



[Escuela de código]

Apuntes de clase:

Introducción a Python

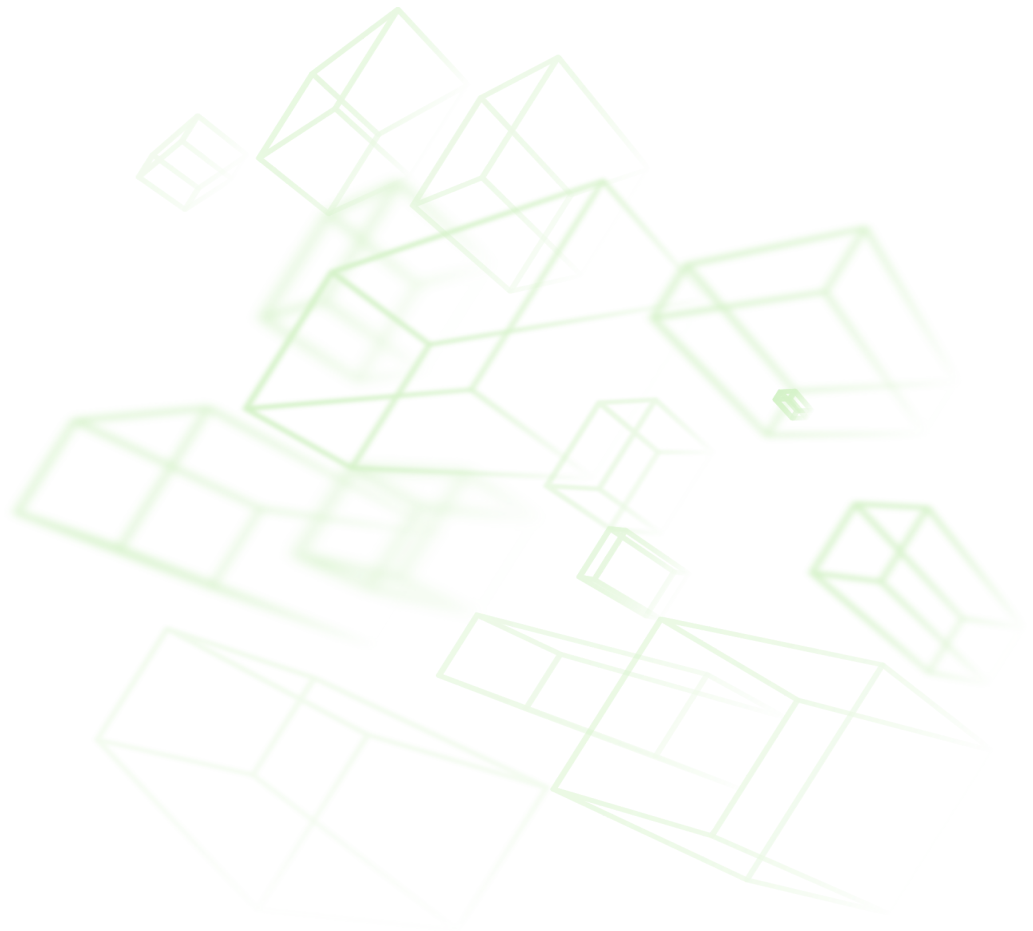


Tabla de contenido

Campus	2
Curso Python Inicial	2
Desafíos	4
Inicios de la programación	6
¿Qué es un programa?	6
Los primeros programas y lenguajes	7
El mundo Python	8
¿Qué es Python?	8
¿Qué necesito para comenzar a programar en Python?	8
Que lo diferencia con otros lenguajes	9
Herramientas y facultades de Python	10
Científico de datos	11
Inteligencia Artificial	11
Backend - Web Apps	12
Desarrollo de Dispositivos (IoT)	12
Herramientas destacadas que hacen esto posible	13
Compilador (contenido extra)	14
Cómo funciona en otros lenguajes	14
El intérprete de Python	14
¡Hola Mundo!	16
Preparación del entorno	16
Descarga del repositorio "hola_mundo_python"	16
Ejecución del programa	17
Desafíos de esta semana	18
¡Hasta la próxima!	19
Links de interés	19

Campus

Este y otros apuntes más lo podrán encontrar en el campus de Inove, ingresando al siguiente link:

<https://inovecode.com/campus>

Curso Python Inicial

El curso de Python Inicial está compuesto por 24 horas de cursada (2 meses) más el mínimo tiempo recomendado por fuera para profundizar en la materia (24 horas de ejercitación). La cursada a su vez se divide en horas enfocadas a contenidos teóricos y de ejercitación. El objetivo de cada clase es que se lleven un concepto nuevo que puedan aplicarlo y despejar sus dudas en el mismo día, para que luego puedan continuar ampliando y fijando la idea en casa con ejercitación. Se espera que dediquen tiempo de cada semana a realizar desafíos que los ayudará a comprender mejor la teoría y el día de mañana serán valiosas horas de experiencia que los ayudaran en su vida personal.

