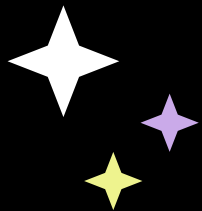
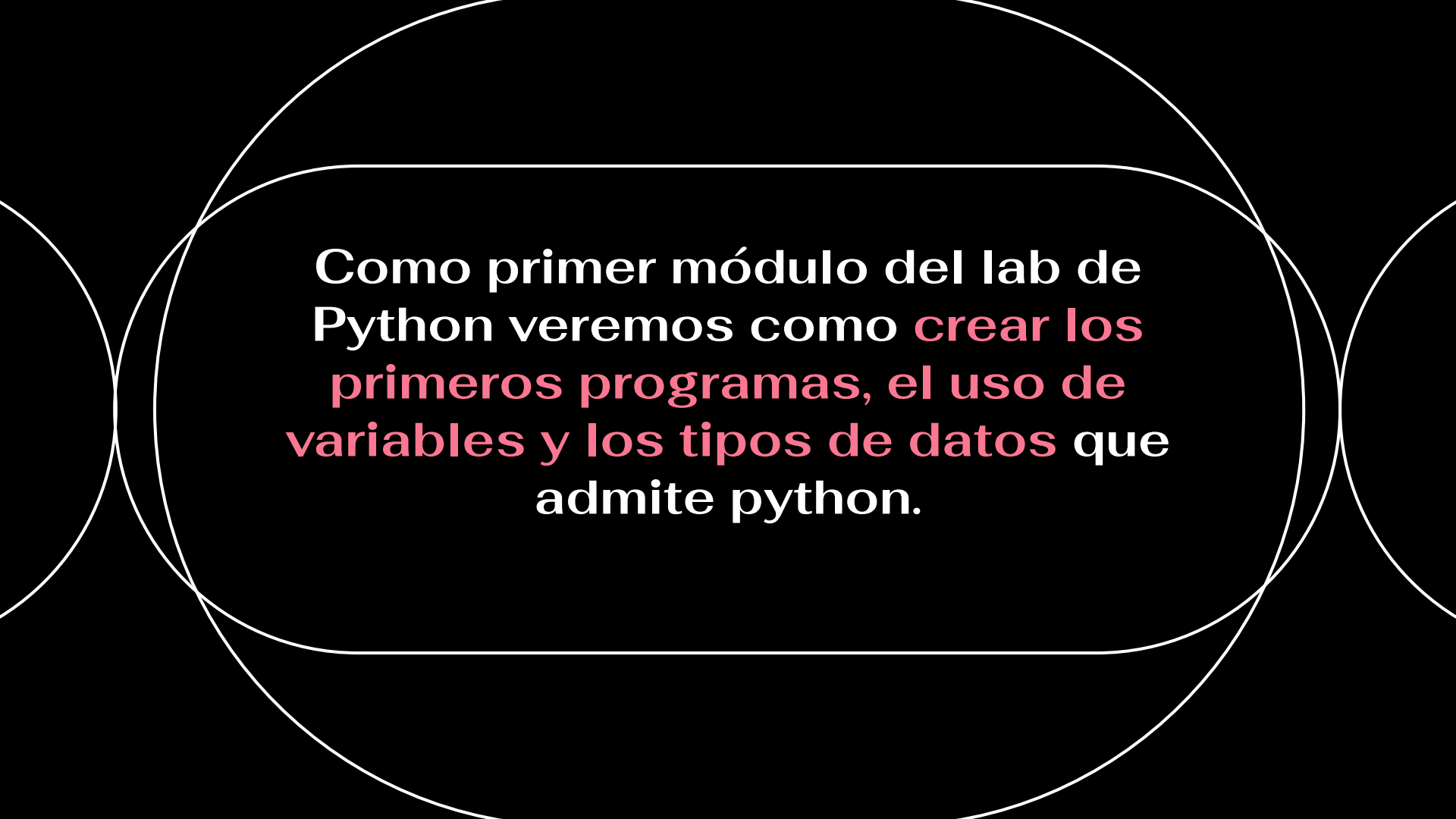


Intro a PYTHON



AED 2025





**Como primer módulo del lab de
Python veremos como crear los
primeros programas, el uso de
variables y los tipos de datos que
admite python.**

En la carpeta del módulo 1 encontrarás:

1. Guía Modulo 1 (presentación)

En esta presentación te contamos qué material hay disponible y hacemos un repaso de los temas del módulo.

