

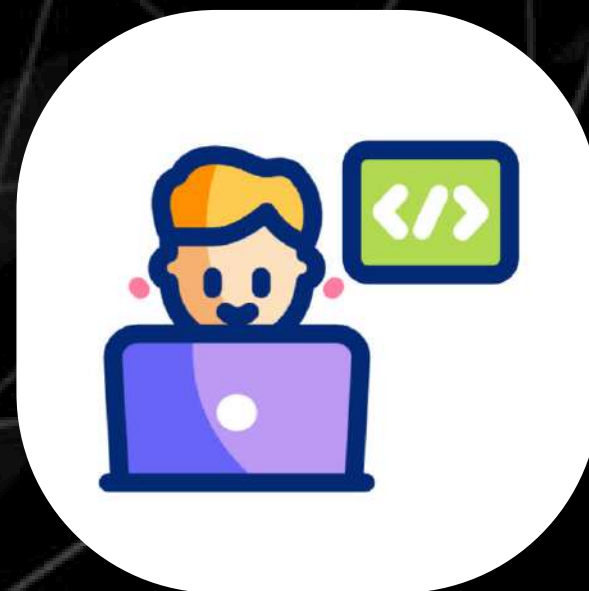
# **CURSO DE PYTHON DESDE CERO**



# CUÁLES SON LOS TIPOS DE DATOS?

Python tiene varios tipos de datos básicos que se utilizan para almacenar y manipular información.

Los principales son:



# LOS PRINCIPALES SON:

Números (Numbers)

Textos

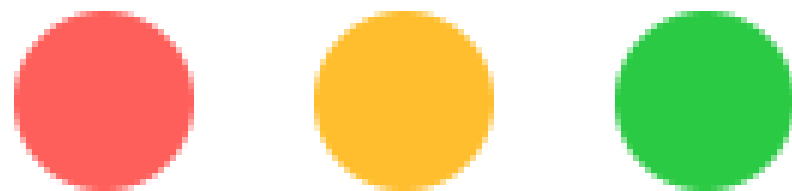
Booleanos

Listas y Colecciones

Ningún valor

# NÚMEROS (NUMBERS)

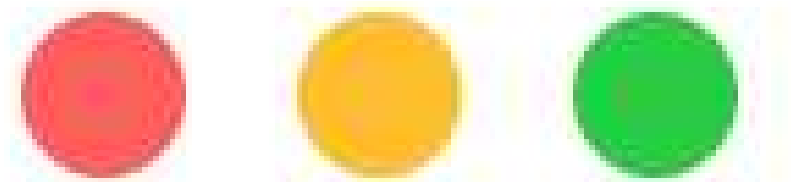
✓ int – Enteros: Números sin decimales.



1 edad = 25

# NÚMEROS (NUMBERS)

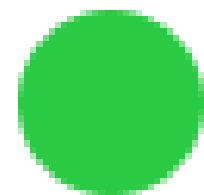
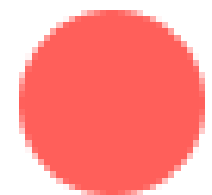
✓ float – Flotantes: Números con decimales.



```
1 precio = 19.99
```

# NÚMEROS (NUMBERS)

- ✓ complex – Complejos: Tienen parte real e imaginaria.



1

$z$

=

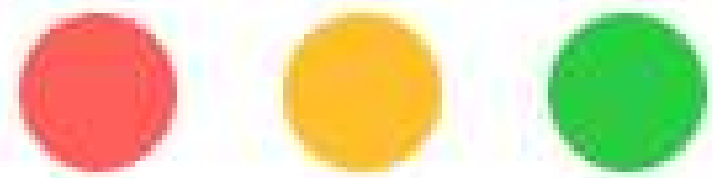
2

+

3j

# TEXTOS

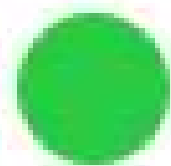
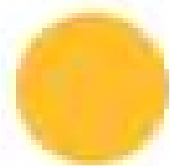
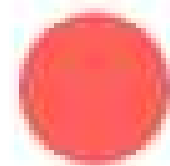
- ✓ str – Cadenas de caracteres: Sirve para almacenar texto.



```
1 nombre = "Python"
```

# BOOLEANOS

✓ bool – Verdadero o Falso: Indicadores lógicos.

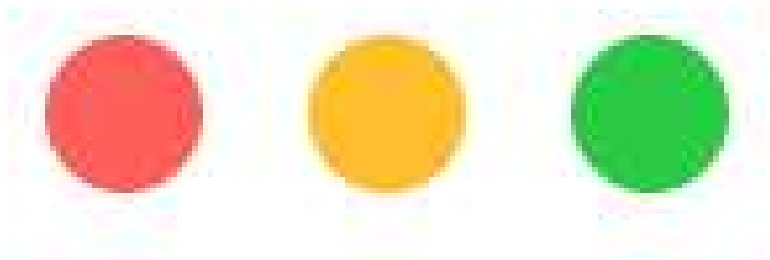


```
1 es_mayor = True
```

```
2 lloviendo = False
```

