

Programación en Python I



Ing. Wilmer Garzón Alfonso

wilmer.garzon@escuelaing.edu.co

www.wilmergarzon.com.co



Agenda

- Modularidad
- Funciones
- Funciones en Python
- Parámetros
- Funciones recursivas
- Referencias
- Módulos en Python



Programación Modular

- Consiste en dividir un programa en módulos (subprogramas) con el fin de hacerlo más legible y manejable.
- Permite solucionar problemas de programación más grandes y complejos.
- Un problema complejo se divide en varios subproblemas más simples.



Programación Modular

- Esta técnica se conoce como: **DIVIDE y VENCERAS**
- Un módulo es un programa, llamado función. Es código en Python que permite llevar a cabo una tarea específica.
- Las funciones tienen la capacidad de comunicarse entre si, con el objetivo de compartir los resultados.



Ventajas

- Por la independencia de las funciones, un programa se puede diseñar con mayor facilidad.
- Se puede modificar una función sin afectar a las demás.
- Las funciones solamente se escriben una vez, aunque se necesiten en distintas ocasiones a lo largo del programa. Esto permite la reutilización de código en nuestros programas.



Funciones

- Matemáticamente una función es una operación que toma uno o mas valores llamados **argumentos** y produce un valor denominado **resultado**

$$f(x) = \frac{x}{1 + x^2}$$

Diagram illustrating the components of the function notation $f(x)$:

- nombre** (name) points to f .
- parámetro** (parameter) points to x .

- Una función puede tener varios argumentos



Funciones

- Al día de hoy, ya hemos utilizado funciones definidas en Python, como por ejemplo:
 - len, range, chr, upper, lower, ord, int, str,.....
- Para el uso de estas funciones se deben especificar de forma CORRECTA los parámetros requeridos:
 - len("casa")
 - ~~➤ len(8)~~



Funciones en Python

- Las funciones se definen de la siguiente manera:

Nombre de la función Parámetros Obligatorio

```
def my_function(arg1,...,argn):
```

```
→ print("Within the function...")  
→ .....  
→ .....  
→ .....
```

Todas las instrucciones que forman parte de la función

```
print("Outside the function...")
```

Instrucciones fuera de la función



Funciones en Python

Función llamada saludo

```
def saludo():  
    print("Buen día")
```

```
saludo()
```

```
>>>  
Buen día
```

```
def saludo():  
    print("Buen día")  
  
for x in range(1,6):  
    saludo()
```

```
>>>  
Buen día  
Buen día  
Buen día  
Buen día  
Buen día
```

```
def saludo(nombre):  
    print("Buen día", nombre)  
  
for x in range(1,6):  
    print("Saludando")  
    n="Juan" + str(x)  
    saludo(n)
```

```
>>>  
Saludando  
Buen día Juan1  
Saludando  
Buen día Juan2  
Saludando  
Buen día Juan3  
Saludando  
Buen día Juan4  
Saludando  
Buen día Juan5
```



Funciones en Python

- Las funciones se definen de la siguiente manera:

Nombre de la función Parámetros Obligatorio

```
def my_function(arg1,...,argn):
```

```
→ print("Within the function...")  
→ .....  
→ .....  
→ .....
```

Todas las instrucciones que forman parte de la función

```
print("Outside the function...")
```

Instrucciones fuera de la función



Funciones en Python

Datos

```
from sys import stdin
```

```
def operar(n1,n2):  
    s = n1+n2  
    return n1+n2
```

```
def main():  
    cases = int(stdin.readline())  
    while cases!=0:  
        a,b = [ x for x in stdin.readline().split() ]  
        print(operar(int(a),int(b)))  
        cases -= 1
```

```
main()
```

```
C:\Python34>python prueba.py < file.in  
-18  
0  
15  
-5  
12
```



Parámetros

- Es un valor que la función espera recibir cuando sea invocada.
- Una función puede esperar cero, uno o más parámetros, los cuales están separados por coma.
- Los parámetros que una función espera, serán utilizados por ésta, dentro de su algoritmo, a modo de **variables de ámbito local**.



Parámetros

```
def sumar(a,b):  
    r = a + b  
    return r  
  
def main():  
    n1 = int(input("No1:"))  
    n2 = int(input("No2:"))  
    rta = sumar(n1,n2)  
    print("La suma es: ", rta)  
  
main()
```

```
>>>  
No1:2  
No2:3  
La suma es: 5
```



Parámetros

```
def sumar(a,b):  
    r = a + b  
    return r  
  
def main():  
    n1 = int(input("No1:"))  
    n2 = int(input("No2:"))  
    rta = sumar(n1,n2)  
    print("La suma es: ", rta)  
    print("El numero1 es: ", a)  
  
main()
```

```
>>>  
No1:2  
No2:3  
La suma es: 5  
Traceback (most recent call last):  
  File "C:/Users/laboratorio/Downloads/pp.py",  
    main()  
  File "C:/Users/laboratorio/Downloads/pp.py",  
    print("El numero1 es: ", a)  
NameError: name 'a' is not defined
```

