

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general. Fue creado por Guido van Rossum y su primera versión fue lanzada en 1991. Python se destaca por su sintaxis clara y legible, lo que facilita la escritura y mantenimiento del código. Es un lenguaje versátil utilizado en diversos campos como desarrollo web, inteligencia artificial, análisis de datos, automatización, entre otros.

Estructuras de control de flujo:

i. Condicionales:

- `if`: Permite ejecutar un bloque de código si una condición es verdadera.
- `else`: Define un bloque de código a ejecutar en caso de que la condición del `if` sea falsa.
- `elif`: Se utiliza para evaluar múltiples condiciones en orden.

ii. Bucles:

- `for`: Itera sobre una secuencia (como listas, tuplas o cadenas de texto).
- `while`: Ejecuta un bloque de código mientras una condición sea verdadera.
- `break`: Sale prematuramente de un bucle.
- `continue`: Salta a la siguiente iteración del bucle.

Orden de evaluación de expresiones y funciones:

Las expresiones en Python se evalúan de izquierda a derecha. En cuanto a las funciones, los argumentos se evalúan de izquierda a derecha antes de llamar a la función.

Tipos de datos y mecanismos para la creación de nuevos tipos:

- Tipos básicos: `int`, `float`, `str`, `bool`.
- Colecciones: Listas, tuplas, conjuntos, diccionarios.
- Tipos de datos especializados: `datetime`, `timedelta`.
- Mecanismos de creación de nuevos tipos:
 - Clases: Python es orientado a objetos y permite la creación de nuevas clases para definir tipos personalizados.
 - Herencia: Permite heredar propiedades y comportamientos de una clase a otra.
 - Polimorfismo: Se logra mediante el uso de interfaces y métodos que pueden tener diferentes implementaciones en clases derivadas.

Sistema de tipos:

- Dinámico: Los tipos de variables son asignados dinámicamente durante la ejecución del programa.
- Fuertemente tipado: No se permite la mezcla de tipos incompatibles sin conversión explícita.
- Equivalencia de tipos: Python utiliza "duck typing", donde el tipo de un objeto se determina por su comportamiento en lugar de por su tipo explícito.
- Compatibilidad: Se deben respetar las reglas de tipos en operaciones, evitando combinaciones no permitidas.

- Inferencia de tipos: Python es capaz de inferir el tipo de una variable en tiempo de ejecución, pero no es estáticamente tipado como algunos lenguajes. Puede realizar inferencia en entornos específicos como en algunas IDEs o herramientas estáticas.