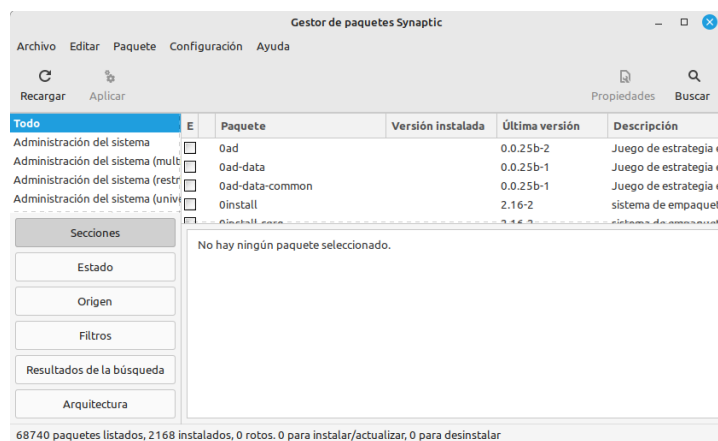


# Preparar SublimeText 4 – Linux Mint

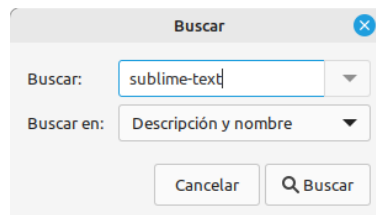
Sublime es un editor de texto de código libre que admite numerosas modificaciones. En este tutorial transformaremos Sublime en un ligero y eficaz IDE con el que programar en Python utilizando un Linux (Mint) como sistema operativo.

## 1. Descarga e instalación

Abre el **gestor de paquetes synaptic**. Tras identificarte con tus credenciales de administrador se abrirá la ventana principal:



Haz clic en el **botón Buscar** y escribe la expresión **sublime-text**:



Se mostrará en la ventana de la derecha el paquete sublime-text. Haz clic sobre su nombre con el clic derecho del ratón y selecciona la opción **marcar para instalar**. Y a continuación haz clic en el **botón Aplicar**. Se abrirá una nueva ventana informando de los detalles de la instalación. Haz clic en el **botón Aplicar**. Al terminar la instalación ya podrás ejecutar SublimeTex.

En el caso de Linux Mint, se habrá creado (si todavía no existía) un grupo nuevo de programas en el **menú de inicio** llamado **programación**.

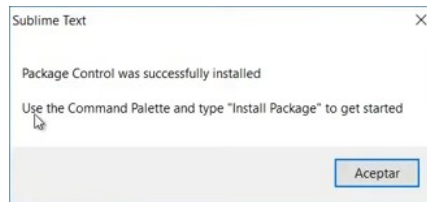
## 2. Configuración sencilla Python

Terminada la instalación configuraremos varias opciones para optimizar su funcionamiento.

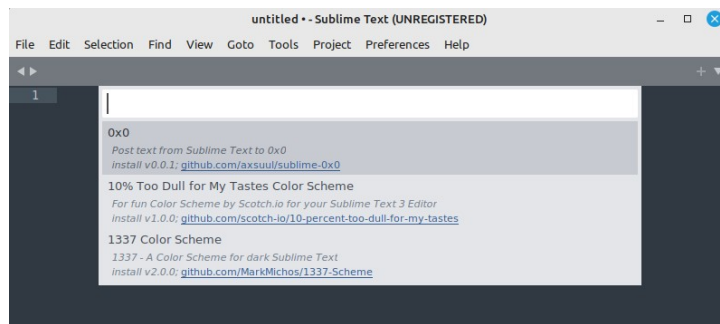
Al arrancar el programa selecciona en el menú superior: **Tools > Command Palette...**

Se abrirá la ventana que permite introducir comandos. Escribe **install Package Control**. Esto instalará el gestor de paquetes de Sublime, que permite buscar, instalar y mantener actualizados componentes a Sublime mejorando su funcionalidad.

