A graphic on the left side of the slide consisting of four stacked, 3D rectangular blocks in purple, orange, yellow, and blue. An orange arrow points to the right, passing through the middle of the blocks.

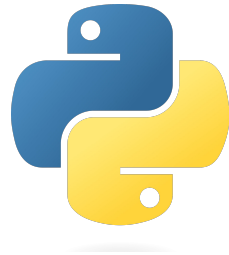
Agencia de
Aprendizaje
a lo largo
de la vida

CODO A CODO INICIAL

Clase 16

Python 4

Condicionales II



Les damos la bienvenida

Vamos a comenzar a grabar la clase

Clase 15

Python 3 - Condicionales I

- Estructuras condicionales.
- Uso de "if" y "else" para tomar decisiones en un programa.
- Diferencias de sintaxis entre Python y PSeint

Clase 16

Python 4 - Condicionales II

- Estructuras condicionales.
- Uso de "if", "elif" y "else" para tomar decisiones en un programa.
- Uso de operadores lógicos.
- Uso de Match.
- Diferencias de sintaxis entre Python y PSeint

Clase 17

**Python 5 - Funciones I:
Introducción**

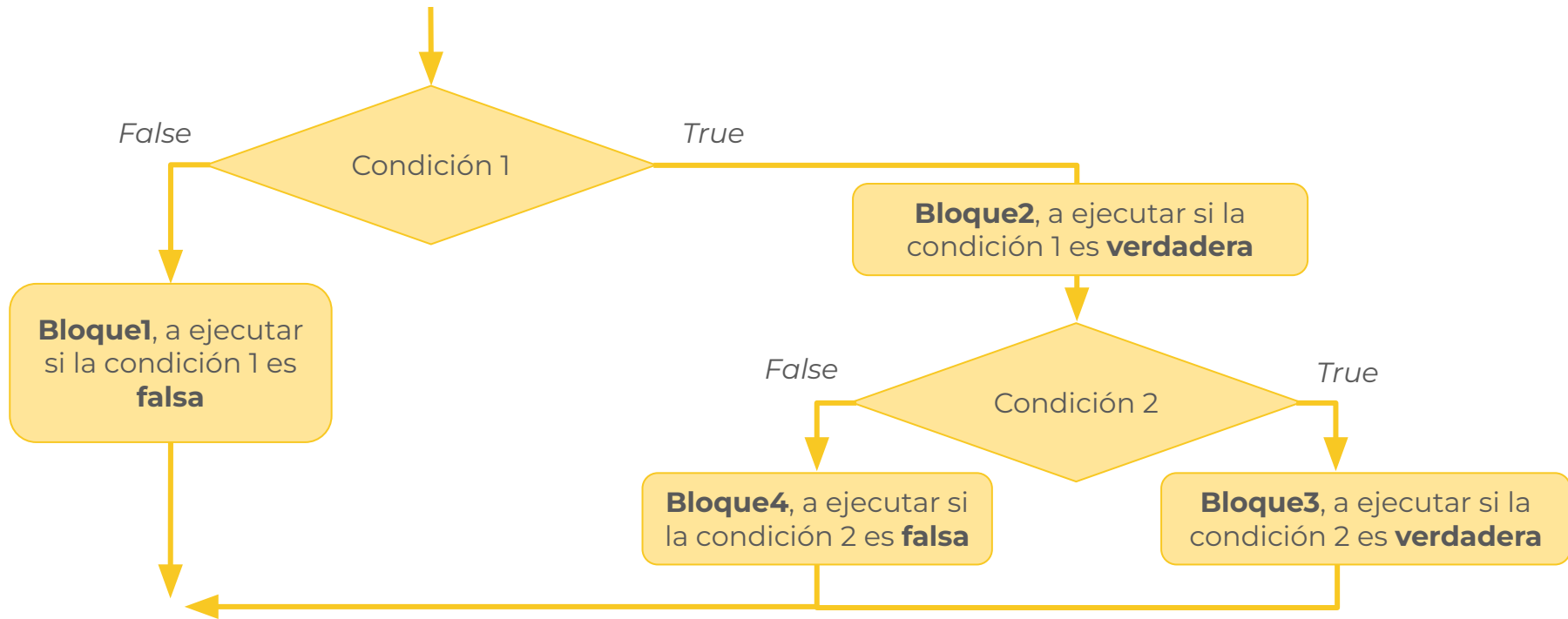
- Definición
- Propósito. Ventajas
- Funciones integradas en el lenguaje (len(), print(), input(), etc.)
- Sintaxis para declarar funciones.
- Ejemplos de funciones sencillas.

