

Módulo 1: Introducción al Mundo de la Programación ✨

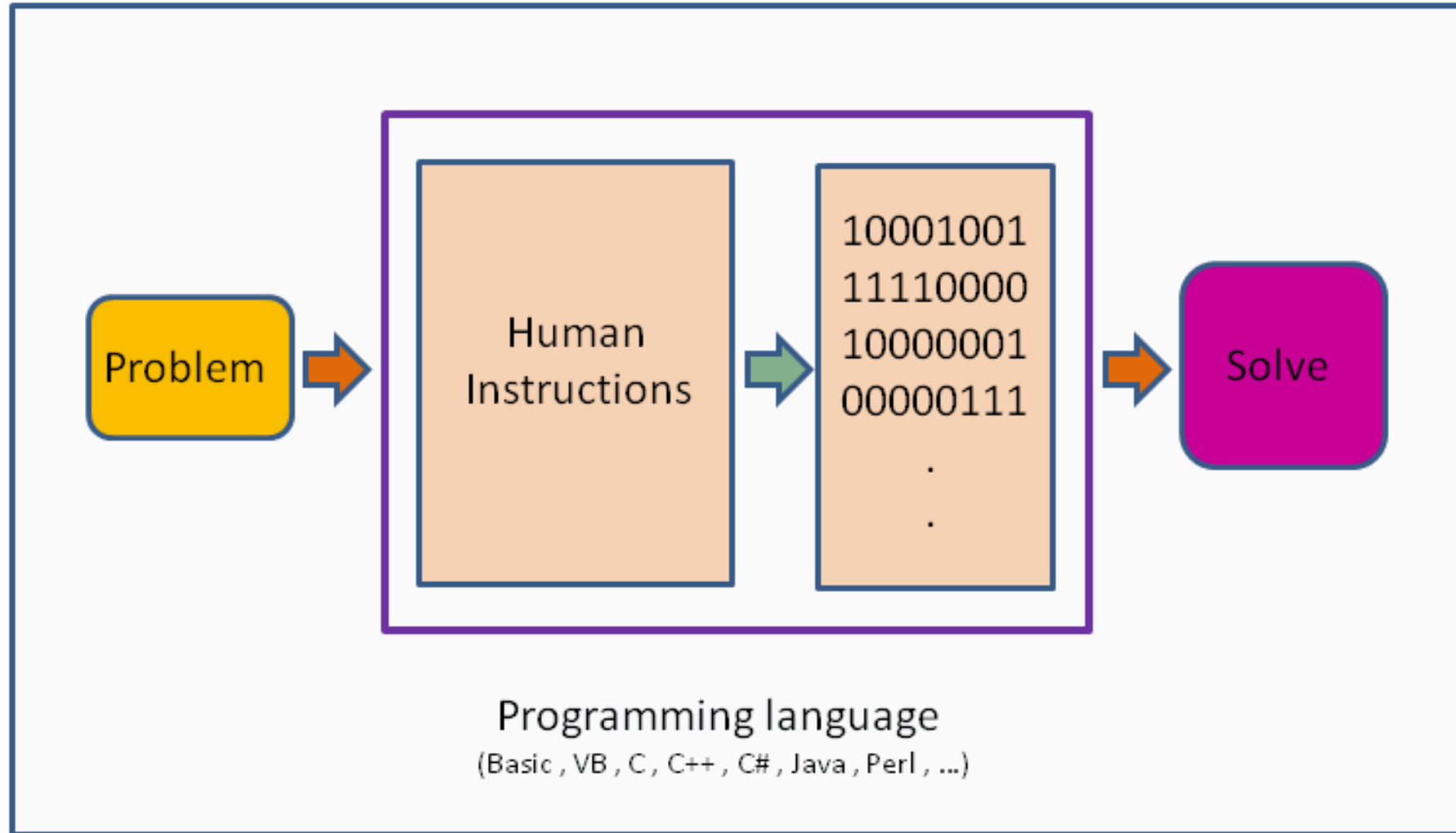
Prof. Carlos Cedeño

Este documento da inicio al Módulo 1. Si está leyendo esto, ha comenzado su introducción al mundo de la programación. Este curso está diseñado para descubrir el potencial de la programación con Python, independientemente de su campo de estudio o profesión (arte, humanística, sociales, ingeniería, medicina, etc).

¿Qué es la Programación y su Relevancia?

