

A thick blue vertical bar is positioned on the left side of the slide, extending from the top to the bottom.

INTRODUCCIÓN A PYTHON

¿Qué es un Programa?

Podemos definir un programa, en su **forma más básica**, como una secuencia de **declaraciones** o **sentencias** que han sido **diseñadas** para hacer algo.

Normalmente un programa está hecho para poder **satisfacer un requerimiento (resolver un problema)**. Por ejemplo, si quisiéramos contar cuantas veces aparece la palabra "una" en la siguiente frase, ¿Como lo haríamos?

"Es una locura odiar a todas las rosas porque una te pinchó. Renunciar a todos tus sueños porque uno de ellos no se realizó."

Seguramente habrá diferentes soluciones para poder resolver esto, una podría ser **recorrer** todo el texto contando cuantas veces **aparece** esta palabra. Pero también podríamos haber recorrido el texto de fin a principio, aunque parezca anti intuitivo, en la práctica puede haber casos que necesitemos leer el texto de una **manera diferente**.

¿Cómo se construye un Programa?

Un programa se puede construir utilizando sus **elementos** más **básicos**. A estos se los llama **estructuras**.

Esas estructuras no son exclusivas de un **lenguaje de programación**, sino que forman parte de cualquiera de los mismos, desde el código máquina hasta los lenguajes de **alto nivel**.

entrada: Obtener datos del "mundo exterior". Es la materia prima de nuestros programas.

salida: Mostrar los resultados del programa. Puede haber muchas formas diferentes de mostrarlos.

ejecución secuencial: Ejecutar una sentencia tras otra como si fuera una receta.

ejecución condicional: Comprobar ciertas condiciones y luego ejecutar u omitir una secuencia de sentencias.

ejecución repetida: Ejecutar un conjunto de sentencias varias veces, normalmente con algún tipo de variación.

reutilización: Escribir un conjunto de instrucciones una vez, darles un nombre y así poder reutilizarlas luego cuando se necesiten en cualquier momento.

¿Qué es un Problema?

Un problema es un asunto o cuestión que se debe solucionar o aclarar, una contradicción o un conflicto entre lo que es y lo que debe ser, una dificultad o un inconveniente para el alcance de un fin o un disgusto, una molestia o una preocupación.

En resumen, es una **necesidad.**

Como vimos anteriormente este problema podemos resolverlo a través de **programa**, el cual puede ser ejecutado por un ser humano, o aún mejor, una máquina. Para ello tendremos que aprender **a interactuar** con la misma, y es ahí donde entran los **lenguajes de programación**.

¿Qué es Python?

Python es un *lenguaje de programación* creado por **Guido van Rossum** a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses "*Monty Python*".

Python puede ser clasificado como un lenguaje **INTERPRETADO**, de **ALTO NIVEL**, **MULTIPLATAFORMA**, de **TIPADO DINÁMICO** y **MULTIPARADIGMA**.

A diferencia de la mayoría de los lenguajes de programación, Python nos provee de reglas de estilos, a fin de poder escribir código fuente más **legible** y de manera **estandarizada**.

Pero, ¿Qué quiere decir que sea un lenguaje interpretado, de alto nivel, multiplataforma, de tipado dinámico y multiparadigma?

Lenguaje de programación

Es un lenguaje informático, diseñado para expresar **órdenes e instrucciones** precisas, que deben ser llevadas a cabo por una computadora. El mismo puede utilizarse para **crear programas** que controlen el comportamiento físico o lógico de un ordenador. Está compuesto por una serie de símbolos, reglas **sintácticas** y **semánticas** que definen la estructura del lenguaje.

La **sintaxis** de un lenguaje de programación se define como el conjunto de reglas que deben seguirse al escribir el código fuente de los programas para considerarse como correctos para ese lenguaje de programación.

Es decir "**como se escribe**".

La **semántica** en un lenguaje de programación se define como el significado desde un punto de vista matemático de la evaluación de cadenas sintácticamente legales definidas por el lenguaje de programación específico. Esto se puede mostrar describiendo la relación entre la entrada y la salida de un programa.

Es decir "**como se lee**".

Lenguajes compilados vs. Lenguajes interpretados

Par entender mejor qué es Python es un lenguaje interpretado, tenemos que saber también que es un lenguaje compilado. Es por ello que vamos a realizar una pequeña comparativa entre un compilador y un intérprete de que hace cada tipo de lenguaje. Esto nos dará una visión general de cada uno por separado. Para luego definir correctamente qué es un lenguaje Interpretado.



