

A graphic on the left side of the slide consisting of four stacked horizontal bars in purple, orange, yellow, and blue. The text "Agencia de Aprendizaje a lo largo de la vida" is written across these bars. The orange bar has a white arrow pointing to the right.

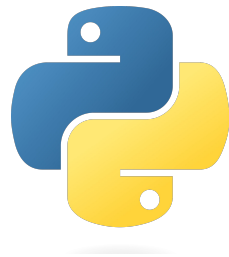
Agencia de  
Aprendizaje  
a lo largo  
de la vida

# **CODO A CODO INICIAL**

## **Clase 15**

Python 3

# Condicionales I



# Les damos la bienvenida

Vamos a comenzar a grabar la clase

## Clase 14

**Python 2 - Tipos de Datos, Entrada y Salida de Información**

- Exploración de tipos de datos
- Cadena como un "contenedor" de caracteres
- Entrada y salida de información por consola
- Ejemplos prácticos de entrada y salida

## 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 Switch.
- Diferencias de sintaxis entre Python y PSeint

# Estructuras de control

A diario actuamos de acuerdo a la evaluación de condiciones, incluso de manera inconsciente. Si el semáforo está en verde, cruzamos la calle. Si no, esperamos. También es frecuente evaluar más de una condición a la vez: Si llega la factura de un servicio y tengo dinero, entonces lo pago.

En Python utilizamos las **estructuras de control de flujo condicionales** para tomar decisiones similares, evaluando expresiones en las que suelen intervenir variables, para determinar qué parte del código que hemos escrito se va a ejecutar.

# Bloques de código (Indentación)

El código puede agruparse en bloques, que delimitan sentencias relacionadas. Python, al igual que PSeInt, utiliza la **indentación o sangrado**, que consiste en mover el bloque de código hacia la derecha insertando **espacios o tabuladores** al principio de la línea, dejando un margen a su izquierda.

Un bloque comienza con un nuevo sangrado y acaba con la primera línea cuyo sangrado sea menor. La guía de estilo de Python recomienda usar espacios en lugar de tabulaciones. Para realizar el sangrado, se suelen utilizar 4 espacios.

No todos los lenguajes de programación necesitan de una indentación, aunque sí se estila implementarla a fin de otorgar mayor legibilidad al código fuente. Pero **en el caso de Python, la indentación es obligatoria**, ya que de ella dependerá su estructura.

# Bloques de código (Indentación)

```
def suma_numeros(numeros): # Bloque 1
    suma = 0                # Bloque 2
    for n in numeros:       # Bloque 2
        suma += n           # Bloque 3
        print(suma)         # Bloque 3
    return suma             # Bloque 2
print()                    # Bloque 1
```

Si bien aún no sabemos exactamente qué hace el código anterior, se pueden ver los bloques de instrucciones indicados mediante los tabuladores sobre el margen izquierdo. Es posible incluir un bloque dentro de otro, para crear estructuras complejas.

# Estructuras de control

En programación, las estructuras de control permiten modificar el flujo de ejecución de las instrucciones de un programa. Con ellas se puede ejecutar un grupo u otro de sentencias, según se cumpla o no una condición (**if**).

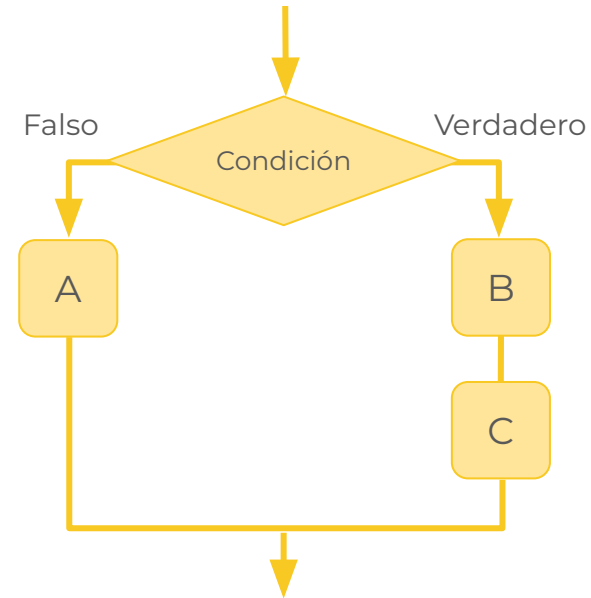
También existen estructuras de control que permiten ejecutar un grupo de sentencias mientras se cumpla una condición (**while**) o repetir un grupo de sentencias, que veremos más adelante, un número determinado de veces (**for**).