Condicionales anidados

Las **estructuras condicionales anidadas** en Python se refieren a la inclusión de una o más instrucciones **"if"** dentro de otra instrucción **"if"** o **"else"**. Esto permite evaluar múltiples condiciones en cascada para tomar decisiones más complejas en el código.

En el anidamiento de estas estructuras es clave el uso de las sangrías o indentaciones para delimitar el inicio y fin de un bloque, además de la **prueba de escritorio**, que consiste en darle valores al programa para saber cómo se comporta.

Estructuras condicionales anidadas



Estructuras condicionales anidadas

En una **estructura condicional anidada**, cada **else** se corresponde con el **if** más próximo que no haya sido emparejado, y deben tener la misma indentación.

En el esquema anterior, se evalúa primero la condición 1. En caso de ser falsa se ejecuta el **Bloque1** que se encuentra en su **else**, y finaliza la ejecución de la estructura.

En caso de que la condición 1 sea verdadera, se ejecuta el **Bloque2** y se procede a evaluar la condición 2, que se encuentra dentro (anidada) del primer **if**.

Si la segunda condición resulta verdadera se ejecuta el **Bloque3**, si resulta falsa se ejecuta el **Bloque4**.

Estructuras condicionales anidadas | Ejemplo

```
usuarioRegistrado = True
contrasenaCorrecta = True

if usuarioRegistrado == True: # Condición 1
    B print("Usuario registrado. Por favor, ingrese su contraseña.")
    if contrasenaCorrecta: # Condición 2
        C print("Contraseña correcta. ¡Bienvenido al sistema!")
    else:
        D print("Contraseña incorrecta. Acceso denegado.")
else:
    A print("Usuario no registrado. Por favor, regístrese para acceder.")
```

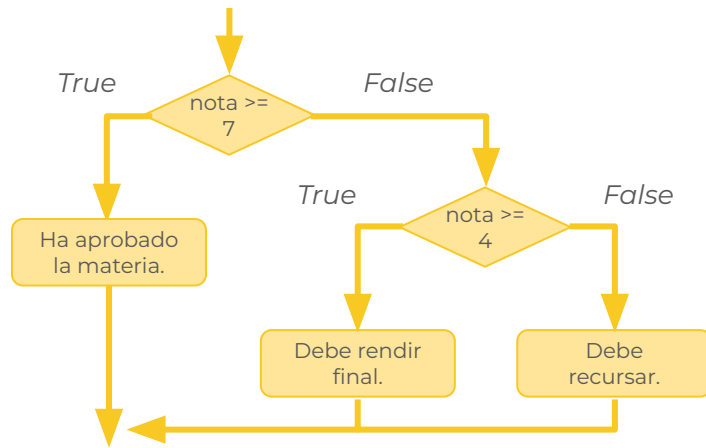
- A. Bloque1 (si es **falsa** la condición 1).
- B. Bloque2 (si es **verdadera** la condición 1).
- C. Bloque3 (si es **verdadera** la condición 1 y es **verdadera** la condición 2)
- D. Bloque3 (si es **verdadera** la condición 1 y es **falsa** la condición 2)

Estructuras condicionales anidadas | Otro ejemplo

En este ejemplo se solicita una nota y se determina el resultado del estudiante. Si la nota ingresada es mayor o igual a 7, ha aprobado. Si está entre 4 y 6 (inclusive) debe rendir examen. Si es menor a 4 debe recursar.

```
nota = int(input("Ingrese la nota: "))

if nota >= 7:
    print("Ha aprobado la materia.")
else:
    if nota >= 4:
        print("Debe rendir examen.")
    else:
        print("Debe recursar.")
```



Estructuras condicionales | if .. elif .. else

A menudo solemos hacer una pregunta a partir de la respuesta de una pregunta anterior. Python puede implementar este comportamiento usando la estructura **if .. elif .. else**.

