

Bienvenido al curso de Python 🐍 Nivel 1: Python Básico

Diseñado por: @pastranauwu

Curso: Python para principiantes

Objetivo: Dominar los fundamentos del lenguaje, estructuras de datos, control de flujo, funciones, POO básica (polimorfismo, herencia) y manejo de errores.

1.1 Introducción a Python y Tipos de Datos

✓ Teoría:

Python es un lenguaje interpretado, dinámico, multiplataforma y orientado a objetos.

Tipos primitivos:

`int, float, bool, str, list, tuple, set, dict`

Conversiones:

`int("5"), float("3.2"), str(25), list("hola")`

🔧 Ejemplo:

```
edad = 23
nombre = "Eduardo"
promedio = 8.7
es_estudiante = True

lista = [1, 2, 3, 4, 5]
tupla = (1, 2, 3)
diccionario = {"nombre": "Eduardo", "edad": 23, "promedio": 8.7}
```

PROF

1.2 Estructuras de Control

✓ Teoría:

- **Condicionales:** `if, elif, else`
- **Bucles:** `for, while, break, continue, range()`

🔧 Ejemplo:

```
for i in range(5):
    if i == 3:
        print("Llegaste a 3")
```

```
        continue
    print(i)
```

1.3 Funciones

✓ Teoría:

- Declaración: `def`
- Argumentos: posicionales, por defecto, `*args`, `**kwargs`
- Retorno con `return`
- Funciones lambda

Ejemplo:

```
def saludar(nombre="Invitado"):
    return f"Hola, {nombre}!"

print(saludar("Eduardo"))

# Lambda
cuadrado = lambda x: x ** 2
```

1.4 Listas y List Comprehension

✓ Teoría:

- Métodos comunes: `.append()`, `.remove()`, `.sort()`, `.pop()`
- **List comprehension:**

```
[x for x in range(10) if x % 2 == 0]
```

Ejemplo:

```
numeros = [1, 2, 3, 4, 5]
pares = [x for x in numeros if x % 2 == 0]
```

1.5 Diccionarios, Sets y Tuplas

✓ Teoría:

- **Diccionarios (`dict`):** clave-valor, `.get()`, `.items()`, `.keys()`

- **Sets** (**set**): elimina duplicados
- **Tuplas** (**tuple**): inmutables

Ejemplo:

```
persona = {"nombre": "Eduardo", "edad": 23}
print(persona["nombre"])

frutas = {"manzana", "pera", "manzana"}
```

1.6 Manejo de Errores

Teoría:

- `try`, `except`, `finally`, `raise`
- Múltiples excepciones

Ejemplo:

```
try:
    resultado = 10 / 0
except ZeroDivisionError:
    print("Error: división entre cero.")
```

1.7 Programación Orientada a Objetos (POO)

Teoría:

- `class`, `__init__`, atributos y métodos
- **Herencia**
- **Polimorfismo**: sobrescritura de métodos (override)

Ejemplo:

```
class Animal:
    def hablar(self):
        return "Hace sonido"

class Perro(Animal):
    def hablar(self):
        return "Ladra"

a = Perro()
print(a.hablar())
```

