

# CREATE FUNCTIONS

### 3- توظيف النوال ضمن الحلول البرمجية.

## ■ ما هي الدوال؟

الدالة هي كتلة من الجمل البرمجية الجاهزة مسبقاً، يتم استدعاؤها عند اللزوم، عادة ما ترافقها عوامل خاصّة بها تسمى "parameters" لاستخدامها للوصول إلى النتيجة المطلوبة. عادة ما تقوم الدالة بإرجاع النتيجة إلى البرنامج الرئيسي الذي تم استدعاؤها منه.

كما أنّها تساهم في حلّ المسائل المعقّدة بحيث يتمّ تقسيم المشكل الرئيسي إلى مشكلات بسيطة يتمّ حلّها بطريقة منفصلة ثمّ تجميعها للحصول على الحلّ النهائي.

## PSEUDOCODE

### 1- قالب العام للدالة ضمن الكود المزيّف PSEUDOCODE:

FUNCTION *function\_name* (*par1, par2, ...*)

Instructions

.....

END FUNCTION

## PYTHON

### 2- قالب العام للدالة ضمن لغة البرمجة PYTHON:

```
def function_name(parameters):
```

```
    instructions
```

```
    .....
```

أ. دالة بدون إرجاع نتيجة:

i. يبدأ بناء الدالة بـ **def**

ii. اسم الدالة

iii. متغيرات الدالة إن وجدت

iv. ترك مسافة بادئة وكتابة الأوامر instructions.

سأنشئ دالة تقوم بإظهار رسالة فقط ولا أحتاج منها إرجاع أية نتيجة.

```
File Edit Format Run Options Window Help
def msg():
    print("This is a function without parameters.")
```

- لكي تنفذ هذه الدالة سأقوم باستدعائها كآتي:

```
File Edit Format Run Options Window Help
def msg():
    print("This is a function without parameters.")

#call of the function msg()

msg()
```

استدعاء الدالة يأتي  
بعد تعيينها.

- أنفذ البرنامج فأتحصل على النتيجة الآتية:

```
This is a function without parameters.
```

ب. دالة مع إرجاع نتيجة:

```
File Edit Format Run Options Window Help
def sumTwo(a, b):
    s=a+b
    return s
#### نهاية الدالة ####
```

1- سأنشئ دالة لحساب مجموع عددين.

2- عدد عوامل الدالة parameters هو 2 (a و b) وهو عدد الأعداد التي سأجمعها.

3- return s: تمثل نتيجة الدالة التي

ستتحول إلى الجزء الرئيسي من البرنامج.

```
##### الجزء الرئيسي من البرنامج #####
x=5
y=4
z=sumTwo(x, y) # استدعاء الدالة
print(z)
```

4- استدعاء الدالة

من الجزء الرئيسي  
من البرنامج.

5- يجب استدعاء  
الدالة بنفس عدد

العوامل parameters التي أنشأت بها وبنفس الترتيب.

6- يتم تعويض عوامل الدالة الرئيسية بالعوامل التي استدعيت بها.

7- أنفذ البرنامج وألاحظ النتيجة ← ستكون النتيجة = 9

مثال 3

1- سأنشئ دالة لحساب كتلة الجسم:

أ. سأحتاج إلى متغير الطول height بالمتر.

ب. سأحتاج إلى الوزن weight بالكيلو.

ج. معادلة حساب كتلة الجسم BodyMass =

$$\text{BodyMass} = \text{weight} / (\text{height} * \text{height})$$

2- إنشاء الدالة:

```
File Edit Format Run Options Window Help
def bodym(h, w):
    b = w / (h*h)
    return b
```

3- الجزء الرئيسي من البرنامج:

```
### الجزء الرئيسي من البرنامج ###
weight = input("Enter your weight in kg:")
height = input("Enter your height in meter:")
BM=bodym(float(height), int(weight))
print(BM)
result(BM)
```

4- لا أنسى أن القيم المدخلة عن طريق input تكون من نوع نصي، لذلك قمت بتحويلها عند استدعاء الدالة (height) من نوع عشري لذلك حولته إلى float و weight من نوع عدد صحيح فحوّلته إلى int)

