



TIC



# Actividad 6

Estructuras de Control en Python

# Estructuras de Control en Python



TIC



## Estructuras de Control if:

La estructura de control 'if' en Python se utiliza para tomar decisiones en base a condiciones. Permite ejecutar un bloque de código si una expresión es verdadera (True) y, opcionalmente, ejecutar un bloque alternativo si la expresión es falsa (False). Aquí tienes la estructura básica y algunos ejemplos:

## Estructura Básica del 'if':

```
if condicion:  
    # Código a ejecutar si la condición es verdadera  
else:  
    # Código a ejecutar si la condición es falsa
```

# Estructuras de Control en Python



TIC



## Ejemplo Simple:

```
edad = 18

if edad >= 18:
    print("Eres mayor de edad.")
else:
    print("Eres menor de edad.")
```

En este ejemplo, el programa evalúa la condición `edad >= 18`. Si es verdadera, imprime "Eres mayor de edad.". Si es falsa, imprime "Eres menor de edad.".



TIC



# Estructuras de Control en Python

## Uso de 'elif' para Múltiples Condiciones:

En este caso, se utiliza 'elif' (contracción de "else if") para evaluar múltiples condiciones en orden. La primera condición que es verdadera se ejecutará, y si ninguna es verdadera, se ejecutará el bloque 'else'.

```
nota = 75

if nota >= 90:
    print("A")
elif 80 <= nota < 90:
    print("B")
elif 70 <= nota < 80:
    print("C")
else:
    print("No aprobado")
```



TIC



# Estructuras de Control en Python

## Uso de Operadores Lógicos:

Aquí se emplean operadores lógicos (and, not) para combinar condiciones. El bloque de código dentro del 'if' se ejecutará si la temperatura es mayor que 20 grados y no está lloviendo.

```
temperatura = 25
lluvia = True

if temperatura > 20 and not lluvia:
    print("Es un día agradable para salir.")
else:
    print("Mejor quédate en casa.")
```





Recuerda que la indentación es crucial en Python. Los bloques de código dentro de **'if'**, **'elif'** y **'else'** deben estar indentados correctamente para indicar su pertenencia a la estructura condicional. La estructura de control **'if'** es fundamental para la toma de decisiones en Python y es una herramienta poderosa en la construcción de algoritmos y programas.



TIC



# Estructuras de Control en Python

## El bucle for

En Python se utiliza para iterar sobre una secuencia (que puede ser una lista, una tupla, un diccionario, un conjunto o cualquier objeto iterable). Permite ejecutar un bloque de código repetidamente para cada elemento en la secuencia.

## Estructura Básica del Ciclo For:

```
for variable in secuencia:  
    # Cuerpo del bucle  
    # Se ejecutará para cada elemento en la secuencia  
    # Puedes usar la variable para acceder al elemento  
    # ...
```

+ info



TIC



# Estructuras de Control en Python

**Variable:** Es la variable que toma el valor de cada elemento en la secuencia en cada iteración del bucle.

**Secuencia:** Es la colección de elementos sobre la cual se iterará.

Ejemplo Práctico:

```
frutas = ["manzana", "plátano", "uva"]

for fruta in frutas:
    print(fruta)
```

En este ejemplo, el bucle for itera sobre la lista frutas, y en cada iteración, la variable fruta toma el valor de uno de los elementos de la lista. El bloque de código dentro del bucle (print(fruta)) se ejecuta para cada elemento, imprimiendo cada fruta en una línea.

+ info



TIC



# Estructuras de Control en Python

## Funciones Relacionadas con el Ciclo For:

`range()`: La función `range()` se utiliza comúnmente con el bucle `for` para generar una secuencia de números. Puedes proporcionar uno, dos o tres argumentos a `range()`.

- **`range(5)`**: Genera una secuencia de 0 a 4.
- **`range(2, 8)`**: Genera una secuencia de 2 a 7.
- **`range(1, 10, 2)`**: Genera una secuencia de 1 a 9 con un paso de 2.

+ info



TIC



# Estructuras de Control en Python

**enumerate():** La función `enumerate()` se utiliza para obtener tanto el índice como el valor de los elementos en una secuencia.

```
frutas = ["manzana", "plátano", "uva"]  
  
for indice, fruta in enumerate(frutas):  
    print(f"Índice: {indice}, Fruta: {fruta}")
```

Notas adicionales

+ info



- Puedes usar cualquier nombre para la variable en el bucle (for elemento in secuencia).
- Los dos puntos (:) indican el inicio de un bloque de código que se ejecutará en cada iteración.
- La indentación es crucial en Python y define el alcance del bloque de código dentro del bucle.
- El bucle for es una herramienta poderosa para procesar elementos en una secuencia y es ampliamente utilizado en programación Python.

```
# Estructura de control 'for'  
for i in range(5):  
    print(i)
```



TIC



# Estructuras de Control en Python

## El bucle while

En Python se utiliza para repetir un bloque de código mientras una condición sea verdadera. A diferencia del bucle for, que itera sobre una secuencia específica, el bucle while se ejecutará mientras la expresión condicional sea evaluada como True.

### Estructura Básica del Ciclo While:

```
while condicion:  
    # Cuerpo del bucle  
    # Se ejecutará mientras la condición sea verdadera  
    # ...
```

+ info



TIC



# Estructuras de Control en Python

**condicion:** Es una expresión booleana que determina si el bucle debe continuar ejecutándose.

Ejemplo Práctico:

```
contador = 0

while contador < 5:
    print(f"Contador: {contador}")
    contador += 1
```

En este ejemplo, el bucle while se ejecutará mientras la variable contador sea menor que 5.

En cada iteración, se imprime el valor actual de contador y se incrementa en 1. El bucle se detendrá cuando la condición (contador < 5) sea falsa.

+ info



TIC



# Estructuras de Control en Python

## Funciones Relacionadas con el Ciclo While:

**Break:** La palabra clave break se utiliza para salir del bucle while antes de que la condición sea falsa. Puede usarse con una condición específica dentro del bucle.

```
contador = 0

while True:
    print(f"Contador: {contador}")
    contador += 1
    if contador >= 5:
        break
```

En este ejemplo, el bucle while se ejecuta indefinidamente hasta que la condición `contador >= 5` se cumple, momento en el que se utiliza `break` para salir del bucle.

+ info



TIC



# Estructuras de Control en Python

**Continue:** La palabra clave continue se utiliza para pasar a la siguiente iteración del bucle sin ejecutar el resto del código dentro del bucle en esa iteración.

```
contador = 0

while contador < 5:
    contador += 1
    if contador == 3:
        continue
    print(f"Contador: {contador}")
```

En este ejemplo, cuando contador es igual a 3, continue salta a la siguiente iteración sin imprimir el mensaje.

Notas adicionales

+ info



- La condición se evalúa antes de cada iteración, y el bucle se ejecutará mientras la condición sea verdadera.
- Asegúrate de que la condición eventualmente se vuelva falsa para evitar bucles infinitos.
- El bucle while es útil cuando no sabes cuántas veces se repetirá el bloque de código y dependes de una condición para decidir cuándo detener la ejecución.

```
# Estructura de control 'while'  
contador = 0  
while contador < 3:  
    print(contador)  
    contador += 1
```



TIC



# Estructuras de Control en Python

## Diccionario y Búsqueda de Clave:

```
persona = {"nombre": "Ana", "edad": 22, "ciudad": "Barcelona"}
if "edad" in persona:
    print("La edad de la persona es:", persona["edad"])
else:
    print("No se encontró la edad.")
```

+ info



TIC



▶ TALENTO  
TECH

