

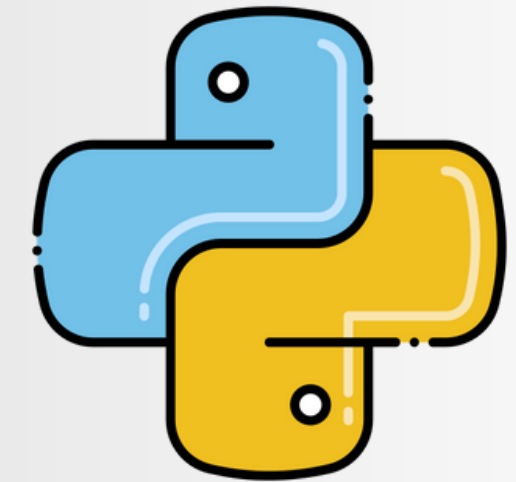
أساسيات لغة البايثون

المحتويات

- ماهي الدوال و ما دورها ؟
- كيفية إستدعاء الدوال.
- طريقة إنشاء الدوال.
- أمثلة لترسيخ الفهم

الدوال

- Functions -



أساسيات لغة البايثون

• ماهي الدوال و ما دورها ؟

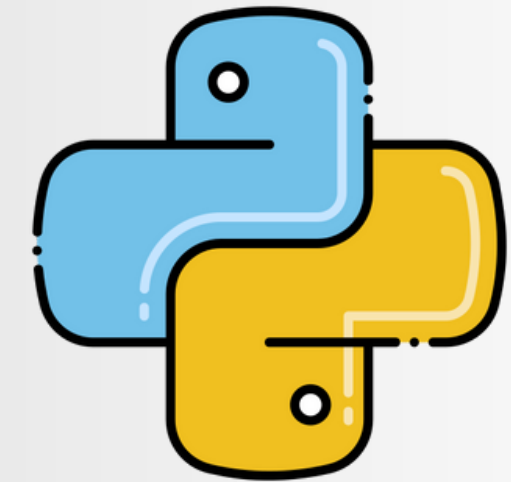
الدوال هي مجموعة من الأوامر مجمعة في كتلة واحدة لتنفيذ وظيفة معينة

أنواع الدوال

- 1 - الدوال المدمجة في البايثون
- 2 - الدوال المعروفة من قبل المبرمج

الدوال

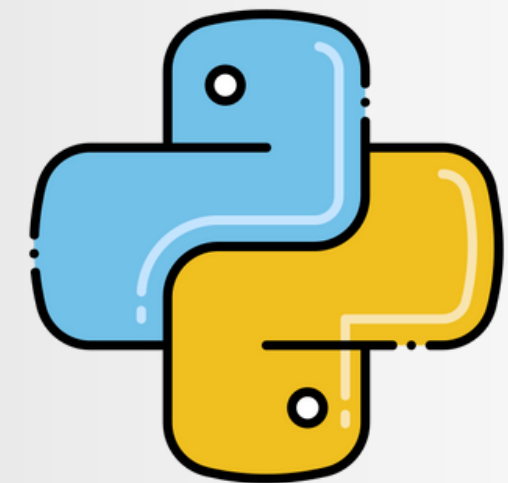
- Functions -



أساسيات لغة البايثون

الدوال

- Functions -



• ماهي الدوال و ما دورها ؟

الدوال المدمجة

الدوال المدمجة تستدعى دون الحاجة الى اعادة برمجتها .
مثال :

• طريقة إستدعاء الدوال .

لاستدعاء الدالة نقوم بكتابة اسمها في سطر جديد .

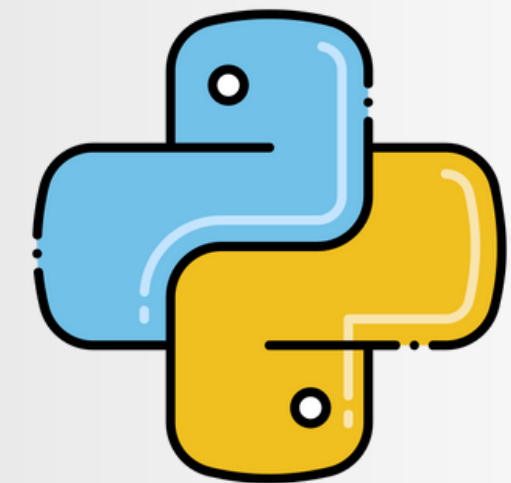
```
print()  
range()  
len()
```

طباعة نتيجة الكود أو البرنامج
التكرار على بيانات رقمية محددة ببداية و نهاية
حساب عدد عناصر متغير ما

أساسيات لغة البايثون

الدوال

- Functions -



- طريقة إنشاء الدوال.