2. Guia Instalación

Primer paso, para que puedas correr programas de python en tu computadora.

3. Guía teórica

Material teórico detallado de lo que necesitas saber de python para comenzar.

4. Guía de errores

Resumen de los errores más comunes a la hora de utilizar python, cómo reconocerlos y solucionarlos.

En la carpeta del módulo 1 encontrarás:

5. Notebooks (google colab)

Las **notebooks** nos permiten correr de manera aislada código python desde la web.

Dejamos una serie de ejemplos (los mismos de la guía teórica) y ejercicios resueltos para que puedas ejecutarlo directamente y ver su resultado.

Acceso en este link:

[5_modulo1_lab_python.ipynb](#)

6. Cuestionario

Cuestionario de auto-evaluación para medir si adquirieron los conocimientos esperados para este módulo

Acceso en este link:

<https://forms.gle/geoZgVdjsHDUEJvH8>

[Tutorial de Colab.ipynb](#)



PYTHON



Python es un lenguaje de programación de **alto nivel** y de **propósito general**.

- Sintaxis simple, en **ingles**
- Open Source and **community driven**
- Es un lenguaje **interpretado**
- Es **multiplataforma y multiparadigma**

Además contamos con muchas **librerías** para python lo que lo hacen útil en distintas áreas: *big data, programación web , inteligencia artificial, automatización, systems scripting* etc.



