

Entornos de Programación para Python



Septiembre 2025

Juan Romero Sánchez

IES Trassierra



Introducción

- Python es un lenguaje versátil y ampliamente utilizado.
- Antes de programar, debemos conocer las opciones de entornos disponibles.

Tipos de Entornos

| Alternativas para Comenzar a Programar en Python | |
|--|--|
| Opción 1 | Intérprete y Editor de Texto Puro: Solo se instala el intérprete. Se usa un editor de texto simple (ej: Notepad, no Word). Recomendado para fanáticos de la consola o acceso remoto sin escritorio. |
| Opción 2 | Entorno de Desarrollo Integrado (IDE): La mejor opción para empezar a programar en serio. Ofrecen herramientas que facilitan el desarrollo: correctores, depuradores, escritura predictiva, etc. |
| Opción 3 | Entorno de Desarrollo Web: Permite trabajar desde un navegador. El código se ejecuta en un servidor (en la nube o localmente). Ideal para colaboración y acceso simple. Ejemplos: Replit, Jupyter Notebook, Google Colab. |
| Recomendación Inicial | Empezar con un IDE ligero y esencial como Thonny. |

IDEs Populares

PyCharm (El Gigante Profesional)

| PyCharm: El Estándar Profesional | |
|----------------------------------|--|
| Desarrollador | JetBrains |
| Características Clave | <ul style="list-style-type: none">- Soporte Avanzado: Autocompletado inteligente, refactorización, depuración robusta, integración con Git/Docker.- Análisis de Código: Detección de errores en tiempo real y sugerencias de optimización.- Personalización: Temas, <i>plugins</i> extensos, multiplataforma. |
| Ventajas (Pro) | Ideal para proyectos grandes/complexos (web, ML). Herramientas avanzadas para colaboración. |
| Desventajas (Pro) | Alto consumo de recursos (RAM/CPU). Curva de aprendizaje inicial. |
| Versiones y Precio | Community (Gratis) para aprendizaje/proyectos pequeños. Professional (Suscripción) para uso profesional. |

IDEs Populares

Visual Studio Code (VS Code): El Editor Multilenguaje

| Visual Studio Code (VS Code): El Editor Multilenguaje | |
|---|---|
| Desarrollador | Microsoft |
| Características Clave | <ul style="list-style-type: none">- Extensiones: Gran ecosistema de complementos (oficial de Python, IntelliSense, Jupyter).- Ligereza: Bajo consumo de recursos, rápido.- Integración: Git, Docker, soporte para múltiples lenguajes (ideal para proyectos mixtos). |
| Ventajas (General) | Altamente personalizable. Excelente para desarrollo colaborativo (Live Share) y DevOps. Gratuito. |
| Desventajas (General) | Requiere configuración inicial (extensiones, linters, entornos virtuales). Menos integrado nativamente que PyCharm para Python puro. |
| Recomendado | Desarrolladores que usan múltiples lenguajes o prefieren un entorno ligero y modular . VSCodium es la alternativa 100% <i>Open-Source</i> . |

IDEs Populares

Visual Studio Code (VS Code): El Editor Multilenguaje

| Visual Studio Code (VS Code): El Editor Multilenguaje | |
|---|---|
| Desarrollador | Microsoft |
| Características Clave | <ul style="list-style-type: none">- Extensiones: Gran ecosistema de complementos (oficial de Python, IntelliSense, Jupyter).- Ligereza: Bajo consumo de recursos, rápido.- Integración: Git, Docker, soporte para múltiples lenguajes (ideal para proyectos mixtos). |
| Ventajas (General) | Altamente personalizable. Excelente para desarrollo colaborativo (Live Share) y DevOps. Gratuito. |
| Desventajas (General) | Requiere configuración inicial (extensiones, linters, entornos virtuales). Menos integrado nativamente que PyCharm para Python puro. |
| Recomendado | Desarrolladores que usan múltiples lenguajes o prefieren un entorno ligero y modular . VSCodium es la alternativa 100% <i>Open-Source</i> . |

IDEs Populares

Entornos para Ciencia de Datos: Jupyter y Spyder

| Entornos para Ciencia de Datos: Jupyter y Spyder | |
|--|--|
| Jupyter Notebook / JupyterLab | <ul style="list-style-type: none">- Enfoque: Interactivo y documentado. Ideal para prototipado, análisis de datos y ML.- Características: Celdas de código y texto (Markdown), visualizaciones integradas.- Ventaja: Ideal para compartir (GitHub, Colab). |
| Spyder | <ul style="list-style-type: none">- Enfoque: Programación científica. Integrado en la distribución Anaconda.- Características: Interfaz similar a MATLAB/RStudio.- Explorador de variables en tiempo real (crucial en ciencia de datos).- Ventaja: Ligero, excelente para análisis estadístico y visualización. |
| Anaconda (Distribución) | Distribución gratuita de Python y R. Incluye Conda (gestor de paquetes/entornos) y librerías clave (NumPy, pandas). Ideal para preconfigurar entornos para ciencia de datos. |

