

# CONFERENCIA 1

## DE "HOLA, MUNDO" A TU PRIMERA CALCULADORA

# ¿QUÉ ES PROGRAMAR?

Es darle **instrucciones precisas** a la computadora para que resuelva un problema.

Imagina que estás escribiendo una receta de cocina, pero para una máquina que sigue cada paso al pie de la letra.

( Pizarra: Idea clave de la programación)

# ¿POR QUÉ PYTHON? 🐍

- **Sintaxis Simple y Legible:** Se parece mucho al inglés, ideal para empezar.
- **Poderoso y Versátil:** Usado por empresas como Google, Netflix e Instagram para todo, desde desarrollo web hasta Inteligencia Artificial.
- **Gran Comunidad:** Mucha ayuda y bibliotecas disponibles.

## HOY TRABAJAREMOS DE DOS MANERAS

1. **Intérprete Interactivo (Consola):** Ideal para experimentar y probar código rápidamente.
2. **Archivos de Script (.py):** Para escribir programas completos, que puedan ser compartidos con otros usuarios.

# NUESTRO PRIMER PROGRAMA

El ritual de iniciación de todo programador.

( Intérprete: El "Hola, Mundo")

( Intérprete: Calculadora)

# VARIABLES

Una variable es como una **caja con una etiqueta** donde guardamos un valor para usarlo más tarde.

( **Intérprete y Pizarra: Concepto de variable**)

**¡Concepto Clave!** El código se ejecuta de arriba hacia abajo, línea por línea. El valor de una variable es una "foto" en un momento dado.

( **Pizarra e Intérprete: El orden secuencial de las operaciones)**

## TIPOS DE DATOS PRIMITIVOS

- `int`: Números enteros (-3, 0, 25)
- `float`: Números con decimales (-3.14, 9.99)
- `bool`: Valores de verdad (True o False)
- `str`: Cadenas de texto ("Hola", 'Python')

(  **Intérprete: La función `type()`** )



# OPERACIONES CON TIPOS PRIMITIVOS


Podemos realizar operaciones según el tipo de dato.

- **Aritméticos (`int`, `float`):** `+`, `-`, `*`, `/`, `//`, `%`, `**`
- **Comparación (producen `bool`):** `==`, `!=`, `>`, `<`, `>=`, `<=`
- **Strings (`str`):** `+` (unir), `*` (repetir)
- **Lógicos (`bool`):** `and`, `or`, `not`

( **Intérprete: Ejemplos de operadores**)


# INTERACTUANDO CON EL USUARIO

La función `input ( )` pausa el programa y espera que el usuario escriba algo.

( **Intérprete: La función `input ( )`**)

## EL "PROBLEMA" DE `input()`

¡Cuidado! `input()` siempre devuelve el dato como un **string** (**str**), incluso si escribimos números.

( Intérprete y Pizarra: El "problema" de `input()` con los números)

## CONVERSIÓN DE TIPOS

Podemos convertir un tipo de dato en otro usando funciones como `int()` y `float()`.

( **Intérprete: La solución con conversión de tipos**)

# F-STRINGS

La forma más limpia y recomendada de incluir variables dentro de un string es usando **f-strings**.

Solo pon una **f** antes de las comillas y las variables entre llaves **{ }**.

( **Intérprete: Formateo con f-strings**)

## MINI CALCULADORA

¡Vamos a juntar todo! Crearemos un programa completo en un archivo.

**Objetivo:** Pedir dos números al usuario y mostrar el resultado de las 4 operaciones básicas.

( Código en vivo (.py): Proyecto Mini Calculadora)

- Usamos **variables** para guardar datos.
- Todo dato tiene un **tipo** (`int`, `float`, `bool`, `str`).
- Usamos **operadores** para manipular los datos.
- `print()` muestra información, `input()` la pide.
- ¡Debemos **convertir** los datos de `input()` si queremos hacer cálculos!
- Trabajamos en el **intérprete** (para probar) y en **archivos .py** (para programas).

# PRÓXIMOS PASOS

Ahora que nuestros programas pueden calcular...

En la próxima clase, aprenderán a **tomar decisiones** usando `if`, `elif` y `else`.



**¿PREGUNTAS?**