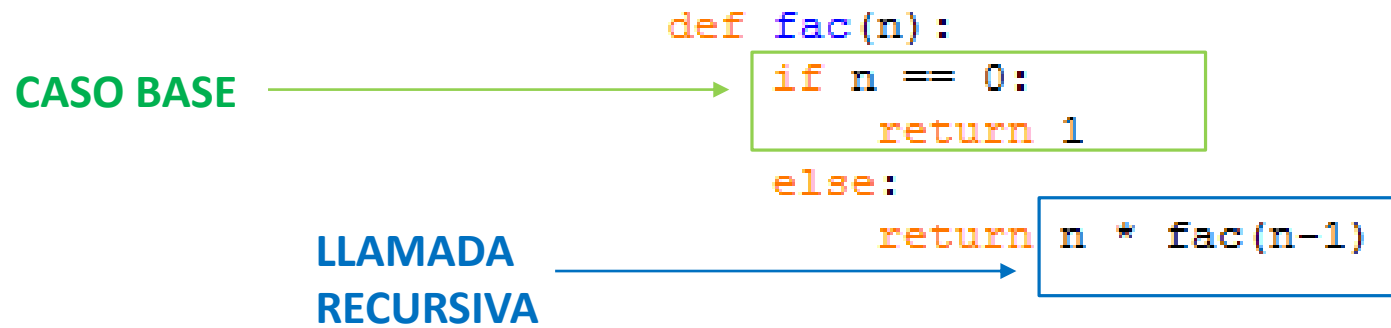


Funciones recursivas

- Son funciones que se invocan a sí mismas en su definición.
- Debe existir un caso base o condición de escape que permite terminar la recursión.
- El uso es de mucho cuidado, puesto que si no se detiene la computadora termina sobrecargada.



Factorial (n!)



```
>>> fac(1)  
1  
>>> fac(10)  
3628800  
>>> fac(0)  
1  
>>> fac(5)  
120
```




Jugar

```
def jugar(intento=1):  
    respuesta = input("¿De qué color es un limón? ")  
    if respuesta != "verde":  
        if intento < 3:  
            print("\nFallaste! Inténtalo de nuevo")  
            intento += 1  
            jugar(intento)  
        else:  
            print("\nPerdiste!")  
    else:  
        print("\nGanaste!")
```

```
>>> jugar()  
¿De qué color es un limón? rojo  
  
Fallaste! Inténtalo de nuevo  
¿De qué color es un limón? azul  
  
Fallaste! Inténtalo de nuevo  
¿De qué color es un limón? verde  
  
Ganaste!
```



Módulos en Python

- Se componen de funciones y definiciones realizadas por el usuario que cumplen una tarea específica.
- Los módulos se crean en un archivo en Python con extensión `.py`
- Los módulos pueden ser importados, permitiendo reutilizar código. Se utiliza la instrucción `import`



Módulos

Nombre del Archivo

```
Python 3.4.1: module1.py C:/Python34/module1.py
File Edit Format Run Options Windows Help

def sumar(a,b):
    return a+b

def restar(a,b):
    return a-b

def multiplicar(a,b):
    return a*b

def dividir(a,b):
    if b==0:
        print("Imposible dividir por CERO")
        return -9999
    else:
        return a/b
```

Importar el módulo

```
from module1 import *

def menu():
    print("Menú Principal")
    print("1.Sumar\n2.Restar\n3.Multiplicar\n4.Dividir\n5.Salir")
    opc = int(input("Ingrese la opción deseada:"))
    return opc

def main():
    n1 = int(input("Primer Número:"))
    n2 = int(input("Segundo Número:"))
    rta = 0
    opc = menu()
    if opc==1:
        rta = sumar(n1,n2)
    elif opc==2:
        rta = restar(n1,n2)
    elif opc==3:
        rta = multiplicar(n1,n2)
    elif opc==4:
        rta = dividir(n1,n2)
    else:
        print("Fue un placer servirle...")

    print("La respuesta es:",rta)

main()
```

Hacer el llamado a las funciones del módulo importado



Referencias

- Introduction to Computation and Programming Using Python, revised and expanded edition, John V Guttag, MIT Press.
- Python Programming: An Introduction to Computer Science, John Zelle.
- MITx's Introduction to Computer Science and Programming Using Python.
- <http://paginaspersonales.deusto.es/dipina/teaching.html>
- <https://courses.edx.org/courses/course-v1:UTAx+CSE1309x+2016T1/>
- <https://www.python.org/>
- <http://www.camilorocha.info/>