Al terminar se mostrará:

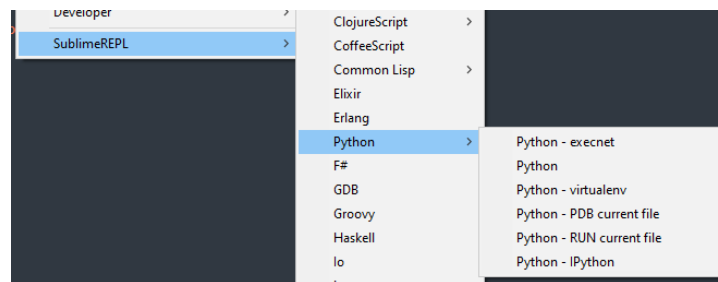


La ventana indica que debes volver a seleccionar en el menú **Tools** la opción **Command Palette...** y teclear ahora la expresión **“Install Package”**. Con ello se abrirá la ventana que permite instalar paquetes:



En ella escribe `sublimeREPL`. Esto instalará el plugin de ese nombre, que es interprete de Python (y de otros lenguajes).

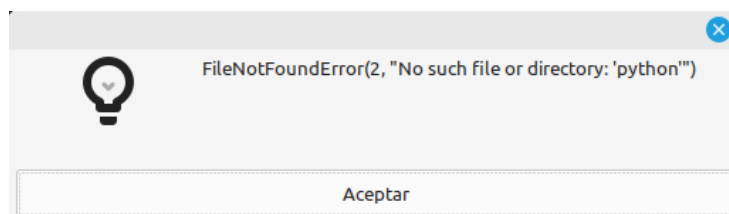
Una vez terminada la instalación, el menú **Tools** mostrará un nuevo elemento: **SublimeREPL** y dentro de él, la opción **Python**. A su vez dentro de ella verás varias opciones:



Elegimos la opción **Python**. Se mostrará una ventana de edición. Ya podemos trabajar.

## 2.1 Posible error en Linux

Es posible que en linux, al elegir python se muestre un error:



En ese caso debes seguir estos pasos:

Abre una ventana de tu **administrador de archivos**, situate en la carpeta correspondiente a **tu usuario** dentro del **directorio home** (donde se guardan todos tus archivos) y pulsa la combinación de teclas **ctrl+h** para mostrar los **ficheros ocultos** (su nombre comienza por un punto).

En mi caso, mi nombre de usuario es `joses`, por lo tanto se muestra:



Entra dentro de la carpeta **.config**, ahora en la carpeta **sublime-text > Packages > SublimeREPL > config > Python** y abre el archivo **Main.sublime-menu**. Selecciona su contenido y borrarlo.

En su lugar pega el código:

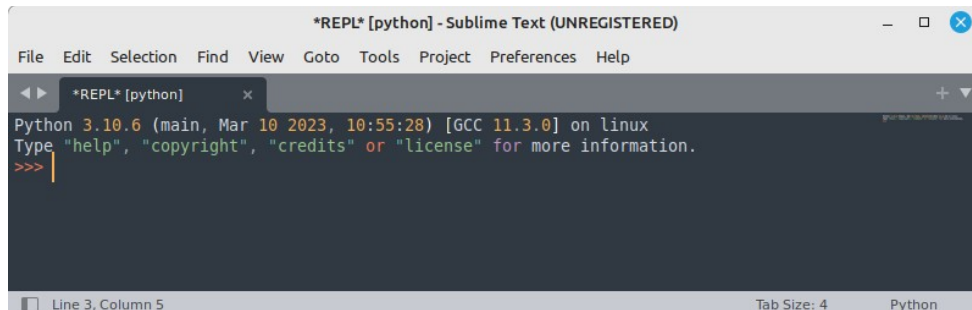
```
[
  {
    "id": "tools",
    "children":
    [
      {
        "caption": "SublimeREPL",
        "mnemonic": "r",
        "id": "SublimeREPL",
        "children":
        [
          {
            "caption": "Python",
            "id": "Python",
            "children":
            [
              {
                "command": "repl_open",
                "caption": "Python",
                "id": "repl_python",
                "mnemonic": "p",
                "args": {
                  "type": "subprocess",
                  "encoding": "utf8",
                  "cmd": ["python3", "-i", "-u"],
                  "cwd": "$file_path",
                  "syntax": "Packages/Python/Python.tmLanguage",
                  "external_id": "python",
                  "extend_env": {"PYTHONIOENCODING": "utf-8"}
                }
              },
              {
                "command": "python_virtualenv_repl",
                "id": "python_virtualenv_repl",
                "caption": "Python - virtualenv"},
              {
                "command": "repl_open",
                "caption": "Python - PDB current file",
                "id": "repl_python_pdb",
                "mnemonic": "d",
```

```
"args": {
  "type": "subprocess",
  "encoding": "utf8",
  "cmd": ["python3", "-i", "-u", "-m", "pdb", "$file_basename"],
  "cwd": "$file_path",
  "syntax": "Packages/Python/Python.tmLanguage",
  "external_id": "python",
  "extend_env": {"PYTHONIOENCODING": "utf-8"}
},
{"command": "repl_open",
 "caption": "Python - RUN current file",
 "id": "repl_python_run",
 "mnemonic": "d",
 "args": {
  "type": "subprocess",
  "encoding": "utf8",
  "cmd": ["python3", "-u", "$file_basename"],
  "cwd": "$file_path",
  "syntax": "Packages/Python/Python.tmLanguage",
  "external_id": "python",
  "extend_env": {"PYTHONIOENCODING": "utf-8"}
},
{"command": "repl_open",
 "caption": "Python - IPython",
 "id": "repl_python_ipython",
 "mnemonic": "p",
 "args": {
  "type": "subprocess",
  "encoding": "utf8",
  "autocomplete_server": true,
  "cmd": {
    "osx": ["python", "-u",
"${packages}/SublimeREPL/config/Python/ipy_repl.py"],
    "linux": ["python3", "-u",
"${packages}/SublimeREPL/config/Python/ipy_repl.py"],
    "windows": ["python", "-u",
"${packages}/SublimeREPL/config/Python/ipy_repl.py"]
  },
  "cwd": "$file_path",
  "syntax": "Packages/Python/Python.tmLanguage",
  "external_id": "python",
  "extend_env": {
    "PYTHONIOENCODING": "utf-8",
    "SUBLIMEREPL_EDITOR": "$editor"
  }
}
}
]]
]
```

```
}  
}  
]
```

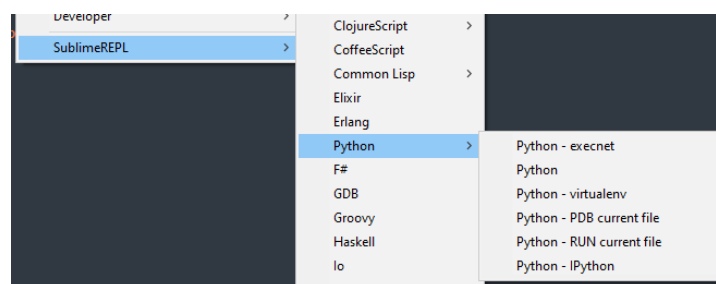
y guarda el archivo.

Inicia de nuevo SublimeText, comprueba que el error no aparece y se abre una consola Python:



Error resuelto gracias al tutorial: <https://www.youtube.com/watch?v=vFq95boV7R0>

### 3. Trabajar con un archivo .py



Elegimos la opción **Python**. Se mostrará una ventana de edición. Ya podemos trabajar.

### 4. Trabajar con un archivo .py

En primer lugar, de cara a ser ordenado, crea en tu disco duro una carpeta en la que guardar tus futuros proyectos y archivos. Abre en Sublime dicha carpeta y en su interior crear un archivo con extensión .py. Para guardar, abrir... archivos será suficiente con utilizar el menú File como en cualquier otra aplicación.

A modo de práctica crea una carpeta en tu disco duro e intenta guardar en su interior un módulo al que llamarás ensayo.py.

Como código de esta aplicación puedes escribir un clásico de la programación:

```
print("Hello World!!!!)
```

Para poder ejecutar cualquier módulo .py, selecciona en el menú superior la opción **Tools**, **Build System** y comprueba que tienes seleccionado el lenguaje **Python** o la opción **Automatic**.

A continuación dentro del menú **Tools** selecciona **Build** o utiliza la combinación de teclas **Ctrl+B**.

Sin embargo esta forma de trabajar tiene un problema importante:

**Pega de Sublime.** Solamente permite introducir información por teclado (input) si utilizamos la consola, para eso hay que ir a Tools, SublimeREPL, Python, Python – Run current file.