# Estructuras de control

Antes de comenzar con las estructuras de control condicionales, debemos recordar que en el código de un programa podemos encontrar bloques de ejecución **secuencial**, donde las instrucciones se ejecutan una después de la otra, en el orden en que están escritas, es decir, *en secuencia*.

Las estructuras **condicionales** ejecutan un bloque de instrucciones u otro, o saltan a otro según se cumpla o no una condición.



*Estructura de  
selección o decisión*

# Ejecución en secuencia

Las acciones se llevan a cabo en sucesión, cada instrucción se ejecuta de manera consecutiva, y el flujo de control se desplaza de una acción a la siguiente en orden lineal, de arriba hacia abajo. Todas las instrucciones se procesan una única vez.

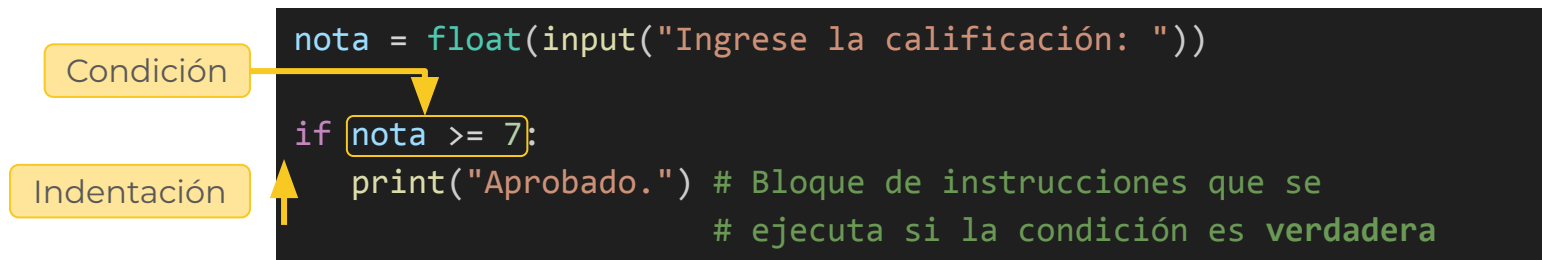
```
#Programa Suma: suma dos números enteros ingresados por teclado
nro1 = int(input("Ingrese el primer número: "))
nro2 = int(input("Ingrese el segundo número: "))
suma = nro1 + nro2
print("La suma es:", suma)
```

El código del ejemplo pide un número, luego pide otro, realiza la suma de ambos valores guardando el resultado en suma y finalmente muestra un mensaje por pantalla.

# Estructuras condicionales | if

Las estructuras condicionales tienen como objetivo ejecutar un bloque de instrucciones u otro en base a una condición que puede ser verdadera o falsa. La palabra clave asociada a esta estructura es **if**.