Python Inicial 24 horas cursada (2 meses) 15 horas teoría 9 horas práctica + 24 horas ejercitación 48 horas experiencia	Certificado de Asistencia Completar todos los cuestionarios para validar presentismo (hay un cuestionario por clase)  0 de 0 Intentados
	Certificado de Conocimientos alcanzados <ul style="list-style-type: none">• Deberá completar al menos uno de los proyectos disponibles en el sistema de desafíos.• Puede elegir cualquier proyecto, incluso cambiarlo o realizar más de uno de los disponibles (si los hubiera).

Cada cursada finaliza con un trabajo práctico integrador que deben presentar para poder obtener la certificación del curso, caso contrario recibirán un certificado de asistencia si han completado todos los cuestionarios del campus.

Cuanto mayor tiempo le dediquen a los desafíos planteados para hacer durante la semana, más fácil será para ustedes el desarrollo del proyecto final.

Estas son algunas de las herramientas que se verán durante la cursada:

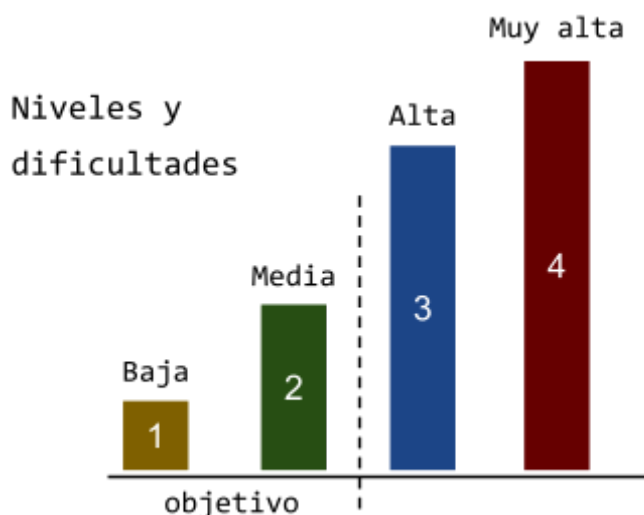


Este curso se enfoca en el estudio de conceptos de programación general, Escribir código limpio y escalable. Utilizar editores de código, resolución de problemas, depuración de programas. Construir la base necesaria para el desarrollo de programas.

Desafíos

Los desafíos se dividen en distintas dificultades:

- **Nivel 1 (dificultad baja):** Desafíos propuestos para afirmar los conceptos básicos de la clase, requieren muy poca dedicación.
- **Nivel 2 (dificultad media):** Desafíos que podrá realizar el alumno sin experiencia previa, dedicando un tiempo considerable y realizando las debidas dudas o consultas en los foros de comunidad del curso (¡consulte sus dudas!).
- **Nivel 3 (dificultad alta):** Desafíos que superan el objetivo mínimo del curso, requieren mucha más investigación y exploración personal. Son desafíos planteados para alumnos con experiencia previa en programación o para aquellos alumnos con mucho tiempo disponible para profundizar lo visto en clase.
- **Nivel 4 (dificultad muy alta):** Son desafíos similares a los que pueden encontrarse en una entrevista de trabajo, recomendamos realizarlos cuando ya tenga una sólida base de los temas abordados en el curso. Los puede realizar más adelante como calentamiento antes de postularse a una oferta de trabajo.



La realización de los desafíos es un camino que transita el alumno, la prioridad es que pueda realizar los ejercicios de nivel dos ya que son aquellos que garantizan haber entendido el concepto de lo visto en clase y el objetivo mínimo del curso esperado.

Los desafíos de mayor dificultad se deben realizar cuando el alumno no tiene ninguna duda sobre lo realizado en los otros desafíos. Recomendamos siempre priorizar la realización de los desafíos que comprenden el objetivo del curso y haber comprendido la unidad antes de continuar con los desafíos de la siguiente unidad.

La realización del proyecto es la recta final del camino transitado por el alumno, recomendamos haber realizado y comprendido a la perfección los desafíos que comprenden al objetivo mínimo del curso antes de comenzar a desarrollar el proyecto, ya que integra todo lo desarrollado durante la cursada.