La sección de código dentro del **elif** se ejecuta cuando la condición del primer **if** ha resultado ser falsa y la condición del **elif** es verdadera. Si la condición del **elif** es falsa, entonces se ejecuta el código del bloque **else**.

Además, en situaciones más complejas se pueden utilizar **múltiples instancias de elif**, dando lugar a estructuras condicionales elaboradas y que permiten resolver prácticamente cualquier situación, utilizando varios bloques de instrucciones diferentes.

Estructuras condicionales | if .. elif .. else

Volviendo al ejemplo de las notas, supongamos que debemos **agregar más condiciones** y anidar múltiples estructuras condicionales.

En este ejemplo, de acuerdo a la nota aparecen distintos mensajes. Si bien podemos anidar múltiples **if ... else**, Python nos ofrece la posibilidad de utilizar **elif**.

```
nota = int(input("Ingrese la nota: "))

if nota == 10:
    print("Desempeño excelente!")
elif nota >= 8:
    print("Muy buen desempeño!")
elif nota >= 6:
    print("Buen desempeño")
elif nota >= 4:
    print("Desempeño regular")
else:
    print("Debes esforzarte más")
```

Estructuras condicionales | if .. elif .. else

Explicación del ejemplo anterior:

```
nota = int(input("Ingrese la nota: "))  
  
if nota == 10:  B  
    print("Desempeño excelente!") C  
elif nota >= 8:  
    print("Muy buen desempeño!") D  
elif nota >= 6:  
    print("Buen desempeño") D  
elif nota >= 4:  
    print("Desempeño regular") D  
else:  
    print("Debes esforzarte más") E
```

A

- A. Estructura if...elif...else.
- B. Condición del if...
- C. Instrucciones que se ejecutan si la condición del primer if es **verdadera**.
- D. Instrucciones que se ejecutan si la condición del primer if es **falsa** y la del elif es **verdadera**.
- E. Instrucciones que se ejecutan si las condiciones anteriores son falsas.

Otro ejemplo de estructura condicional anidada

```
clima = input("¿Cómo está el clima hoy? (soleado/nublado/lloviendo): ")

if clima == "soleado":
    print("¡Hay sol! Podrías llevar ropa ligera.")
    temperatura = float(input("¿Cuántos grados hace?: "))
    if temperatura > 30:
        print("¡Hace mucho calor! Un sombrero y protector solar podrían ser útiles.")
    elif temperatura >= 20:
        print("La temperatura es agradable.")
    else:
        print("Hace calor, pero no demasiado. Un suéter ligero podría ser útil.")
elif clima == "nublado":
    print("Está nublado. Un suéter podría ser necesario.")
else:
    print("Está lloviendo. ¡No olvides el paraguas y una chaqueta impermeable!")
```

Uso de operadores lógicos

Se pueden utilizar **operadores lógicos** para tomar una decisión basada en múltiples condiciones, reduciendo la cantidad de **ifs anidados**:

```
if condicion1 and condicion2:  
if condicion1 or condicion2:
```

Las **tablas de verdad** muestran los valores de verdad de una proposición en función del valor lógico de sus operadores:

Cond 1	Cond 2	Y (and)
V	V	V
V	F	F
F	V	F
F	F	F

Cond 1	Cond 2	O (or)
V	V	V
V	F	V
F	V	V
F	F	F

Cond	NO (not)
V	F
F	V

Uso de operadores lógicos | and

Veamos un ejemplo de un **condicional** con **and**:

En un aviso del diario piden **ingenieros en sistemas con 5 años de experiencia como mínimo**, para ocupar un puesto laboral. A la convocatoria se presenta:

- Un **licenciado en sistemas** con **6 años de experiencia**: NO LO TOMAN, pues la primera condición es falsa.
- Un **ingeniero en sistemas** con **4 años de experiencia**: NO LO TOMAN, pues la segunda condición es falsa.
- Un **analista programador** con **4 años de experiencia**: NO LO TOMAN, pues las 2 condiciones son falsas.
- Un **ingeniero en sistemas** con **7 años de experiencia**: LO TOMAN, pues las 2 condiciones son verdaderas.