La **programación** es el proceso de proporcionar **instrucciones claras y precisas** a una computadora para que realice tareas específicas.

Considere la analogía de una receta detallada para un robot de cocina: cada paso debe ser explícito y en un lenguaje que la máquina pueda procesar.



Relevancia de la Programación en Diversos Campos:

- **Resolución de Problemas:** Aprender a programar fomenta el pensamiento lógico y la habilidad para descomponer problemas complejos en componentes manejables. Esta es una habilidad aplicable en diversos campos.
- **Automatización:** Tareas repetitivas que consumen tiempo pueden ser automatizadas mediante la programación.
- **Creación y Desarrollo:** Permite la creación de herramientas personalizadas, desde páginas web simples y pequeños juegos, hasta visualizaciones de datos complejas.
- **Comprensión del Entorno Digital:** En un mundo cada vez más tecnológico, entender los fundamentos de la programación ofrece una perspectiva valiosa sobre cómo funcionan las herramientas digitales.

Cómo es el flujo para resolver un problema mediante la programación?

El Núcleo de la Programación: Los Algoritmos 🧠

Antes de escribir código en un lenguaje de programación, es necesario un plan: un **algoritmo**.

Un **algoritmo** es una **secuencia finita y ordenada de pasos** bien definidos, que describe cómo resolver un problema específico o realizar una tarea.

Ejemplos de la vida cotidiana incluyen:

- **Una receta de cocina:**

- i. Precalentar el horno a 180°C.
- ii. Mezclar los ingredientes especificados en un recipiente.
- iii. Verter la mezcla en un molde.
- iv. Hornear durante el tiempo indicado.

*Cada instrucción **es un paso** definido del algoritmo.*

¿Faltan instrucciones?

<https://www.youtube.com/watch?v=FN2RM-CHkul>

Características de un algoritmo efectivo:

- **Preciso:** Cada paso debe ser inequívoco y claramente definido.
- **Definido:** Con la misma entrada, el algoritmo siempre debe producir el mismo resultado.
- **Finito:** Debe constar de un número limitado de pasos y concluir.

Es probable que ya utilice algoritmos en su vida diaria sin identificarlos formalmente. La programación se enfoca en formalizar estos procesos para la ejecución por una computadora.