**Este es un proceso engorroso.** La solución es asignar a esa acción una combinación de teclas. Para ello en el menú superior seleccionamos **Preferences > Key Bindings**.

En la zona de la derecha y entre los corchetes pega el código:

```
{ "keys": ["ctrl+alt+b"], "command": "run_existing_window_command", "args":  
  {  
    "id": "repl_python_run",  
    "file": "config/Python/Main.sublime-menu"  
  }  
}
```

A continuación, guarda el documento.

Con esto la combinación de teclas Ctrl+Alt+b ejecutará la aplicación en la consola (normalmente en una pestaña nueva). Puedes poner la combinación de teclas que quieras (intenta no sobrescribir alguna de las ya existentes).

Otra combinación útil es Ctrl+w para cerrar la consola (la ventana activa).

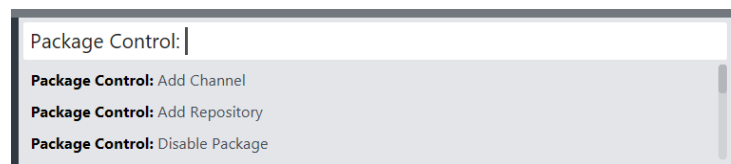
## 5. Utilizar la consola de Terminus

La opción anterior es muy sencilla de instalar, pero tiene dos pegas:

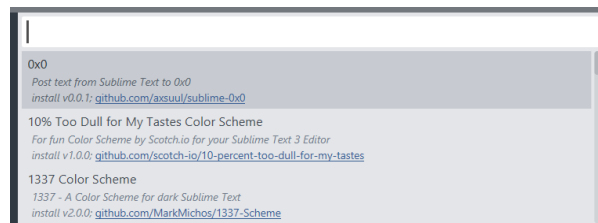
1. Cada vez que ejecutamos una aplicación, el resultado se muestra en una ventana emergente, al final se van acumulando y todo queda muy confuso.
2. La tecla enter del teclado numérico no funciona.

Vas a instalar y utilizar otra consola más eficiente instalando el paquete Terminus.

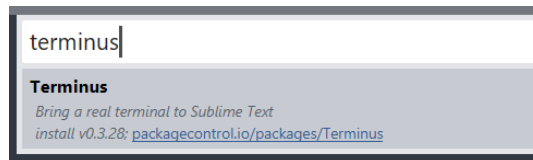
En primer lugar instalarás Terminus, para ello selecciona en el **menú superior > Preferences > Package Control**:



Teclea **install package**, se abrirá la ventana de instalación:



Escribe **terminus** y acepta:



Se iniciará la instalación.

Inicio el proceso de configuración siguiendo lo descrito en el tutorial:

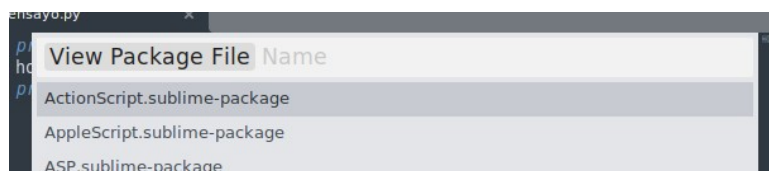
<https://www.youtube.com/watch?v=etIJMVivVgg>

Una vez terminada la instalación, debes crear un nuevo Build System, basado en el que viene por defecto en Sublime para compilar Python.

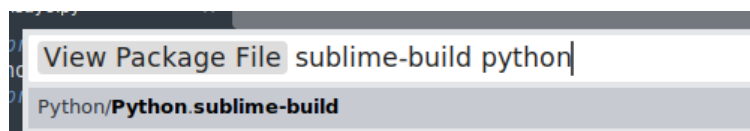
**Build system:** Conjunto de configuraciones y comandos que permite compilar, ejecutar o realizar otras tareas relacionadas con el desarrollo de software.

Para ello abrimos la paleta de comandos ("Command Palette"), **Menú principal > Tools > Command Palette...**

Escribimos **vpf** (View Package File) y hacemos enter. Esto nos permitirá mostrar todos los recursos y paquetes que tengamos funcionando en nuestro SublimeText:



Estamos buscando el build system que viene por defecto para hacer funcionar Python en Sublime. Para filtrar y que solo se muestren los build systems escribimos sublime-build y como queremos que nos muestre los build system de python añadimos la palabra python al filtro. Se mostrará y seleccionaremos Python.sublime-build.



