

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Entrada y Salida estándar.



INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Entrada y Salida estándar.

Entrada por teclado: la función `input()`

La función `input()` permite obtener texto escrito por teclado. Al llegar a la función, el programa se detiene esperando que se escriba algo y se pulse la tecla **Intro**.

```
print("¿Cómo se llama?")
nombre = input()
print(f"Me alegro de conocerle, {nombre}")
```

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Entrada y Salida estándar.

Entrada por teclado: la función `input()`

De forma predeterminada, la función **`input()`** convierte la entrada en **una cadena**, aunque escribamos un número. Si intentamos hacer operaciones, se producirá un error.

```
edad = int(input('Teclear edad: ')) # entrada de entero
peso = float(input('Teclear peso: ')) # entrada de flotante
nombre = input('Teclear nombre: ') # entrada de cadena
print(nombre, edad, 'años', peso, 'kg') # muestra datos
```

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Entrada y Salida estándar.

Salida por pantalla: la función `print()`

En Informática, la "salida" de un programa son los datos que el programa proporciona al exterior.

La función **`print()`** se utiliza para mostrar información en la **salida estándar** que, normalmente, se corresponde con la pantalla de un ordenador. El texto a mostrar se escribe como argumento de la función. Las cadenas se pueden delimitar tanto por comillas dobles (") como por comillas simples (').

```
print("Hola")
```

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Entrada y Salida estándar.

Salida por pantalla: la función `print()`

La función `print()` admite varios argumentos seguidos. En el programa, los argumentos deben separarse por comas. Los argumentos se muestran en el mismo orden y en la misma línea, separados por espacios:

```
print("Hola", "Adiós")
```

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Entrada y Salida estándar.

Salida por pantalla: la función print()

Cuando se trata de dos cadenas seguidas, se puede **no** escribir comas entre ellas, pero las cadenas se escribirán seguidas, sin espacio en blanco entre ellas:

```
print("Hola" "Adiós")
```

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Entrada y Salida estándar.

Salida por pantalla: la función `print()`

Al final de cada `print()`, Python añade automáticamente un salto de línea. Si se quiere que Python no añada un salto de línea al final de un `print()`, se debe añadir al final el argumento `end=""`:

```
print("Hola.", end=" ")  
print("Adiós")
```

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Entrada y Salida estándar.

Salida por pantalla: la función `print()`

La función `print()` permite incluir variables o expresiones como argumento, lo que nos permite **combinar texto y variables**:

```
nombre = "Pepe"  
edad = 25  
print("Me llamo", nombre, "y tengo", edad, "años.")
```

```
semanas = 4  
print("En", semanas, "semanas hay", 7 * semanas, "días.")
```


INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Entrada y Salida estándar.

Salida por pantalla: la función `print()`

La función `print()` muestra los argumentos **separados por espacios**, lo que a veces no es conveniente.

Python 3.6 se añadió una nueva notación para cadenas llamada **cadenas "f"**, que simplifica la inserción de variables y expresiones en las cadenas. Una cadena "f" contiene **variables y expresiones entre llaves ({}), que se sustituyen directamente por su valor**. Las cadenas "f" se reconocen porque **comienzan por una letra f antes de las comillas de apertura**. Si no se escribe la letra f antes de la cadena, Python no sustituye los valores de las variables ni calcula las expresiones.

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Entrada y Salida estándar.

Salida por pantalla: la función print()

```
nombre = "Pepe"  
print("¡Hola,", nombre, "!")
```

```
nombre = "Pepe"  
edad = 25  
print(f"Me llamo {nombre} y tengo {edad} años.")
```

```
semanas = 4  
print(f"En {semanas} semanas hay {7 * semanas} días.")
```

INTRODUCCIÓN A LA PROGRAMACIÓN

Python – Entrada y Salida estándar.

“Fin del tema”