# LISTAS Y COLECCIONES

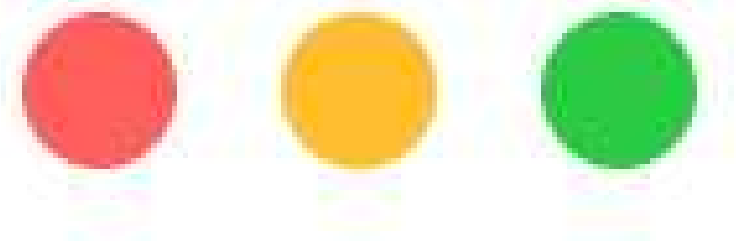
- ✓ list – Listas: Colección mutable (se puede cambiar) y ordenada.



```
1 frutas = ["manzana", "pera", "uva"]
```

# LISTAS Y COLECCIONES

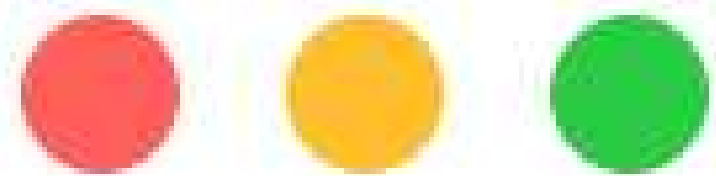
- ✓ tuple – Tuplas: Colección inmutable y ordenada.



```
1 numeros = (10, 20)
```

# LISTAS Y COLECCIONES

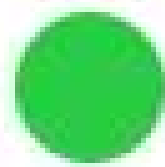
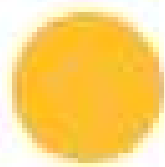
✓ set – Conjuntos: Colección desordenada, sin elementos duplicados.



```
1 numeros = {1, 2, 3}
```

# LISTAS Y COLECCIONES

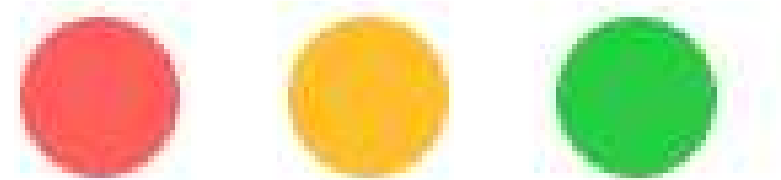
✓ dict – Diccionarios: Almacenan datos en pares clave: valor.



```
1 persona = {"nombre": "Ana", "edad": 30}
```


# NINGÚN VALOR

✓ NoneType: Representa ausencia de valor.



```
1 resultado = None
```

# ¿PARA QUÉ SIRVE CADA DATO?

Tipo de dato	Para qué sirve	Ejemplo práctico	
int (entero)	Para contar, hacer operaciones matemáticas, manejar cantidades que no tienen decimales.	Edad, número de productos, puntajes.	
float (decimal)	Para representar valores con decimales, medidas, cálculos más precisos.	Precio de un producto, temperatura, distancia.	
complex (número complejo)	Para cálculos matemáticos avanzados (ingeniería, física).	Representar números como <code>3+2j</code> .	
str (texto)	Para almacenar y mostrar texto. Cadenas como nombres, mensajes, direcciones, etc.	"Hola", nombres de usuario, descripciones.	
bool (booleano)	Para manejar condiciones lógicas: verdadero o falso.	Saber si un usuario está activo: <code>True</code> o <code>False</code> .	

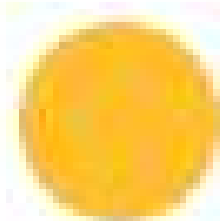
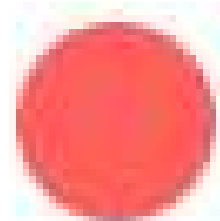
# ¿PARA QUÉ SIRVE CADA DATO?

<code>list</code> (lista)	Para guardar <b>varios elementos</b> en un solo lugar, y poder modificarlos.	Lista de frutas, lista de tareas, resultados de una consulta.
<code>tuple</code> (tupla)	Para almacenar una secuencia de datos <b>que no quieres que cambien</b> .	Coordenadas de un punto: <code>(10, 20)</code> , fecha <code>(2025, 11, 16)</code> .
<code>set</code> (conjunto)	Para guardar elementos sin repeticiones; útil para eliminar duplicados o hacer operaciones <b>de conjunto</b> .	<code>{"rojo", "azul", "verde"}</code> , quitar duplicados de una lista.
<code>dict</code> (diccionario)	Para representar <b>objetos</b> , almacenar datos con estructura <b>clave: valor</b> . Muy útil en programación real.	Datos de usuario: <code>{"nombre": "Ana", "edad": 30}</code>
<code>NoneType</code> ( <code>None</code> )	Para indicar "sin valor", "no definido" o "vacío".	Resultado pendiente, variable sin inicializar, final de una función.



# VER TIPO DE DATO EN PYTHON

Puedes comprobar el tipo de cualquier valor  
con:



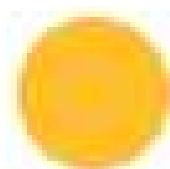
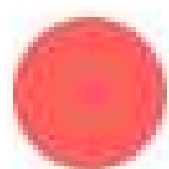
1

```
type(valor)
```



# VER TIPO DE DATO EN PYTHON

Ejemplo:



```
1  type(10)           # int
2  type("Hola")       # str
3  type([1, 2, 3])    # list
```

**NOS VEMOS EN UN PRÓXIMO  
VIDEO DE ESTE CURSO,  
SALUDOS 🚀**