Se abrirá el archivo. Al ser un archivo nativo no se puede editar.

En mi caso el código es:

```
{
  "cmd": ["python3", "-u", "$file"],
  "file_regex": "^[ ]*File \\(...*?\\)", line ([0-9]*)",
  "selector": "source.python",

  "env": {"PYTHONIOENCODING": "utf-8"},

  "windows": {
    "cmd": ["py", "-u", "$file"],
  },

  "variants":
  [
    {
```

```
        "name": "Syntax Check",
        "cmd": ["python3", "-m", "py_compile", "$file"],

        "windows": {
            "cmd": ["py", "-m", "py_compile", "$file"],
        }
    }
}
}
```

Selecciono el contenido, lo copiamos y vamos a al menú **Tools> Build system > New build System...**

Pegamos (eliminamos lo que haya) y lo guardamos en la carpeta que nos ofrecerá sublime. Le asignamos un nombre que luego será el que aparecerá en la lista de build systems. Le asigno el nombre

```
Python Terminus.sublime-build
```

Insertamos tras el corchete inicial las siguientes líneas:

```
{
    "target": "terminus_exec",
    "cancel": "terminus_cancel_build",
    "focus": true,
    "timeit": false,
```

- La primera línea hace que Terminus le dice a Sublime que Terminus es la aplicación a utilizar.
- La segunda línea le dice a sublime como detener la ejecución de este build.
- La tercera lleva el foco de la ejecución a la consola.
- La cuarta hace que no se muestre el tiempo de ejecución.

Para ejecutar un archivo .py bastará con ir al menú Tools> Build with... y seleccionamos la opción Python Terminus.

A partir de ese momento ya podremos ejecutar un .py simplemente haciendo Tools> Build.

En el tutorial también se cuenta otras más opciones.... Consola en una pestaña... o ventana dividida.

### Configurando Command Palette:

Selecciona menú superior: **Preferences> Package Setting> Terminus> Command Palette:**

En la ventana de la derecha (que se llamará Default.sublime-commands escribe:

```
[
  {
    "caption": "Terminal (panel)",
    "command": "terminus_open",
    "args" : {
      "cmd": "bash",
      "cwd": "${file_path:${folder}}",
      "title": "Command Prompt",
      "panel_name": "Terminus"
    }
  }
]
```



```
},  
]
```

Guarda este archivo y ciérralo.

### Configurando key bindings:

Definimos las combinaciones de teclas a utilizar como atajos. El siguiente código permitirá:

Combinación de teclas	Acción
ctrl+alt+b (originalmente es alt+n)*	Ejecuta la aplicación
alt+a	Muestra el panel
alt+t	Abre la consola
alt+,	Cierra la consola
crtl+alt+m	Muestra/oculta el menú superior.

\* He cambiado alt+n por Ctrl+alt+b por dejar la misma combinación que tenemos con la consola sencilla, pero se pueden modificar todas estas combinaciones

Desde el menú superior selecciona: **Preferences > Package Setting >Terminus > Key Bindings**

Sustituimos el código por:

```
[  
  {  
    "keys": ["ctrl+alt+b"], "command": "build"  
  },  
  { "keys": ["alt+a"], "command": "show_panel", "args": {"panel": "console",  
"toggle": true}  
  },  
  {  
    "keys": ["alt+t"],  
    "command": "terminus_open",  
    "args": {  
      "cmd": "bash",  
      "cwd": "${file_path:${folder}}",  
      "panel_name": "Terminus"  
    }  
  },  
  {  
    "keys": ["alt+,"], "command": "terminus_close",  
    "context": [{"key": "terminus_view"}]  
  },  
  {  
    "keys": ["ctrl+alt+m"], "command": "toggle_menu"  
  }  
]
```

Arranca Sublime Text, abre **Tools> Build With...** y selecciona **Python Terminus**. Con ello la consola Terminus será la opción predefinida mientras no selecciones otra opción.

Si ahora seleccionas **Tools> Build** se ejecutará la consola. En ella se admitirán datos a través de la orden input al hacer clic con cualquiera de las dos teclas enter.