Uso de operadores lógicos | and

```
puesto = "Ingeniero en sistemas"
experiencia = 6

if puesto == "Ingeniero en sistemas" and experiencia >= 5:
    print("Ha obtenido el puesto laboral")
else:
    print("No cumple con los requisitos")
```

En este ejemplo solamente podrán obtener el puesto laboral aquellos que sean **“Ingeniero en sistemas”** y tengan **5 o más años de experiencia laboral**.

Si alguno de los dos requisitos no se cumplen la leyenda que aparece es *“No cumple con los requisitos”*.

Uso de operadores lógicos | or

Veamos un ejemplo de un **condicional** con **or**:

Tengo invitados en casa y voy a comprar 1 kilo de helado. Sé que los únicos gustos que comen son chocolate o vainilla. Después de ir a varias heladerías encontré:

- Hay **chocolate** pero **no hay vainilla**. LO COMPRO, pues la primera condición es verdadera.
- Sólo hay **vainilla**, **no chocolate**. LO COMPRO, pues la segunda condición es verdadera.
- Hay **chocolate** y **vainilla**. LO COMPRO, pues las dos condiciones son verdaderas.
- Hay **crema americana** y **dulce de leche**. NO LO COMPRO, pues ninguna de las condiciones es verdadera.

Uso de operadores lógicos | or

```
sabor1 = "Chocolate"
sabor2 = "Vainilla"

if sabor1 == "Chocolate" or sabor2 == "Vainilla":
    print("Compro el kilo de helado")
else:
    print("No compro el kilo de helado")
```

En este ejemplo el kilo de helado se compra si se cumple con **alguno** de los requisitos de sabor: debe ser de Chocolate o de Vainilla.

Si ninguno de los dos requisitos se cumple la leyenda que aparece es “*No compro el kilo de helado*”.

Condicional múltiple | Uso de Match

PSeint dispone del condicional “Según...”. Python, a partir de la versión 3.10, posee “**Match**”, una *declaración de coincidencia* que permite realizar acciones diferentes según los valores de una variable. Utiliza **declaraciones de caso** (case statements) para especificar los casos en los que exista coincidencia.

Se puede utilizar “_” como *declaración de caso* para especificar un bloque de instrucciones a ejecutar cuando ninguno de los casos anteriores se cumple.

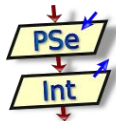
```
dia = input("Introduzca un número de día: ")
match dia:
    case "1":
        print("Es lunes")
    case "2":
        print("Es martes")
    case "3":
        print("Es miércoles")
    case "4":
        print("Es jueves")
    case "5":
        print("Es viernes")
    case "6":
        print("Es sábado")
    case "7":
        print("Es domingo")
    case _:
        print("Día no válido")
```

Condicionales anidados | PSeint vs. Python

Algoritmo condicionalAnidado

```
Escribir "¿Temperatura actual?: "  
Leer temperatura  
Si temperatura >= 30 Entonces  
    Escribir "Hace calor"  
SiNo  
    Si temperatura >= 20 Entonces  
        Escribir "Está agradable"  
    SiNo  
        Escribir "Hace frío"  
Fin Si  
Fin Si
```

FinAlgoritmo



```
temperatura = float(input("¿Temperatura  
actual?: "))  
  
if temperatura >= 30:  
    print("Hace calor")  
else:  
    if temperatura >= 20:  
        print("Está agradable")  
    else:  
        print("Hace frío")
```



Este programa solicita la temperatura y determina si es hace calor, está agradable o hace frío en base a valores predefinidos.

Operadores lógicos | PSeint vs. Python

Algoritmo operadoresLogicosAnd

```
    Escribir "Ingrese la edad: "  
    Leer edad  
    Escribir "¿Tiene registro al día?  
1-SI, 0-NO: "  
    Leer registro
```

```
    Si edad >= 18  $\wedge$  registro == 1
```

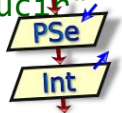
Entonces

```
    Escribir "Puede conducir"
```

SiNo

```
    Escribir "No puede conducir"
```

Fin Si



FinAlgoritmo

```
edad = int(input("Ingrese la edad: "))  
registro = int(input("¿Tiene registro al  
día? 1-SI, 0-NO: "))
```

```
if edad >= 18 and registro == 1:  
    print("Puede conducir")  
else:  
    print("No puede conducir")
```



Este programa solicita la edad y verifica si el usuario tiene el registro de conducir al día. Si la edad es igual o mayor a 18 y el registro está actualizado (valor 1), imprime "Puede conducir". En caso contrario, imprime "No puede conducir".

