



Funciones

Función

- Una función es un bloque de código
- Cada bloque de código contiene instrucciones relacionadas a ejecutar una tarea

Ventajas

- Nos permiten organizar el código en piezas pequeñas
- Permite organizar el código
- Evita repetir instrucciones y permite reutilizar código

Sintaxis:

```
def <nombre_funcion>(parámetros opcionales):  
    <instrucciones>  
  
<nombre_funcion>()
```

Tipos de funciones

- Built-in: las que trae el propio lenguaje
- User-defined: las que crea el usuario

Ejemplos de función

Una función que printea *"hello"* en consola:

```
def say_hello():  
    print("Hello!")  
  
say_hello()
```

Una función recibiendo un parámetro:

```
def welcome_user(user):  
    print(f"hello, {user}")  
  
welcome_user("Desire") # Printea "hello, Desire"
```

Una función que calcula el perímetro de un cuadrado:

```
def sqr_perimeter(side, units):  
    perim = side * 4  
    print(f"El parámetro es: {perim} {cm}")  
  
sqr_perimeter(2, "cm")
```

Parámetros

Hay dos formas de enviar parámetros a una función.

```
# En orden
def hi_user(name, surname):
    print(f"Hello, {name} {surname}")

hi_user("Desire", "Brown")
```

```
# Utilizando el nombre del parámetro
def hi_user(name, surname):
    print(f"Hello, {name} {surname}")

hi_user(name = "Desire", surname = "Brown")
```

Argumento y parámetro es lo mismo, pero argumento es el valor en sí enviado en el parámetro.

"Name" sería el parámetro, "Desire" sería el argumento.

Devolver valores

```
def sqr_perimeter(side, units):
    perim = side * 4
    return perim

def sqr_area(side):
    area = side * side
    return area

print(sqr_perimeter(5, "cm"))
print(sqr_area(2))

# Si necesitamos guardar el valor que se devuelve
```

```
res_area = sqr_area(5)
print(res_area)
```

Los valores pueden guardarse o simplemente mostrarse.

```
# Retornar más de un valor en una función -> Devuelve una tupla
def sqr_area_perim(side):
    area = side * side
    perim = side * 4
    return area, perim

area, perim = sqr_area_perim(5);

# Si la función devuelve más de un valor pero la estamos asignando
# a una sola variable, guardará una tupla en dicha variable
just_one = sqr_area_perim(3);

print(just_one) # Devolverá (9, 12)
```

Documentar funciones

Hay dos opciones:

- Descripción corta en una sola línea, entre comillas triples
- Descripción larga

Descripción corta

```
def sqr_area_perim(side):
    """ Calcula el área y perímetro de un cuadrado """
    area = side * side
    perim = side * 4
    return area, perim
```

Al pasar el ratón por encima de la función, el IDE muestra la documentación.

Descripción larga

```
def sqr_area_perim(side):  
    """ Calcula el área y perímetro de un cuadrado  
  
    Esta función recibe el valor de lado de un cuadrado y a  
    este calcula y retorna:  
        - su área  
        - su perímetro  
  
    Args:  
        lado (int): medida del lado del cuadrado  
  
    Returns:  
        perímetro (int): el perímetro del cuadrado  
        area (int): el área del cuadrado  
    """  
    area = side * side  
    perim = side * 4  
    return area, perim
```

Formatos más comunes de documentación en Python:

- NumPy/SciPy docstrings
- PyDoc
- EpyDoc
- Google Docstrings

Cada dev puede escoger el estilo que más le guste. En equipo, todo el equipo debe acordar el mismo tipo de documentación para apoyar la legibilidad del proyecto.

@Desiré Marrón