

# LENGUAJES DE PROGRAMACIÓN

**PYTHON**

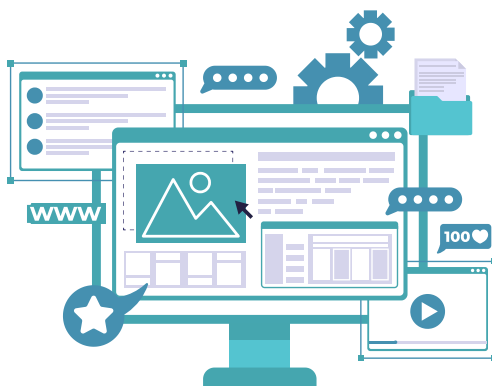




## Sintaxis de las Variables y los Operadores en Python

La programación en Python es una habilidad fundamental en el mundo actual, ya que este lenguaje de programación se ha convertido en uno de los más populares y ampliamente utilizados. Al aprender Python, es esencial comprender la sintaxis de las variables y los operadores, ya que forman la base para realizar cálculos, almacenar y manipular datos.

En esta lección, exploraremos la sintaxis de las variables y los operadores en Python. Comenzaremos revisando los conceptos básicos de las variables y cómo se utilizan para almacenar datos. Luego, nos adentraremos en los operadores, que nos permiten realizar diversas operaciones matemáticas y de manipulación de datos en Python.





## Variables en Python

Una variable es un nombre que se utiliza para almacenar un valor en la memoria de la computadora. En Python, no es necesario declarar explícitamente el tipo de variable, ya que se determina automáticamente según el valor asignado. Para asignar un valor a una variable, se utiliza el signo de igual (=). Veamos un ejemplo:

```
edad = 25
```

En este ejemplo, hemos creado una variable llamada "edad" y le hemos asignado el valor entero 25. A partir de este momento, podemos utilizar la variable "edad" para realizar cálculos o mostrar su valor. Por ejemplo:

```
print(edad) # Imprime el valor de la variable edad
```

En Python, los nombres de las variables son sensibles a mayúsculas y minúsculas, lo que significa que "edad" y "Edad" se considerarían variables diferentes.

Es importante elegir nombres descriptivos y significativos para las variables, ya que esto facilita la comprensión del código. Además, los nombres de las variables no pueden comenzar con un número y no deben contener espacios en blanco.



## Operadores Aritméticos

Los operadores aritméticos en Python nos permiten realizar operaciones matemáticas básicas. A continuación, se presentan los operadores aritméticos más comunes en Python:

**Suma:** se utiliza el operador "+". Por ejemplo:

```
a = 5
b = 3
resultado = a + b
print(resultado) # Imprime 8
```

**Resta:** se utiliza el operador "-". Por ejemplo:

```
a = 5
b = 3
resultado = a - b
print(resultado) # Imprime 2
```

**Multiplicación:** se utiliza el operador "\*". Por ejemplo:

```
a = 5
b = 3
resultado = a * b
print(resultado) # Imprime 15
```



**División:** se utiliza el operador `"/"`. Por ejemplo:

```
a = 10
b = 2
resultado = a / b
print(resultado) # Imprime 5.0
```

**División entera:** se utiliza el operador `"//"`. Este operador devuelve el resultado de la división como un número entero, ignorando la parte decimal. Por ejemplo:

```
a = 10
b = 3
resultado = a // b
print(resultado) # Imprime 3
```

**Módulo:** se utiliza el operador `"%"`. El operador módulo devuelve el resto de la división entre dos números. Por ejemplo:

```
a = 10
b = 3
resultado = a % b
print(resultado) # Imprime 1
```



## Operadores de Asignación

Los operadores de asignación se utilizan para asignar valores a las variables. A continuación, se presentan algunos ejemplos de operadores de asignación en Python:

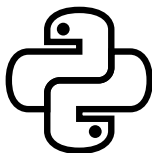
Asignación simple: se utiliza el operador "=" para asignar un valor a una variable. Por ejemplo:

```
edad = 25
```

**Asignación con suma:** se utiliza el operador "+=" para sumar un valor a la variable existente y asignar el resultado a la misma variable.

**Por ejemplo:**

```
contador = 0  
contador += 1 # Equivale a contador = contador + 1  
print(contador) # Imprime 1
```







**Asignación con resta, multiplicación y división:** sse utilizan los operadores "-", "\*", "/" de manera similar a la asignación con suma.

**Por ejemplo:**

```
a = 10
a -= 5 # Equivale a a = a - 5
print(a) # Imprime 5

b = 3
b *= 2 # Equivale a b = b * 2
print(b) # Imprime 6

c = 10
c /= 2 # Equivale a c = c / 2
print(c) # Imprime 5.0
```





## Operadores de Comparación

Los operadores de comparación se utilizan para comparar dos valores y devuelven un resultado booleano (True o False) según si la comparación es verdadera o falsa. A continuación, se presentan algunos ejemplos de operadores de comparación en Python:

**Igualdad:** se utiliza el operador "==". Por ejemplo:

```
a = 5
b = 5
resultado = a == b
print(resultado) # Imprime True
```

**Desigualdad:** se utiliza el operador "!=". Por ejemplo:

```
a = 5
b = 3
resultado = a != b
print(resultado) # Imprime True
```

**Mayor que:** se utiliza el operador ">". Por ejemplo:

```
a = 5
b = 3
resultado = a > b
print(resultado) # Imprime True
```





**Menor que:** se utiliza el operador "<". Por ejemplo:

```
a = 5
b = 3
resultado = a < b
print(resultado) # Imprime False
```

**Mayor o igual que y menor o igual que:** se utilizan los operadores ">=" y "<=". Por ejemplo:

```
a = 5
b = 5
resultado = a >= b
print(resultado) # Imprime True
```

```
c = 3
d = 5
resultado = c <= d
print(resultado) # Imprime True
```





En esta lección, hemos explorado la sintaxis de las variables y los operadores en Python. Hemos aprendido cómo asignar valores a variables, utilizar operadores aritméticos para realizar cálculos y operadores de comparación para comparar valores. Estos conceptos son fundamentales para comprender y escribir programas en Python.

A medida que avancemos en nuestro aprendizaje de Python, nos encontraremos con otros tipos de variables y operadores más avanzados. Sin embargo, los conceptos básicos de variables y operadores que hemos cubierto en esta introducción nos proporcionarán una base sólida para construir nuestro conocimiento y habilidades en programación.

Recuerda practicar con ejemplos adicionales y experimentar con diferentes combinaciones de variables y operadores para fortalecer tu comprensión de la sintaxis de Python. En las siguientes lecciones, exploraremos conceptos más avanzados, como estructuras de control y funciones, que nos permitirán crear programas más complejos y completos.

