

Programación con python

Retomando conocimientos básicos

Términos Clave

Python es un lenguaje de programación y de scripting de propósito general que es utilizado por millones de desarrolladores en todo el mundo. Python se utiliza más comúnmente para:

- Aprendizaje Automático (Machine Learning)
- Inteligencia Artificial
- Ciencia de Datos
- Aplicaciones Financieras
- Desarrollo Web
- Scripting
- Automatización de Tareas

Tipos de datos en Python

- **Int**, el tipo de dato int puede representar todos los números enteros, positivos o negativos...., -2, -1, 0, 1, 2, ... son todos int en Python.
- **Float**, el tipo de dato float representa números de punto flotante, positivos o negativos.-2.1, .2, 7., -90.234, 0.222321, 3.14 son todos floats en Python. Incluso números como 2.0, que son enteros, se consideran floats debido a la presencia del punto decimal.
- **String**, el tipo de dato str representa una secuencia de caracteres. En Python, comienzan y terminan con ", "" o "". Ejemplos de strings válidos en Python: "hello", "2.0", 'str', """False""", ""true""".
- **Bool**, el tipo de dato bool representa uno de dos valores: True o False.

Variables

Una variable puede considerarse como un contenedor que almacena un valor. Como programador, puedes crear tus propias variables que almacenan diferentes tipos de datos.

El siguiente es un ejemplo de una variable:

```
hello_world = "esta variable contiene una cadena"
```

En Python, los nombres de variables deben:

- No comenzar con un número.
- No contener ningún carácter especial, excepto guiones bajos (_).
- No contener espacios.

Variables

Una variable puede considerarse como un contenedor que almacena un valor. Como programador, puedes crear tus propias variables que almacenan diferentes tipos de datos.

Los siguientes son ejemplos de variables:

```
hello_world = "esta variable contiene una cadena"  
N1 = 1  
n2 = 1.5  
seguir = True
```

En Python, los nombres de variables deben:

- No comenzar con un número.
- No contener ningún carácter especial, excepto guiones bajos (_).
- No contener espacios.

Input

La función input está incorporada directamente en Python como un medio para recopilar datos del usuario desde la línea de comandos.

Un punto importante es que siempre devolverá un objeto de tipo str, el cual deberá convertirse a un int (por ejemplo) si esperas que la entrada del usuario sea un número entero.

Ejemplo:

```
user_name = input("¿Cuál es tu nombre? ")
user_age = int(input("¿Cuál es tu edad? "))
print("Hola " + user_name + "!")
print("Tienes " + str(user_age) + " años")
print("El próximo año tendrás " + str(user_age + 1) + " "
años")
```

Operadores aritméticos

- **Operador.** Un operador es un símbolo que indica una operación a realizar. `+`, `-`, `/`, `*`, `**`, `//` son todos operadores.
- **Operando.** Un operando es un objeto sobre el que actúa un operador. En la expresión `2 + 3`, tanto `2` como `3` son operandos, mientras que `+` es el operador.
- **División Entera.** La división entera es aquella en la que se descarta la parte fraccionaria (residuo). En Python, la división entera se realiza usando el operador `//`. Por ejemplo, la expresión `11 // 3` evalúa a `3`.

- **Módulo.** El módulo (o resto) es el residuo después de dividir un número entre otro. En Python, el módulo se calcula usando el operador %. Por ejemplo, la expresión `10 % 3` evalúa a 1.
- **Concatenación.** En Python, la concatenación es el proceso de combinar o unir cadenas (strings). La concatenación se puede realizar usando el operador +. Por ejemplo, la expresión `"1" + "word"` evalúa a `"1word"`.

fstring

Las f-strings (formatted string literals) son una forma rápida y legible de insertar variables o expresiones dentro de cadenas de texto.

Se escriben anteponiendo la letra f a las comillas de la cadena.

Dentro de la cadena, las variables o expresiones se colocan entre llaves {}.

Python convierte automáticamente los valores a texto, por lo que no es necesario usar str().

Ejemplo:

```
user_name = input("¿Cuál es tu nombre? ")
user_age = int(input("¿Cuál es tu edad? "))
print(f"Hola {user_name}!")
print(f"Tienes {user_age} años")
print(f"El próximo año tendrás {user_age + 1} años")
```

Ventaja: el código es más claro y conciso que con la concatenación tradicional.

Comentarios

Los comentarios son partes del código que son ignoradas por el intérprete de Python.

Su propósito es explicar o documentar secciones del programa para que sean más fáciles de entender por otras personas (o por uno mismo en el futuro).

Comentario de una sola línea (Single-line comment)

En Python, un comentario de una sola línea se escribe con el símbolo #.

```
# Este programa pide el nombre al usuario y lo saluda
nombre = input("¿Cuál es tu nombre? ")
print("Hola", nombre)
```

Comentarios de varias líneas (Multi-line comment)

"""

Este programa determina si un número es par o impar.

1. Se solicita un número al usuario.
2. Se calcula el residuo con %.
3. Se imprime si es par o impar.

"""

```
numero = int(input("Escribe un número: "))
```

```
if numero % 2 == 0:  
    print("El número es par")  
else:  
    print("El número es impar")
```

Ejercicio 1

Cree un programa en Python para capturar la información de una venta en una tienda donde solicite al usuario:

- Nombre del cliente
- Descripción del producto
- Cantidad de artículos (no hay fracciones de productos)
- Precio del artículo

Con esa información calcule y muestre:

- El subtotal
- El impuesto al valor agregado (IVA 16%)
- El total (subtotal + IVA)

NO USE INTELIGENCIA ARTIFICIAL PARA SU TRABAJO
ENTREGUELA EN LA SECCION DE TAREAS

Ing. Luis Francisco Contreras González

Nombre del cliente: Luis Francisco
Descripción del producto: Articulo de prueba
Cantidad de artículos: 10
Precio por artículo: 155.50

----- DETALLE DE LA VENTA -----

Cliente: Luis Francisco
Producto: Articulo de prueba
Cantidad: 10
Precio unitario: \$ 155.5
Subtotal: \$ 1555.0
IVA (16%): \$ 248.8
Total a pagar: \$ 1803.8