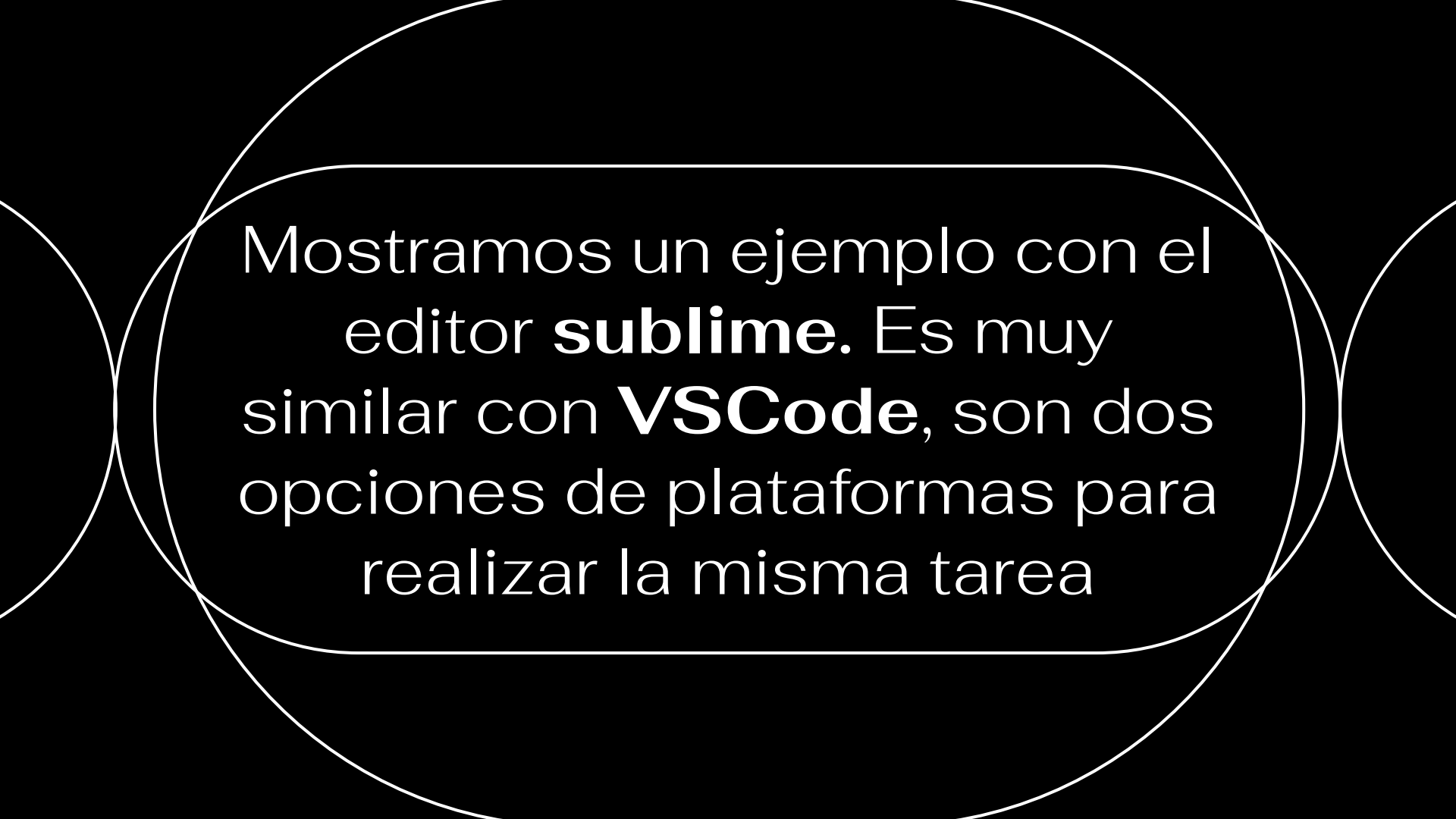


Entorno de trabajo



Existen varias maneras de crear un programa y ejecutar nuestro código python pero en este curso trabajaremos sin utilizar programas extras (como un editor o una IDE) para lograr un mejor entendimiento y poder tener mayor control sobre esta tecnología





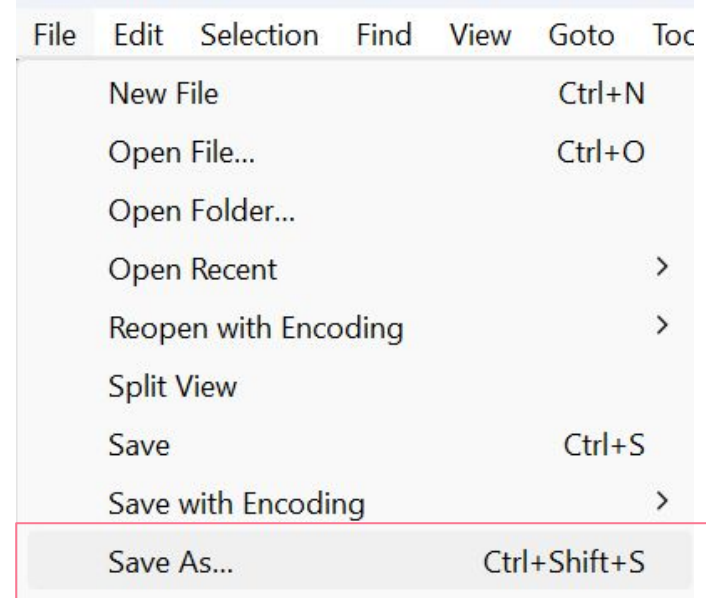
Mostramos un ejemplo con el editor **sublime**. Es muy similar con **VSCode**, son dos opciones de plataformas para realizar la misma tarea

1. Abrir sublime text

1.1



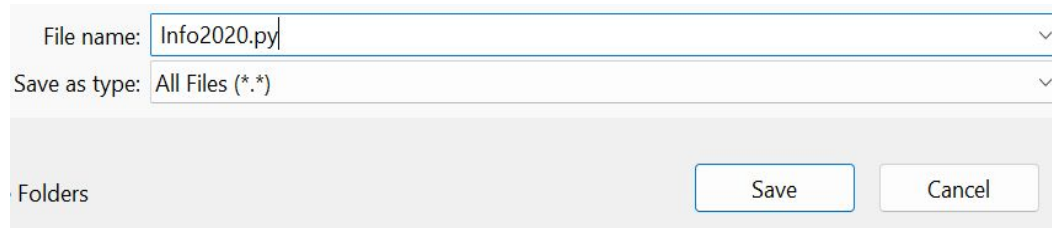
1.2



¡IMPORTANTE!