De los Algoritmos al Código: Los Lenguajes de Programación

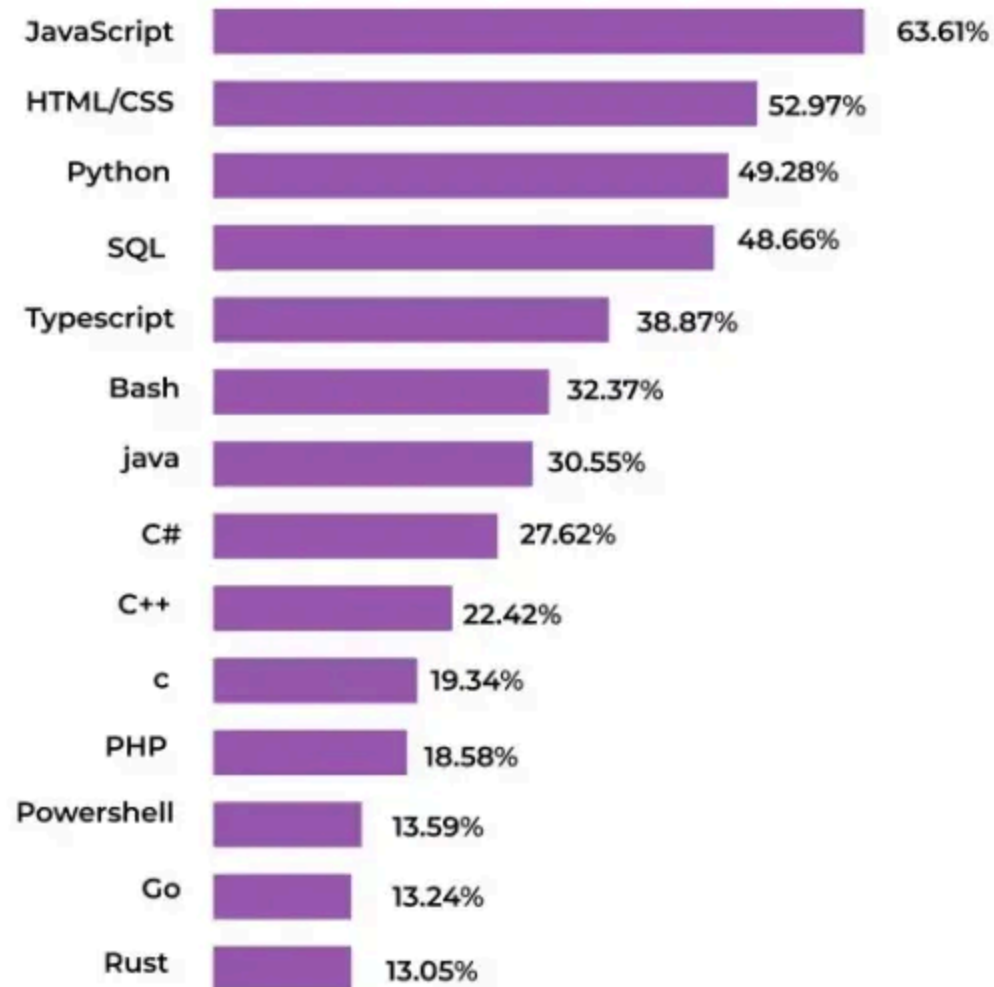


Con un algoritmo definido, el siguiente paso es comunicarlo a la computadora. Esto se realiza mediante **lenguajes de programación**, que son lenguajes formales diseñados para instruir a las máquinas.

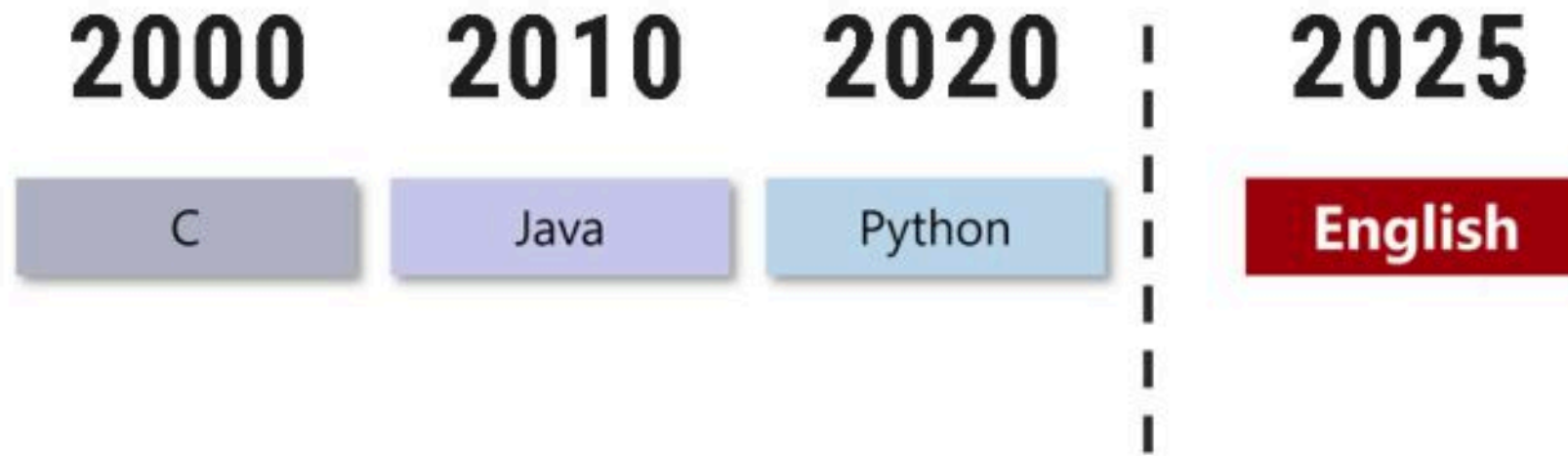
¿Qué es el Código Fuente?

Las instrucciones que se escriben en un lenguaje de programación se conocen como **código fuente** (o simplemente "código"). Esencialmente, es el texto que los programadores redactan, siguiendo las reglas gramaticales (sintaxis) del lenguaje elegido, para detallar los pasos que la computadora debe seguir.

