



UNIVERSIDAD
NACIONAL DE
HURLINGHAM

Introducción a Python

Parte 2

Profesor

Sergio Gonzalez

Funciones

- Muchas veces, partes del programa se repiten
- Es una buena práctica, separar estas partes para crear "Funciones"
 - Bloques de código que realizan tareas específicas
 - El programa principal los "llama"
 - Salto en la ejecución

Funciones

- El programa principal las “llama”
- Comunicación:
 - Entrada: Se envían datos desde el programa principal a la función
 - Salida: Se envían datos desde la función al programa principal

Definición

- Nombre: Debe ser indicativo
- Parámetros: Variables que se usan como entrada
- Salida

```
def nombreFuncion(parametro1:tipo , parametro2:tipo, ...) -> tipoSalida:  
    <bloque código>  
    return salida
```

Invocación (Llamado)

- Utilizando su nombre y los argumentos entre paréntesis, coincidiendo en cantidad, orden y en tipo con los parámetros de la definición

```
salida = nombreFuncion(argumento1, argumento2, ...)
```

Ámbito de variables

- Ámbito (*scope*): Parte del programa donde la existe una variable
- Variables locales
 - Declaradas y utilizadas dentro de una función
 - Se pueden compartir nombres
 - No accesibles desde afuera

Paso de parámetros

- Tipos:
 - Paso por valor
 - Paso por referencia

Paso por valor

- La función recibe el valor de la variable
- Modificaciones a dicha variable, solo tienen efecto dentro de la función

```
def funcion1(param1, param2):  
    <acciones>  
    return salida  
  
varSalida = funcion1(5, "hola")  
  
var1 = 10  
var2 = "casa"  
varSalida2 = funcion1(var1, var2)
```


Paso por referencia

- Parámetros de Entrada / Salida
- Modificaciones a variables dentro de la función se mantienen fuera de ella

Funciones en Python

- Tipos primitivos: Solo se permite paso por valor
- Tipos **NO** primitivos: Solo se permite paso por “referencia”
- Se pueden poner valores por defecto y así poder cambiar el orden de los parámetros

Funciones en Python

- Función suma de dos números enteros

```
def suma(x:int , y:int) -> int:  
    salida = x + y  
    return salida
```

```
def suma(x:int = 0 , y:int = 0) -> int:  
    salida = 0  
    salida = x + y  
    return salida
```

```
n = suma (5 , 4)  
n = suma (x = 5 , y = 4)  
n = suma (y = 4 , x = 5)  
x, y = 5 , 4  
n = suma (y = y , x = x)
```