



ProfesorenVideo.com

Funciones en Python

Qué es una función ?: Es un bloque de instrucciones válidas que se invocan (o llaman) por su nombre, y se declaran así:

```
def nombre_de_funcion ( [ parámetros ] ) :  
    # instrucción  
    # instrucción  
    # instrucción  
    # instrucción  
    [ return expresión ]
```

[] = indican que es opcional



ProfesorenVideo.com

Funciones en Python

palabra clave

nombre de función

lista de parámetros separados por 'coma'

```
def nombre_de_funcion ( [parámetros] ) :
```

```
# instrucción
```

```
# instrucción
```

```
# instrucción
```

```
# instrucción
```

cuerpo de la función

```
[ return expresión ]
```

retorno de la función

[] = indican que es opcional



ProfesorenVideo.com

Funciones en Python

```
def reporte_de_ventas( ) :  
    os.system("cls")  
    print ( " \n\n          REPORTE DE VENTAS")  
    print ( " Código Vendedor   Nombre   Monto   Fecha   ")  
    print ( " =====   =====   =====   =====   ")  
    while ( condicion ) :  
        instruccion  
        instruccion  
        instruccion
```


Funciones en Python

Cuál es su razón de ser ? Las funciones sirven para:

- **Reutilizar el código:** Definir un bloque de código una vez y reutilizarlo cuantas veces sea necesario.
- **Facilitar la lectura y comprensión del código:** Al dividir un programa en partes más pequeñas cada bloque se dedica a una tarea más específica. Esto hace que el código sea más manejable y escalable.
- **Promover la Abstracción:** Las funciones permiten abstraer procesos complejos mediante una interfaz sencilla. Hacen el código más fácil de entender y usar.
- **Facilita la depuración y corrección de errores:** hacer mejoras o buscar errores siempre será más fácil hacerlo si tenemos bloques de código chicos e independientes.
- **Promueve la usabilidad y el desarrollo iterativo / incremental:** Permiten escribir código genérico reutilizable en otras partes del programa o en otros proyectos.



ProfesorenVideo.com

Funciones en Python

254 líneas de código

```
# import ...  
# import ...  
diccDatos = {.....}  
listaProductos = [.....]
```

Instrucción
Instrucción
Instrucción

Proceso 1 (50 líneas)

Instrucción
Instrucción
Instrucción

Proceso 2 (50 líneas)

Instrucción
Instrucción
Instrucción

Proceso 3 (50 líneas)

Instrucción
Instrucción
Instrucción

Proceso 1 (50 líneas)

...

Instrucción
Instrucción
Instrucción

Proceso 2 (50 líneas)

159 líneas de código

funcion_1

Instrucción
Instrucción (50)
Instrucción

funcion_2

Instrucción
Instrucción (50)
Instrucción

funcion_3

Instrucción
Instrucción (50)
Instrucción

```
# import ...  
# import ...  
diccDatos = {.....}  
listaProductos = [.....]
```

Llamadas a
funciones

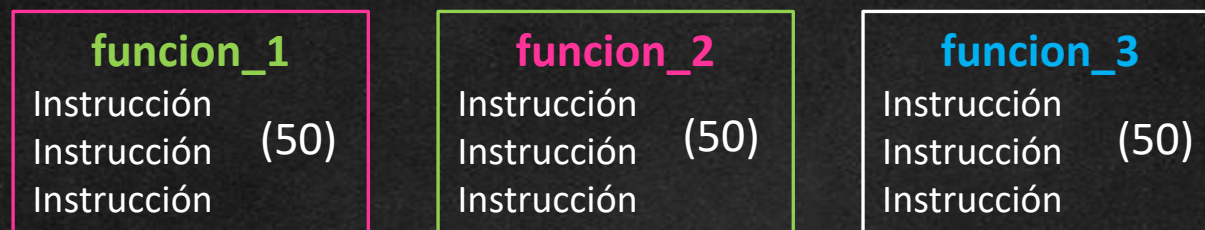
funcion_1
funcion_2
funcion_3
funcion_1
funcion_2

cuerpo
Principal (**main**)



ProfesorenVideo.com

Funciones en Python



import ...
import ...
diccDatos = {.....}
listaProductos = [.....]