La extensión de los archivos que guardan código de python

.py



1.3



Hello, World!

Escribamos nuestro primer programa

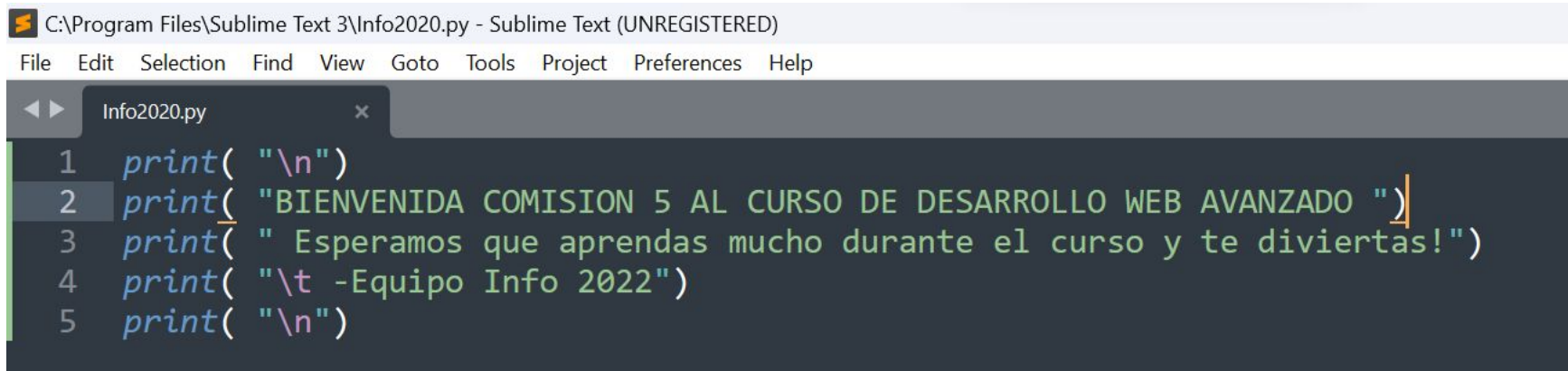
¿Como mostramos por pantalla lo que
escribimos en nuestro programa?

Sentencia `print()`

```
print("Hola mundo! Este es mi primer  
programa! :) ")
```

Lo que este entre "" se
mostrara textualmente

2. Escribir un programa en python



```
C:\Program Files\Sublime Text 3\Info2020.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

Info2020.py x
1 print( "\n")
2 print( "BIENVENIDA COMISION 5 AL CURSO DE DESARROLLO WEB AVANZADO ")
3 print( " Esperamos que aprendas mucho durante el curso y te diviertas!")
4 print( "\t -Equipo Info 2022")
5 print( "\n")
```

`\t` y `\n` son caracteres con significados especiales

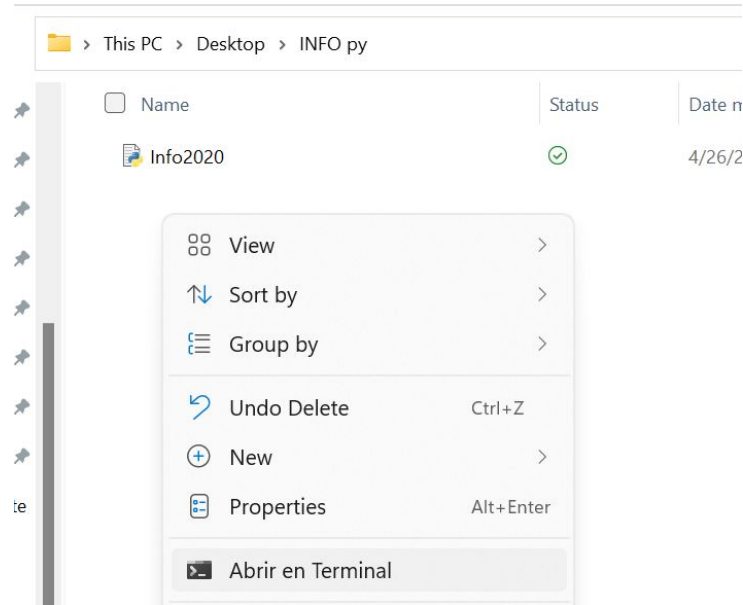
- `\n` representa un salto de línea en lo que se muestra en pantalla, se usa para indicar el fin de una línea de texto y el inicio de una línea nueva
- `\t` representa la tabulación