الدوال المعروفة من قبل المبرمج

- الدوال متعددة الأسطر البرمجية

لإنشاء دالة متعددة الأسطر البرمجية نقوم بكتابة كلمة محجوزة **def** ثم نعطي اسم معبر عن وظيفة الدالة ثم نفتح قوسين عاديين () ثم نضع نقطتين فوق بعض و نضغط على زر **enter** لبرمجة مهام الدالة مع ترك أربع مسافات في السطر الجديد . مثال

```
def open_file():  
    print('file opened !!')  
open_file()
```

```
> def open_file(): ...  
open_file()  
print('Done !!')  
open_file()
```

أساسيات لغة البايثون

الدوال المعروفة من قبل المبرمج

- الدوال متعددة الأسطر البرمجية

تحتمل الدالة معاملات أو قيم افتراضية تدعى parameters . تكتب بين قوسي الدالة .

- يجب أن تكون أسماء القيم متبرة أيضا

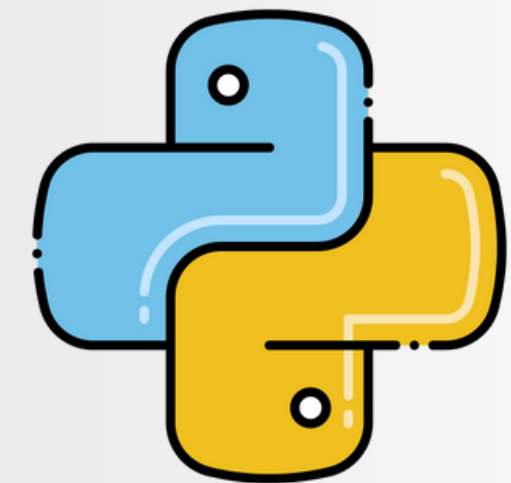
أمثلة :

```
def welcome_msg(name):  
    print('Welcome Mr/Ms : ' + name)  
  
welcome_msg('PyQonsole')
```

Welcome Mr/Ms : PyQonsole

الدوال

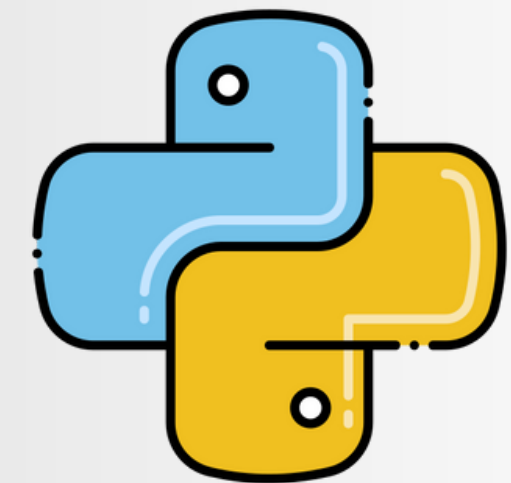
- Functions -



أساسيات لغة البايثون

الدوال

- Functions -



الدوال المعروفة من قبل المبرمج

• الدوال متعددة الأسطر البرمجية

أمثلة :

```
def result(number):  
    print(1 + number)
```

```
result(10)
```

```
def add(myList):  
    print([17,2, 2025] + myList)
```

```
add([1, 0, 0])
```

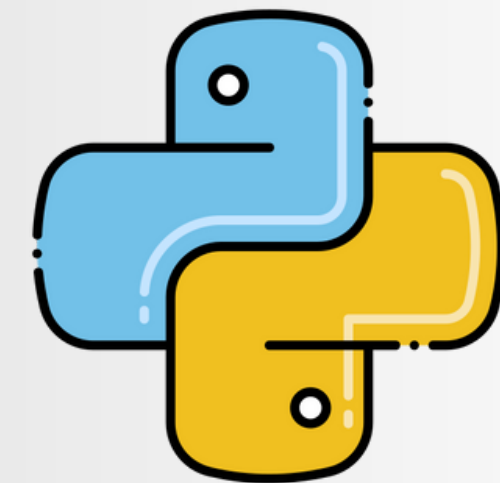
```
def pop_item(list_var):  
    list_var[-1] = ''  
    return list_var  
  
print(pop_item(['PyQonsole', 'Sami']))
```

في هذا المثال استخدمنا كلمة محجوزة **return** والتي تقوم بإرجاع القيمة و حفظها و استدعاء الدالة يتطلب طباعة الدالة ككل عكس الدوال السابقة التي تقوم بالطباعة فور استدعاءها

أساسيات لغة البايثون

الدوال

- Functions -



الدوال المعروفة من قبل المبرمج

أمثلة :

```
def user_info(first_name = 'Admin', last_name = 'Root'):  
    print('- First Name : ' + first_name + '\n- Last Name : ' + last_name)
```

