

Conceptos básicos y sintaxis de Python

Ideas clave

- Python es un lenguaje de programación interpretado, con orientación a objetos y de tipado dinámico.
- En un mismo equipo se pueden crear distintos entornos virtuales con los que poder tener diferentes proyectos y versiones de Python.
- Python se caracteriza por la indentación de código, no usa llaves como otros lenguajes de programación.
- Los tipos básicos en Python son:
 - **Variables, números** (enteros o de coma flotante), **strings** y **booleans**
 - **Listas**: colecciones ordenadas de elementos de tipo arbitrario y tamaño variable.
 - **Tuplas**: secuencias *inmutables* de objetos arbitrarios
 - **Diccionarios**: Colecciones de pares de clave-valor.
 - **Sets**: conjuntos de objetos *mutables* no ordenados
- Modificadores de flujo:
 - Condicionales: **IF...ELIF...ELSE**, operador ternario.
 - Bucles: **FOR**, **WHILE**
- Entrada y salida de datos: En Python podemos introducir datos por medio de la función **input()** y sacar datos mediante la función **print()**
- Una función es un trozo de código, reutilizable, que realiza una acción concreta y que tendrá dos partes, por un lado, la declaración de la función y por otro la llamada a la función. Una función no se ejecuta si no se la llama. Existen numerosas funciones predefinidas en Python ya creadas y listas para ser usadas, no obstante, nosotros podemos crear las nuestras propias.
- La modularización nos permite tener nuestro código ordenado. La idea es crear funciones en diferentes archivos .py e importarlas para usarlas en el lugar de nuestro código que sea necesario.
- En todo programa se pueden producir errores, lo que se conoce como excepciones. Es nuestra labor como programadores prever dichas excepciones y anticiparnos a ellas. En Python disponemos de 4 sentencias que podemos utilizar para manejar excepciones:
 - **try/except**: Intercepta y recupera excepciones disparadas por Python o por nuestro código.
 - **try/finally**: Realiza tareas de limpieza ocurran las excepciones o no.
 - **raise**: Dispara una excepción manualmente en el código.
 - **assert**: Dispara una excepción condicionalmente.