

# Convenciones de Estilo para Python

---

Las convenciones de estilo de programación son guías utilizadas por los programadores para tener una manera consistente de escribir los programas. Su objetivo principal es mejorar la legibilidad del código. La presente guía está diseñada para ser utilizada en un curso básico de programación y la mayor parte de su contenido está basada en las guías de estilo de Python.

## IDENTIFICADORES

- Utilizar **identificadores (nombres) descriptivos** para variables, funciones, parámetros y constantes nombradas.
- Los nombres de identificadores de variables, funciones y parámetros deben utilizar solamente **letras minúsculas, dígitos y guion bajo (\_)**. Las letras no deben contener acentos, diéresis ni tildes.
- Si un identificador está compuesto por **dos o más palabras** éstas se deben separar utilizando el carácter de **guion bajo (\_)**.

Ejemplos:

- **nombre\_alumno**
  - **dias\_trabajados\_por\_semana**
- Solamente es válido utilizar **identificadores de un solo carácter** en los siguientes casos:
  - Cuando sea por una convención muy bien establecida. Por ejemplo:
    - **Índices en un ciclo (i, j, k)**
    - **Coordenadas (x, y).**
  - Cuando el alcance de la variable sea muy corto (máximo dos líneas de código).
  - **I (ele minúscula), O (o mayúscula), l (i mayúscula)** no deben usarse como identificadores dado que es muy fácil confundirlos con los números 1 (uno) y 0 (cero) en ciertos tipos de letra.
- Una literal numérica con decimales debe estar formada por **al menos un dígito antes y un dígito después del punto decimal**.

Ejemplos:

- **5.2**
- **403.25**

## ESTRUCTURA

- El límite máximo de caracteres en una línea es de **79**.
  - Si una línea requiere de más caracteres se puede utilizar la **diagonal invertida (\)** para romperla en dos o más partes. Por ejemplo:

```
suma = primer_dato + segundo_dato + tercer_dato
```

Se puede convertir en:

```
suma = primer_dato \  
+ segundo_dato \  
+ tercer_dato
```

- Otra opción es usar **paréntesis**:

```
suma = (primer_dato  
+ segundo_dato  
+ tercer_dato)
```

- Se deben usar la tecla de **tabulación (Tab)** o **cuatro espacios consecutivos** para indicar un nivel de indentación.
- No debe haber más de un estatuto por línea de código.
- Los elementos de un estatuto deben estar **separados por un espacio en blanco**.

```
5 % 2 + 6 - 5 / 2
```

## COMENTARIOS Y DOCUMENTACIÓN:

- Todas las funciones deben contener una cadena de documentación para describir brevemente su comportamiento. Por ejemplo:

```
"""Calcula la velocidad a partir de la distancia y el tiempo."""  
def velocidad(distancia, tiempo):  
    return distancia / tiempo
```

- Se deben **eliminar** todos aquellos **comentarios que contengan código**, a menos de que sean parte de la documentación.