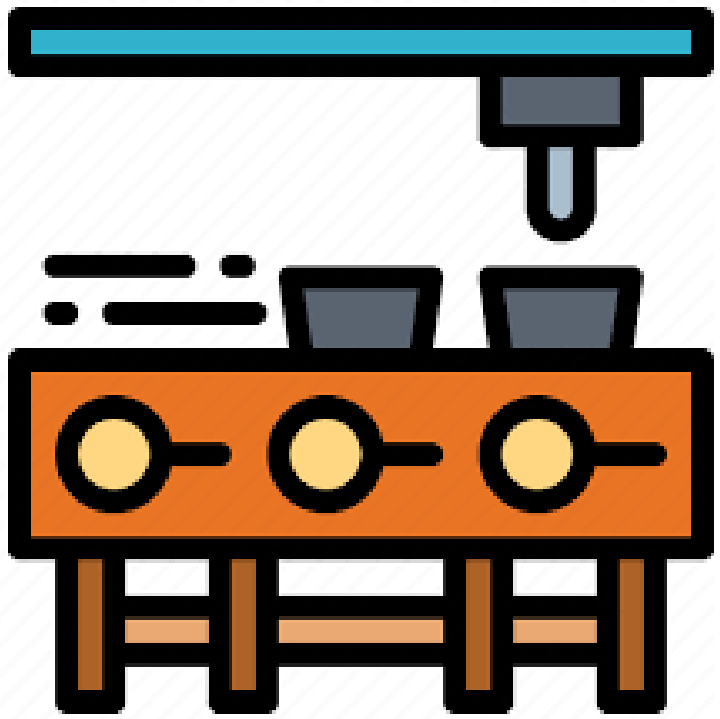
Functions

OBJECTIVE

- What is Function
- How to create Function
- Advantages of Function

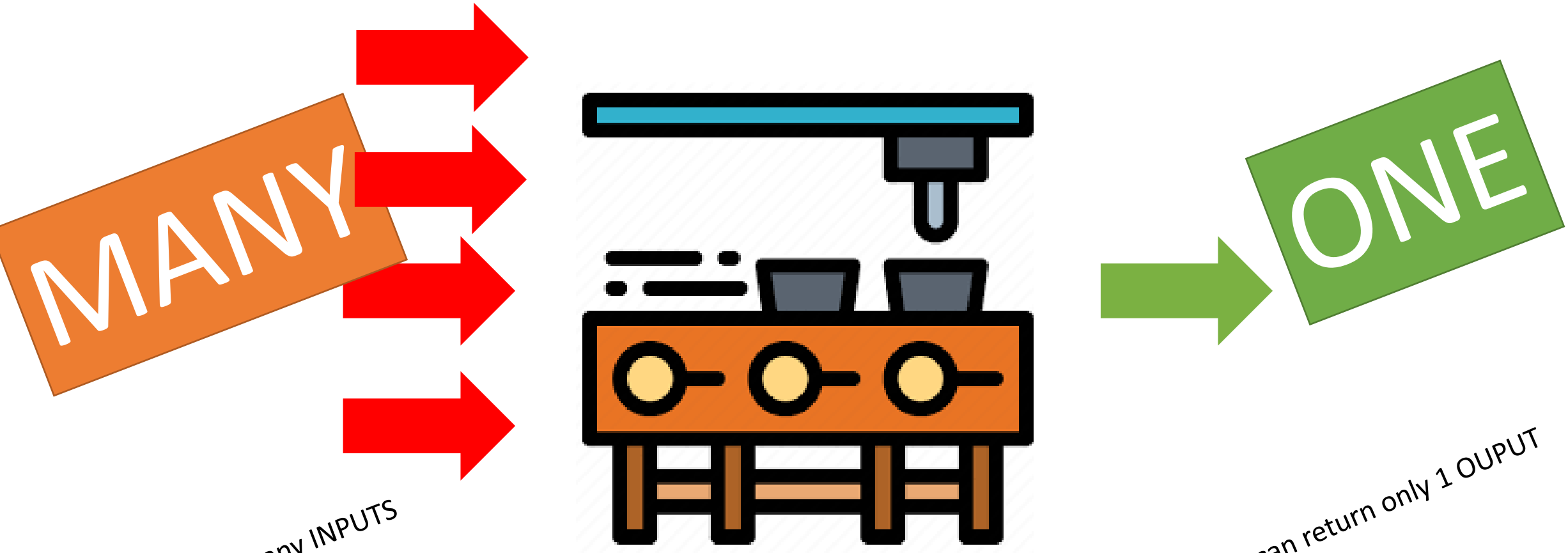
A function is like a machine…

The machine
Takes inputs

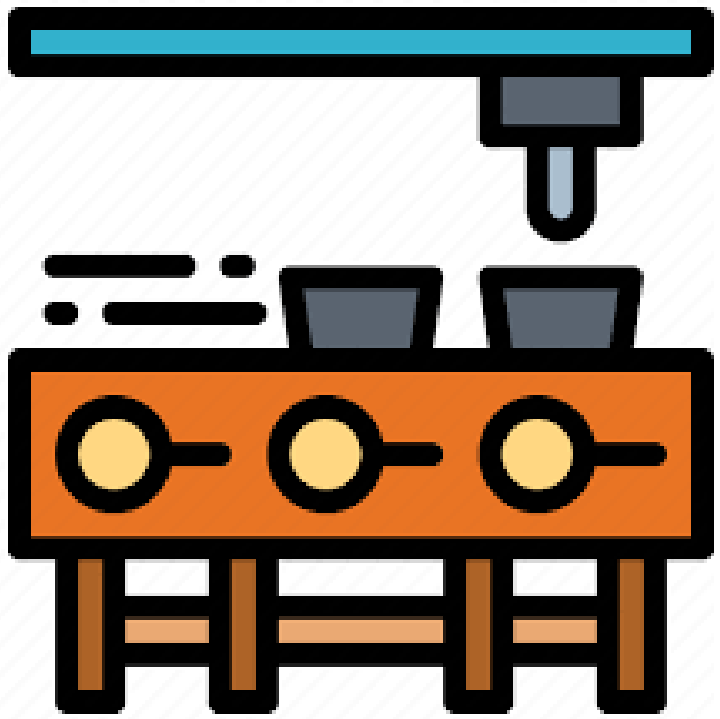


The machine
returns an output

…A machine which returns only 1 product



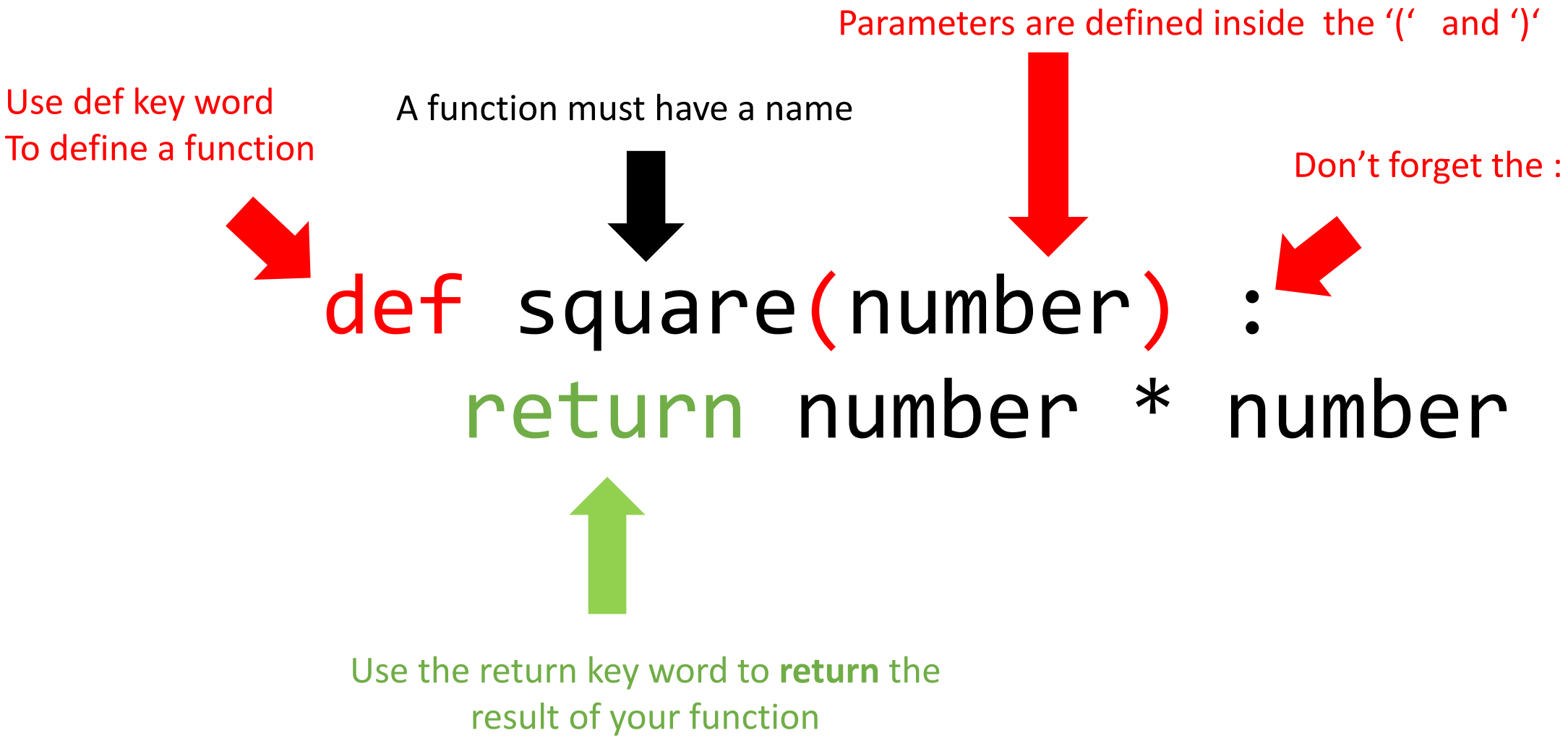
…Or a machine that returns nothing



You can enter many INPUTS

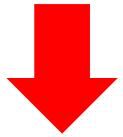
Here we return nothing

How to DEFINE a function ?



Can we have many parameters ?

If many parameters, they are separated using a COMA ‘,’



```
def addNumbers(number1 , number2) :