INVESTIGA QUE OTROS CARACTERES ESPECIALES COMO ESTE HAY

3. Ejecutar nuestro programa

Para esto debemos correr en la consola de comandos nuestro programa


Vamos a la carpeta donde está nuestro programa y damos click derecho y seleccionamos ***abrir en terminal***.



cmd

All Apps Documents Web More ▾

Best match

 **Command Prompt**
App

Apps


 Node.js command prompt >

 Anaconda Prompt (miniconda3) >

Search the web

 cmd - See web results >

 cmd administrator >






 cmdconfig >

 cmdr >



Command Prompt

App

-  Open
-  Run as administrator
-  Open file location
-  Pin to Start
-  Pin to taskbar

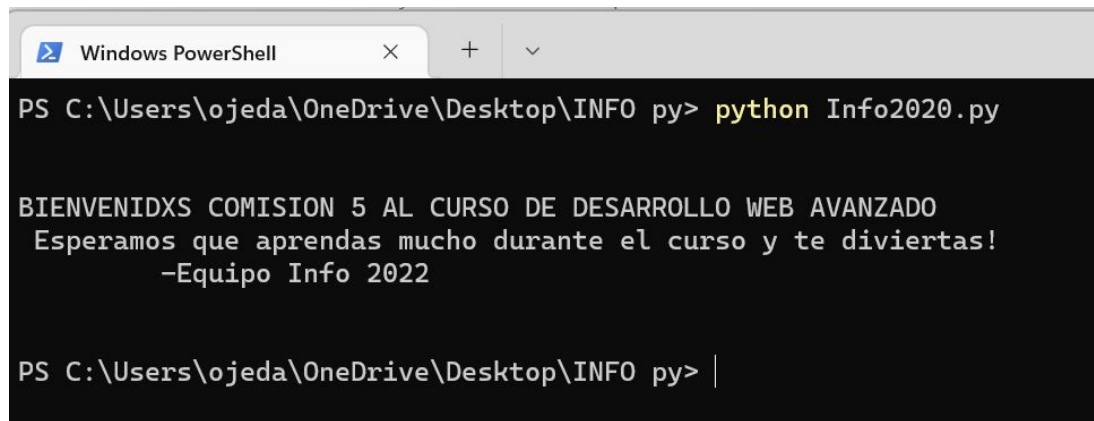


3. Ejecutar nuestro programa

Para esto debemos correr en la consola de comandos nuestro programa

Vemos que estamos **ubicados en la carpeta donde tenemos el programa!**
para ejecutar el programa escribimos *python* seguido del nombre del archivo

```
Python Info2022.py
```



```
Windows PowerShell
PS C:\Users\ojeda\OneDrive\Desktop\INFO py> python Info2020.py

BIENVENIDXS COMISION 5 AL CURSO DE DESARROLLO WEB AVANZADO
  Esperamos que aprendas mucho durante el curso y te diviertas!
    -Equipo Info 2022

PS C:\Users\ojeda\OneDrive\Desktop\INFO py> |
```



VARIABLES



Las variables serán la forma de identificar, de forma sencilla, un dato que se encuentra almacenado en la memoria de la computadora. Luego lo podemos usar en distintas partes del programa

```
<nombre_variable> = <valor> | por ej: x = 1
```





VARIABLES



Son elementos que nos permiten **guardar un dato** para que pueda ser **accesible** fácilmente, **manipulado y transformado** a lo largo de un programa.