A medida que vaya realizando los desafíos obtendrá puntos de experiencia, como recompensa por su esfuerzo.

- ❖ Su esfuerzo será recompensado con puntos de experiencia
- ❖ Podrá canjear sus puntos por descuentos y beneficios
- ❖ Sumando exp puede postularse para trabajar dentro de inove desde el lado académico y de desarrollo
- ❖ El sistema lo recompensará si mantiene su objetivo semanal (racha)

Recuerde que hay tiempos de entrega estimados para los desafíos de cada unidad. Si respeta esos tiempos y mantiene su "racha" alcanzando el objetivo mínimo cada semana, el sistema lo recompensará con más puntos aún (puntos de bonus)

En caso que desee volver a consultar cuales son los tiempo de entrega estipulados para los desafíos, les recomendamos volver a leer los términos y condiciones (sección "Entrega de actividades"):

<http://inovecode.com/terminos-y-condiciones/>

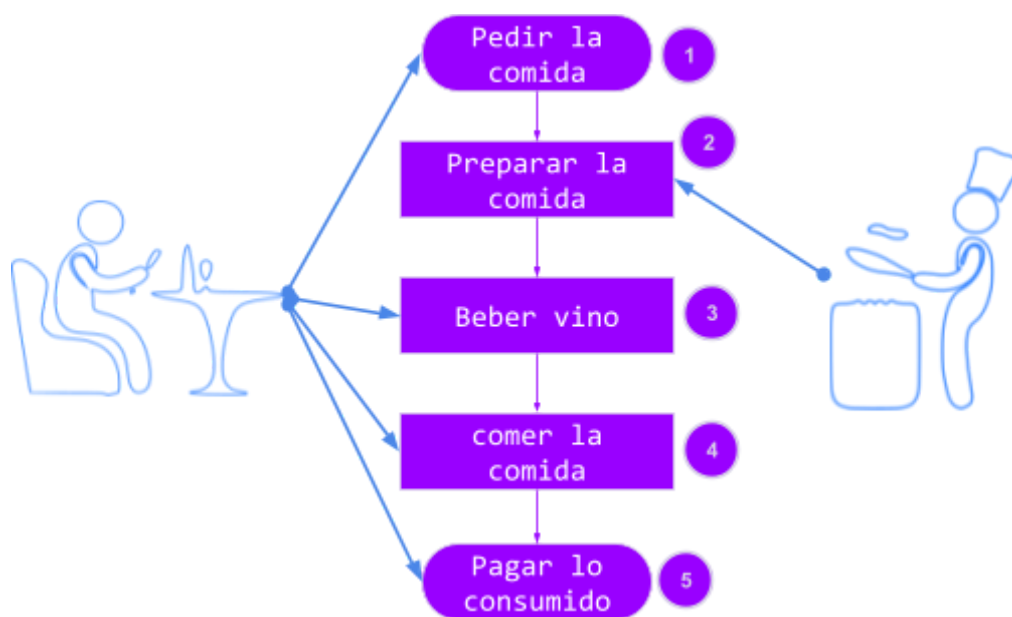
Inicios de la programación

¿Qué es un programa?

La programación es *"organizar una secuencia de pasos ordenados a seguir para hacer cierta cosa"*. Este término puede utilizarse en muchos contextos, por ejemplo: "Vamos a programar una salida para este fin de semana largo".

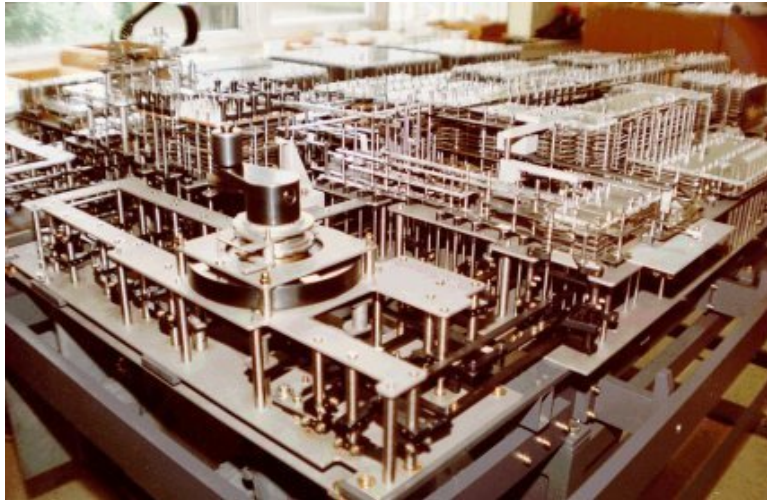
El programa es aquel que se encarga de realizar esas secuencias de pasos para cumplir con el objetivo.