    return number1 + number2
```

Can we have NO parameters ?

You should keep the ‘(‘ and ‘)‘ but with nothing inside

```
def stupidFunction() :
    return 44* 44
```

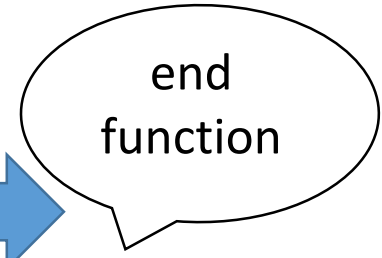
Can we have NO return ?


```
def nicePrint(myText) :
    print("---" + myText + "---")
```

we do something during the function but we don’t return anything

Can we have an instruction after return ?


```
def addNumbers(number1 , number2) :
    return number1 + number2
    number1 = 4
```



We will never go here

So…

Function

- NAME — Mandatory !
- PARAMETERS — Optional, zero to many
- RETURN — Optional, zero to one

A function is like a small program

When you enter in it, it's like when you start a program

```
def getNumberOfA(text) :
    number = 0
    for i in range len(text):
        if text[i] == "A":
            number = number  + 1

    return number
```

Start function

end function

How to CALL a function ?

1- Define the parameters for your function call

```
myResult = square(5)
```

2- Get the result of the function call

Let’s see step by step !

Here I will call a function many times

Here I define a function

Let’s see step by step !

Here I will call a function many times

Here I define a function

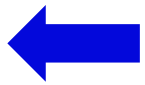
```
def addNumbers(number1 , number2) :
    return number1 + number2
```

Let’s see step by step !

