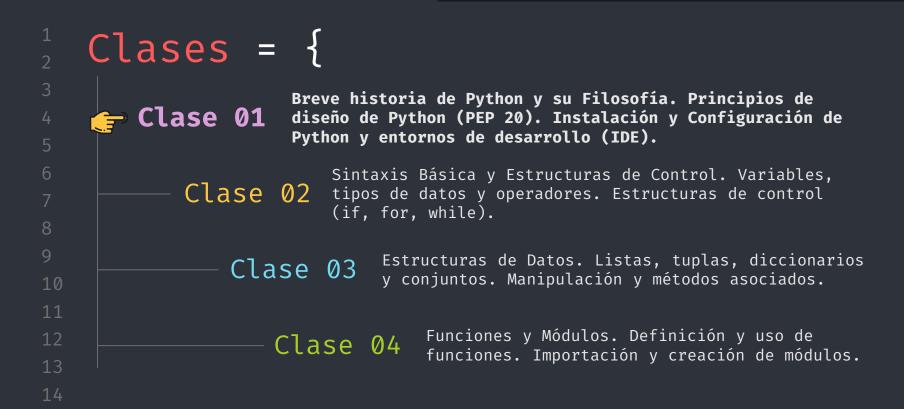
```
Clase 01 = {
 Presentación = [Les
 damos la bienvenida al
 curso
```





```
'Instructor'
Maximiliano Martin
Simonazzi {
       < E-Mail >
      < maxisimonazzi@gmail.com >
       < Github >
       < www.github.com/maxisimonazzi >
       < LinkedIn >
       < www.linkedin.com/in/maxisimonazzi >
```



Sobre el curso {

Este curso introductorio de Python y MicroPython está diseñado para proporcionar a los participantes una introducción a la programación con Python para que los participantes puedan iniciarse en el emocionante mundo de la programación de microcontroladores con MicroPython. A lo largo del curso, los participantes explorarán y aprenderán a aplicar los conocimientos adquiridos en una placa del tipo ESP32. El enfoque principal será teórico pero con el desarrollo de ejercicios prácticos para finalizar con un pequeño proyecto electrónico.



```
Objetivo {
```

Los participantes del curso podrán, finalizado el mismo, comprender los fundamentos de la programación en Python, entender los principios de MicroPython, conocer sus diferencias con CircuitPython, programar la plaqueta ESP32 o Raspberry Pi Pico W 1/2, explorar conectividad de redes con la placa, etc.



Régimen de cursado {

El curso requiere una dedicación total de 60 horas reloj, de las cuales 24 están distribuidas en 12 clases de 2 horas cada una y las restantes 36 es tiempo que cada alumno dedicara entre clases para repasar conceptos, practicar e investigar. Las clases del curso serán sincrónicas y las mismas serán los días sábados a partir del día 10 de Mayo en horario de 10:00 a 12:00 (con excepción de las 2 clases extras que coordinaremos durante la cursada). La modalidad será a distancia.



Metodología a utilizar {

Las clases serán Teóricas y Prácticas. En cada sesión se impartirán conceptos para explicar los fundamentos de Python y MicroPython, utilizando presentaciones y ejemplos prácticos en vivo. Estas sesiones se complementarán con material de lectura y recursos en línea para que el estudiante pueda replicar lo aprendido en clases. El aprendizaje pretendido es orientado a proyecto, por lo que los estudiantes deberán desarrollar un proyecto en equipo de 2 o 3 personas.



```
Asistencia y aprobación {
```

```
# Condiciones para aprobar
```

```
# Asistencia no se tiene en cuenta para el certificado. # Aprobación: Entrega del proyecto integrador.
```

Se entrega Certificado de Aprobación a quienes hayan cumplido con el régimen de aprobación. Aquellos que no cumplan con el régimen, recibirán un certificado de Asistencia.







```
¿Por que Python? {
10
```









```
¿Qué es Python? {
     # Python es un lenguaje de alto nivel de programación
     interpretado cuya filosofía hace hincapié en la
     legibilidad de su código. Se trata de un lenguaje de
     programación multiparadigma. Es un lenguaje
     interpretado, dinámico y multiplataforma
     # Administrado por Python Software Foundation, posee
     una licencia de código abierto, denominada Python
10
     Software Foundation License. Python se clasifica
     constantemente como uno de los lenguajes de
     programación más populares
```

¿Qué es Python? { Multiparadigma Multiplataforma < Soporta programación < Su código puede correr imperativa, en Windows, Linux, Mac, estructurada, POO, Microcontroladores, etc. funcional, etc > Se puede reusar codigo > Tipado Fuerte y Interpretado Dinamico < El código no se compila < El tipo de las a lenguaje máquina, sino variables se decide en que se ejecutan las tiempo de ejecución. No instrucciones a medida que se permite operar tipos se las lee > diferentes >

```
Sobre Python {
  # Sucesor espiritual del lenguaje ABC
  # El nombre proviene del programa humorístico
  británico Monty Python
 # El desarrollo comienza en 1989
  # 20 de Febrero de 1991 publica el código con v.0.9.0
 # En Enero 1994 se publica la v.1
```