En el siguiente ejemplo se muestra lo que podría ser un programa simplificado de una persona consumiendo en un restaurante. El programa es una secuencia de tareas o acciones, que pueden ir de lo más micro o detallado como "tomar el tenedor, pinchar un trozo de comida" a lo más macro "Comer la comida".



Los primeros programas y lenguajes

La programación por computadora comienza con la primera computadora programable en 1936, una calculadora que tenía el tamaño de una mesa entera la cual se programaba con tarjetas perforadas:



La "Z1" era una máquina de aproximadamente 1000 kg de peso, que consistía en unas 20000 piezas. Era una computadora programable, basada en números binarios de punto flotante y un sistema de conmutación binario.

Las operaciones lógicas (cerebro) se realizaban con relés eléctricos y La única unidad eléctrica (corazón) era un motor eléctrico para dar la frecuencia de reloj de 1 Hz (ciclo por segundo) a la máquina.

El lenguaje de programación de esta máquina estaba dado con tarjetas perforables, que indican a la máquina que tiró de operación realizar y con qué números.

El mundo Python

¿Qué es Python?

Python es un lenguaje de programación creado en los 90, del tipo interpretado o script. Se utiliza para crear programas mediante un lenguaje que luego será interpretado por la máquina o dispositivo y transformado en una secuencia de pasos o órdenes que se ejecutarán para cumplir un determinado fin. El fin de un lenguaje de programación es poder escribir en lenguaje "humano" esas órdenes que luego el sistema traducirá a lenguaje máquina (como lo eran en un principio las tarjetas perforadas de la máquina Z1).

¿Qué necesito para comenzar a programar en Python?

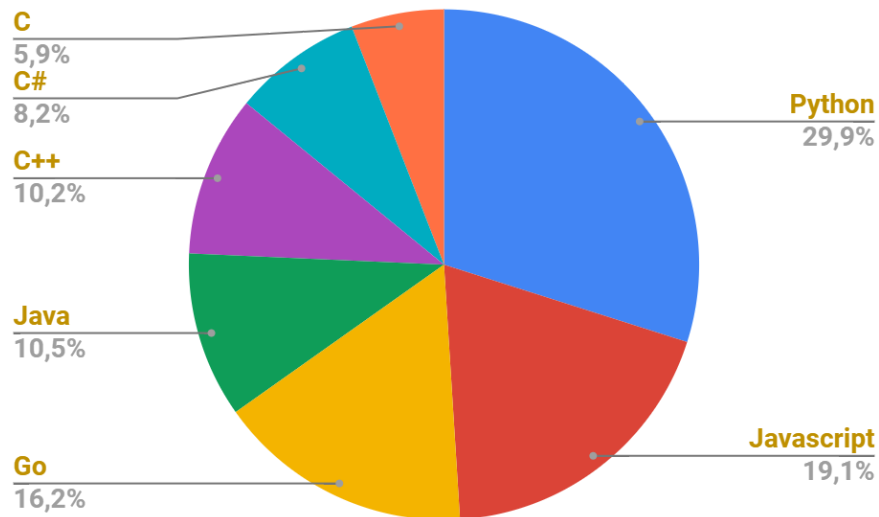
Para poder programar en Python necesitaremos un dispositivo en donde escribir nuestro código, descargarnos el intérprete para dicho sistema y un editor de código o IDE donde escribir nuestro programa y ensayar.

La preparación del entorno de Python puede realizarse en tan solo 10 minutos, a diferencia de un lenguaje compilado donde puede llevar todo un día de trabajo (por tener que resolver dependencias del sistema).