Here I will call a function many times

```
a = 15
b = 23

c = addNumbers(a, b)

d = addNumbers(10, 5)
```

First call to my function



Second call to my function

Here I define a function

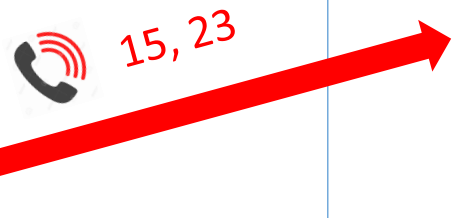
```
def addNumbers(number1 , number2) :
        return number1 + number2
```

Let’s see step by step !

Here I will call a function many times

```
a = 15
b = 23

c = addNumbers(a, b)

d = addNumbers(10, 5)
```

15 23



Here I define a function

```
def addNumbers(number1 , number2) :
    return number1 + number2
```

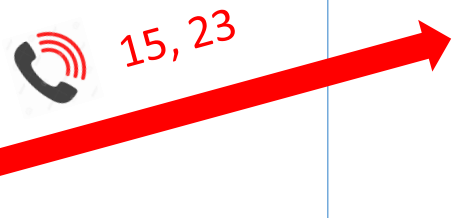
1- Call function addNumbers with parameters 15 and 23

Let’s see step by step !

Here I will call a function many times

```
a = 15
b = 23

c = addNumbers(a, b)

d = addNumbers(10, 5)
```

15 23



Here I define a function

15 23

```
def addNumbers(number1 , number2) :
    return number1 + number2
```

2- Execute function and compute the return value

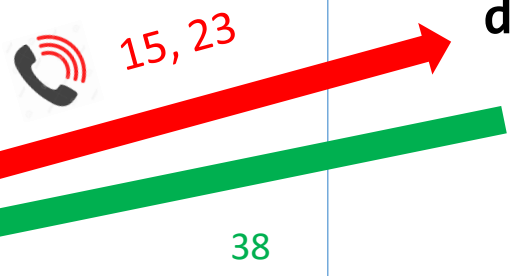
Let’s see step by step !

Here I will call a function many times

```
a = 15
b = 23

c = addNumbers(a, b)

d = addNumbers(10, 5)
```

38

15 23



Here I define a function

15 23

```
def addNumbers(number1 , number2) :
    return number1 + number2
```

3- Exit function with the return value 38

Let’s see step by step !

Here I will call a function many times

```
a = 15
b = 23

c = addNumbers(a, b)

d = addNumbers(10, 5)
```

Here I define a function

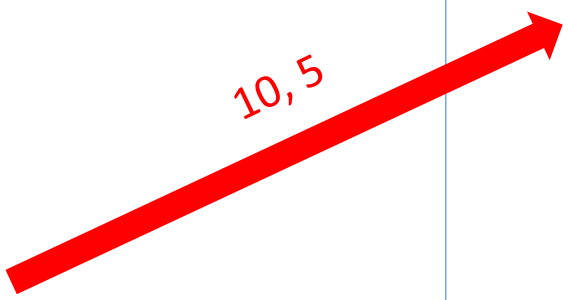
```
def addNumbers(number1 , number2) :
    return number1 + number2
```

3- Call function addNumbers with parameters 10 and 5

Let’s see step by step !

Here I will call a function many times

```
a = 15
b = 23

c = addNumbers(a, b)

d = addNumbers(10, 5)
```



Here I define a function

10 5

```
def addNumbers(number1 , number2) :
    return number1 + number2
```

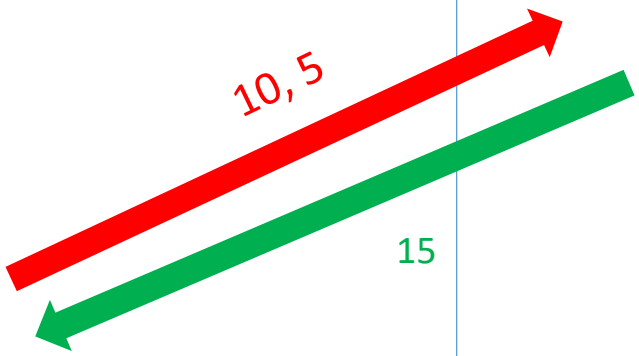
2- Execute function and compute the return value

Let’s see step by step !