VARIABLES



Para **crear** y **darle valores** usamos el operador de **asignación** → =

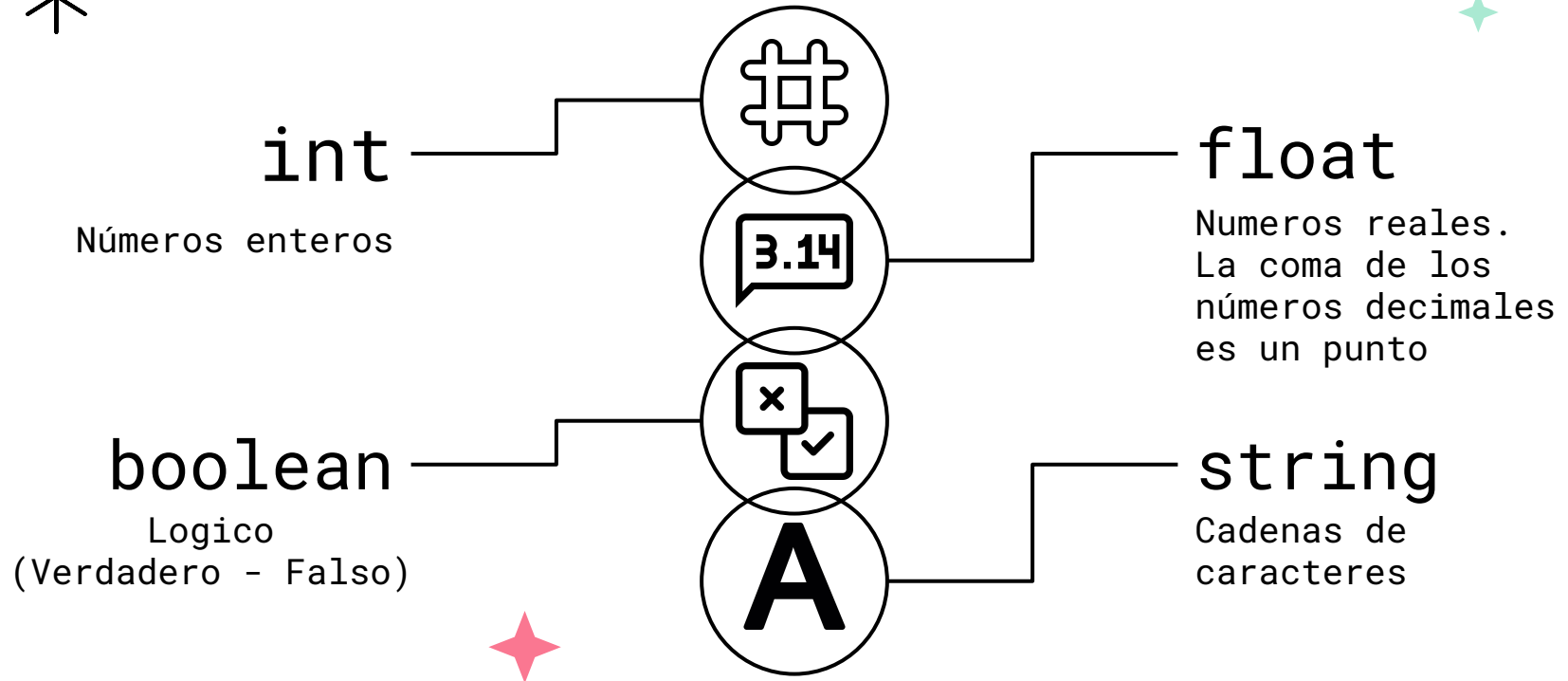
```
<nombre_variable> = <valor> | por ej: x = 1
```

El **operador de asignación** enlaza un **nombre**, en el lado izquierdo del operador, con un **valor** en el lado derecho



TIPOS DE DATOS ELEMENTALES

*



* TIPOS DE DATOS ELEMENTALES



```
entero = 5  
real = 5.5  
cadena_caracteres = "hola"  
print(cadena_caracteres)
```



¿Qué hacemos al escribir código en python?

Al escribir **código** en un **lenguaje de alto nivel** (en este caso Python) lo que hacemos es *hablar un lenguaje más fácil* de comprender para nosotros, y que luego será **traducido a lenguaje de máquina** (lo que entiende la computadora).



Lenguajes de alto nivel (Python, Java, C++,etc)

Instrucciones similares al lenguaje humano
(en ingles). Independiente de la máquina.