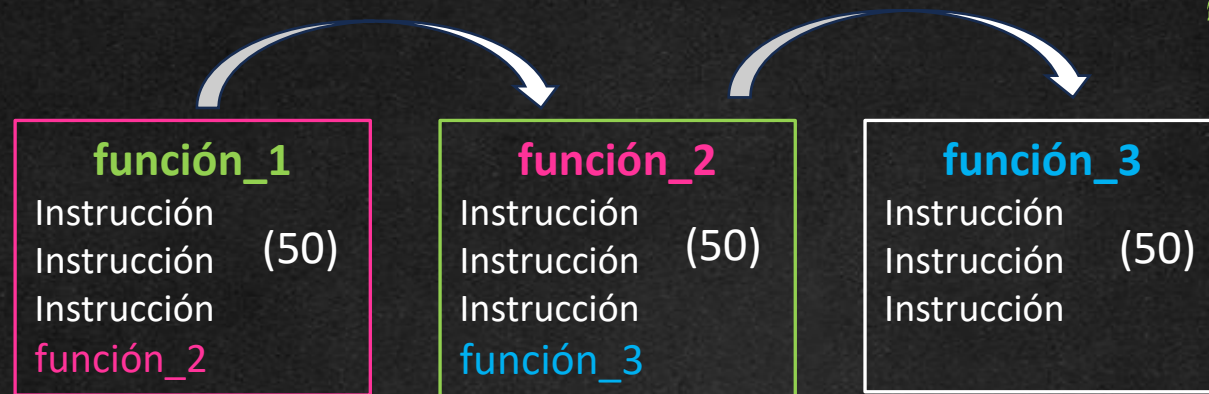
funcion_1
funcion_2
funcion_3
funcion_1
funcion_2

cuerpo
principal

Todo programa inicia por
aquí

**Por dónde arranca un programa
con funciones?**

Funciones en Python



Las funciones se pueden llamar
(invocar) entre sí?

```
# import ...  
# import ...  
diccDatos = {.....}  
listaProductos = [.....]
```

Cuán grande en código debe ser
una función?

```
funcion_1  
funcion_2  
funcion_3  
funcion_1  
funcion2
```

} cuerpo principal

Todo programa inicia por
aquí

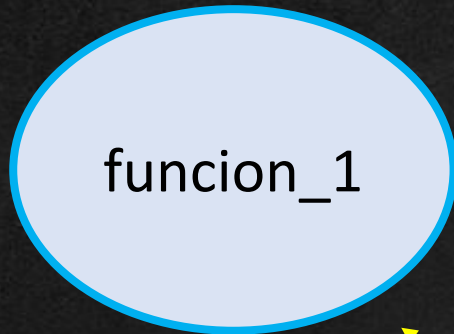


ProfesorenVideo.com

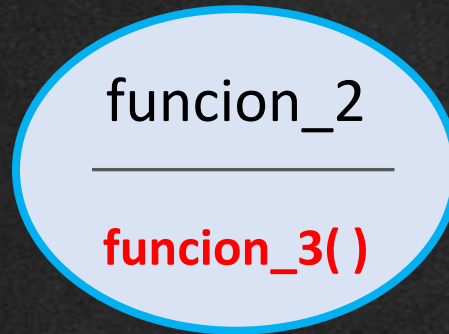
Funciones en Python

Cómo se declaran y dónde ?

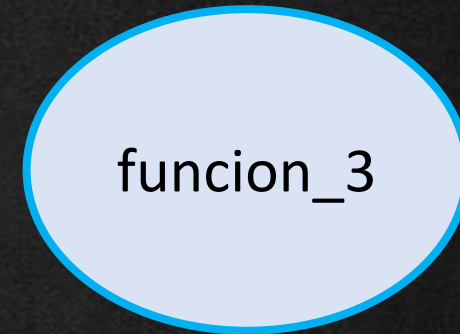
sub funcion_1() :



sub funcion_2() :



sub funcion_3() :



Programa principal

funcion_1()
funcion_2()



