

Interacción con el Usuario en Python

En Python, se puede interactuar con el usuario a través de dos funciones principales:

1. **print()**: Para mostrar mensajes o resultados.
2. **input()**: Para recibir datos del usuario a través del teclado. Siempre lo que se ingresa se interpreta como una cadena.

1. Mostrando mensajes al usuario con print()

La función **print()** permite mostrar información o mensajes en la consola. Es muy versátil y puede manejar diversos tipos de datos (cadenas, números, listas, etc.).

Ejemplo 1: Mostrando un mensaje simple

```
print("Hola, bienvenido al programa.")
```

Salida:

Hola, bienvenido al programa.

Ejemplo 2: Mostrando variables

Puedes incluir el valor de una variable dentro del mensaje usando **f-strings** o concatenación.

Opción 1: f-strings

```
nombre = "Ana"  
print(f"Hola, {nombre}, ¿cómo estás?")
```

Opción 2: Concatenación

```
nombre = "Ana"  
print("Hola, " + nombre + ", ¿cómo estás?")
```

Salida en ambos casos:

Hola, Ana, ¿cómo estás?

Ejemplo 3: Mostrando múltiples valores

Puedes mostrar múltiples valores dentro de una única instrucción **print()** separándolos por comas.

```
edad = 25  
nombre = "Carlos"  
print("El usuario", nombre, "tiene", edad, "años.")
```

Salida:

El usuario Carlos tiene 25 años.

2. Solicitando datos al usuario con input()

La función `input()` permite que el programa espere una entrada del usuario. La información ingresada por el usuario se trata como **texto** (cadena de caracteres) por defecto.

Ejemplo 1: Solicitar un nombre

```
nombre = input("Por favor, introduce tu nombre: ")
print(f"Hola, {nombre}, gracias por participar.")
```

Salida (suponiendo que el usuario introduce "Juan"):

```
Por favor, introduce tu nombre: Juan
Hola, Juan, gracias por participar.
```

Ejemplo 2: Convertir entradas a otros tipos de datos

El valor que devuelve `input()` siempre es una **cadena de texto**, por lo que si necesitas trabajar con números, debes convertirlo.

Convertir a número entero (int):

```
edad = input("¿Cuántos años tienes? ")
edad = int(edad) # Convertir a entero
print(f"Tienes {edad} años.")
```

Salida:

```
¿Cuántos años tienes? 25
Tienes 25 años.
```

Convertir a número decimal (float):

```
altura = input("¿Cuál es tu altura en metros? ")
altura = float(altura) # Convertir a número decimal
print(f"Mides {altura} metros.")
```

Salida:

```
¿Cuál es tu altura en metros? 1.75
Mides 1.75 metros.
```

Ejemplo 3: Sumar dos números ingresados por el usuario

```
numero1 = input("Introduce el primer número: ")
numero2 = input("Introduce el segundo número: ")
```

```
# Convertir las entradas a números enteros
numero1 = int(numero1)
numero2 = int(numero2)
```

```
# Mostrar la suma de los dos números
suma = numero1 + numero2
print(f"La suma de {numero1} y {numero2} es {suma}.")
```

Salida:

Introduce el primer número: 10
Introduce el segundo número: 5
La suma de 10 y 5 es 15.

3. Ejemplo completo: Programa interactivo

A continuación, se muestra un ejemplo de cómo puedes combinar las funciones `print()` e `input()` para crear un programa interactivo.

```
# Saludar al usuario y pedir su nombre
print("¡Hola! Bienvenido a nuestro sistema.")
nombre = input("Por favor, introduce tu nombre: ")

# Pedir la edad del usuario
edad = input("¿Cuántos años tienes? ")
edad = int(edad) # Convertir la entrada a número entero

# Calcular cuántos años tendrá el usuario en 5 años
edad_futura = edad + 5
print(f"Hola, {nombre}. En 5 años tendrás {edad_futura} años.")

# Pedir dos números al usuario y mostrarlos sumados
print("Vamos a hacer una pequeña suma.")
numero1 = int(input("Introduce el primer número: "))
numero2 = int(input("Introduce el segundo número: "))

suma = numero1 + numero2
print(f"La suma de {numero1} y {numero2} es {suma}.")
```

Salida ejemplo:

¡Hola! Bienvenido a nuestro sistema.
Por favor, introduce tu nombre: Carlos
¿Cuántos años tienes? 30
Hola, Carlos. En 5 años tendrás 35 años.
Vamos a hacer una pequeña suma.
Introduce el primer número: 10
Introduce el segundo número: 20
La suma de 10 y 20 es 30.

Conclusión

- **`print()`** te permite mostrar mensajes y resultados al usuario.
- **`input()`** te permite pedir información al usuario y capturar la entrada como texto, que luego puedes convertir a otros tipos de datos según sea necesario.

