

Python Básico

Prof. Ariel Palazzesi

Comenzamos a grabar la clase

Módulo 1:

Introducción a la informática y programación básica en Python

Clase 02

Variables/Operadores

1. La función print().
2. Variables. Creación y asignación de valores.
3. Tipos de datos simples: enteros, punto flotante, lógicos y cadenas de caracteres.
4. Operadores aritméticos.

Clase 03

Ingreso de datos

1. Entrada de datos: función input().
2. Funciones de conversión de tipos de datos: int(), float(), str(), etc.
3. Determinación de tipos: función type().
4. Comentarios en el código.

Clase 04

Cadena de caracteres

1. Manipulación de cadenas de caracteres.
2. Funciones y métodos comunes: len(), *, split(), replace(), join() e in
3. f-strings.

The background is a dark grey field with a subtle hexagonal grid pattern. Overlaid on this are several stylized, glowing blue elements: diagonal lines of parallel bars, thin circuit-like lines with small circular nodes, and angular, stepped geometric shapes that resemble futuristic architectural or technological components.

`input()`

La función input()

Código Python

```
alumno = input("Nombre: ")
```

Variable

Función
input

Mensaje

Terminal

Nombre: _

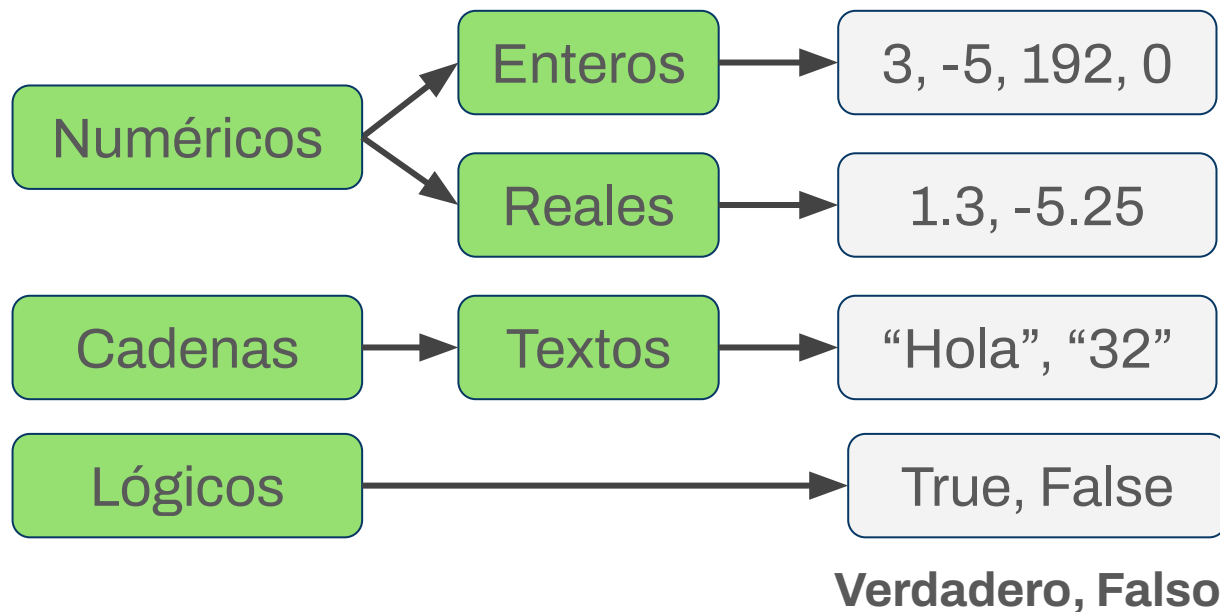
Salida en
la terminal

Conversión de tipos de datos

Tipos de datos



Variable



Conversión de tipos de datos

Código Python

```
base = 10
altura = "5"
area = base * altura / 2
print("Area:", area)
```

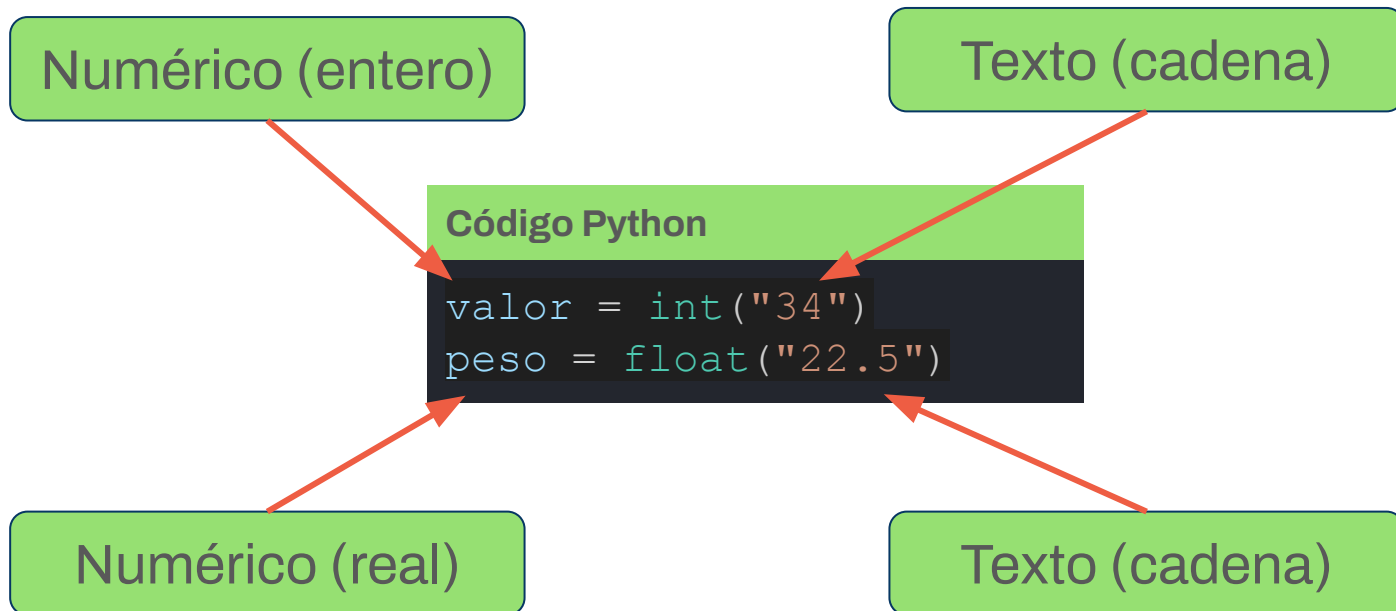
Numérico (entero)

