين مثلا :

```
دالة ج = د(س , ص ) = 6س + 7 ص
د(3, 5) = 6 * 5 + 7 * 3
```

و قد تتعدد المدخلات و تكون دالة ج دالة فى (س و ص) و هذا تبسيط جدا لمفهوم الدوال فهى مجال واسع فى علم الرياضيات .

الدوال فى البايثون

لكل دالة إسم معين تستدعيها به فمثلا دالة تحديد النوع type

```
>>> type("32")
<type 'str'>
# لاحظ أنه حدد النوع كنص لانها بين علامتى تنصيص
>>> srings = type("32")
>>> print strings
<type 'str'>
>>> type(32)
<type 'int''>
```

كتبنا إسم الدالة و بعدها قوسين بينهما المدخل المراد تطبيق الوظيفة المعينة عليه و هى تحديد نوع المتغير .

```
>>> type(32.01)
<type 'float'>
```

هنا حدد المتغير 32.01 كرقم عشرى float .

دالة int تحول الرقم العشرى float أو الرقم الذى بين علامات تنصيص إلى رقم صحيح integer و لا تقبل النصوص.

```
>>> int("32")
32
>>> int(3.99999)
3
>>> int("World")
ValueError: invalid literal for int() with base 10: 'World'
```

وكذلك دالة float تأخذ متغير العدد الصحيح و تحوله إلى عشرى والعدد العشرى الذى بين علامتي التنصيص و لاتقبل النصوص أيضا .

```
>>> float(32)
32.0
>>> float("3.14159")
3.14159
```

ودالة str التى تغير نوع المتغيرات إلى نوع نصي.

```
>>> x=5
>>> x=str(x)
>>> type(x)
<type 'str'>
>>> str(5.555)
'5.555'
```

ما الفائدة من تحويل الرقم الصحيح إلى عشرى ؟

أكتب هكذا و تأكد من الناتج

```
>>> minute = 59
>>> minute/60
0
```

الناتج صفر لماذا ؟ لأن المفسر يقوم بإخراج النتائج بالأعداد الصحيحة فقط و ما الحل ؟ الحل هو جعل نوع المتغير minute كنوع عشرى بالدالة float .

```
>>> float(minute) / 60
0.98333333333333328
```

أو كتابة المقسوم عليه 60 هكذا 60.0 فيعرف المفسر أن نوع هذا المتغير رقم عشرى .

```
>>> minute = 59
>>> minute / 60.0
0.983333333333
```

هذه بعض دوال البايثون و هى كثيرة جدا .