

# Taller de Python para ciencias de datos

Pablo Belmar Apablaza





## Un poco de mi

---

- Cofundador de TECHIND, start-up enfocada a tecnologías para la industria
- Machine Learning and Business Intelligence UDEC
- Certificado HCIA-AI
- [pabloestebanbelmar@gmail.com](mailto:pabloestebanbelmar@gmail.com)



# Temas

Entornos de desarrollo

Fundamentos del lenguaje Python

Python para Data Science

Certificaciones HUAWEI



## Entornos de desarrollo

El interprete de Python puede ser bajado para variadas plataformas de sistemas operativos desde el sitio web oficial. (<https://www.python.org/>)

Plataforma Anaconda la cual integra multiples bibliotecas desarrolladas por terceros y es ampliamente utilizada en Inteligencia Artificial y en computación científica. (<https://www.anaconda.com/>)



ANACONDA

## Advanced Installation Options

Customize how Anaconda integrates with Windows

### Advanced Options



Add Anaconda to the system PATH environment variable

Not recommended. Instead, open Anaconda with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.



Register Anaconda as the system Python 3.6

This will allow other programs, such as Python Tools for Visual Studio, PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically

# IDE's

Jupyter Notebook: ambiente de computación interactiva basada en la Web.  
<https://jupyter.org/>, <https://colab.research.google.com/>

Visual Studio Code: Editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. <https://code.visualstudio.com/>

Spyder: es un ambiente de desarrollo integrado de libre distribución multiplataforma para programación científica en lenguaje Python.  
<https://www.spyder-ide.org/>



# Fundamentos del lenguaje Python

Python es uno de los lenguajes de programación más populares y es el más ampliamente utilizado en el campo de la Inteligencia Artificial (IA)

Python es un lenguaje de programación avanzado universal y completamente de libre distribución (open-source).



# Fundamentos del lenguaje Python

---

El matemático e informático  
Holandés Guido Van Rossum creo  
Python a partir del año 1991.

Python es un lenguaje de  
programación multiparadigma  
(orientado a objeto, funcional,  
imperativo, por eventos,  
declarativa, entre otros)



## Ventajas y desventajas de Python

### Ventajas:

- Es un lenguaje interpretado y dinámico.
- Utiliza una sintaxis simple y estructuras elegante, lo cual lo hace fácil de aprender.
- Tiene una gran colección de bibliotecas desarrolladas por terceros.
- Puede invocar código escrito en otros lenguajes.

### Desventajas:

- Su velocidad de ejecución es lenta.

# Sintaxis básica

## PEP 8

---

- PEP 8: Python Enhancement Proposal 8 es una guía de estilo de como se debe escribir un código en Python, la cual ayuda a mejorar su legibilidad y elegancia. (<https://pep8.org/>)



# Librerías mas utilizadas

Numpy

Pandas



# NumPy

---

**N**umerical **P**ython ha sido por mucho tiempo un pilar de la computación numérica en Python.

---

Esta Biblioteca provee estructuras de datos, algoritmos y módulos necesarios para la mayoría de las aplicaciones científicas que involucra manipulación de datos numéricos.

---

Provee un eficiente y rápido arreglo multidimensional a través del objeto ndarray.

---

Provee funciones para realizar cálculos con arreglos u operaciones matemáticas entre arreglos utilizando la técnica element-wise que es más eficiente.

---

Operaciones de algebra lineal, trasformada de Fourier y generación de números randómicos.





# NumPy

---

Una API para C que permite a extensiones de Python y código nativos en C y C++ poder acceder a estructuras de datos NumPy y así facilitar los cálculos.

---

El principal uso que se le da a NumPy en análisis de datos es de ser un contenedor para datos que deben ser pasados entre algoritmos y bibliotecas.

---

Para trabajar con datos numéricos los arreglos de Numpy son mucho más eficientes para almacenar y manipular datos que las estructuras de datos nativas de Python.

---

Lenguajes de bajo nivel como C y Fortran pueden trabajar con datos almacenados en Numpy sin tener que copiarlos en alguna otra representación de memoria.



# Pandas

---

El nombre pandas viene de la combinación de la frase panel data que es un término que se utiliza en econometría para describir set de datos estructurados multidimensionales y de la frase Python data analysis.

---

pandas ofrece estructuras de datos y funciones de alto nivel diseñadas para trabajar con datos estructurados o tabulados de manera rápida, fácil y descriptiva.

---

Los dos objetos más utilizados en pandas son el DataFrame, que es una estructura de datos tabular orientada a las columnas donde se puede etiquetar tanto las filas como las columnas; y las Series que es un objeto arreglo unidimensional etiquetado.



# Pandas

---

pandas combina el alto rendimiento de los arreglos ofrecidos en NumPy con la flexibilidad de manipulación de datos de las planillas electrónicas, como Excel, o bases de datos relacionales, como SQL.

---

Con pandas se puede redimensionar, separar elementos en una y dos dimensiones, agregar elementos y seleccionar subconjuntos de datos.

---

Como la manipulación, preparación y limpieza de datos es una habilidad muy importante en el análisis de datos, pandas es una de las bibliotecas más utilizadas en esta área.



# IPython y Jupyter

---

El proyecto IPython comenzó el 2001 con el propósito de crear un interprete de Python mejor al nativo.

---

IPython fomenta un flujo de trabajo hacia el paradigma ejecutar-explorar más que en el típico flujo de trabajo de editar-compile-ejecutar de la mayoría de los lenguajes de programación.

---

Considerando que la mayoría de la programación en análisis de datos involucra las tareas de exploración, prueba y error e iteración. IPython puede ayudar a terminar este trabajo de manera mucho más rápida.

---

En Jupyter notebook también se puede utilizar contenido en Markdown u en HTML, lo cual permite crear documentos atractivos con código y texto.





# Etapas

---

**Interacción con el mundo real o exterior:** proceso de lectura y escritura con una variedad de formatos de archivos, medios de almacenamiento y dispositivos de entrada (por ejemplo, sensores).

**Preparación:** limpieza, manipulación, combinación, normalización, redimensionar, separación en 1 o 2 dimensiones (slicing and dicing), y transformar datos para el proceso de análisis.

**Transformación:** aplicar operaciones matemáticas o estadísticas a un grupo de datos para crear nuevos datos. Esto con el objetivos de mejorar la interpretación y procesamiento del conjunto de datos para facilitar su proceso de análisis.

# Etapas

---



**Modelado y computo:** Conectar los datos a modelos estadísticos, algoritmos de machine learning o alguna otra herramienta computacional.



**Presentación:** crear visualizaciones gráficas estáticas o dinámicas o resúmenes en texto.

# Acuerdos



---

- La comunidad de Python a adoptado un número de acuerdo en cuanto a los nombre a utilizar par los módulos importados más comúnmente utilizados.
  - `import numpy as np`
  - `import pandas as pd`
  - `import matplotlib as plt`
  - `import seaborn as sns`
  - `import statsmodel as sm`
  - `import scipy as sp`



# Acuerdos

---

Por ejemplo, `np.arange`, hace referencia a la función `arange` en NumPy.

Estos acuerdos se realizar porque desde el punto de vista del desarrollo de software en Python se considera una mala práctica el importar un módulo completo (`from numpy import *`), sobre todo para bibliotecas tan grandes como NumPy.



# Vamos!