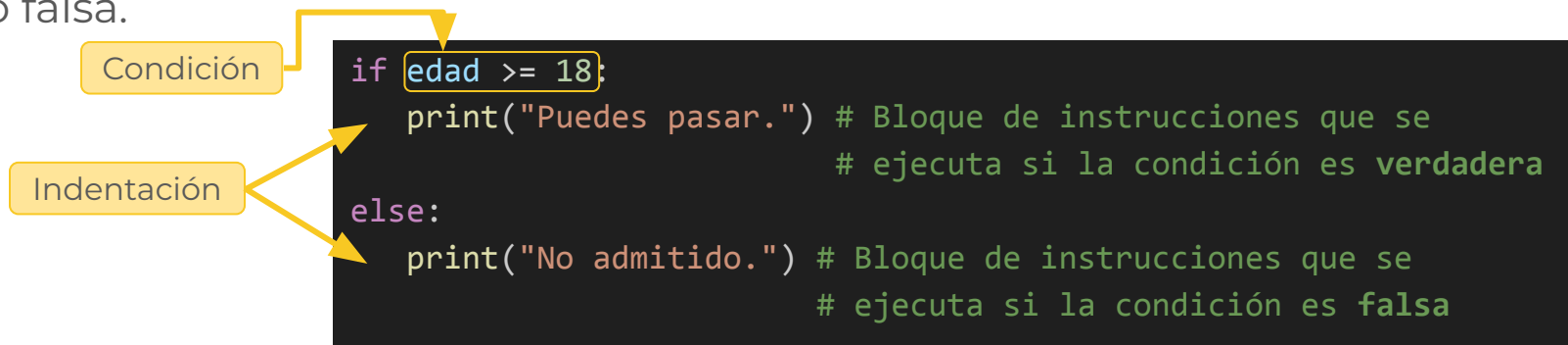
Si la condición es **True** se ejecuta el bloque dentro del **if**. Luego, independientemente del valor de verdad de la condición, el programa continúa con la ejecución del resto del programa.



# Estructuras condicionales | if ... else

En Python podemos utilizar la cláusula **else** para determinar un grupo de instrucciones que se ejecutará en caso de que la evaluación de la condición resulte ser **falsa**.

Con este agregado, una estructura **if** tiene la posibilidad de ejecutar un bloque de instrucciones u otro, dependiendo de si la condición es verdadera o falsa.



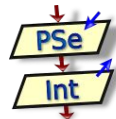
# Estructuras condicionales | PSeint vs. Python

En este ejemplo vemos la sintaxis de una estructura condicional **Si .. SiNo / If ... Else** en Pseint y Python, respectivamente:

**Algoritmo** estructuraCondicional

```
    Escribir "Ingrese su nota: "  
    Leer nota  
    Si nota >= 6 Entonces  
        Escribir "Está aprobado"  
    SiNo  
        Escribir "No está aprobado"  
    Fin Si
```

**FinAlgoritmo**



```
nota = float(input("Ingrese su nota: "))  
if nota >= 6:  
    print("Está aprobado")  
else:  
    print("No está aprobado")
```



*El programa solicita un nota y determina si es mayor o igual a 6. En caso afirmativo el estudiante estará “aprobado” en caso contrario estará “No aprobado”.*

# Comparaciones con texto

Además de utilizar valores numéricos, en las condiciones podemos comparar cadenas de texto. En código se ingresa el importe y la forma de pago (A o B). Esta elección modifica el valor final del importe:

```
importe = int(input("Ingrese el monto de la compra: "))
formaPago = input("¿Qué forma de pago prefiere? A: Contado; B: Otra: ")
if formaPago == "A":
    importe = importe * 0.9
else:
    importe = importe * 1.15
print("El importe a pagar es:", importe)
```

# Desafíos

# Desafío 1: Comparación de cadenas

Escribe un programa que solicite al usuario dos cadenas de caracteres. Por ejemplo: dos nombres.

El programa debe ser capaz de comparar ambas cadenas y mostrar el mensaje “*Son iguales*” si ambas cadenas lo son, o mostrar el mensaje “*No son iguales*” si no lo son. Ejemplo:

```
Ingrese una cadena: PSeint
Ingrese otra cadena: PSeint
Las cadenas son iguales.
```

```
Ingrese una cadena: Python
Ingrese otra cadena: Phyton
Las cadenas no son iguales.
```



## Desafío 2: Tope máximo de extracción

Escribe un programa para un cajero automático de un banco que permita ingresar un importe y determine si la extracción puede hacerse o el usuario debe realizarla por la caja. El tope máximo es de \$ 20000. Ejemplo:

```
Ingrese importe a extraer: 18000  
La extracción puede realizarse.
```

```
Ingrese importe a extraer: 23000  
La extracción debe realizarse por caja.
```

# Material extra

# Artículos de interés

Material extra:

- [Bloques de código y sintaxis en Python](#) | El libro de Python
- [If y else en Python](#) | Oregoom
- [Condicionales en Python](#) | Programación.net

Videos:

- [Estructura if](#), [estructura if...else](#) y [ejemplo de uso](#) | Tutoriales sobre ciencia y tecnología

# 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**