في هذا المثال قمنا بإنشاء دالة تقوم بطباعة أسماء افتراضية . فعند استدعاء الدالة دون تغيير القيم الافتراضية . فالنتيجة ستكون

• استدعاء الدالة

```
user_info()
```

```
- First Name : Admin  
- Last Name : Root
```

• نتيجة الدالة

أما عند استدعاء الدالة و تغيير القيم الافتراضية ستكون النتيجة كالآتي

```
user_info(first_name='PyQonsole', last_name='Khalil')
```

```
- First Name : PyQonsole  
- Last Name : Khalil
```

الدوال المعروفة من قبل المبرمج

• الدوال متعددة الأسطر البرمجية

أمثلة :

```
def nothing(*args, **kwargs):  
    print(args, kwargs)
```

```
nothing('PyQonsole', 'Is', 'Cool', [17,2,2025], ytchannel = 'PyQonsole')
```

```
('PyQonsole', 'Is', 'Cool', [17, 2, 2025]) {'ytchannel': 'PyQonsole'}
```

في هذا المثال استخدمنا كلمتين مجوزتين *args و **kwargs عند إنشاء الدالة حيث تعني

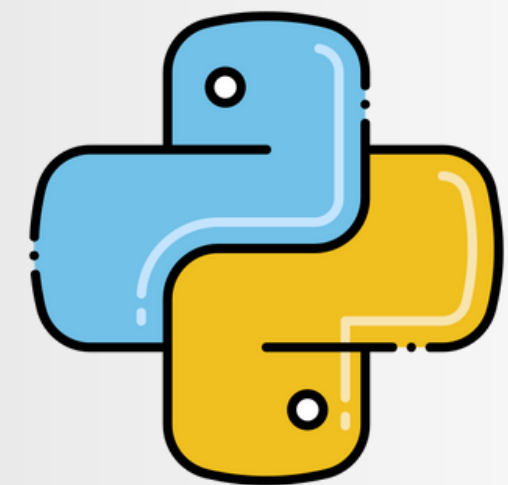
args : arguments

kwargs : key word arguments

أساسيات لغة البايثون

الدوال

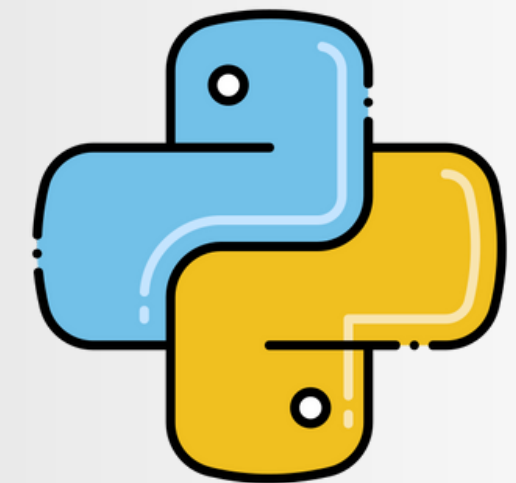
- Functions -



أساسيات لغة البايثون

الدوال

- Functions -



الدوال المعروفة من قبل المبرمج

- الدوال متعددة الأسطر البرمجية

ملاحظة :

ترتيب المعاملات و القيم الافتراضية يكون كما في
المثال أدناه

```
def nothing(a, b, *args, **kwargs):
```

حيث نضع القيم الافتراضية أولاً ثم نضيف المعاملات
أو args و kwargs
فإن اختلاف الترتيب سيؤدي إلى إظهار خطأ
بالبرنامج.

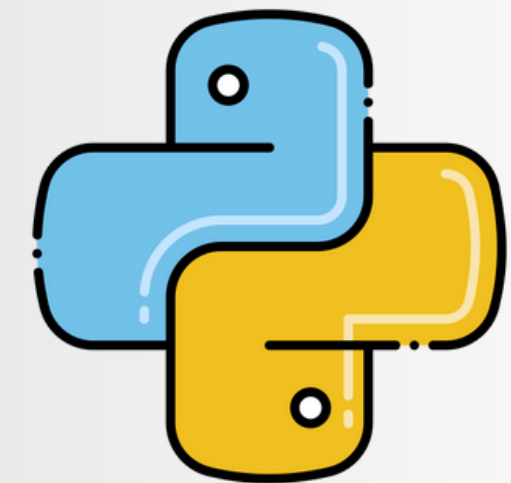
أساسيات لغة البايثون

• طريقة إنشاء الدوال.

• الدوال ذات السطر الواحد

الدوال

- Functions -



```
Lambda : print('Subbed !!')
```

```
mul = lambda number_1, number_2: number_1 * number_2  
print(mul(5, 5))
```

```
names = lambda myList: myList  
  
for name in names(['PyQonsole', 'Khalil', 'Sami']):  
    print(name)
```