ProfesorenVideo.com

Funciones en Python

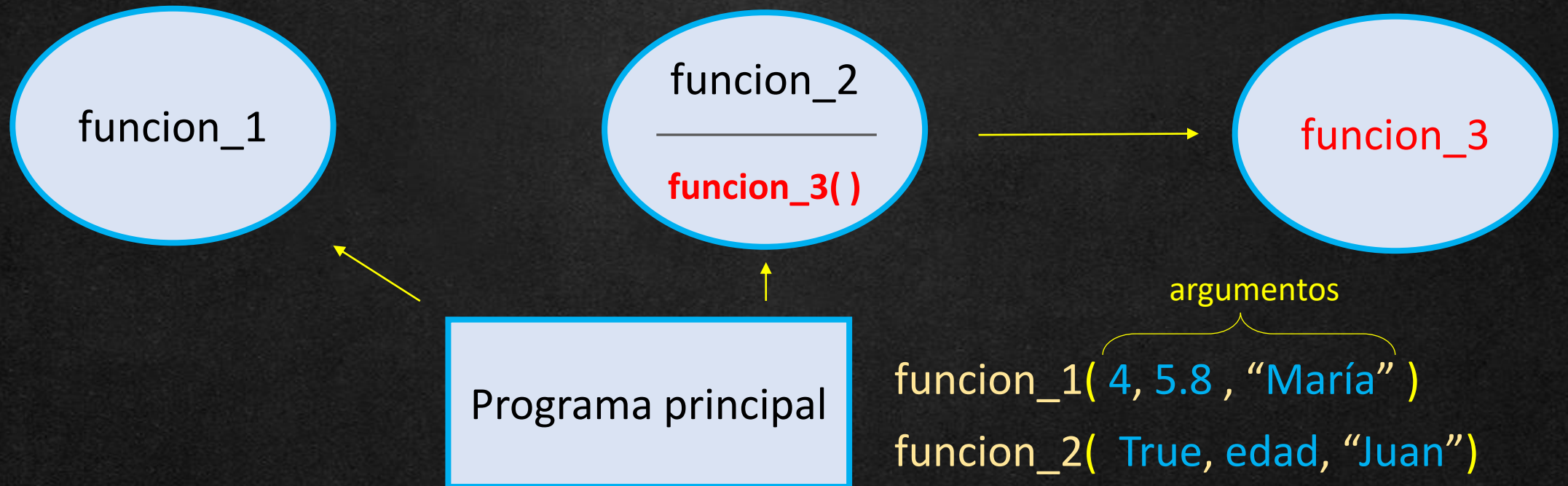
Qué son los argumentos?: Son los valores enviados hacia las funciones en las **llamadas**.

Qué son los parámetros?: Son las “**variables**” que reciben los valores en las funciones.

sub funcion_1(para1, para2, para3) :

sub funcion_2(para1, para2, para3) :

sub funcion_3() :



Funciones en Python

Y las funciones cómo devuelven (o retornan) valores *hacia el punto de retorno*?:

Eso se hace a través del **return** o también a través de los **parámetros**.

sub funcion_1(para1, para2, para3) :

sub funcion_2(para1, para2, para3) :

sub funcion_3() :

instruccion
instruccion
instruccion
return valor

Instrucción
funcion_3()
instruccion

instruccion
instruccion
instruccion
return valor

Programa principal

argumentos

funcion_1(4, 5.8 , "María")

funcion_2(True, edad, "Juan")



ProfesorenVideo.com

Funciones en Python

Qué son variables Globales y variables Locales?:

Variables Globales: Se declaran fuera de las funciones y su ámbito es **TODO** el programa.

Variables Locales: Se declaran dentro de una función y su ámbito es la **sólo** la función.

sub funcion_1() :

```
m = k + 1  
print( m )
```

sub funcion_2() :

```
m = k + 25  
print( m )
```

sub funcion_3() :

```
s = 60  
h = m - k  
print( s, h )
```

Error

```
k=50      # Global  
s = 22    # Global  
.....
```

Programa principal



ProfesorenVideo.com

Muchas gracias por su atención
y
vayamos a programar ...