5- إذا علمت أنّ:

أ. الوزن المثالي يكون بين 18.5 و 24.9

ب. مرحلة السمنة من 25.0 وما فوق

ج. مرحلة سوء التغذية من 18.4 وأقلّ

6- أنشئ، أسفل الدالة الأولى، دالة جديدة تسمى result تظهر وصف نتيجة حساب كتلة الجسم وفق المعايير أعلاه.

```
def result(bm):
    if bm>=18.5 and bm < 25.0:
        print("excelent")
    elif bm>=25.0:
        print("Overweight")
    else:
        print("Underweight")
```

7- استدعاء دالة result يكون بعد استدعاء دالة bodym.

```
### الجزء الرئيسي من البرنامج ###
weight = input("Entyer your weight in kg:")
height = input ("Enter your height in meter:")
BM=bodym(float(height), int(weight))
print(BM)
result(BM)
```

## 8- البرنامج كاملاً:

```

File Edit Format Run Options Window Help
def bodym(h,w):
    b = w/(h*h)
    return b

def result(bm):
    if bm>=18.5 and bm < 25.0:
        print("excellent")
    elif bm>=25.0:
        print("Overweight")
    else:
        print("Underweight")

### الجزء الرئيسي من البرنامج ###
weight = input("Enter your weight in kg:")
height = input("Enter your height in meter:")
BM=bodym(float(height), int(weight))
print(BM)
result(BM)

```

مثال 4

- 1- سأنشأ دالة باسم **CreateList** لإدخال عناصر سلسلة **list** مكونة من أعداد صحيحة تُحدّد عددها.
- 2- سأنشأ دالة باسم **SumList** لحساب مجموع عناصر السلسلة.
- 3- إظهار النتيجة.

```

File Edit Format Run Options Window Help
1 ##### Create the list #####
2 def CreateList():
3     ls=[]
4     x=input("Length of the list:")
5
6     for i in range(int(x)):
7         item=input("Enter an item of the list:")
8         ls.append(int(item))
9
10    return ls
11
12 ##### Sum of the list #####
13 def SumList(ls):
14     sum=0
15     for i in range(len(ls)):
16         sum=sum+ls[i]
17     print("The sum is:",sum)
18
19 ##### Main #####
20 ls=CreateList()
21 print(ls)
22 SumList(ls)

```

## نشاط 1

أكتب برنامجاً بلغة Python يمكّنك من برمجة نموذج لآلة حاسبة متّبعاً الخطوات الآتية:

- 1- يتمّ إدخال العددين اللذين سيتمّ إجراء العملية الحسابية عليهما من خلال الجزء الرئيسي من البرنامج.
- 2- يتمّ إدخال المعامل الحسابي المطلوب: (+ أو - أو / أو \*)
- 3- في حال إدخال علامة الجمع: يتمّ استدعاء دالة الجمع وتنفيذ العملية وإرجاع النتيجة إلى البرنامج الرئيسي.
- 4- في حال إدخال علامة الطرح: يتمّ استدعاء دالة الطرح وتنفيذ العملية وإرجاع النتيجة إلى البرنامج الرئيسي.
- 5- وكذلك الحال لباقي العوامل الحسابية.
- 6- يتمّ عرض نتيجة العملية الحسابية من البرنامج الرئيسي من خلال القيمة المُرجّعة من الدالة المعنية.

## نشاط 2

- 1- عيّّن ضمن سلسلة list مجموعة هامة من الكلمات.
- 2- اختر كلمة عشوائية من القائمة.
- 3- ادخل حرفاً من الحروف.
- 4- أظهر عدد مرات تواجد هذا الحرف في الكلمة ومواقعها.
- 5- حلّ المسألة مستعيناً بالدوال والخوارزميات اللازمة.

مثال: الكلمة: manama

الحرف: a

النتيجة: عدد مرات تواجد الحرف a ضمن الكلمة manama = 3

المواقع index: (2, 4, 6)