



# Taller de Programación Web

## Funciones



Subsecretaría de  
**Empleo**  
Chaco Gobierno de todos



Ministerio de  
**Producción, Industria y Empleo**  
Chaco Gobierno de todos



**CHACO**  
Gobierno de todos



## INTRODUCCIÓN

### Funciones

Una **función** es un fragmento de código con un nombre asociado que **realiza una serie de tareas y devuelve un valor**. A los fragmentos de código que tienen un nombre asociado y no devuelven valores se les suele llamar **procedimientos**.

En Python no existen los procedimientos, ya que **cuando el programador no especifica un valor de retorno la función devuelve el valor None** (nada), equivalente al null de Java.

El uso de funciones es un componente muy importante del paradigma de la programación llamada estructurada, y tiene varias ventajas:

- **Modularización:** permite segmentar un programa complejo en una serie de partes o módulos más simples, facilitando así la programación y el depurado.
- **Reutilización:** permite reutilizar una misma función en distintos programas.



## Sentencia def

La sentencia **def** es una definición de función usada para crear funciones definidas por el usuario.

Una **definición de función** es una sentencia ejecutable. Su ejecución enlaza el nombre de la función en el namespace local actual a un objeto función (un envoltorio alrededor del código ejecutable para la función). Este objeto función contiene una referencia al namespace local como el namespace global para ser usado cuando la función es llamada.

*La definición de la función no ejecuta el cuerpo de la misma; esto es ejecutado solamente cuando la función es llamada.*

En Python las funciones se declaran de la siguiente forma:

```
def mi_funcion(param1, param2):  
    print (param1 print param2)
```

Es decir, la palabra clave **def** seguida del nombre de la función y entre paréntesis cero o más argumentos como entrada separados por comas, en otra línea, indentado y después de los dos puntos tendríamos las líneas de código que conforman el conjunto de instrucciones a ser ejecutadas por la función.

También podemos encontrarnos con una cadena de texto como primera línea del cuerpo de la función, estas cadenas se conocen con el nombre de **docstring(cadena de documentación)** y sirven, como su nombre indica, a modo de documentación de la función. Hay herramientas que usan las docstrings para producir automáticamente documentación en línea o imprimible, o para permitirle al usuario que navegue el código en forma interactiva; es una buena práctica incluir docstrings en el código que uno escribe, por lo que se debe hacer un hábito de esto.

```
def mi_funcion(param1, param2):  
    """Esta función imprime los dos valores pasados como parámetros"""  
    print (param1)  
    print (param2)
```

Como se mencionó anteriormente en la definición de la función lo único que hacemos es asociar un nombre al fragmento de código que conforma la función, de forma que podamos



ejecutar dicho código más tarde referenciándolo por su nombre. Para llamar a la función (ejecutar su código) se escribiría:

```
mi_funcion("Hola", 2)
```

Es decir, el nombre de la función a la que queremos llamar seguido de los valores que queramos pasar como parámetros entre paréntesis.

| Código   |
|--|
| <pre>Imprimir los valores pasados como parámetro.  def mi_funcion(param1, param2):     """Esta función imprime los dos valores pasados como     parámetros"""     print(param1)     print(param2)  mi_funcion("Hola", 2)</pre> |
| Resultado  |
| <pre>Hola 2</pre>  |

## Qué son los parámetros?

Seguí revisando en el aula y revisemos de qué se trata