

Instalar Thonny – Linux Mint

Thonny es un entorno de programación Python sencillo, ordenado y funcional para estudiantes que están comenzando a programar.

1 Instalación de Thonny (Linux)

Abre el terminal y escribe:

```
sudo apt update  
sudo apt install thonny
```

Una vez instalado, puedes abrir Thonny desde el menú de aplicaciones o escribiendo `thonny` en la terminal

2 Instalación de Thonny (Windows)

2.1 Descargar Thonny

- Ve al sitio web oficial de Thonny: <https://thonny.org>.
- Haz clic en el botón de descarga para **Windows**. Esto descargará el instalador adecuado para tu versión de Windows.

2.2 Ejecutar el Instalador

- Una vez que se haya descargado el archivo `.exe`, haz doble clic sobre él para ejecutar el instalador.
- Si ves un mensaje de advertencia de seguridad, selecciona **Ejecutar**.
- El instalador te guiará a través de los pasos de instalación. Asegúrate de dejar las opciones predeterminadas si no tienes preferencias específicas.
- Después de que la instalación se complete, puedes hacer clic en **Finish**.
- Ahora puedes abrir Thonny desde el menú de inicio de Windows.

3 Configurar la interfaz (en inglés)

Justificación del idioma:

Aunque sea un curso en español, usar la interfaz en **inglés** desde el inicio te acostumbra a manejar el vocabulario técnico en el que normalmente se encontrará la documentación en foros, librerías y entornos profesionales. Esto facilitará el aprendizaje a largo plazo.

3.1 Cambiar el idioma a inglés

- Menú **Tools** → **Options** → **General** → **Language**. Selecciona **English** y reinicia Thonny.

2.2 Activar las vistas necesarias

Acción	Menú	Resultado
Mostrar Árbol de archivos	View → Files	Panel izquierdo con carpetas y archivos
Mostrar Variables	View → Variables	Panel derecho que muestra las variables al ejecutar/debuggear
Mostrar Consola (Shell)	View → Shell	Consola en la parte inferior
Mostrar Números de línea	View → Line numbers	Números en la izquierda del editor

3.2 Mejorar la visibilidad

- **Tools** → **Options** → **Editor**
 - ✓ Activar "**Save before running**" (guarda automáticamente antes de ejecutar)
 - ✓ Cambiar **tamaño de fuente** (12 o 14 pt según la pantalla)
- **Tools** → **Options** → **Font & Theme**
 - ✓ Elegir tema claro (Light) o tema oscuro (Dark), según preferencia

4 Primer programa de prueba

1. Crear un nuevo archivo (**File** → **New**)
2. Guardarlo en una carpeta de trabajo (**File** → **Save As...**)
3. Escribir el siguiente código:

```
a = 5
b = 7
suma = a + b
print(suma)
```

4. Ejecutar con **Run** → **Run current script** (o pulsar **F5**)

En la consola (Shell) aparecerá 12 como resultado.

5 Usar el depurador visual (Debugger)

El depurador permite ver paso a paso cómo se ejecuta el código y cómo cambian las variables.

1. Cargar el mismo programa anterior
2. Ejecutar con **Run** → **Debug current script** (icono del escarabajo  o pulsar **Ctrl + F5**)
3. Usar los botones **Step over** y **Step into** para avanzar

4. Observar cómo las variables **a**, **b** y **suma** aparecen en la ventana **Variables** con sus valores

6 Organizar los proyectos

Recomendación para clase:

- Crear una carpeta por tema/proyecto (por ejemplo, **tema1**, **tema2**)
- Guardar siempre los scripts dentro de la carpeta correspondiente
- Abrir y navegar las carpetas usando el panel **Files** (árbol de archivos)

7 Resumen de la interfaz ideal

Elemento	Ubicación
Árbol de archivos (Files)	Izquierda
Código (Editor)	Centro
Consola (Shell)	Abajo
Variables (Variables)	Derecha
Números de línea	Activados en el editor

8 Consejos para clase

- Acostumbrarse a utilizar **comentarios** desde el principio (**# comentario**)
- **Guardar frecuentemente** y usar **nombres descriptivos** para los archivos
- **Comprobar el valor de las variables** usando el depurador visual
- Recordar: **trabajar siempre en inglés** para aprender vocabulario técnico útil