Que lo diferencia con otros lenguajes

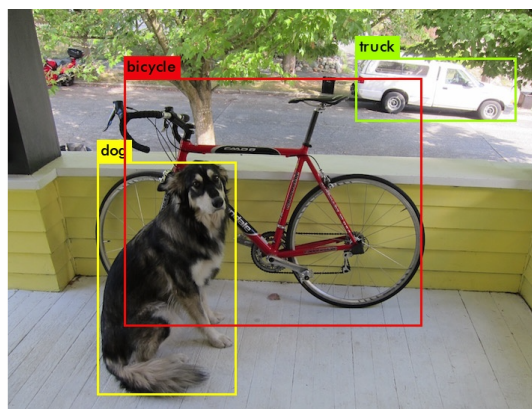
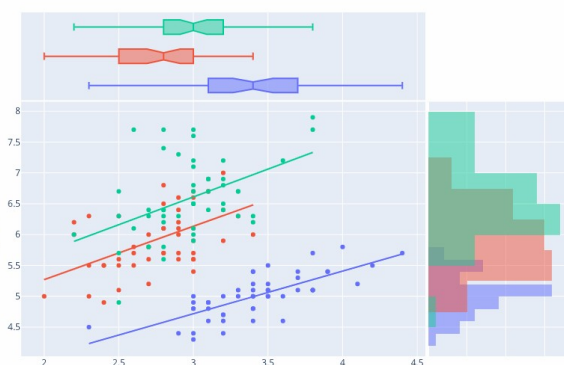
La sintaxis de Python es tan sencilla y cercana al lenguaje natural que es el lenguaje más utilizado para comenzar a programar. La gran cantidad de librerías disponibles y la potencia del lenguaje, entre otros, hacen que desarrollar una aplicación en Python sea sencillo, muy rápido y, lo que es más importante, divertido.



Herramientas y facultades de Python

A lo largo de la cursa se estudiarán herramientas de Python que nos permitan realizar diferentes roles en el ámbito personal, como pueden ser:

- **Científico de datos o Inteligencia de negocios:** Realizar análisis y representación de datos.
- **Inteligencia artificial:** Reconocimiento de imágenes, patrones y clasificación.
- **Desarrollador Backend o aplicaciones Web:** Desarrollo de la infraestructura de base de datos y servicios de API Rest.
- **Desarrollador de Dispositivos o Plataformas Cloud:** Desarrollo de software para dispositivos conectados a la nube y su administración en plataformas cloud.



Científico de datos

Los científicos de datos son una nueva estirpe de expertos en datos analíticos que poseen habilidades técnicas para resolver problemas complejos. Son parte matemáticos, parte científicos en computación y parte observadores de tendencias.

Los científicos de datos permiten encontrar patrones de formas alternativas al los métodos tradicionales, logrando por ejemplo encontrar la cura a una enfermedad analizando millones de resultados de sangre que serían imposibles estudiar por un humano en toda su vida.

La inteligencia de negocios o business intelligence (BI) es el conjunto de procesos, aplicaciones y tecnologías que facilitan la obtención rápida y sencilla de datos provenientes de los sistemas de gestión empresarial para su análisis e interpretación, de manera que puedan ser aprovechados para la toma de decisiones y se conviertan en conocimiento para los responsables del negocio.

En otras palabras, se utilizan tecnologías como Python para analizar la evolución de un negocio, una tienda, un proveedor de servicios, para tomar las decisiones adecuadas que permitan sacar la mejor ganancia de mi producto o servicio.

Un científico de datos o inteligencia de negocios tiene conocimientos en programación, estadística, big data y conocimiento del dominio del problema.

Inteligencia Artificial

Python es uno de los lenguajes más utilizados en la inteligencia artificial (IA), aprendizaje automático y deep learning. Su vasta cantidad de herramientas, librerías y comunidad detrás de este lenguaje permite llevar algoritmos de inteligencia artificial a otro nivel para combatir problemáticas complejas. La IA es esencialmente cualquier inteligencia mostrada por una máquina que conduce a una solución óptima o subóptima, dado un problema. Luego, el aprendizaje automático lleva esto a un paso más allá al utilizar algoritmos para analizar datos y aprender de ellos para la toma de decisiones.

Por ejemplo, si en un auto autónomo tuviéramos que enseñarle como se ve una persona para que las diferencia de los autos y los elementos en las calles para evitar un accidente, sería muy difícil darle una descripción que abarque a todas las personas en el mundo. Este tipo de casos complejos se resuelven con inteligencia artificial, en donde la máquina es entrenada viendo millones de imágenes de personas dejando que la computadora interprete de la mejor manera cómo diferenciar a una persona de otro objeto.

Backend - Web Apps

La programación backend está orientada al desarrollo y mantenimiento de la base del sistema que hace que todo funcione. Por ejemplo, se encarga del intercambio de información entre los clientes y la aplicación, procesar la información para poder presentarla al cliente en un formato determinado. Diseñar y mantener las bases de datos de todo el sistema acorde a la aplicación y el negocio de la compañía. Un programador backend avanzado es el arquitecto del sistema en sí, permite que el producto sea escalable y permite ser utilizado por miles de usuarios sin perjudicar la performance.

El desarrollo backend se utiliza para construir APIs, aplicaciones Webs y cualquier sistema en la nube.

