

Fundamentos de Python:

Un Análisis de Ejercicios Prácticos
(De Cero a Mini-Sistemas)

¡Hola, futuros programadores!

Soy un especialista en Python. Hoy no solo veremos código; veremos **CÓMO** pensar como programadores analizando 4 ejercicios clave.

Los Pilares de Hoy



Concepto 1: Variables y Datos

Captura y cálculo de datos básicos.
(Ejercicio 1: Factura con IVA)



Concepto 2: Lógica y Filtros

Validar entradas y filtrar
información útil.
(Ej. 2 y 4: Validación y
Recomendaciones)



Concepto 3: Estructuras y Librerías

Organizar datos (Dicts) y usar
herramientas (Pandas).
(Ej. 3 y 4: Películas y Ventas)

Concepto 1: Variables y Operadores

Todo empieza aquí. Una variable es una "caja" con nombre para guardar datos (un precio, un nombre, una cantidad).

Usamos **operadores** (+, -, *, /) para transformar esos datos.

Ejemplo (Ejercicio 1):

```
precio_unitario = 15.0
```

```
cantidad = 2
```

```
subtotal = precio_unitario * cantidad
```


El Poder del `input()`

`input()`

La Puerta de Entrada

Capturando Datos del Usuario


El código no es estático. `input()` pausa el programa y espera que el usuario escriba algo. Es la forma más simple de interactividad.

Importante: ¡ `input()` siempre devuelve texto (string)! Por eso usamos `float()` o `int()` para convertirlo a número.

Concepto 2: Lógica y Validación

Un buen programa no se "rompe" fácilmente. Usamos `if` (si...) para tomar decisiones y validar los datos **antes** de usarlos.

 ¿Qué pasa si el usuario ingresa "-5" como cantidad? El cálculo fallaría o daría un resultado sin sentido.

 **La Solución (Ejercicio 2):** Usamos un `if` para "blindar" el código.

```
if largo ≤ 0 or ancho ≤ 0:
```

 **El Principio:** "Programación Defensiva". Protegemos nuestro código de datos inesperados o incorrectos.

Concepto 3: Estructuras de Datos

Diccionarios { }

(Ejercicio 4) Un diccionario guarda datos usando un sistema de "llave" y "valor". Es perfecto para agrupar datos relacionados.

```
películas = { "Interestelar": 0 }
```

(Llave: "Interestelar", Valor: 0)

Librerías (Pandas)

(Ejercicio 3) ¡No tienes que programarlo todo! Una librería es código que alguien más escribió. **Pandas** es una librería para manejar datos en tablas.

Es como tener un Excel súper-poderoso dentro de Python.

Sobre las Librerías...

“

No reinventes la rueda, a menos
que quieras aprender a hacer
ruedas.

”

— Dicho popular en programación (adaptado)

Usar `pandas` para un reporte de ventas es usar la mejor "rueda" disponible.

Concepto 4: Bucles y Menús



El Bucle `while True:`

¿Cómo hacemos que un programa se ejecute hasta que el usuario decida salir? Con un bucle (loop).

`while True:` crea un **bucle infinito** que repite el "menú principal" (Ejercicios 3 y 4).

El bucle solo se detiene cuando el usuario elige "Salir" y usamos la palabra clave `break` para "romperlo".

