

Programación con



Tema 1 - Introducción

Alma Mallo Casdelo - alma.mallo@udc.es

Características de Python

- Lenguaje de programación de propósito general de código abierto.
- **Interpretado**, no compilado:
 - Penalización en velocidad de ejecución.
 - Tipado dinámico en tiempo de ejecución.
 - No hay necesidad de gestionar el uso de memoria RAM.
- Sintaxis sencilla con la intención de **facilitar entender el código**.
- Enorme cantidad de librerías, algunas muy eficientes porque están implementadas en C++:
 - Atenúa el problema generado por el hecho de que Python sea interpretado.
 - Se pueden programar algoritmos complejos en **muy pocas líneas** de código fuente.

Historia y estado actual

- Python lo diseñó Guido Van Rossum, nacido en 1956 en Haarlem, Países Bajos.
- Eligió el nombre de “Python” porque era fan de los Monty Python.
- La primera versión del lenguaje se publicó en 1991.
- Licencia de Python: **Python Software Foundation License** (PSFL) - basada en la licencia BSD
- **Python Software Foundation**: organización encargada de apoyar el desarrollo de Python.



https://en.wikipedia.org/wiki/Guido_van_Rossum#/media/File:Guido_van_Rossum_OSCON_2006.jpg

Versiones de Python

- Dos versiones principales: **Python 2** y **Python 3**.
 - No son 100% compatibles.
 - Python 2 fue declarado obsoleto el 01/01/2020
 - Si tenemos que programar en algún proyecto con Python 2, se puede programar código compatible con Python3 usando Python-Future.
 - Implica importar unos módulos determinados (que reemplazan aquellos elementos de Python 2 diferentes en Python 3).
- En este curso utilizaremos Python 3.

Intérpretes de Python



El intérprete de referencia es de la Python Software Foundation, conocido también como **CPython**.

Es el que se instala por defecto, el más utilizado y el único que soporta todos los elementos del lenguaje. Está desarrollado en C.

Es el intérprete que utilizaremos en este curso.

Intérpretes de Python



Desarrollado en C# para facilitar la integración con el framework de Microsoft .NET



Desarrollado en Java para facilitar el uso en conjunción con programas en dicho lenguaje. Corre sobre una JVM (Java Virtual Machine). Solo soporta Python 2.

Intérpretes de Python



Utiliza un compilador JIT (Just-In-Time) para que la ejecución de los programas sea más rápida.

Es un subconjunto de CPython.



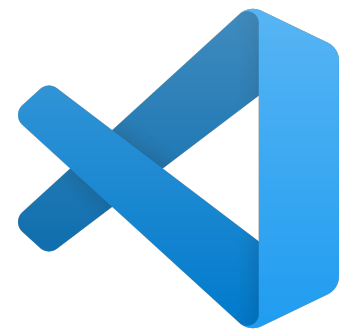
Incluye un pequeño subconjunto de la biblioteca estándar de Python.

está optimizado para ejecutarse en microcontroladores.

- IPython (*Interactive Python*) empezó como un *shell* con más posibilidades que el (limitado) que acompaña a CPython, pero ha derivado en un proyecto mucho mayor (Jupyter).
- Actualmente, IPython cubre la parte de Python de dicho proyecto y proporciona:
 - Un *shell* interactivo que, internamente, utiliza CPython.
 - Un kernel para Jupyter.
 - Soporte para visualización de datos interactiva.

- Conjunto de herramientas, formatos y protocolos para dar soporte a la programación interactiva, independientemente del lenguaje, con énfasis en la facilidad de uso, la visualización y la ejecución distribuida.
- Su uso es normalmente vía web, lo que facilita además la colaboración.
- Muy utilizado por la comunidad educativa y científica.
- El concepto fundamental es el *notebook* (cuaderno), que es un fichero que incluye código fuente (Python, por ejemplo) y texto libre que acompaña al código (Markdown o HTML). Cada bloque de texto o de código fuente se denomina celda.
- **Jupyter Notebook** es la aplicación web que permite ejecutar notebooks desde un navegador web.
- **JupyterLab** es un IDE completo, también web, sustituto de Jupyter Notebook. Tiene un sistema de plugins (extensiones) con su propio gestor. Como IDE, incorpora un depurador, un visualizador de datos, múltiples pestañas, etc.

Visual Studio Code



Visual Studio
Code

- Es un entorno de desarrollo integrado (IDE) disponible para Windows, macOS y Linux, con soporte para múltiples lenguajes de programación, gratuito y de código abierto.
- Infinidad de plugins de todo tipo e integración con software de control de versiones.
- Es el IDE que utilizaremos en este curso.

Visual Studio Code: conceptos de uso

- Se pueden editar ficheros de forma independiente.
- Se pueden abrir carpetas enteras:
 - Permite manejar más fácilmente proyectos / paquetes software.
 - Es necesario abrir una carpeta para usar el depurador que nos permitirá encontrar fallos en el código fuente.
- Se pueden definir *workspaces*:
 - Un *workspace* agrupa de forma lógica varias carpetas (pueden estar físicamente en sitios distintos).
 - Facilita trabajar con varios proyectos software o proyectos grandes divididos en varios componentes claramente identificados.
 - ¡¡Una misma carpeta puede pertenecer a varios *workspaces*!!

Visual Studio Code: conceptos de uso

- Al contrario que en la mayoría de los IDE, no se trabaja directamente con el concepto de “proyecto software” (este puede ser una carpeta o un *workspace*).
- Se puede configurar el Visual Studio Code a nivel de carpeta, de *workspace* o de forma global.
- En caso de que existan diferentes configuraciones en varios de esos niveles, la prioridad es en el orden mencionado (de más concreto a más general).

pip (*Package Installer for Python*)

- Python tiene su propio gestor de paquetes (pip) para instalar (y desinstalar) módulos de Python desde <https://pypi.org> (Python Package Index repository)
- Algunos comandos de pip:
 - pip help —> Muestra ayuda
 - pip list —> Muestra paquetes instalados
 - pip show paquete —> Muestra información de “paquete”
 - pip install paquete —> Instala “paquete”
 - pip uninstall paquete —> Desinstala “paquete”
 - pip install -U paquete —> Actualiza “paquete”
- Estos comandos los podemos ejecutar desde cualquier terminal.

Referencias

- <https://docs.python.org/2/>
- <https://docs.python.org/3/>
- <https://python-future.org>
- <https://ironpython.net>
- <https://www.jython.org>
- <https://ipython.org>
- <https://jupyter.org>
- <https://www.markdownguide.org>
- <https://code.visualstudio.com>
- <https://www.pypy.org/>