

Visual Studio Code: un editor open-source, ligero pero potente

Mathieu Kessler

Departamento de Matemática Aplicada y Estadística
Universidad Politécnica de Cartagena

<https://code.visualstudio.com>



- 1 Disponible para macOS, Linux y Windows
- 2 Tiene soporte para cientos de lenguajes de programación
- 3 Incluye muchas funcionalidades muy útiles como coloreado de sintaxis, balanceado de paréntesis, auto-indentación, selección rectangular, capsulas de código.
- 4 Permite hacer debugging
- 5 Incluye una interfaz para control de versiones (Git)
- 6 Alto grado de personalización posible, existen muchas extensiones.

- 1 Disponible para macOS, Linux y Windows
- 2 Tiene soporte para cientos de lenguajes de programación
- 3 Incluye muchas funcionalidades muy útiles como coloreado de sintaxis, balanceado de paréntesis, auto-indentación, selección rectangular, capsulas de código.
- 4 Permite hacer debugging
- 5 Incluye una interfaz para control de versiones (Git)
- 6 Alto grado de personalización posible, existen muchas extensiones.

Se trata de un editor cada vez más popular para Python

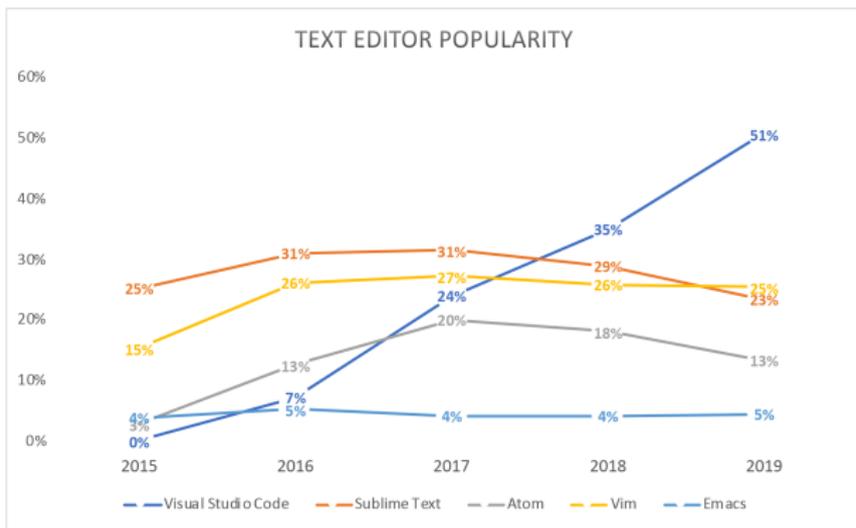
- 1 Disponible para macOS, Linux y Windows
- 2 Tiene soporte para cientos de lenguajes de programación
- 3 Incluye muchas funcionalidades muy útiles como coloreado de sintaxis, balanceado de paréntesis, auto-indentación, selección rectangular, capsulas de código.
- 4 Permite hacer debugging
- 5 Incluye una interfaz para control de versiones (Git)
- 6 Alto grado de personalización posible, existen muchas extensiones.

Se trata de un editor cada vez más popular para Python

Nota:

Visual Studio Code es completamente diferente de Microsoft Visual Studio. No os confundáis!

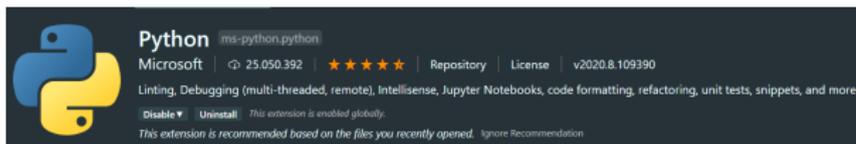
En la encuesta anual para desarrolladores de Stack Overflow, hay una pregunta sobre editor de texto usado:



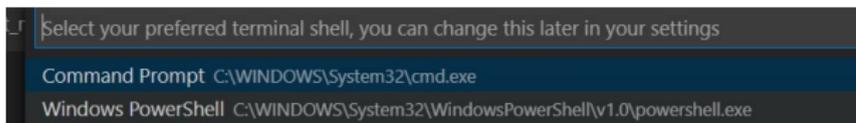
Esta gráfica está extraída del blog post de lectura muy recomendable: *The era of Visual Studio Code*, Roben Kleene, 2020-09-21.

<https://blog.robenkleene.com/2020/09/21/the-era-of-visual-studio-code/>

- Descargad y lanzad el ejecutable de instalación de <https://code.visualstudio.com>.
- Abrid el market place de extensiones a través del icono en la barra lateral y buscad la extensión de Python (el autor es el propio Microsoft)



- Cambiaremos el shell por defecto. Para ello, abrid la paleta de comandos (Ctrl - Shift- P), y teclad "Terminal Select shell". VS Code lo completará, escoged la consola estándar.



Ahora pasamos a crear un “workspace”

Workspace en VS Code

Un workspace in VS Code es el contenedor para un proyecto, incluye todos los ficheros y carpetas relevantes.

Muchas de las opciones de configuración de VS Code se pueden establecer a nivel de workspace.