Ejercicios:

- 1)** Declara una variable que contenga un nombre, después muestra un mensaje de bienvenida por consola con el nombre asignado a la variable. Por ejemplo: si asigno "Fernando", me aparezca "Bienvenido Fernando".
- 2)** Modifica la aplicación anterior, para que nos pida el nombre que queremos introducir.
- 3)** Declara dos variables numéricas (con el valor que desees), muestra por consola la suma, resta, multiplicación, división y módulo (resto de la división).
- 4)** Declara 2 variables numéricas (con el valor que desees), he indica cual es mayor de los dos. Si son iguales indicarlo también. Ves cambiando los valores para comprobar que funciona.
- 5)** Lee un número por teclado e indica si es divisible entre 2 (resto = 0). Si no lo es, también debemos indicarlo.
- 6)** Lee un número por teclado que pida el precio de un producto (puede tener decimales) y calcule el precio final con IVA. El IVA sera una constante que sera del 21%.
- 7)** Muestra los números del 1 al 100 (ambos incluidos). Usa un bucle while.
- 8)** Haz el mismo ejercicio anterior con un bucle for.
- 9)** Muestra los números del 1 al 100 (ambos incluidos) divisibles entre 2 y 3. Utiliza el bucle que desees.
- 10)** Lee un número por teclado y comprueba que este número es mayor o igual que cero, si no lo es lo volverá a pedir (do while), después muestra ese número por consola.
- 11)** Escribe una aplicación con una variable que contenga una contraseña cualquiera. Después se te pedirá que introduzcas la contraseña, con 3 intentos. Cuando aciertes ya no pedirá más la contraseña y mostrara un mensaje diciendo "Acceso Correcto". Piensa bien en la condición de salida (3 intentos y si acierta sale, aunque le queden intentos). Si no acierta en ninguno de los 3 intentos, mostrar el mensaje "El acceso se ha bloqueado después de los 3 intentos". Fin programa.
- 12)** Crea una aplicación que nos pida un día de la semana y que nos diga si es un día laboral o no (siendo sábado y domingo no laborales). Usa un switch para ello. Valida que el número ingresado sea un valor entre 1 y 7, caso contrario solicite el valor nuevamente. Debe investigar cuales son las alternativas que se pueden codificar para reemplazar o emular una estructura switch por ejemplo implementando diccionarios. Dado que Python no posee esta estructura por defecto.

13) Pide un número por teclado e indica si es un número primo o no. Un número primo es aquel solo puede dividirse entre 1 y sí mismo. Por ejemplo: 25 no es primo, ya que 25 es divisible entre 5, sin embargo, 17 si es primo. Un buen truco para calcular la raíz cuadrada del numero e ir comprobando que si es divisible desde ese número hasta 1. NOTA: Si se introduce un número menor o igual que 1, directamente es no primo.

14) Codifique un programa de consola en Python que permita realizar las siguientes acciones:

Generar un número aleatorio entre 0 y 100, para ello use la siguiente función:

`random.randint(0, 100)`

Una vez generado el número codifique la lógica necesaria para encontrar el número aleatorio ayudando al usuario informando al mismo si el número ingresado es mayor o menor al número aleatorio buscado. Una vez encontrado el número muestre la cantidad de intentos necesarios para lograrlo.

Ejemplo: Número aleatorio generado: 63

Ingrese un número entre 0 y 100.

Numero Ingresado: 50

Respuesta: Es muy bajo

Ingrese un número entre 0 y 100.

Numero Ingresado: 75

Respuesta: Es muy alto

Ingrese un número entre 0 y 100.

Numero Ingresado: 60

Respuesta: Es muy bajo

Ingrese un número entre 0 y 100.

Numero Ingresado: 65

Respuesta: Es muy alto

Ingrese un número entre 0 y 100.

Numero Ingresado: 63

Respuesta: Correcto, numero encontrado, cantidad de intentos 5

15) Dados los siguientes criterios de divisibilidad:

Criterios de divisibilidad del 2

Para saber si un número es divisible entre dos hay que comprobar que sea par

Criterios de divisibilidad del 3

Sumamos las cifras del número y si el resultado de la suma es un número múltiplo de 3, entonces el número sí es divisible por 3.

Ejemplo

Como ya sabemos que 45 es divisible por 3 vamos a comprobar que la suma de sus cifras es un múltiplo de 3.

Sumamos sus cifras: $45 \rightarrow 4 + 5 = 9$

9 es divisible por 3 por lo tanto 45 también es divisible por 3.

Criterio de divisibilidad del 5

Para saber si un número es divisible entre 5, dicho número tiene que acabar en 0 o 5

Criterios de divisibilidad del 6

Si se cumplen los criterios del 2 y del 3, entonces también es divisible por 6

Criterio de divisibilidad del 9

Un número es divisible entre 9 cuando la suma de sus dígitos es 9 o múltiplo de 9.

Por ejemplo, vamos a comprobar si 2610 es un múltiplo de 9.

$2 + 6 + 1 + 0 = 9$, por lo tanto 2610 es divisible por 9.

Criterio de divisibilidad del 10

Para saber si un número es divisible entre 10, éste tiene que acabar en 0.

Codifique un programa en Python que solicite el ingreso de un número entero y determine cuáles son los criterios de divisibilidad que cumple.