Lenguaje de bajo nivel

Lenguaje ensamblador. Utiliza instrucciones
parecidas al inglés. Depende del **procesador**

Lenguaje de máquina

Codigo binario [0-1]. Instrucciones que entiende
la máquina. Lenguaje propio de cada máquina.

HARDWARE
(Parte física de la computadora)

```
package main

import "fmt"

func main() {
    fmt.Printf("hello, world")
}
```

Lenguaje de alto nivel que
entiende el programador



```
0101010111101110001101
0100010100010101001010
0101010010101010000101
0011010001010100011110
0110010100101010101001
1110001101010010010001
```

Lenguaje de máquina que
entiende el procesador



PYTHON



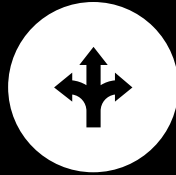
Python es un lenguaje de programación de **alto nivel** y de **propósito general**.

- Sintaxis simple, en **ingles**
- Open Source and **community driven**
- **Es un lenguaje interpretado**
- Es **multiplataforma** y **multiparadigma**

Además contamos con muchas **librerías** para python lo que lo hacen útil en distintas áreas: *big data, programacion web, inteligencia artificial, automatizacion, systems scripting*

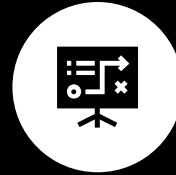


Según la forma en la que son **traducidos**, los lenguajes de programación se clasifican en:



Interpretados

Las instrucciones se traducen (por un **interprete**) al **ejecutar el programa**

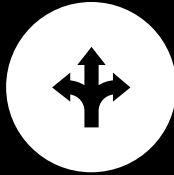


Compilados

Traducen a lenguaje máquina (0-1), creando un **archivo** traducido **antes de ejecutar el programa**. Hay un paso previo de proceso de compilado

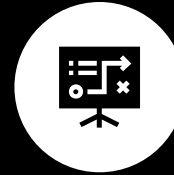


Según la forma en la que son **traducidos**, los lenguajes de programación se clasifican en:



Interpretados

10x más lentos que los compilados
Logramos **independencia de plataforma** y **menor tamaño** de programa
Optimizado para *hacerle la vida más fácil al programador*



Compilados

No son multiplataforma
Son mas eficientes
Al tener un paso previo, esta optimizado para el **momento de ejecución**



PYTHON



¿Que ventajas nos ofrece que sea un lenguaje **interpretado**?

- No necesitamos pasar por el proceso de compilado al desarrollar y probar, el ciclo de desarrollo es mas rapido
- Puede usarse en cualquier sistema operativo o entorno que tenga un **intérprete para python**
(*Windows, MacOS, Linux, Android, Web*)





PYTHON



Python es un lenguaje de programación de **alto nivel** y de **propósito general**.

- Sintaxis simple, en **ingles**
- Open Source and **community driven**
- Es un lenguaje **interpretado**
- **Es multiplataforma y multiparadigma**

Además contamos con muchas **librerías** para python lo que lo hacen útil en distintas áreas: *big data, programación web, inteligencia artificial, automatización, systems scripting etc.*



Multiparadigma

¿Qué son los paradigmas de programación?

Representan un enfoque particular o filosofía para la construcción del software.

Python soporta más de un **paradigma**

- Imperativo
- Funcional
- Orientado a Objetos

Multiplataforma

El código fuente puede ser interpretado y ejecutado en varios sistemas operativos

Python soporta diversas **plataformas** para **ejecutar software** escrito en python

- Linux
- MacOS
- Windows

✦ Según la forma en la que hacen **control de tipos**, los lenguajes de programación se clasifican en:



Tipado estatico

cada variable y parámetro tiene un tipo fijo y es elegido por el programador. (si declaro un número no puede guardar una palabra)

Control de tipos en tiempo de compilación

Tipado dinamico

Una variable o elemento puede tomar valores de distintos tipos en diferentes momentos (numérico, carácter, booleano)

Control de tipos en tiempo de ejecución



Fuertemente tipado



Si bien las variables o elementos en python pueden cambiar de valor a distintos tipos

