



Fundamentos de programación

Python

Para el análisis de datos

Sesión 1

Reseña del lenguaje Python

- **Creación:** Python fue creado en 1991 por el programador Guido van Rossum.
- **Propósito:** Lenguaje de programación de alto nivel y de propósito general.
- **Diseño:** Se enfoca en la simplicidad, legibilidad y accesibilidad.

Características clave:

- **Sintaxis clara y concisa:** Más fácil de escribir y entender que otros lenguajes.
- **Ideal para principiantes:** Perfecto para aprender conceptos básicos de programación sin complicaciones.
- **Uso en equipos:** Facilita la colaboración y revisión de proyectos por su estructura accesible.

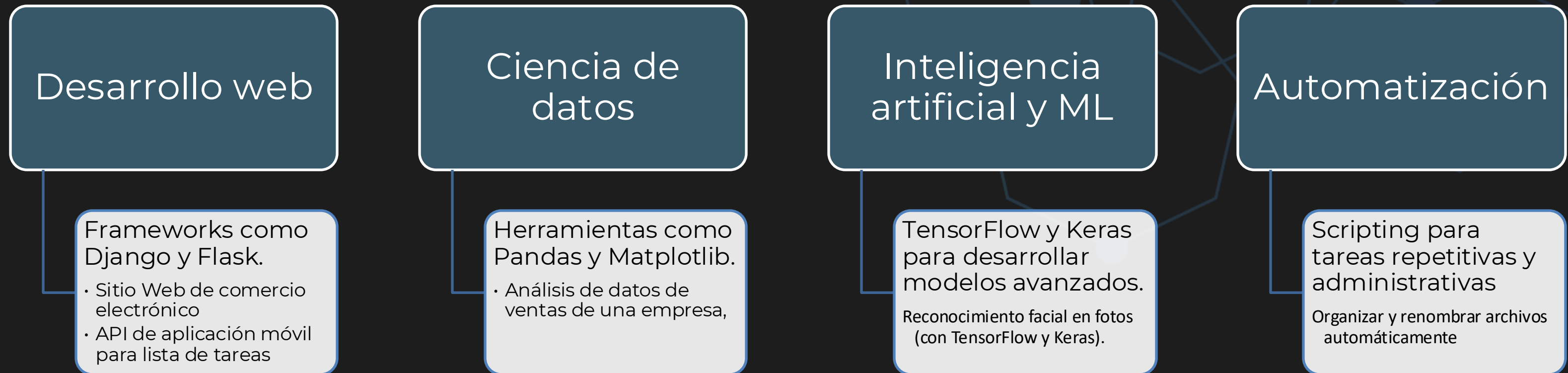
Ventajas:

- **Accesibilidad:** Atractivo tanto para principiantes como para desarrolladores experimentados.
-

Propósito y Accesibilidad de Python

- **Creación:** Diseñado para hacer la programación accesible a todos, desde principiantes hasta expertos.
 - **Facilidad de uso:** Sintaxis intuitiva y cercana al lenguaje natural, ideal para quienes comienzan a programar.
 - **Propósito general:** Python no se limita a un sector, es versátil y utilizado en múltiples áreas.
-

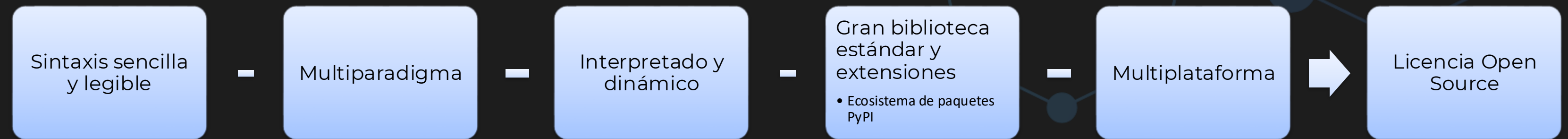
Adaptabilidad en diversas Disciplinas



Eficiencia, Productividad y Comunidad

- **Eficiencia:** Python facilita la creación rápida de prototipos y ajustes en proyectos complejos.
- **Comunidad global:** Python es **código abierto**, promoviendo la colaboración y el apoyo mutuo.
- **Recursos:** Amplia documentación y foros de ayuda disponibles para todos los niveles.