El atajo Ctrl+alt+b (tal y como hemos configurado en los Key Bindings) ejecutará la aplicación. Si quiero utilizar otra combinación puedo sustituirla por la elegida.

### Aplicar estilo a la consola de Terminus

---

Terminus permite elegir entre infinidad de estilos de colores para la consola. Selecciona la que más te guste. Simplemente hay que ir a **Tools > Command Palette...** escribir **terminus** y elegir la opción **Select Theme**. Se abrirá una lista y vamos probando. A mi personalmente me gusta el tema “solarized-dark-higher-contrast”.

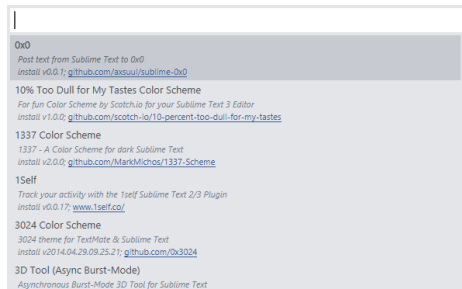
## 6. Instalar Anaconda IDE

Complemento que convierte Sublime Text en un IDE con las propiedades:

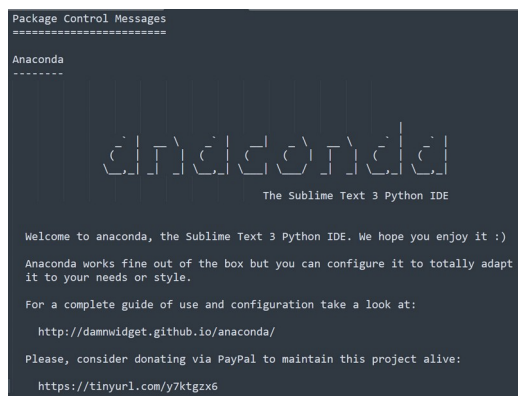
- Autocompletado de código Python.
- Muestra errores de sintaxis y PEP8 (guía de estilos de Python).
- Ofrece documentación de Python.

Para instalarlo selecciona en el menú superior **preferences > Package Control**.

Escribe la orden **Install Package**:



Escribe Anaconda en la caja de texto y acepta para iniciar la instalación. Al terminar se muestra la ventana:

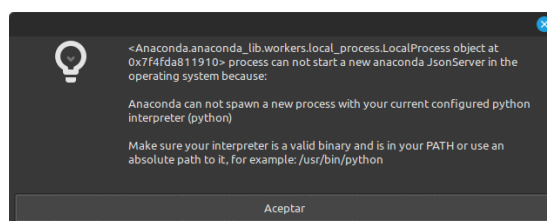


Reinicia Sublime y Anaconda estará operativo. Observa como:

- Al comenzar a escribir una palabra reservada (print...) te da opciones.
- Al escribir el nombre de un objeto muestra un desplegable con todos sus métodos.
- Muestra la información de Python sobre esa función (esta opción está activa por defecto y es incómoda, en el punto 6.4 aprenderás a desinstalar esta opción).

### 6.1 Error al instalar Anaconda en Linux

Al intentar instalar Anaconda, se genera el error:



Anaconda no encuentra la dirección en la que está instalado Python.

Aceptamos el mensaje y vamos a abrir una ventana de consola.

Para averiguar donde está instalado python 3 tecleamos (como administrador):

```
which python3
```

En mi caso me indica:

```
/usr/bin/python3
```

Necesito saber que versión es así que voy probando. En mi caso es la 3,10

```
which python3.10
```

En mi caso me indica:

```
/usr/bin/python3.10
```

Vamos a Sublime Text, seleccionamos en el menú superior: **Preferences > Package Settings > Anaconda > Settings – Default** y buscamos la línea:

```
"python_interpreter": "python",
```

Cambiamos “python” por la dirección en la que está nuestro Python

```
"python_interpreter": "/usr/bin/python3.10",
```

Guardamos el archivo y listo.

