

Introducción a Python

Parte 2

Profesor

Sergio Gonzalez

Funciones - Procedimientos

- Muchas veces, partes del programa se repiten
- Es una buena practica, separar estas partes para crear "Funciones" o "Procedimientos"
- Realizan tareas especificas
- El programa principal los "llama"
- Parámetros: Lista de valores de entrada
- Salidas

Definición

- Nombre identificadorio
- Parámetros: Lista de valores de entrada
- Salida

```
función nombre-función(par1,par2,par3...)  
    <acciones>  
fin función
```

par1,par2, ,

lista de parámetros formales o argumentos: son nombres de identificadores que sólo se utilizan dentro del cuerpo de la función.

nombre-función

nombre asociado con la función, que será un nombre de identificador válido sobre el que se almacenará el resultado.

<acciones>

instrucciones que constituyen la función, y que deben contener al menos una acción que asigne un valor como resultado de la función.

Invocación (Llamado)

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

nombre-función (lista de parámetros actuales)

nombre función

función que es invocada.

lista de parámetros actuales

constantes, variables, expresiones, valores de funciones, nombres de subprogramas, que se corresponden con los parámetros formales, que hemos visto durante en la declaración de funciones.



Escribir la función: $y = x^n$ (potencia n de x , tanto si es positiva, como negativa) y utilizarla para calcular $1/(2.5)^3$

función potencia (x, n)
 ↑ ↑
 inicio _____ *parámetros formales (real y entero)*
 ↓ _____ *función interna (valor absoluto)*
 $y \leftarrow 1$
desde $i = 1$ **hasta** $abs(n)$ **hacer**
 $y \leftarrow y * x$
fin_desde
si $n < 0$ **entonces** $y \leftarrow 1/y$
 potencia $\leftarrow y$
fin

Su utilización sería:

$z \leftarrow potencia(2.5, -3)$
 ↑ ↑
 _____ *parámetros actuales*

Ámbito de variables

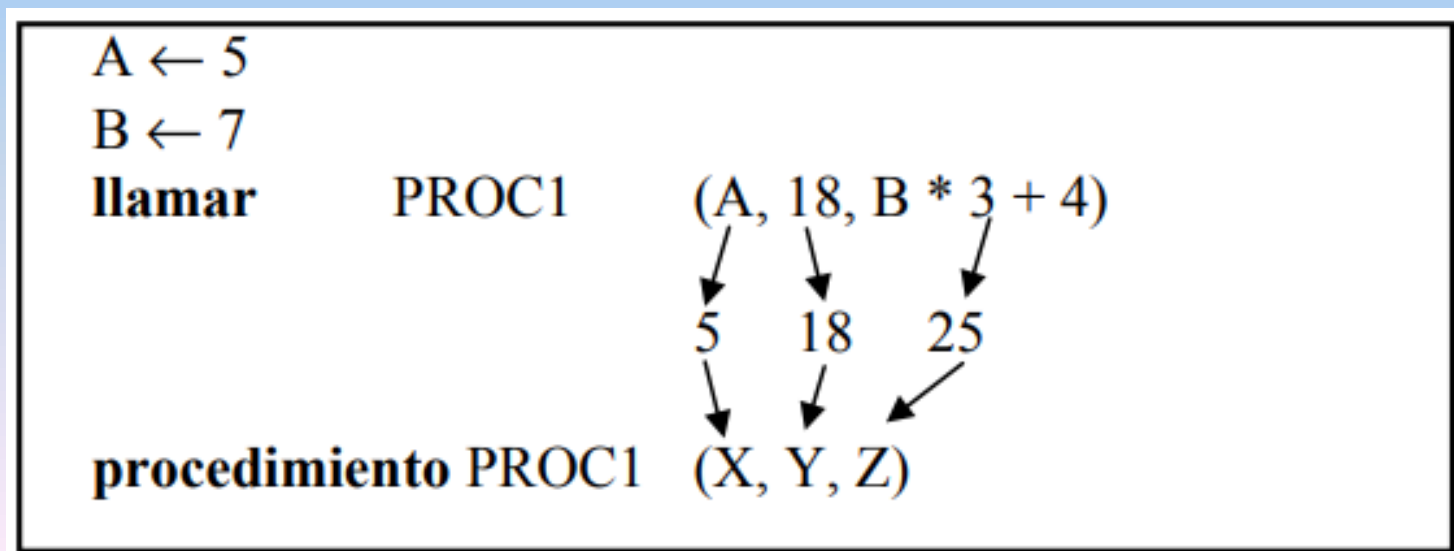
- La ejecución de una función no debe afectar al resto del programa
- Desarrollos independientes
- Variable local
 - Declarada y utilizada solo dentro de la función
 - Compartir nombres
 - Valores no accesibles desde afuera
- Ámbito (*scope*): Parte del programa donde la variable se reconoce como tal

