

Variables y Expresiones

Algoritmos y Programación Básica (CC2005) - 2026

Variables y Expresiones

Semestre 02, 2025

Problemática

- Queremos resolver problemas con datos.
- Los datos cambian.
- No siempre conocemos los valores antes de ejecutar el programa.

Calcular el promedio de un estudiante:

- El nombre cambia.
- Las notas cambian.
- El resultado cambia.

Es necesaria una forma de "guardar información"

Datos

Definición

Un dato es una pieza básica de información que un programa puede manipular.

- Números
- Texto
- Valores lógicos
- 42
- 3.14
- "Hola"
- True

Cada uno representa un tipo distinto de información.

Variables

Definición

Una variable es un nombre que hace referencia a un valor almacenado en memoria.

Permite trabajar con datos sin conocer su valor de antemano.

Una variable es como:

- Una caja etiquetada.
- Un contenedor con nombre.
- Un espacio donde guardamos algo.

Ej

```
mensaje = "Hola Mundo"
```

- **mensaje** es la variable.
- **Hola Mundo** es el valor almacenado.

Asignación

Definición

Asignar significa guardar un valor en una variable.

```
variable = valor
```

Ej

```
edad = 18
```

La variable **edad** ahora contiene el valor 18.

Reasignación

```
edad = 18  
edad = 19
```

El valor anterior se reemplaza.

La variable siempre guarda el valor más reciente.

Una variable solo puede almacenar 1 valor a la vez.

Reglas para nombrar variables

- Deben iniciar con una letra.
- Pueden contener letras, números y `_`.
- Son sensibles a mayúsculas y minúsculas.
- No pueden usar palabras reservadas.

Ej - incorrecto

```
50marimbas = "ensamble"
```

Ej - correcto

```
nombre_completo = "Ana Pérez"
```

Tipos de datos

El tipo de dato define qué clase de información almacena una variable.

Python asigna el tipo automáticamente.

Tipos principales

- `int` → números enteros
- `float` → números decimales
- `str` → texto
- `bool` → verdadero o falso

Ej

```
numero = 10
precio = 3.5
nombre = "Luis"
es_mayor = True
```

Verificar y convertir tipos

Determinar el tipo

Se usa la función `type()` .

```
type(numero)
```

Devuelve el tipo de dato almacenado en la variable.

Ejemplo:

```
numero = 10
print(type(numero))
```

Resultado: `<class 'int'>`

Conversión de tipos

A veces necesitamos transformar un valor.

```
numero = 3.14
entero = int(numero)
```

- `numero` es un `float`
- `entero` se convierte en `int`
- El valor final es `3`

La conversión elimina la parte decimal.

Otros ejemplos

```
texto = "25"
numero = int(texto)

edad = 20
texto_edad = str(edad)
```

- `int()` convierte a entero
- `str()` convierte a texto
- `float()` convierte a decimal

Operadores aritméticos

Operador	Significado
<code>+</code>	Suma
<code>-</code>	Resta
<code>*</code>	Multiplicación
<code>/</code>	División

Ej

```
a = 5
b = 2
resultado = a + b
```

Operadores relacionales

Operador	Significado
<code>==</code>	Igual
<code>!=</code>	Diferente
<code>></code>	Mayor
<code><</code>	Menor

Siempre devuelven un valor booleano.

Ej

```
edad = 20  
edad >= 18
```

Resultado: `True`

Operadores lógicos

Sirven para crear conexiones y formar expresiones más compuestas.

- `and`
- `or`
- `not`

Para determinar su valor usamos las tablas de la verdad.

Ej

```
edad = 20  
tiene_credencial = True  
edad >= 18 and tiene_credencial
```

Expresiones

Definición

Es la combinación de:

- Datos
- Variables

- Operadores

Que produce un resultado.

Ej

```
x = 10  
y = (2 * x) + 5
```

y es el resultado de una expresión.

Orden de operaciones

PEMDASA

1. Paréntesis
2. Exponentes
3. Multiplicación
4. División
5. Adición
6. Sustracción
7. Asignación

Ej

```
resultado = (5 + 3) * 2
```

Primero se resuelve el paréntesis.

Operaciones con cadenas

Concatenación

Concatenar es prácticamente unir 2 cadenas.

```
nombre = "Ana"
saludo = "Hola " + nombre

print(saludo)
# Hola Ana
```

Repetición

```
print(" " * 3)

#   
```

¿No era lo que esperaban?

Los tipos de datos son importantes ya que dictaminan el comportamiento de los operandos dada una operación.

Problemas comunes

- Usar variables sin definir.
- Confundir `=` con `==`.
- Mezclar tipos incompatibles.

Ej

```
numero = "10"
numero + 5
```

Error por mezclar texto con número.