

HINTS

CÁLCULOS ARITMÉTICOS

En Python podemos hacer cálculos con variables, y luego el resultado asignarlo a alguna de las variables que usamos para realizarlo. De esta forma estaríamos actualizando el valor de la variable.

```
y = y + 1
```

De la misma manera, Python nos permite simplificar la forma de escribir algunos cálculos aritméticos como los descritos anteriormente. Para ello podemos usar los operadores `+=`, `-=`, `*=` y `/=`.

Así, la expresión del ejemplo anterior podríamos también escribirla:

```
1 y += 1
```

DATO BOOLEANO

Es importante tomar en cuenta que los datos tipo bool deben ser escritos la primera letra en mayúscula, y las siguientes en minúscula (False / True) para que el intérprete de Python las interprete correctamente.

VARIABLES COMBINADAS

Para imprimir variables combinadas con texto usando la función `print()` podemos hacerlo de dos formas diferentes.

Concatenación:

La primera de ellas es separar el texto en comillas, y las variables con el símbolo más (+), antes y después de la variable. Para que este método funcione el valor de la variable debe ser tipo cadena.

```
1 cantidad = "dos"
2 print("Hoy comí " + cantidad + " helados")
```

Resultado:

```
Hoy comí dos helados
```

Interpolación:

La interpolación de variables en cadenas en Python se puede hacer de manera sencilla utilizando f-strings (formatted string literals). Las f-strings permiten incluir expresiones dentro de una cadena, encerradas entre llaves {}, y están precedidas por una f antes de las comillas de la cadena.

- Sintaxis Básica:
 - Para interpolar una variable, se coloca una f antes de abrir las comillas.
 - Dentro de las llaves {}, se puede incluir el nombre de la variable o cualquier expresión válida de Python.

```
1 cantidad = 2
2 print(f"Hoy comí {cantidad} helados")
```

Resultado:

```
Hoy comí 2 helados
```

La interpolación de variables es una característica presente en varios lenguajes de programación. Por ejemplo, en JavaScript, se puede realizar utilizando template literals. Los template literals se definen con las comillas invertidas (backticks) ` en lugar de las comillas simples ' ' o dobles " ". Dentro de un template literal, las variables y expresiones se pueden incluir mediante \${}.

```
const nombre = "Ana";
const edad = 30;

const mensaje = `Hola, mi nombre es ${nombre} y tengo ${edad} años.`;

console.log(mensaje);
```