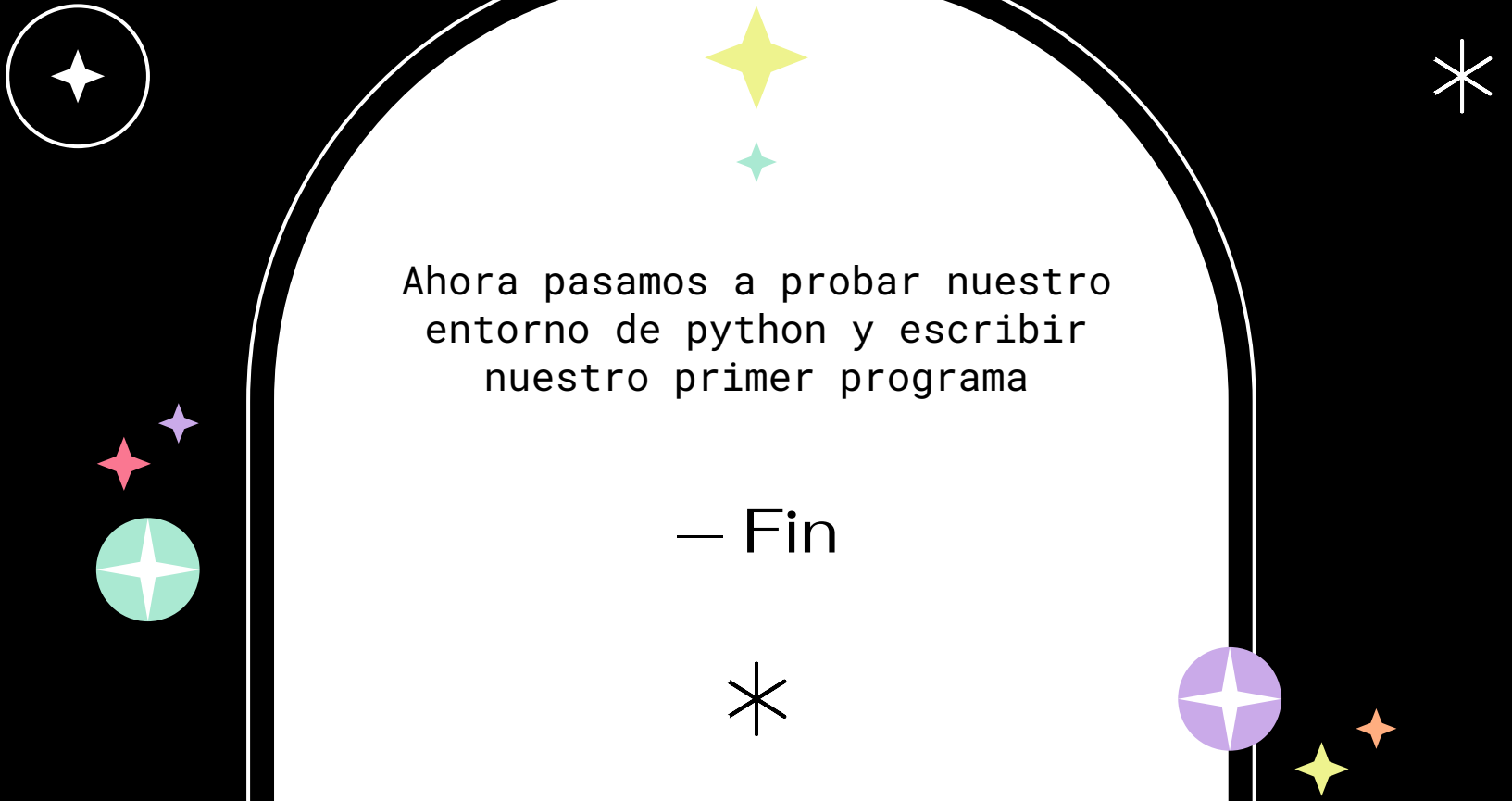
No se permiten violaciones del tipo de dato definido.

```
X = 2
```

```
X = "5"
```

```
R = x + int(y)
```





Ahora pasamos a probar nuestro
entorno de python y escribir
nuestro primer programa

— Fin