10

```
Sobre Python {
 # En el 2000 se lanza Python v.2
 # En el 2008 se lanza Python v.3 ;;;Liberen a la bestia!!!
 # La transición de v.2 a v.3 fue lenta por sus diferencias
  # A día de hoy, nos encontramos en la v.3.14
 # La demanda laboral no para de crecer
 # PEP-8 (estilo) y PEP-20 (zen) son las guías principales.
```

```
Sobre Python {
  # Fácil de aprender (alto impacto en la educación)
  # Alto nivel (sintaxis sencilla, clara, legible)
  # Es super flexible ante diferentes escenarios
  # Creado a finales de los 80'por Guido Van Rossum
  # Completamente libre, Open Source
  # Amplia librería standard
  # Pagina Oficial: www.python.org
```

```
Opiniones a favor {
  # Es un lenguaje optimo para comenzar.
  # Es la navaja suiza de los lenguajes, sin necesidad de configurar
  servidores, compilar, aprender un lenguaje intermedio, etc. Si se
  requiere una dependencia fácil de incorporarla.
  # Gran comunidad.
  # Es el mas usado en AI por mucho (Langchain, LangGraph, Pydantic
  AI) y tiene alta demanda en Startups y empresas tecnológicas.
  # Agentes de AI codifican muy bien en Python o están hechos con
  Python.
```

```
Opiniones en contra {
 # Es lento
 # Esta sobrecargado para que sea flexible
 # Solo sirve para principiantes (es un lenguaje de
  juguete)
 # No sirve para aplicaciones de alto rendimiento
 # No es multitareas real
```

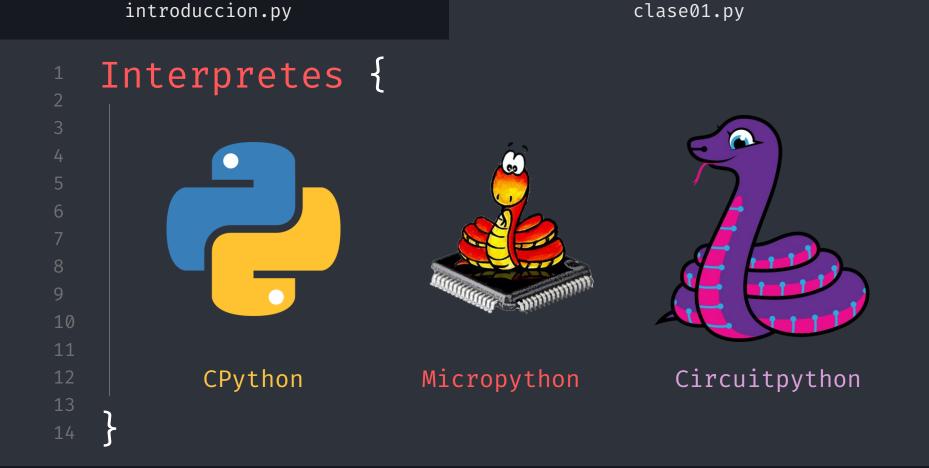
Interpretado vs Compilado {

```
10
```

Los lenguajes compilados utilizan un compilador que procesa el programa completo en una sola vez, generando un código intermedio o código máquina. Estos lenguajes, como C, C++, y C#, ofrecen una eficiencia notable, ya que los errores se identifican después de la verificación completa del programa, asegurando una mayor integridad del código

Interpretado vs Compilado {

Los lenguajes interpretados adoptan un enfoque más lineal mediante el uso de un intérprete que ejecuta el programa línea por línea. Lenguajes como JavaScript y Python pertenecen a este grupo. Aunque son menos eficientes que los lenguajes compilados, la facilidad de depuración y la interpretación línea por línea hacen que sean una opción atractiva







Interpretes {

Circuitpython

- < Circuitpython es un fork de Micropython mantenido
 y llevado adelante por la empresa Adafruit para
 darle soporte a todas sus placas y diseños. >
- < Las diferencias entre Circuitpython y Micropython son prácticamente nulas. Se reduce solo a dar mejor soporte a placas de la empresa, es por ello que los códigos y librerías de ambos pueden ser intercambiados y reutilizados. >
- < Ambos tienen una gran comunidad lo cual es bueno.
 Micropython opera casi exclusivamente en github
 mientras que Circuitpython en discord y su foro >

```
Aprender a programar
es aprender a pensar.
{ Steve Jobs; }
```



```
{ Nos vemos en la
proxima clase }
```