Principales características del lenguaje



Versiones de Python

Python 1.x (1994-2000)

Lanzamiento inicial: enero de 1994.

Sintaxis simple, orientación a objetos y manejo de excepciones.

Introducción de funciones clave como lambda y recolección automática de basura.

Python 2.x (2000-2020)

Introducción de list comprehensions, soporte para Unicode, y mejoras en la gestión de excepciones.

Compatibilidad con diversos módulos como xml.

Se mantuvo como versión estándar hasta 2020.

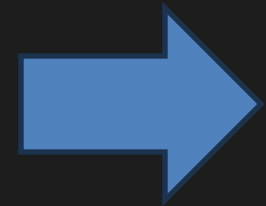
Python 3.x (2008-presente)

Lanzamiento: diciembre de 2008, con mejoras en división, manejo de cadenas Unicode y rendimiento.

Incompatibilidad con Python 2, migración gradual.

Actualizaciones continuas con mejoras en asignación, IA y ciencia de datos.

Entorno de trabajo y herramientas para Python

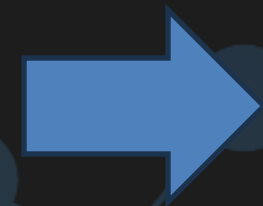


Entorno Anaconda

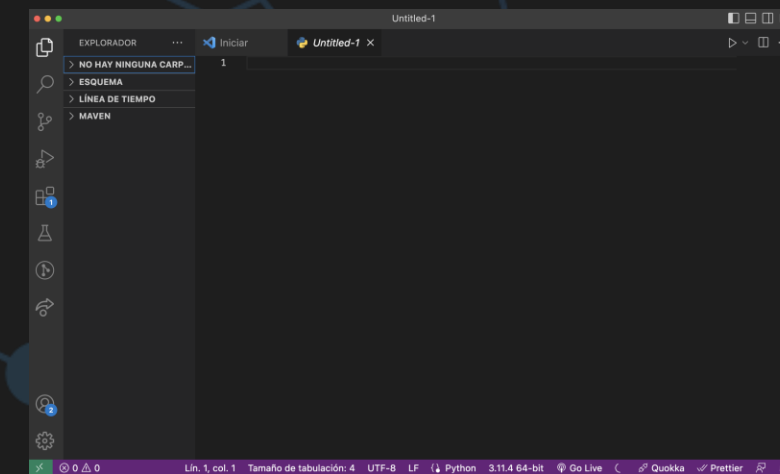


Anaconda es una de las mejores distribuciones de Python. Está orientada a dar soporte integral al desarrollo científico con Python tanto analítico como gráfico. Entre sus características destacan:

- **Aplicación gráfica** que integra las herramientas y versiones de Python orientadas al desarrollo científico.