Most popular programming languages as per Stack Overflow Survey 2023



Most Popular Programming Languages



Procesamiento del Código: Lenguajes Compilados vs. Interpretados

No todos los lenguajes de programación se procesan de la misma manera. Hay dos enfoques principales: Lenguajes **Compilados** VS Lenguajes **Interpretados**

- **Lenguajes Compilados:**

- **Proceso:** El código fuente escrito por el programador es traducido en su totalidad por un programa llamado **compilador**. Este proceso genera un archivo de código máquina (instrucciones que la CPU puede ejecutar directamente), conocido como archivo ejecutable.
- **Analogía:** Similar a traducir un libro completo a otro idioma antes de que el lector acceda a él. La lectura posterior es directa y rápida.
- **Ventajas:** La ejecución de los programas suele ser más rápida, ya que el código ya está traducido al lenguaje nativo de la máquina.
- **Desventajas:** Cualquier modificación en el código fuente requiere una nueva compilación completa. El archivo ejecutable es específico para la arquitectura del sistema operativo para el cual fue compilado.
- **Ejemplos:** C, C++, Go, Rust.

- **Lenguajes Interpretados:**

- **Proceso:** Un programa llamado **intérprete** lee el código fuente línea por línea, traduciendo y ejecutando cada instrucción de manera secuencial y en tiempo real.
- **Analogía:** Comparable a un traductor humano que interpreta una conversación frase por frase, simultáneamente.
- **Ventajas:** Ofrecen mayor flexibilidad y portabilidad (el mismo código puede ejecutarse en diferentes sistemas operativos si el intérprete correspondiente está disponible). Facilitan la prueba y depuración de código.
- **Desventajas:** La ejecución puede ser más lenta en comparación con los lenguajes compilados, debido a la traducción concurrente.
- **Ejemplos:** Python, JavaScript, Ruby, PHP.

Python como Lenguaje Interpretado

Python es un lenguaje **interpretado**. Esta característica contribuye a su facilidad de uso para principiantes, permitiendo escribir y probar código de forma ágil.

Ventajas de Python para Principiantes:

Python es ampliamente recomendado para iniciarse en la programación por varias razones:

- **Sintaxis Clara y Legible:** Su estructura se asemeja al lenguaje natural, facilitando su comprensión.
- **Interpretado:** Permite una experimentación y aprendizaje iterativos.
- **Versátil:** Aplicable en desarrollo web, análisis de datos, inteligencia artificial, automatización de tareas, y más.
- **Amplia Comunidad y Recursos:** Existe una vasta cantidad de documentación, tutoriales y foros de soporte.
- **Enfoque en Conceptos:** Permite concentrarse en los fundamentos de la programación sin la complejidad de otros lenguajes.

La Programación en Acción: Aplicaciones de Python

Es natural preguntarse cómo aplicar estos conocimientos en su campo específico. Python es una herramienta notablemente versátil. A continuación, algunas ideas conceptuales de su aplicación:

- **Nutrición:**
 - **Idea:** Desarrollar una herramienta que calcule el valor calórico de una receta a partir de sus ingredientes y cantidades, o que permita registrar y analizar el consumo diario de nutrientes.
 - **Concepto con Python:** Utilizar variables para almacenar datos de alimentos, listas para agrupar ingredientes, y funciones para realizar cálculos basados en una base de datos nutricional.

- **Diseño Gráfico:**

- **Idea:** Automatizar tareas como el redimensionamiento de un lote de imágenes a un formato específico o la aplicación de filtros de color uniformes.
- **Concepto con Python:** Emplear bibliotecas de manipulación de imágenes (ej. Pillow) para escribir un script que procese múltiples archivos de imagen de forma automática. Esto podría significar un ahorro considerable de tiempo.

- **Ingeniería Industrial / Procesos:**

- **Idea:** Analizar datos de tiempos de ciclo de máquinas en una línea de producción para identificar cuellos de botella o modelar el impacto de cambios en el proceso.
- **Concepto con Python:** Importar datos de producción (ej. desde hojas de cálculo), utilizar Python para realizar cálculos estadísticos (promedios, desviaciones), e incluso generar visualizaciones para identificar áreas de mejora.

Resumen de la clase

En este primer módulo, se han establecido las bases conceptuales:

- Definición y relevancia de la programación.
- Perspectiva histórica de la programación.
- Concepto de algoritmo como plan para la resolución de problemas.
- Introducción a los lenguajes de programación, la distinción entre compilados e interpretados, y las ventajas de Python para principiantes.
- Definición de código fuente.
- Ejemplos conceptuales de la aplicabilidad de Python en diversas disciplinas.