Texto (cadena)

Terminal

```
area = base * altura / 2
TypeError: unsupported operand type(s) for /: 'str' and 'int'
```


Conversión de tipos de datos



Determinar el tipo de dato

Determinar el tipo de dato

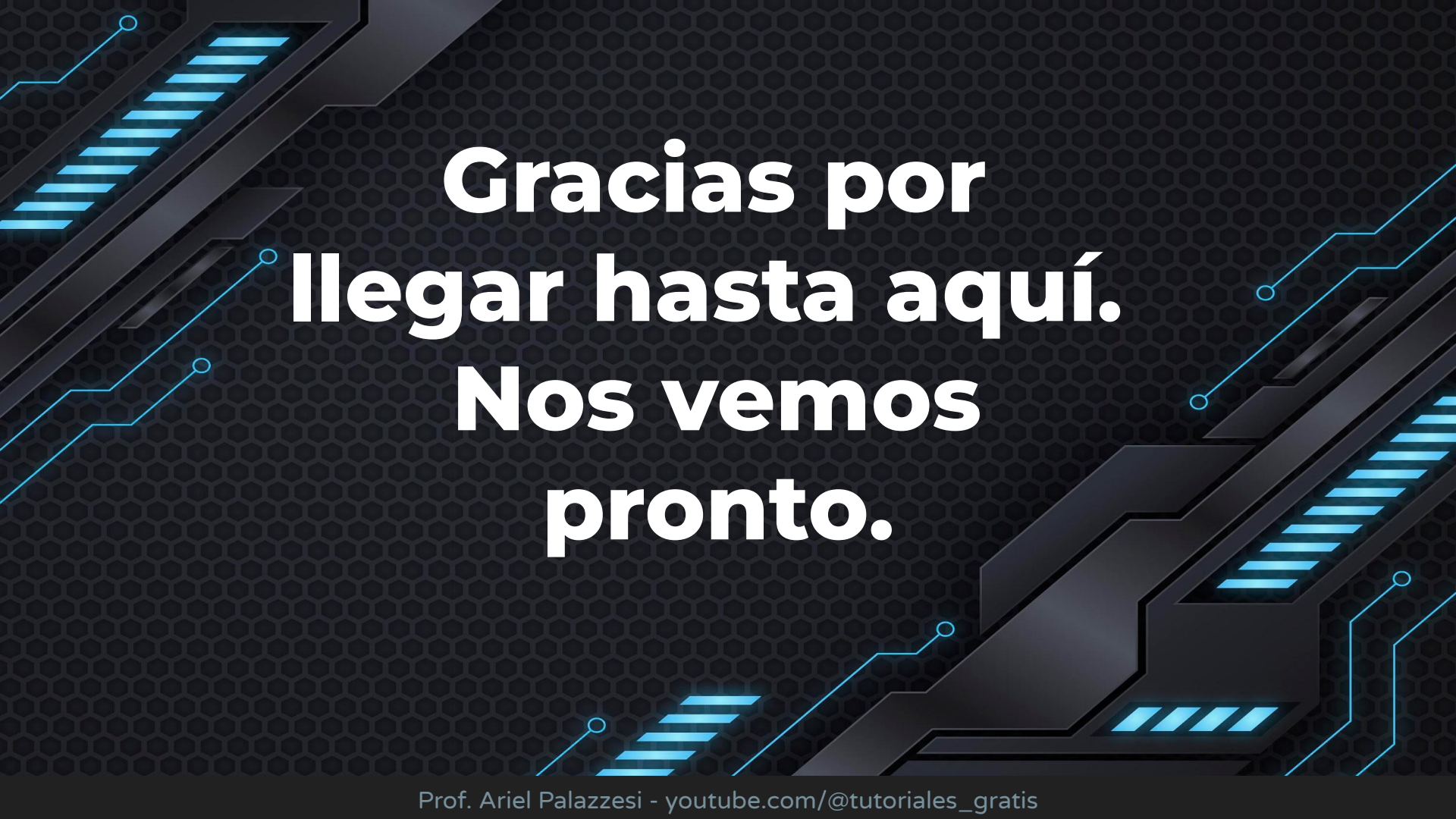
Código Python

```
variable entero = 10
variable texto = "Hola"
variable flotante = 3.14
variable_logica = True

print(type(variable entero))
print(type(variable texto))
print(type(variable flotante))
print(type(variable_logica))
```

Terminal

```
<class 'int'>
<class 'str'>
<class 'float'>
<class 'bool'>
```



**Gracias por
llegar hasta aquí.
Nos vemos
pronto.**