سيكون شكل الملف النصى هكذا .

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

و الناتج سيكون

```
[sherif@localhost ~]$ python mathes.py
 My name is sherif
 Your name
```

لاحظ أنه عندما تستدعى دالة فمن المنطقى أن تكون معرفة مسبقا فالمفسر يبدأ البرنامج من السطر الأول ثم الثانى و هكذا فلابد من الترتيب فى تعريف الدوال و إستدعائها .

دوال بمدخلات argument

لتعريف دالة بمدخلات لابد بوضع بارمتر داخل القوسين و البارمترات هى كالمتغيرات لكنها قاصرة على الدالة فقط فمثلا.

```
def two_man(man):
   print man , man
```

لإستدعاء الدالة ووضع مدخلات مكان البارمتر man نكتب الدالة و بين القوسين أي قيمة بدلا من كلمة man و هذه الكلمة يمكن تغييرها لأى اسم.

```
two_man("sherif")
two_man(555)
```

الناتج بعدها.

```
sherif sherif
555 555
```

يمكن وضع بيان فى المدخل فمثلا.

```
two_man(6*4)
```

أو متغير سبق كتابته.

```
dist=" Fedora and Joatha "
two_man(dist)
```

سيكون الناتج طباعة المتغير مرتين .

و يمكن تعريف دالة بمدخلين مثلا نقوم بعمل دالة لجمع شيئين.

```
def cat_twice(part1, part2):
   cat = part1 + part2
   print cat
```

هنا سيأخذ قيميتن و لتنفيذ الدالة لابد من إدخال قيمتين عددية أو نصية.

```
cat_twice("me "," you")
cat_twice(5,10)
```

سيضم كلمتين me و you و سيجمع الرقمين 5 و 10 جرب أنت بنفسك ، وشكل الملف النصى سيكون هكذا.

```
def cat_twice(part1, part2):
    cat = part1 + part2
    print cat
cat_twice("me "," you")
cat_twice(5,10)
```

لاحظ هنا أن هناك متغير داخل الدالة إسمه cat حاول عمل أمر طباعة له print cat ستجد أن المفسر يعطيك خطأ معناه أن هذا المتغير لم يتم التعرف عليه .. لماذا ؟ لإن المتغيرات داخل الدوال محلية و قاصرة في داخل الدالة فقط و ليس لها أى علاقة بباقى البرنامج .

الدوال المنتجة و الغير منتجة

إستكمالا للبرنامج السابق قم بعمل متغير يساوي الدالة cat_twice(5,10) و سمه مثلا result .

```
result=cat_twice(5,10)
print result
```

ستجد أن قيمة result هى None لكن لماذا ؟ و لماذا الدوال الآخرى كدوال وحدة math تقوم بحفظ القيمة ؟

هناك نوعين من الدوال ، دالة منتجة ودالة غير منتجة فالدالة قاصرة على تنفيذها و لا تقوم بإرجاع قيمة معينة فى الذاكرة و لكن كى تقوم بإرجاع أو حفظ قيمة المخرجات تحتاج إلى وضع أمر return و سنرى بالمثال كيف نستخدمه و الدوال التى كتبنها لا تحتوى على return . سنكتب دالة تقوم بحساب مساحة الدائرة و قانون المساحة هو حاصل ضرب ط فى تربيع نصف القطر.

```
import math
def area(radius):
    temp = math.pi * radius**2
    return temp
area_c=area(5)
print area_c
```

فى هذا البرنامج لن يعطيك قيمة المتغير بـ None بل سيعطيك قيمة حقيقية ويمكن إختصار الدالة هكذا.

```
def area(radius):
   return math.pi * radius**2
```