Diferencias entre compilador e intérprete

Compilador	Intérprete
El compilador trabaja en el programa completo a la vez. Toma todo el programa como entrada.	Un intérprete lleva los programas línea a línea. Toma una declaración a la vez como entrada.
Genera el código intermedio: código máquina o código objeto.	Un intérprete no genera un código intermedio: código máquina.
Los lenguajes compilados son más eficientes pero difíciles de depurar.	Los lenguajes interpretados son menos eficientes pero fáciles de depurar.
El compilador no permite que un programa se ejecute hasta que esté completamente libre de errores.	El intérprete ejecuta el programa desde la primera línea y detiene la ejecución sólo si encuentra un error.
Compile una sola vez y ejecútalo en cualquier momento.	Cada vez que ejecutan el programa, se interpreta línea por línea.
Los errores se informan después de que se comprueba todo el programa, los errores sintácticos y de otro tipo.	El error se informa cuando se encuentra el primer error. El resto del programa permanece sin marcar hasta que se resuelve el error.
El programa compilado ocupa más memoria porque todo el código objeto tiene que residir en la memoria.	El intérprete no genera código de memoria intermedio. Por lo tanto, los programas de intérprete son eficientes en la memoria.

Lenguajes interpretados

Entonces ahora si, definamos los lenguajes interpretados.

Los lenguajes interpretados son aquellos que, a diferencia de los compilados, no requieren de un compilador para ser ejecutados sino de un **intérprete**. Un intérprete, actúa de manera casi idéntica a un compilador, con la salvedad de que ejecuta el programa **directamente**, sin necesidad de generar previamente un **ejecutable**. Lee **línea por línea** mientras ejecuta dichas sentencias, cortando la ejecución si encuentra un **error**, ignorando el resto del código hasta solucionar dicho error.

Lenguajes de alto nivel

Son aquellos cuya característica principal, consiste en una estructura sintáctica y semántica legible, acorde a las capacidades cognitivas humanas. A diferencia de los lenguajes de bajo nivel, son independientes de la arquitectura del hardware, motivo por el cual, asumen mayor portabilidad.

Tipado dinámico

Un lenguaje de tipado dinámico es aquel cuyas variables, no requieren ser definidas asignando su tipo de datos, sino que éste, se auto-asigna en tiempo de ejecución, según el valor declarado.

Luego cuando veamos el apartado de variables, entenderemos un poco más esto.

¿Por qué Python?

Empresas que lo utilizan:



Se utiliza para: Machine Learning, Inteligencia Artificial, Análisis de datos, Desarrollo web, Video Juegos, Aplicaciones, entre otros ambientes.

Herramientas

Editor de código: Se necesita instalar un editor de código o texto, existen varios:

- Visual Studio Code: <https://code.visualstudio.com/>
- Notepad++: <https://notepad-plus-plus.org/downloads/>

Intérprete de Python: es necesario que exista en tu computadora. Podemos descargarlo e instalarlo desde:

- **Microsoft Store (en windows)**: abrir la aplicación y buscar Python, seleccionar la versión que desees.
- **Página oficial**: <https://www.python.org/downloads/>

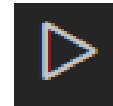
Extensiones en Visual Studio Code: ir a Extensiones, escribir Python, instalar complemento.

Existe distintas versiones de Python: **Python 2** y **Python 3** (actual). Luego existen distintas variantes en cada uno.

Otra opción: utilizar un sitio web donde puedo programar sin instalar nada, <https://replit.com/>

Primer código

Escribimos nuestro primer programa (script). En este ejemplo se utilizará Visual Studio Code (VSC):

- 1- Abrir el editor.
- 2- Crear un nuevo archivo: ir a opciones de **Archivo**. Atajo: Ctrl + N.
- 3- Guardar el archivo: ir a **Archivo** y seleccionar **Guardar**. Atajo: Ctrl + S.
- 4- Colocarle un nombre y en la opción **Tipo**, seleccionar Python. Luego presionamos en **Guardar**.
- 5- Ahora escribimos en el archivo lo siguiente: `print("Hola mundo")`
- 6- Guardamos los cambios: ir a **Archivo** y seleccionar **Guardar**. Atajo: Ctrl + S.
- 7- Presionamos en el botón en la parte superior derecha, el siguiente triángulo: 
- 8- Observamos que debajo se abre una pantalla negra, donde se observa el mensaje "Hola mundo".
- 9- La ejecución finalizó correctamente.
- 10- Hemos logrado ejecutar nuestro primer script.

También puedo hacer una operación matemática... ¿cómo? Dentro de `print(...)` escribimos `1+2`.