## 6.2 Errores con Anaconda

Por defecto Anaconda no muestra los errores. Para que lo haga es necesario configurar la aplicación. En el menú superior selecciona **Preferences > Package settings > Anaconda > Settings-Default**. Se mostrará una ventana de código. Pulsa **Ctrl+F** para mostrar la ventana de búsqueda y busca **anaconda\_linter\_phantoms**.

```
/*  
   If this is set to true, anaconda will show errors inline.  
*/  
"anaconda_linter_phantoms": false,
```

Cambia el valor de la propiedad de false a **true**. Guarda el archivo y listo.

## 6.3 Demasiados errores pep8 mostrados por Anaconda

La norma de estilo pep8 es muy estricta y genera muchos avisos molestos. Puedes elegir entre dos opciones:

1. **Desactivar la opción de revisión de estilo pep8** . Selecciona en el menú superior: **Preferences> Package Settings> Anaconda > Settings Default**. Busca la línea “**pep8**”; **true** y cambia la propiedad true por false.
2. **Desactivar los avisos que te molestan**. Por ejemplo, líneas de más de 80 caracteres.

Personalmente recomiendo la segunda opción ya que, aunque sea molesto en ocasiones, tener esta opción activa te recordará como escribir código según la norma de estilo correcta.

Para desactivar un aviso molesto debes seguir el siguiente procedimiento:

Lee el código asociado al aviso PEP 8, por ejemplo:

```
# Solicito el índice del elemento a mostrar
indice = int(input("Introduce el índice del elemento de la lista que quieres mostrar por pantalla:"))
[W] PEP 8 (E501): line too long (101 > 79 characters)
```

En este caso es el aviso E501. Selecciona: **Preferences > Package Settings > Anaconda > Settings Default** y busca la línea "**pep8\_ignore**". Encontraras entre corchetes una lista de errores pep8 a ignorar. Inicialmente tendrás:

```
"pep8_ignore":
[
  "E309",
  "W503"
],
```

Añade una coma, tras el último error y añade código del nuevo aviso a ignorar:

```
"pep8_ignore":
[
  "E309",
  "W503",
  "E501"
],
```

Tras guardar el archivo Anaconda no te avisará del exceso de longitud en una línea.

## 6.4 Desactivar la información sobre funciones

Por defecto cuando escribimos el nombre de una función o palabra reservada, Anaconda muestra por defecto información sobre la misma. Esto puede llegar a hacer que escribir el código sea confuso. Vas a configurar la aplicación para desactivar esta función por defecto.

En primer lugar debes seleccionar en el menú superior **Preferences > Package Settings > Anaconda > Settings Default**.

Utilizando la herramienta Find (**Menú Find > Find...**) busca la línea "**display\_signatures**" y modifica el valor de la propiedad al valor **false**:

```
"display_signatures": false,
```

A partir de ahora si quieres que Anaconda muestre la información sobre la función que acabas de escribir, tendrás que utilizar la combinación de teclas **Ctrl+Alt+D**.

## 7. Instalar SideBarEnhancements

Este complemento mejora el funcionamiento de la ventana lateral Sidebar. Para instalarlo, selecciona menú **Preferences > Package Control**. Escribe **Install Package** e instala el complemento **SideBarEnhancements**.

Una vez instalado, al hacer clic con el botón derecho del ratón sobre las carpetas y archivos mostrados en el Sidebar podremos ver todas las opciones que nos ofrece la herramienta.

## 8. Añadir iconos según tipo extensión

Puedes mejorar la apariencia de los iconos de los archivos mostrados en el sidebar de Sublime.

En el menú superior y selecciona **Preferences > Package Control** escribe **Install Package** y busca **A File Icon**, haz clic para iniciar la instalación. Se mostrarán unos iconos más agradables.

## 9. Ajustes finos

Selecciona **Menú > Preferences > Settings** y abre el archivo **Preferences.sublime-settings**.

El código al final de apartado permite:

- Para resaltar la línea en la que está situado el cursor
- Modificamos la anchura del cursor a tres puntos
- Aumentamos su altura
- Le damos un poco más de tamaño por abajo
- Interlineado aumentado
- Nombre de las carpetas en negrita

```
"highlight_line": true,  
  "caret_extra_width":3,  
  "caret_extra_top":2,  
  "caret_extra_bottom":2,  
  "line_padding_bottom":1,  
  "line_padding_top":1,  
  "bold_folder_labels":true,
```

Investiga por tu cuenta y realiza las modificaciones que más te gusten.