Desarrollo de Dispositivos (IoT)

Un desarrollador de dispositivos "Internet of things" tiene la capacidad de elaborar programas que corren en mini computadoras o dispositivos de uso específico (como sensores, actuadores, robots) y a su vez tiene la capacidad de integrar el reporte y el estado de todos aquellos dispositivos en la nube.

Para ello el desarrollador de dispositivos o plataformas cloud tiene conocimiento del aspecto físico que se desea resolver (medir alguna magnitud como temperatura o controlar otro dispositivo como pueden ser LEDs, motores o control de acceso) y los conocimientos de Python necesarios para llevar toda esa información a la nube y poder administrarla en plataformas de control, actualización y monitoreo.

Herramientas destacadas que hacen esto posible

Las herramientas que harán esto posible, entre otras, son las siguientes:

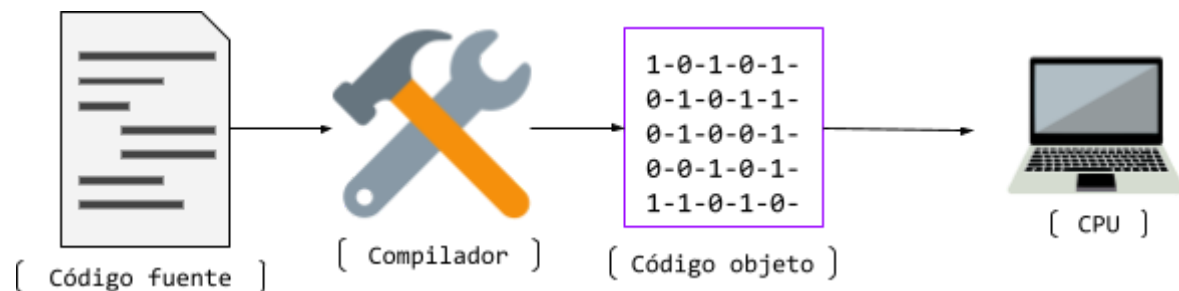


Estas son solo una parte de las herramientas que se estudiarán durante la carrera, y les permitirán desarrollar casi cualquier tipo de trabajo o actividad el día de mañana.

Compilador (contenido extra)

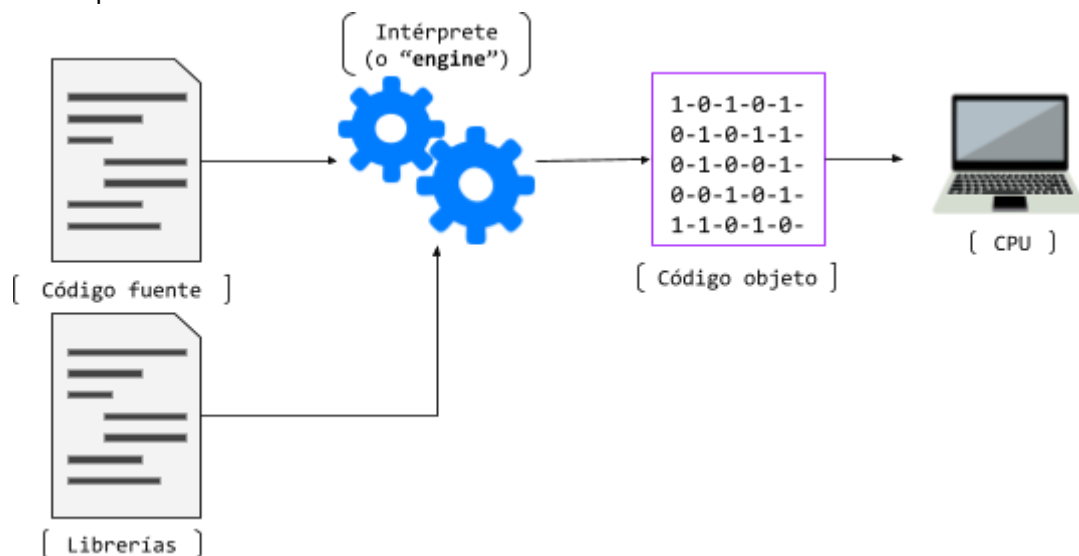
Cómo funciona en otros lenguajes

Para poder generar un programa de computadora se utiliza un "Compilador", el cual se encarga de traducir nuestro programa escrito en la computadora (código fuente) en lenguaje que pueda interpretar la máquina en sí (código máquina), produciendo como resultado una aplicación o programa.