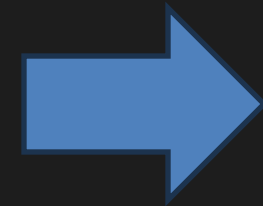


Visual Studio Code

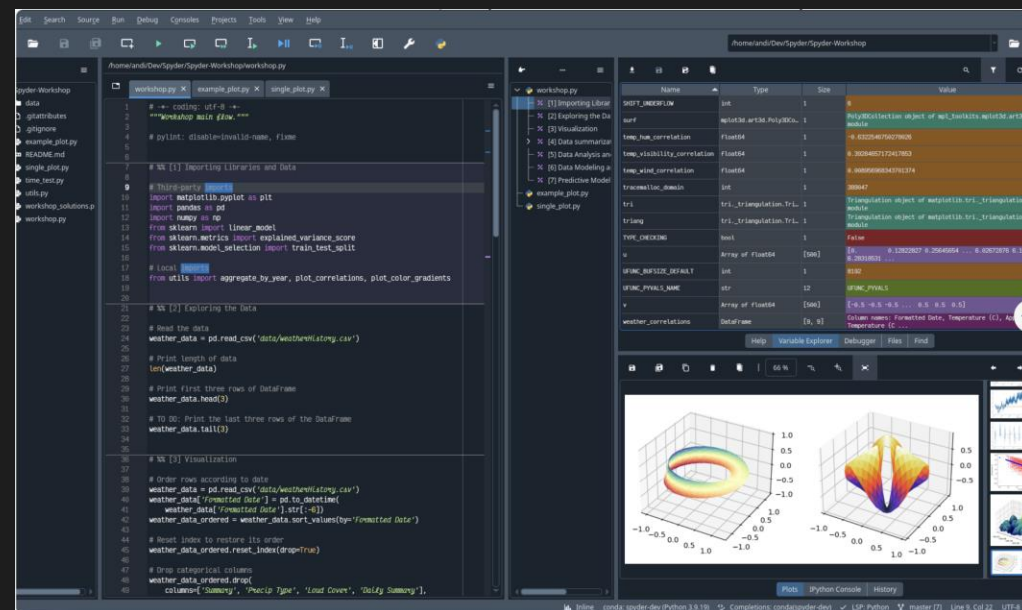


Editor de código fuente independiente que se ejecuta en Windows, macOS y Linux. La elección principal para desarrolladores web y JavaScript, con extensiones para admitir casi cualquier lenguaje de programación.

Entorno de trabajo y herramientas para Python



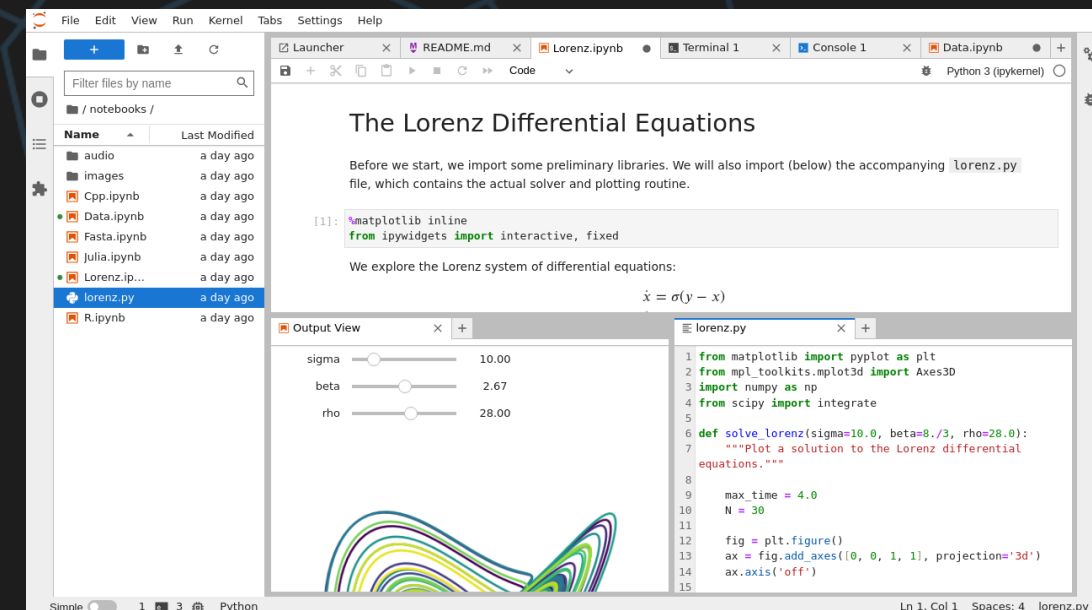
Editor Spyder



Spyder es un IDE de código abierto que combina la sencillez de un editor de código con la potencia de herramientas integradas para la computación científica en Python. Se creó para ofrecer un entorno adaptado a las necesidades de científicos, ingenieros y analistas de datos.

