

TC1028 Pensamiento Computacional para Ingeniería

Convenciones de Estilo para Python

Elaborado por: Ariel Ortiz, Yolanda Martínez

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 la guía de estilo de Python [PEP 8](#).

Identificadores y constantes

- 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 guión bajo (_). Las letras no deben contener acentos, diéresis ni tildes, aunque el editor de texto lo acepte.
- Si un identificador está compuesto por dos o más palabras éstas se deben separar utilizando el carácter de guión bajo (_). Ejemplos: alumno, nombre, nombre_alumno, dias_trabajados_por_semana.
- Los identificadores que se usan como constantes nombradas deben definirse a nivel de módulo (afuera de cualquier función) y se deben escribir con letras mayúsculas usando un guión bajo como separador entre palabras. Por ejemplo: TOTAL, MAX_EMPLEADOS.
- 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 (*scope*) de la variable sea muy corto (máximo dos líneas de código).
- l (ele minúscula), O (o mayúscula), I (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 (*fonts*).
- Una literal numérica con decimales debe estar formada por al menos un dígito antes y un dígito después del punto decimal.

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 la continuación de línea implícita que ocurre cuando está pendiente cerrar un paréntesis, corchete o llave:

```
suma = (primer_dato  
      + segundo_dato  
      + tercer_dato)
```
- Se deben usar cuatro espacios consecutivos para indicar un nivel de indentación. No se debe utilizar el carácter de tabulador (código ASCII 9), también conocido como “hard tab”. Los editores modernos permiten configurarse para solamente usar “soft tabs” en lugar de “hard tabs”. El uso de “soft tabs” significa que se insertan siempre espacios al momento de indentar cuando se presiona la tecla de tabulación (Tab).
- No debe haber más de un estatuto por línea de código.
- Los elementos de un estatuto deben estar separados por un espacio en blanco.

Comentarios y documentación

- Todas las funciones deben contener una cadena de documentación para describir brevemente su comportamiento. Ejemplo:
def velocidad(distancia, tiempo):
 """Calcula la velocidad a partir de la distancia y el tiempo."""
 return distancia / tiempo
- Se deben eliminar todos aquellos comentarios que contengan código, a menos de que sean parte de la documentación.