Here I will call a function many times

```
a = 15
b = 23

c = addNumbers(a, b)

d = addNumbers(10, 5)
```

15



Here I define a function

10 5

```
def addNumbers(number1 , number2) :
    return number1 + number2
```

3- Exit function with the return value 15

What this code will print?

```
def computeAbsolute(number) :
    if number<0:
        return -1 * number
    else
        return number

print(computeAbsolute(-4))
```

A	B	C	D
-4	4	computeAbsolute(-4)	error

Can you complete the parameters/return types ?

`len("45AA")` → `4`

Len function:

Name	len
Parameters	?
Return	?

Can you complete the parameters/return types ?

`print("45AA")` → `<void>`

Print function:

Name	print
Parameters	?
Return	?

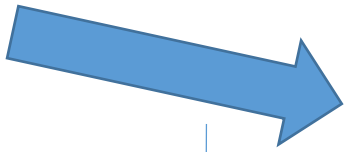
Let’s look at this code

```
print("Good morning Seiha!")
print("Good night sweet dream!")
print("Good Bye!")
print("Good morning Seiha!")
print("Good night sweet dream!")
print("Good Bye!")
print("Good morning Hugo!")
print("Good night sweet dream!")
print("Good Bye!")
print("Good morning Sievny!")
print("Good night sweet dream!")
print("Good Bye!")
```

```
print("Good morning Ronan!")
print("Good night sweet dream!")
print("Good Bye!")

print("Good morning Seiha!")
print("Good night sweet dream!")
print("Good Bye!")

print("Good morning Hugo!")
print("Good night sweet dream!")
print("Good Bye!")

print("Good morning Sievny!")
print("Good night sweet dream!")
print("Good Bye!")
```

In RED the only difference of code

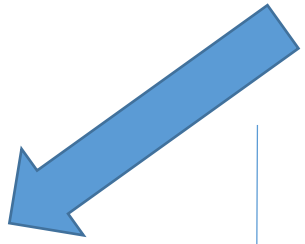
1 - We create a function with 1 parameter (the name)



```
def printWelcome(name):
    print("Good morning " + name + "!")
    print("Good night sweet dream!")
    print("Good Bye!")
```

2 - We call this function with 4 times

```
printWelcome("ronan")
printWelcome(“seiha")
printWelcome(“hugo")
printWelcome(“sievny")
```



```
def printWelcome(name):
    print("Good morning " + name + "!")
    print("Good night sweet dream!")
    print("Good Bye!")
```

Write a function to compute the sum of 2 numbers

Name	computeSum
Parameters	number1, number2
Return	the sum of the 2 numbers

```
result = computeSum(4,5)  # result should be 9

result = computeSum(2,4)  # result should be 6
```

Write a function to compute the minimum of 2 numbers

Name	computeMin
Parameters	number1, number2
Return	the min of the 2 numbers

```
result = computeMin(4,5)  # result should be 4

result = computeMin(3,7)  # result should be 3
```

Write a function to compute the minimum of 2 numbers

Name	computeMin
Parameters	number1, number2
Return	the min of the 2 numbers

```
result = computeMin(4,5)  # result should be 4

result = computeMin(3,7)  # result should be 3
```

What is the result of this code ?

```
def myfunc():
    x = 300
    print(x)

myfunc()
```

A

300

B

Nothing displayed

C

Error x not defined

What is the result of this code ?

```
def myfunc():
    x = 300


myfunc()
print(x)
```

A	B	C
300	Nothing displayed	Error x not defined

A variable created inside a function belongs to the ***local scope*** of that function, and can only be used inside that function

```
def myfunc():
    x = 300
```

X visible only inside myFunc