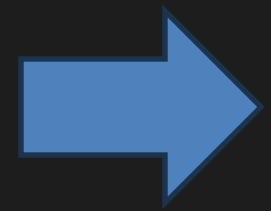


Jupyter Notebook



Jupyter Notebook es una aplicación web de código abierto que permite a los usuarios crear y compartir documentos que contienen código en vivo, ecuaciones, visualizaciones y texto narrativo. Esta herramienta es especialmente popular entre los científicos de datos, analistas y desarrolladores por su capacidad para facilitar el desarrollo, la documentación y la presentación de proyectos tanto de aprendizaje como profesionales.

Entorno de trabajo y herramientas para Python



Google Colab

```
[ ] import pandas as pd
import numpy as np

[ ] # empty dataframe
df_empty = pd.DataFrame()
df_empty

# dataframe with list
df_list = pd.DataFrame(
    [
        ["nombre_01", "apellido_01", 60],
        ["nombre_02", "apellido_02", 14]
    ],
    columns = ["nombre", "hapellido", "ehdad"]
)
df_list
```

	nombre	hapellido	ehdad
0	nombre_01	apellido_01	60
1	nombre_02	apellido_02	14

```
[ ] # dataframe with dct
df_dct = pd.DataFrame(
    {
        "nombre": ["nombre_01", "nombre_02"],
        "hapellido": ["apellido_01", "apellido_02"],
        "ehdad": [60, 14]
    }
)
```

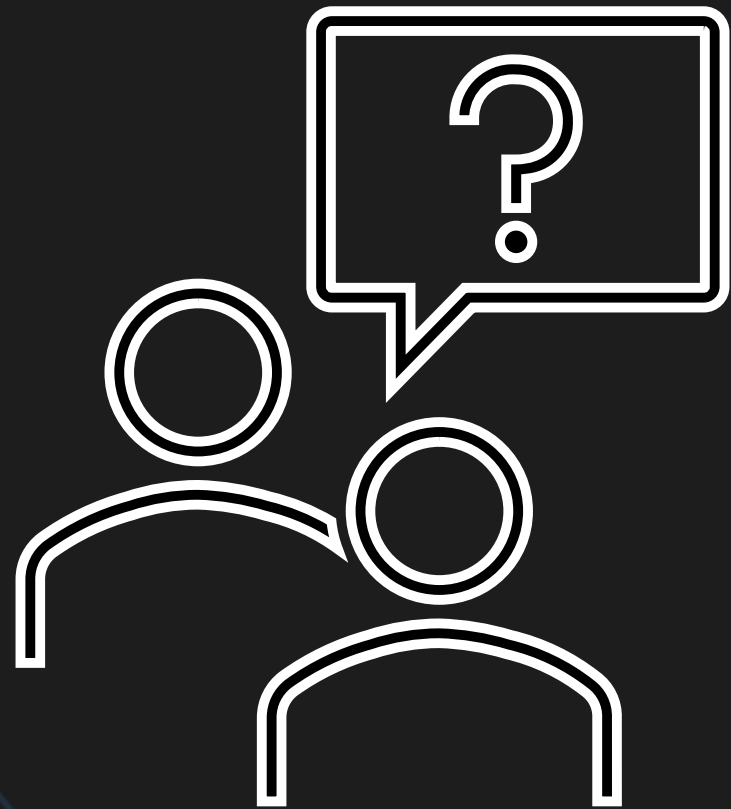
Colab, también conocido como "Colaboratory", te permite programar y ejecutar Python en tu navegador con las siguientes ventajas:

- No requiere configuración
- Acceso a GPUs sin coste adicional
- Permite compartir contenido fácilmente

Colaboratory puede facilitar tu trabajo, ya seas **alumno, científico de datos o investigador de IA**

Preguntas

Sección de preguntas





Fundamentos de programación

Python

Para el análisis de datos

Continúe con las
actividades