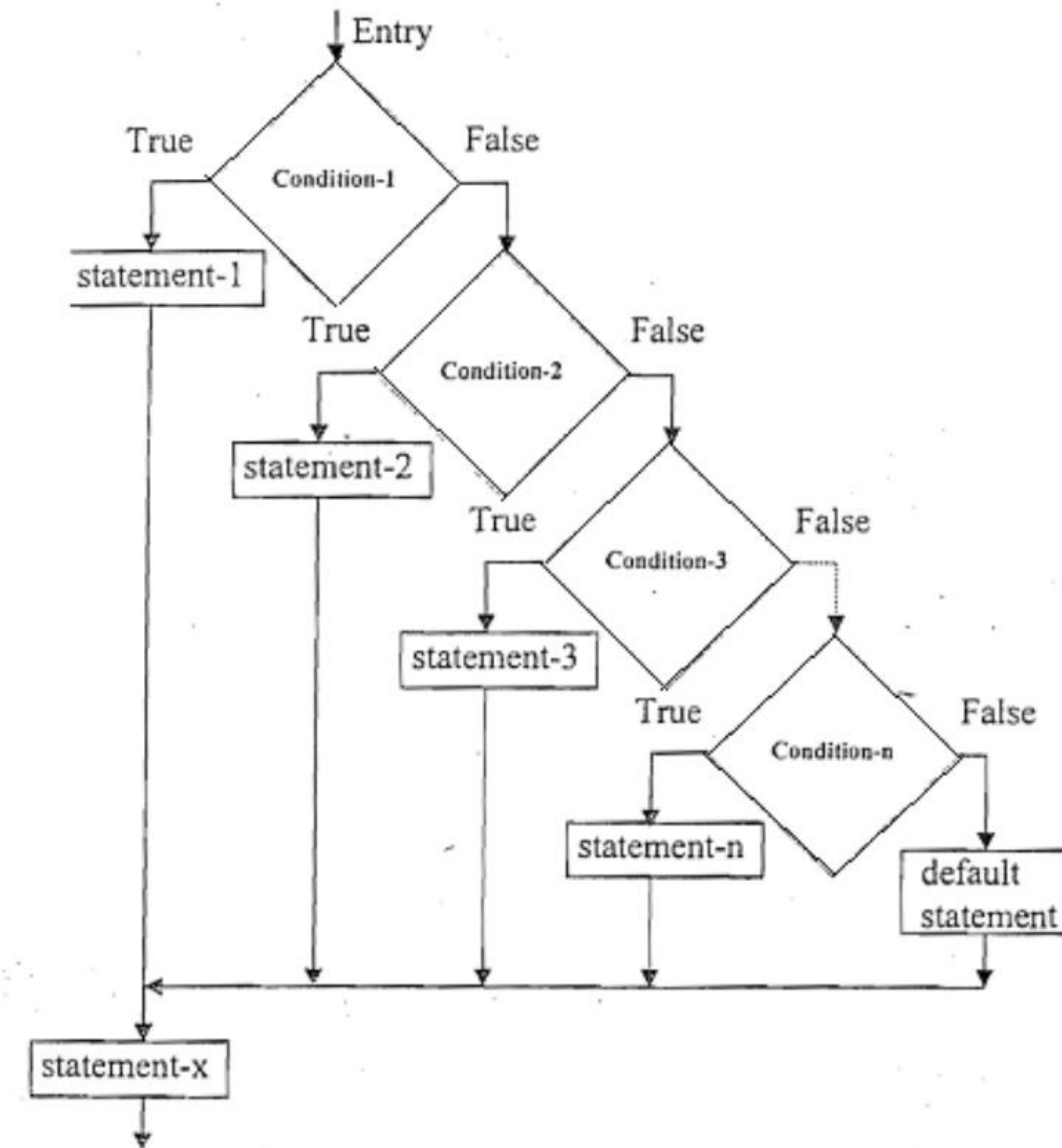
Control de Flujo: match-case

Manejando Menús (El modo moderno)

Nuevo en Python (3.10+), `match-case` es una forma limpia de manejar menús. Es como un `if ... elif ... else ...` pero más legible.

En lugar de múltiples `if` anidados, usamos `match` sobre la `opcion` del usuario:

```
match opcion:
    case '1':
        registrar_venta()
    case '2':
        generar_corte()
    case _: // Default
        print("Opción no válida")
```



Tu Camino Apenas Comienza

Vimos cómo de `precio * cantidad` (Ex 1) llegamos a un sistema de ventas con reportes en Pandas (Ex 3). La clave es combinar estos bloques:

Variables + Lógica + Estructuras + Bucles

¿Preguntas?

El código es su nuevo lenguaje. ¡La única forma de dominarlo es practicando!

¡Sigam programando!

Image Sources



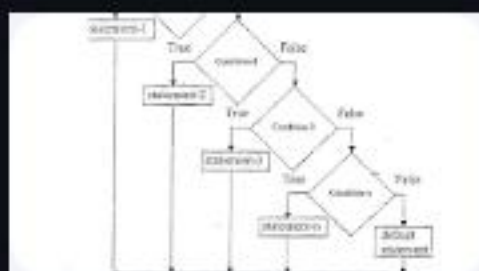
https://img.freepik.com/free-photo/minimalist-blue-white-wave-background_1017-46756.jpg?semt=ais_hybrid&w=740&q=80

Source: www.freepik.com



<https://slidemodel.com/wp-content/uploads/21912-01-infinite-loop-powerpoint-template-16x9-1.jpg>

Source: slidemodel.com



<https://www.codewithc.com/wp-content/uploads/2014/07/if.png>

Source: www.codewithc.com