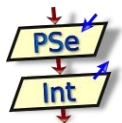
Operadores lógicos | PSeint vs. Python

Algoritmo operadoresLogicosOr

```
    Escribir "Ingrese el nombre del
color:"
    Leer color

    Si color == "Rojo" V color == "Verde"
V color == "Azul" Entonces
        Escribir "Es un color primario"
    SiNo
        Escribir "No es un color
primario"
    Fin Si

FinAlgoritmo
```



```
color = input("Ingrese el nombre del
color: ")

if color == "Rojo" or color == "Verde"
or color == "Azul":
    print("Es un color primario")
else:
    print("No es un color primario")
```



Este programa solicita el nombre del color y determina si pertenece a los colores primarios (rojo, verde y azul) o no, imprimiendo una leyenda en cada caso.

Condicional múltiple | PSeint vs. Python

Algoritmo usoDeSegun

Escribir "Seleccione una opción: 1-
Extraer, 2- Depositar, 3- Consultar Saldo, 4-
Salir:"

Leer opc

Segun opc **Hacer**

1:

Escribir "Ingresando a
extracciones..."

2:

Escribir "Ingresando a
depósitos..."

3:

Escribir "Consultando saldo..."

4:

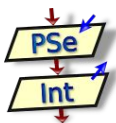
Escribir "Saliendo..."

De Otro Modo:

Escribir "Opción incorrecta!"

Fin Segun

FinAlgoritmo



```
opc = int(input("Seleccione una opción: 1-  
Extraer, 2- Depositar, 3- Consultar Saldo, 4-  
Salir: "))
```

```
match opc:  
    case 1:  
        print("Ingresando a extracciones...")  
    case 2:  
        print("Ingresando a depósitos...")  
    case 3:  
        print("Consultando saldo...")  
    case 4:  
        print("Saliendo...")  
    case _:  
        print("Opción incorrecta!")
```



Este programa simula un cajero automático, ofreciendo distintas leyendas de acuerdo a la selección del usuario.

Desafíos

Desafío 1: Descuentos en tienda de ropa

Escribe un programa en Python que solicite al usuario el monto total de la compra y la cantidad de artículos que está comprando. El programa debe determinar el descuento aplicable según las siguientes reglas:

- Si la cantidad de artículos comprados es mayor o igual a 5 y el monto total es mayor a \$10000, aplica un descuento del 15%.
- Si la cantidad de artículos comprados es menor a 5 pero mayor o igual a 3, aplica un descuento del 10%.
- Si la cantidad de artículos comprados es menor a 3, no se aplica descuento.

Al final, el programa debe imprimir el monto total de la compra después de aplicar cualquier descuento o simplemente el monto original si no hay descuento.

Desafío 2: Categorización de productos según su tipo

Escribe un programa en Python que solicite al usuario ingresar el código de un producto y determine a qué categoría pertenece según las siguientes reglas:

- Si el código comienza con "A", el producto es de tipo "Electrónico".
- Si el código comienza con "B", el producto es de tipo "Ropa".
- Si el código comienza con "C", el producto es de tipo "Alimenticio".
- Si el código no coincide con ninguna de las anteriores, el programa debe mostrar un mensaje indicando que el tipo de producto es "Desconocido".

Importante: No utilizar condicionales anidados para resolver este ejercicio.

Material extra

Artículos de interés

Material extra:

- [Aprende los condicionales en Python con ejemplos](#) | Pythones
- [Match...Case en Python](#) | Johan Moises Ariza Mahecha

Videos:

- [if...elif...else](#) | Tutoriales sobre Ciencia y Tecnología
- [if...elif...else](#) | Programación ATS
- [Match en Python](#) | TecnoBinaria

No te olvides de dar el presente

Recordá:

- Revisar la Cartelera de Novedades.
- Hacer tus consultas en el Foro.
- Realizar el Ejercicio de Repaso.

Todo en el Aula Virtual.

Muchas gracias por tu atención.

Nos vemos pronto