El intérprete de Python

Python es un lenguaje de programación del tipo interpretado o script. Un lenguaje interpretado o de script es aquel que se ejecuta utilizando un programa intermedio llamado intérprete. El intérprete se encarga de generar un código máquina compatible con el sistema de la computadora, nos abstrae de las dependencias del entorno. Sino fuera por el intérprete, cada módulo o librería Python debería ser compilado para el sistema operativo utilizado.



Ejemplo:

Compilador

Si un grupo de extranjeros asistieran a una reunión de trabajo en un país donde no se habla su idioma natal, todos los extranjeros deberían aprender el idioma local para poder comunicarse en la reunión de trabajo. Cuantos más extranjeros viajen, más personas distintas deberían capacitarse en el idioma local.



Intérprete

Si en el ejemplo anterior el grupo de extranjeros llevará un intérprete, esta persona sería la encargada de comunicar todas las ideas del grupo y traducirlas al idioma local. Tiene la ventaja de que sola una persona debe capacitarse para poder comunicarse, es más flexible y eficiente ya que el intérprete se ha preparado para ese fin. La única desventaja es que no puede haber charlas de grupos separados, todos deben hablar a través del intérprete.



Conclusión:

Los lenguajes interpretados son más flexibles y más portables, pero no son recomendables para sistemas críticos de tiempo real donde se requiera extrema velocidad con recursos reducidos. En esos casos es recomendable compilar el programa realizado en Python para lograr la performance necesaria.

¡Hola Mundo!

Cuando aprendemos un nuevo lenguaje de programación se estila comenzar por el clásico "Hola Mundo!" o "Hello World!". Este programa en general consta de hacer que el sistema envíe un mensaje para verificar el correcto funcionamiento de nuestro paquete de instalación y configuración.

Preparación del entorno

Para poder realizar esta actividad primero debemos tener instalado y configurado nuestro entorno de trabajo. Para ello referirse al "Instructivo de instalación: Entorno Python" disponible entre los archivos del campus. De no encontrarse allí, por favor, tenga a bien comunicarse con alumnos@inove.com.ar para su solicitud. Recomendamos a su vez mirar los videos tutoriales de instalación del entorno.

Descarga del repositorio "hola_mundo_python"

Para poder realizar esta actividad deben haber primero haber creado una cuenta de GitHub para poder trabajar. Para ello referirse al "Instructivo de Github: Interfaz web" disponible entre los archivos del campus. De no encontrarse allí, por favor, tenga a bien comunicarse con alumnos@inove.com.ar para su solicitud.

Completada la configuración del entorno, debemos descargar el programa "Hola Mundo!" del repositorio de Inove. El repositorio que descargamos del siguiente:

https://github.com/InoveAlumnos/hola_mundo_python

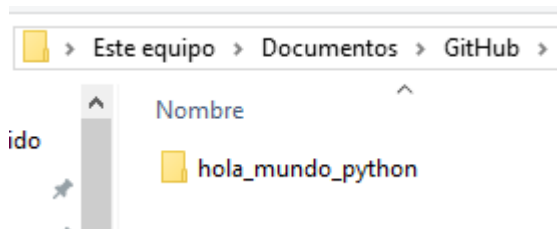
Recomendamos seguir el siguiente video tutorial para realizar la correcta descarga del material:

[Link al video de youtube](#)

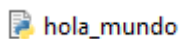
Una vez descargado el repositorio como un archivo "zip" lo deben ubicar en una carpeta donde colocarán de aquí en más los repositorios de este curso y descomprimir los archivos para poder abrirlos en Visual Studio Code.

Ejecución del programa

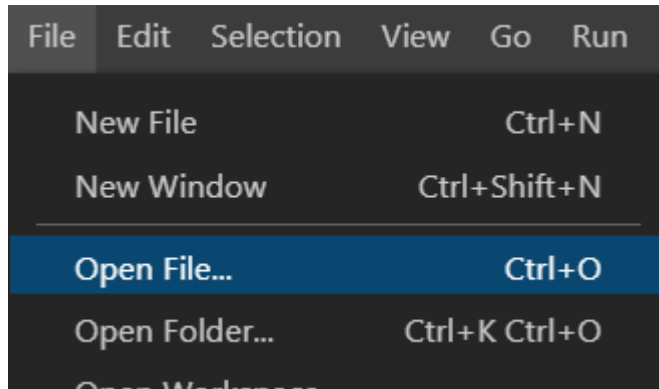
Para poder ejecutar el programa que acabamos de descargar, primero debemos localizar la carpeta donde ubicamos el repositorio descomprimido:



Dentro de la carpeta del repositorio veremos el programa "hola_mundo!", un archivo de extensión ".py" el cual nuestro editor de código "Visual Studio Code" lo asociará con un programa script del tipo Python:



Para abrir el programa realizamos los mismos pasos que cualquier otro programa que hayan usado antes, ingresamos al Visual Studio Code vamos al menú "**File**" y seleccionamos "**Open File**".