Paso de parámetros

- Entrada (Argumentos): Desde el programa principal a la función
- Salida: Resultados de la función, desde la función al programa principal
- Entrada / Salida: Mismo parámetro para mandar argumentos y devolver resultados.
- Tipos:
 - Paso por valor
 - Paso por referencia

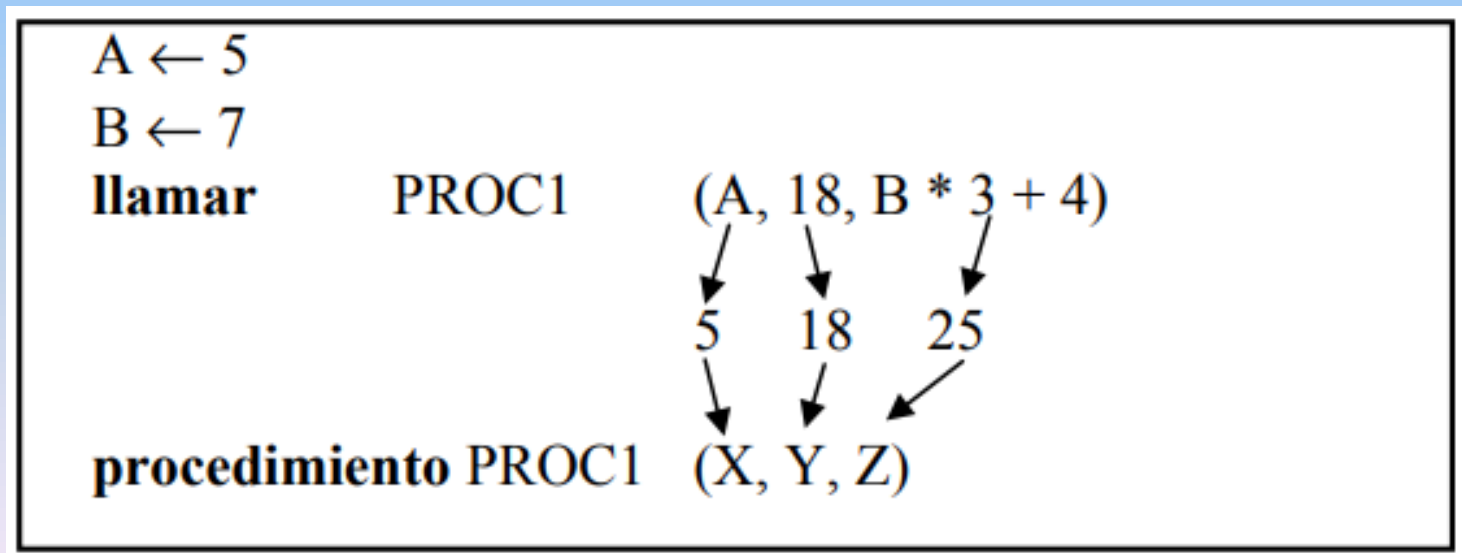
Paso por valor

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



Paso por valor

- Todos los parámetros son solo de entrada
- Salida por retorno de la función



Paso por referencia

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

$A \leftarrow 5$

$B \leftarrow 7$

$C \leftarrow B * 3 + 4$

llamar_a PROC1(A, B, C)

5

7

25

direcciones o
posiciones
de memoria

procedim PROC1(REF X, REF Y, REF Z)



Paso de parametros

algoritmo ABC

inicio

$A \leftarrow 3$

$B \leftarrow 5$

$C \leftarrow 17$

llamar_a SUB(A, B, C)

escribir A,B,C

fin

procedimiento SUB(x,y,z)

inicio

$x \leftarrow x+1$

$z \leftarrow x+y$

fin

(a)

algoritmo ABC

inicio

$A \leftarrow 3$

$B \leftarrow 5$

$C \leftarrow 17$

llamar_a SUB(A, B, C)

escribir A,B,C

fin

procedimiento SUB(REF x,REF y,REF z)

inicio

$x \leftarrow x+1$

$z \leftarrow x+y$

fin

(b)

Funciones en Python

- Tipos primitivos: Solo se permite paso por valor
- No se define tipo de salida
- No se definen tipos de parámetros
- Se pueden poner valores por defecto y así poder cambiar el orden de los parámetros

Funciones en Python

```
def nombreFuncion(parametros):  
    salida = 0  
    <accionesFuncion>  
    return salida
```

Funciones en Python

- Función suma de dos números enteros

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

```
def suma(x = 0 , y):  
    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)
```