Workspace en VS Code

Un workspace in VS Code es el contenedor para un proyecto, incluye todos los ficheros y carpetas relevantes.

Muchas de las opciones de configuración de VS Code se pueden establecer a nivel de workspace.

Pasos:

- 1 En algún lugar en vuestro ordenador, cread la carpeta `ids`
- 2 Desde VS Code, escoged `Add Folder to Workspace`. Esto se puede hacer o bien desde el menú `File` o usando la paleta de comandos (`Ctrl - Shift- P`) introduciendo “Add folder”.

Workspace en VS Code

Un workspace in VS Code es el contenedor para un proyecto, incluye todos los ficheros y carpetas relevantes.

Muchas de las opciones de configuración de VS Code se pueden establecer a nivel de workspace.

Pasos:

- 1 En algún lugar en vuestro ordenador, cread la carpeta `ids`
- 2 Desde VS Code, escoged `Add Folder to Workspace`. Esto se puede hacer o bien desde el menú `File` o usando la paleta de comandos (`Ctrl - Shift- P`) introduciendo "Add folder".

Nota:

Todos los ficheros, programas, datos y figuras asociados a las prácticas de esta asignatura se ubicarán en esta carpeta.

Workspace en VS Code

Un workspace in VS Code es el contenedor para un proyecto, incluye todos los ficheros y carpetas relevantes.

Muchas de las opciones de configuración de VS Code se pueden establecer a nivel de workspace.

Pasos:

- 1 En algún lugar en vuestro ordenador, cread la carpeta `ids`
- 2 Desde VS Code, escoged `Add Folder to Workspace`. Esto se puede hacer o bien desde el menú `File` o usando la paleta de comandos (`Ctrl - Shift - P`) introduciendo "Add folder".

Nota:

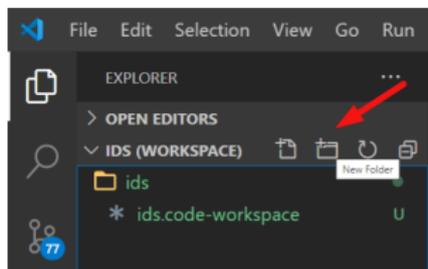
Todos los ficheros, programas, datos y figuras asociados a las prácticas de esta asignatura se ubicarán en esta carpeta.

Comprobad, usando el icono del explorador  en la barra lateral, que un workspace sin título se ha creado y que contiene la carpeta `ids`.

Podéis guardar el workspace ("`Save Workspace as`" en el menú `File`), con el nombre `ids`.

Creamos nuestro primer proyecto Python

Usando el explorador en la barra lateral, hacemos click en la carpeta `ids` y creamos una nueva subcarpeta que llamaremos `src`.



También creamos una subcarpeta llamada `data` y otra llamada `doc`. La estructura de la carpeta `ids` es por lo tanto:

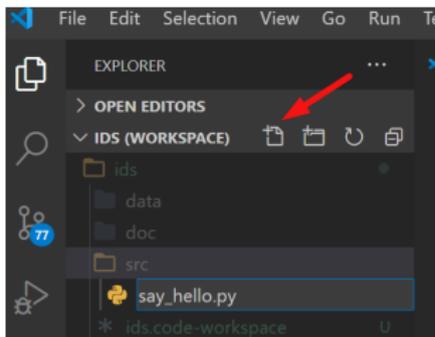
```
ids
  |__ src
  |__ data
  |__ doc
```

Nota

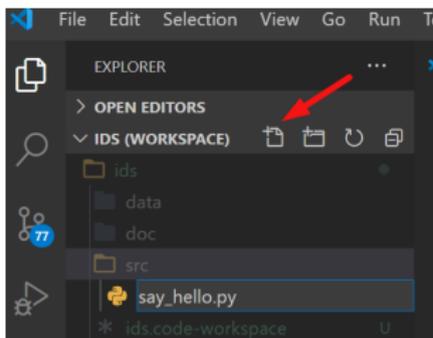
- Todos nuestros programas `.py` se ubicarán en la carpeta `src` mientras que los blocs de notas estarán en la carpeta `doc`.
- Todos los ficheros de datos que leeremos desde Python se ubicarán en la carpeta `data`.

Creamos nuestro primer programa Python

Usando el Explorador en la barra lateral, haced click en la carpeta src y cread el fichero `say_hello.py`.



Usando el Explorador en la barra lateral, haced click en la carpeta `src` y cread el fichero `say_hello.py`.



La estructura de la carpeta `ids` es por lo tanto:

```
ids
|__ src
|    |__ say_hello.py
|
|__ data
|__ doc
```

Creamos nuestro primer program Python

En la primer línea de `say_hello.py`, introducimos

```
1     print('Hello world')
```

Guardad el fichero. Personalmente tengo la opción de auto-guardado activada (Menú File).

