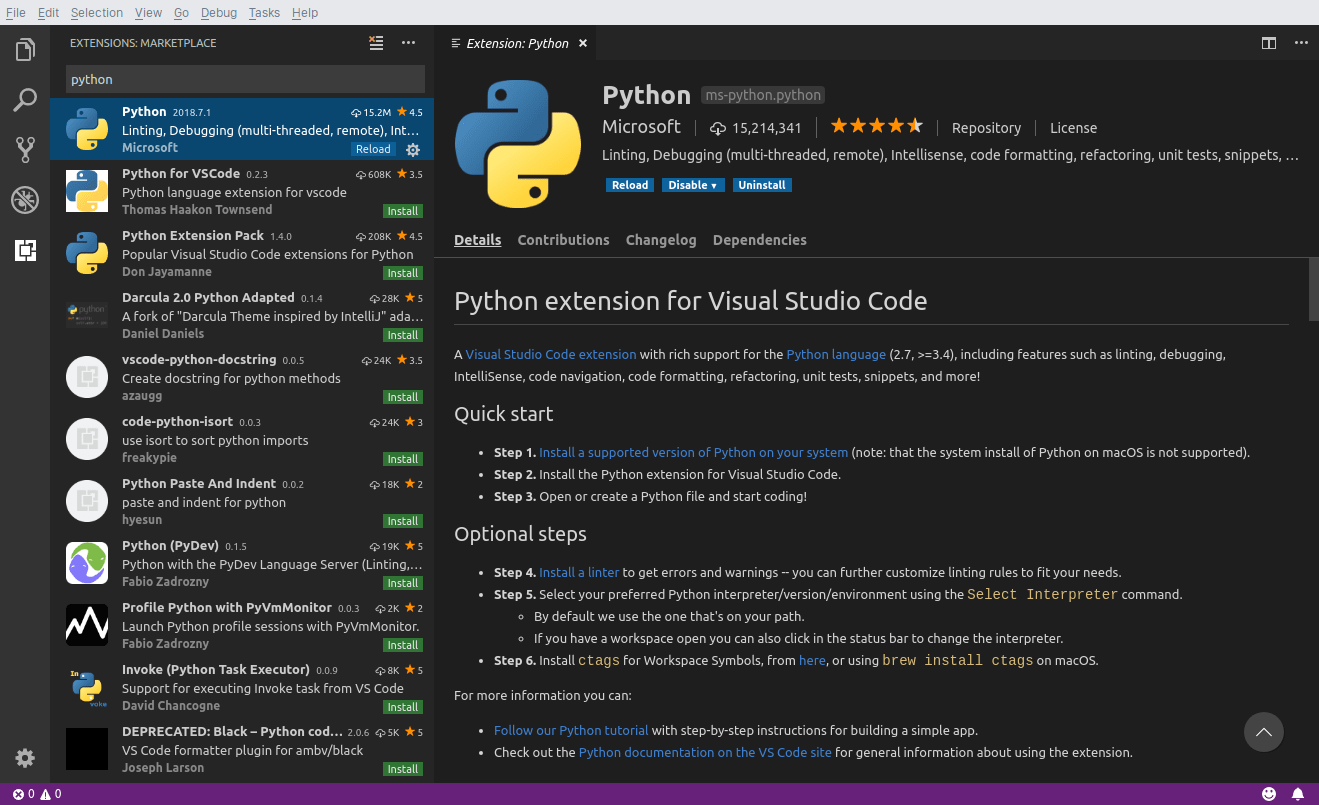
**Cómo configurar Visual Studio Code para programar en Python**

## **Instalar la extensión de Python para Visual Studio Code**

Lo primero que debemos hacer es **instalar la extensión de Python para Visual Studio Code**. Para ello, hacemos clic en el menú de extensiones a la izquierda y tecleamos 'Python'. **Seleccionamos la de Microsoft** y la instalamos.



## **Configurar la extensión de Python en Visual Studio Code**

Ahora que ya tenemos instalada la extensión de Python, podemos pasar a configurarla. Visual Studio Code gestiona las configuraciones mediante archivos **JSON**y se utilizan principalmente dos: un archivo para la configuración global y otro archivo específico para las configuraciones de proyectos.

Para abrir la configuración global podemos utilizar el atajo Control+, (coma) o navegar por '**Archivo>Preferencias>Ajustes**'. Vamos a añadir el siguiente fragmento:

{  
    "python.pythonPath": "/usr/bin/python3"  
}

En este caso, he añadido /usr/bin/python3 porque es la ruta en la que se encuentra mi ejecutable de Python, pero vosotros tendréis que añadir la ruta en la que lo hayáis instalado.

## **Linters de código**

La extensión de Python soporta varios linters de código entre los que se encuentran pep8, flake8 y pylint. El linter por defecto es pylint, pero es posible cambiarlo por ejemplo por Flake8. Para ello, vamos a instalarlo:

pip install flake8

Tal y como se menciona en la [documentación de Flake8](http://flake8.pycqa.org/en/latest/), la configuración global se almacena en ~/.config/flake8 en los sistemas operativos MacOS y Linux, pero también podemos especificar la configuración desde Visual Studio Code:

"python.linting.pylintEnabled": false,   
"python.linting.flake8Path": "/usr/local/bin/flake8",   
"python.linting.flake8Enabled": true,   
"python.linting.flake8Args": ["--max-line-length=90"],

Hay muchos parámetros que podemos pasarle a Flake8 y todos ellos podemos verlos en su [documentación](http://flake8.pycqa.org/en/latest/user/invocation.html).

## **Formateo de código**

Con Visual Studio Code también podemos configurar formateo de código automático. La extensión de Python soporta por defecto autopep8, black y yapf. Vamos a ver cómo podríamos configurar black:

"python.formatting.provider": "black",

"python.formatting.blackPath": "/usr/local/bin/black",   
"python.formatting.blackArgs": ["--line-length=90"],   
"editor.formatOnSave": true,

En este caso **hemos configurado VS Code para que formatee el código cuando guardemos el archivo**, pero si quisiéramos quitar esta opción solo habría que establecerla a false.

## **Tests unitarios**

Muchos **no podemos vivir sin tests unitarios** y, por suerte, la extensión de Visual Studio Code soporta por defecto pytest, unittest y nosetest. Una vez hayas habilitado los tests unitarios en tu proyecto, VS Code los detectará solo y podrás ejecutarlos. Para habilitar pytest añadiríamos lo siguiente a la configuración:

"python.unitTest.pyTestEnabled": true,

"python.unitTest.pyTestPath": "${workspaceRoot}/.venv/bin/pytest",