



inpi

LABORATOIRE
D'INFORMATIQUE
INTELLIGENTE
BISKRA



COMPUTER SCIENCE 2

DR. ADEL ABDELLI

adel.abdelli@univ-biskra.dz



CHAPTER 2

FUNCTIONS AND PROCEDURES

DR. ADEL ABDELLI

1. Introduction

Let's say we have the following array, and we want to calculate the sum of all the elements,

so we do as follow:

```
t = [4,8,7,12,5]
```

```
s = 0
```

```
for i in range(len(t)):
```

```
    s = s + t[i]
```

```
print("the sum is", s)
```

1. Introduction

Now let say I have to other arrays t2 and t3 and I want to do the sum again, so I must do the following:

```
t2 = [4,18,17,12,5]
```

```
s = 0
```

```
for i in range(len(t)):
```

```
    s = s + t[i]
```

```
print("the sum is", s)
```

```
t3 = [14,1,11,10,15]
```

```
s = 0
```

```
for i in range(len(t)):
```

```
    s = s + t[i]
```

```
print("the sum is", s)
```

1. Introduction

So, we got redundances, and we must write many lines of codes, To avoid that there something called functions, so we can solve the problem as follows:

```
def sumArray(t):  
    s = 0  
    for i in range(len(t)):  
        s = s + t[i]  
    print("the sum is", s)
```

```
a = [4,8,7,12,5]  
sumArray(a)
```

Function definition begins with "def." Function name and its arguments.

```
def get_final_answer(filename):  
    """Documentation String"""  
    line1  
    line2
```

Colon.

The indentation matters...
First line with less
indentation is considered to be
outside of the function definition.

Calling a Function


- The syntax for a function call is:

Function code or core or part

```
def myfun(x, y):  
    return x * y
```

Main code

```
myfun(3, 4) #The output will be: 12
```

The diagram illustrates the relationship between function definition and function call. On the left, two vertical double-headed arrows indicate the flow of information. The top arrow connects the label 'Function code or core or part' to the function definition code block. The bottom arrow connects the label 'Main code' to the function call code block. The function definition code block contains two lines: 'def myfun(x, y):' and 'return x * y'. The function call code block contains the line 'myfun(3, 4) #The output will be: 12'.

Function Vs Procedure

In programming there is a difference between a procedure and a function, where a function contain the keyword **return** and the procedure doesn't contain that keyword,

Example:

```
def Mul(x, y):  
    a = x * y  
    print(a)
```

VS

```
def Mul(x, y):  
    a = x * y  
    return a
```

Mul(5, 4)

#output will be: 20

result = Mul(5, 4)

print(result)

#output will be: 20

So, a Python function is like a math function where the function itself get a value:

$$y = f(x)$$

and $f(x) = ax + b$ for example

Default Values for Arguments

- You can provide default values for a function's arguments
- These arguments are optional when the function is called

```
def myfun(b, c=3, d="hello") :  
    return b + c  
myfun(5, 3, "hello")  
myfun(5, 3)  
myfun(5)
```

All of the above function calls return 8

Local and Global variable in a function

The variables of the function are local, and we can not use them outside the function,

Example:

```
def foo(x, y):
```

```
    z = 5
```

```
    return x+y+z
```

```
print(x)    #NameError: name 'x' is not defined
```

```
print(y)    #NameError: name 'x' is not defined
```

```
print(z)    #NameError: name 'x' is not defined
```

```
a = 5
```

```
print(a)    #output: 5
```



**THANK YOU FOR YOUR
ATTENTION**