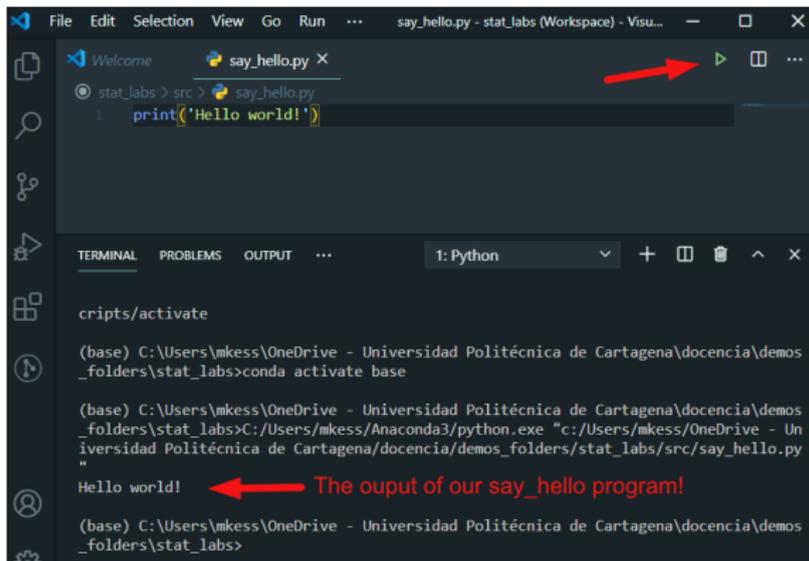
Creamos nuestro primer program Python

En la primer línea de `say_hello.py`, introducimos

```
1 print('Hello world')
```

Guardad el fichero. Personalmente tengo la opción de auto-guardado activada (Menú File).

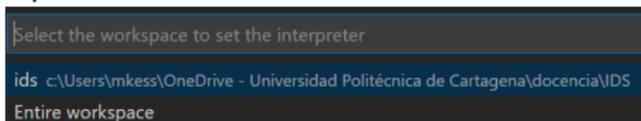
Para ejecutar el programa, podemos hacer click en el botón de “Play” en la esquina superior derecha, o desde la paleta de comandos, invocar *Python: Run Python file in terminal*.



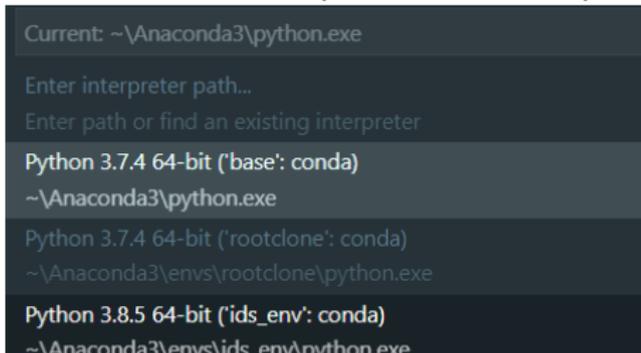
- Es una buena práctica trabajar en un entorno virtual. Lo habéis tenido que crear ya, siguiendo el documento “Crear un entorno virtual con conda”.
- Para cambiarlo para un workspace, podéis hacer click en la versión Python de la parte inferior izquierda de VS Code o usando la paleta de comandos (Ctrl-Shift-P), introduciendo `Select Interpreter`



Escoged el workspace entero:



y seleccionad el entorno virtual `ids` que habéis creado previamente,



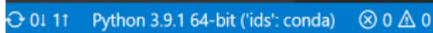
Para comprobar que se ha seleccionado correctamente el entorno

Podéis comprobar que el entorno `ids` está activado en la parte izquierda de la barra inferior:

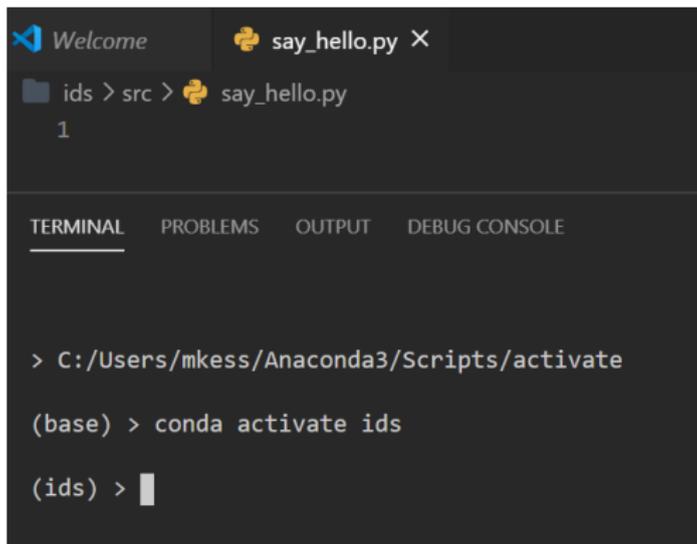


Para comprobar que se ha seleccionado correctamente el entorno

Podéis comprobar que el entorno `ids` está activado en la parte izquierda de la barra inferior:



O podéis abrir un nuevo terminal (Terminal menu ← New Terminal), y podéis comprobar que conda activa el entorno `ids` :



```
Welcome say_hello.py X
ids > src > say_hello.py
1

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE

> C:/Users/mkess/Anaconda3/Scripts/activate

(base) > conda activate ids

(ids) > |
```