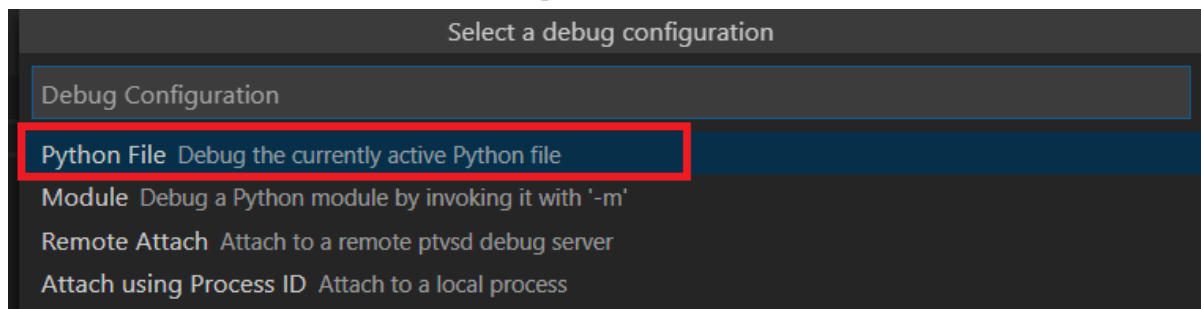


Al abrir el programa con nuestro editor podremos ver que al final del documento se encuentra la siguiente línea de código:

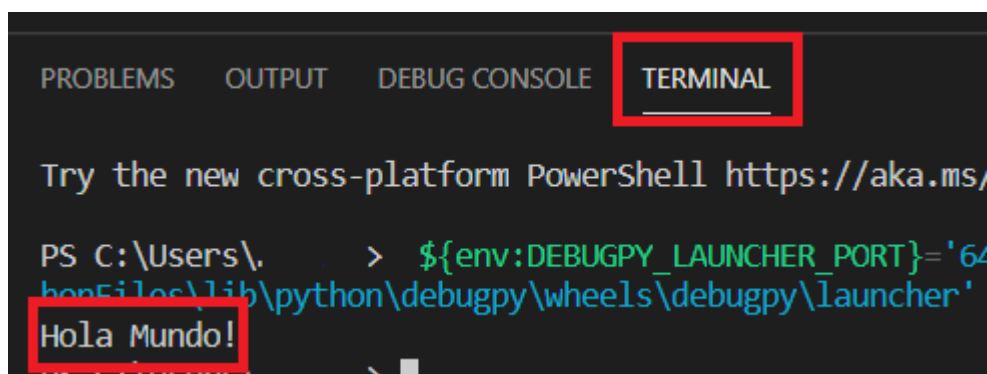
```
print("Hola Mundo!")
```

Esta línea de código lo que hace es "**imprimir**" (print) el mensaje en consola "Hola Mundo!".

Para poder ejecutar el programa presionamos la tecla **"F5"** para iniciar la ejecución de Visual Code Estudio, nos aparecerá el siguiente cartel:



Debemos seleccionar que queremos ejecutar el programa como un **"Python File"**. Al comenzar la ejecución se abrirá la **"Terminal"** en la parte inferior de nuestra pantalla:



Si en su consola pueden visualizar el mensaje "Hola Mundo!" es porque su sistema está listo para comenzar!

Desafíos de esta semana

¡Una vez que tenga el entorno listo puede empezar a trabajar en sus desafíos! Recomendamos ver el video tutorial del uso del sistema de desafíos que se encuentra en el campus (dentro de la sección "Sistema Desafíos") para comenzar con la actividad.

Cualquier duda puede consultar por los foros de comunidad del curso.

¡Hasta la próxima!

Con esto finaliza la introducción a Python, a partir de ahora tienen todo preparado para empezar a realizar programas y continuar con la ejercitación de las clases.

Si desean conocer más detalles sobre el contenido pueden iniciar un tema de discusión en el foro del campus, o visitar los "Links de interés" que se encuentra al final de este apunte.

Links de interés

- [Empresas que utilizan Python en Argentina](#)
- [Documentación oficial Python 3.7](#)
- [Compilador online](#)