IDEs Populares

Entornos para Principiantes y Educativos

| Opciones para Principiantes y Educación | |
|---|---|
| Thonny | <ul style="list-style-type: none">- Puede ser el ideal para Empezar el Curso- Enfoque: Diseñado para la enseñanza.- Características: Depuración sencilla (visualización paso a paso), gestión de entornos simplificada, interfaz amigable.- Recomendado: Principiantes absolutos y educadores. |
| IDLE | <ul style="list-style-type: none">- El Básico Integrado- Enfoque: Viene preinstalado con Python.- Características: Editor simple con resaltado de sintaxis y <i>shell</i> interactivo.- Uso: Pruebas rápidas o primeros pasos, carece de herramientas avanzadas. |
| Mu | <ul style="list-style-type: none">- El Editor Mínimo- Enfoque: Simple, más que un editor, menos que un IDE.- Uso: Ideal para iniciación y sistemas embebidos (micro:bit, ESP32). |

IDEs Populares

Entornos para Principiantes y Educativos

| Opciones para Principiantes y Educación | |
|---|---|
| Thonny | <ul style="list-style-type: none">- Puede ser el ideal para Empezar el Curso- Enfoque: Diseñado para la enseñanza.- Características: Depuración sencilla (visualización paso a paso), gestión de entornos simplificada, interfaz amigable.- Recomendado: Principiantes absolutos y educadores. |
| IDLE | <ul style="list-style-type: none">- El Básico Integrado- Enfoque: Viene preinstalado con Python.- Características: Editor simple con resaltado de sintaxis y <i>shell</i> interactivo.- Uso: Pruebas rápidas o primeros pasos, carece de herramientas avanzadas. |
| Mu | <ul style="list-style-type: none">- El Editor Mínimo- Enfoque: Simple, más que un editor, menos que un IDE.- Uso: Ideal para iniciación y sistemas embebidos (micro:bit, ESP32). |

IDEs Populares

Entornos de Desarrollo Web/Online

| Entornos de Desarrollo Web/Online | |
|-----------------------------------|--|
| Google Colab | <ul style="list-style-type: none">- El Notebook en la Nube de Google- Base: Adaptación y evolución de Jupyter Notebook.- Ventajas Clave: Integración con Google Drive. Soporte para GPU/TPU gratuitas (ideal para Machine Learning). Permite compartir entornos personalizados completos (código, librerías, datos).- Recomendado: Colaboración, ML, uso de infraestructura potente sin depender del equipo local. |
| Replit | <ul style="list-style-type: none">– Plataforma Multi-lenguaje en la Nube– Ventajas: Permite programar Python sin instalación local. Soporte para despliegue de aplicaciones.– Recomendado: Experimentación, proyectos rápidos. |
| Ventajas Generales | Accesibilidad (cualquier navegador), colaboración fácil, sin necesidad de configuración local. |
| Desventajas | Dependencia de conexión a internet. Menos control sobre el entorno y recursos limitados en planes gratuitos. |

Resumen y Conclusión (Recomendaciones)

| Entorno | Profesionales | No Profesionales | Casos de Uso Principales |
|-----------------------------|---|---------------------------------|---|
| PyCharm | Proyectos grandes, desarrollo web, ML | Aprendizaje avanzado | Desarrollo profesional, proyectos complejos |
| VS Code | Flujos personalizados, multi-lenguaje | Aprendizaje versátil | Desarrollo general, ciencia de datos |
| Jupyter | Ciencia de datos, ML, prototipado | Aprendizaje de datos | Análisis de datos, visualización |
| IDLE | Pruebas rápidas, scripts simples | Primeros pasos en Python | Aprendizaje básico |
| Thonny | Enseñanza, tutorías | Principiantes absolutos | Educación, aprendizaje inicial |
| Spyder | Ciencia de datos, análisis, visualización | Aprendizaje de ciencia de datos | Programación científica, análisis de datos |
| Plataformas en Línea | Prototipado, colaboración, ML (Colab) | Experimentación sin instalación | Proyectos rápidos, aprendizaje en la nube |

Resumen y Conclusión (Recomendaciones)

| Resumen y Conclusión (Recomendaciones) | |
|--|--|
| Profesionales | <ul style="list-style-type: none">- PyCharm (Professional): Desarrollo <i>full-stack</i> y proyectos complejos.- VS Code: Versatilidad, multi-lenguaje, flujos personalizados.- Jupyter/Spyder: Imprescindibles para Ciencia de Datos y Análisis. |
| No Profesionales | <ul style="list-style-type: none">- Thonny/IDLE: Ideal para principiantes absolutos.- VS Code / Plataformas Online (Colab): Buenas opciones para estudiantes que avanzan y experimentan. |
| Tendencias 2025 | Accesibilidad (cualquier navegador), colaboración fácil, sin necesidad de configuración local. |
| Desventajas | Auge de la integración de IA (Copilot) y plataformas en la nube (Colab, Replit), haciendo el desarrollo más accesible y colaborativo. |