



Python: Estructuras de Control

Centro de Servicios y Gestión Empresarial
SENA Regional Antioquia



@SENAComunica

www.sena.edu.co

Conceptualización

Estructuras de Control

Las estructuras de control permiten modificar el flujo de ejecución de un programa.

En Python existen tres tipos principales:

- **Condicionales** (if, elif, else)
- **Bucles** (while, for)
- **Control de flujo** (break, continue, pass)
- **Sentencias de casos** (match-case).



Condicionales

Condicionales

Los **Condicionales** son (*if*, *elif*, *else*) y permiten ejecutar un bloque de código si se cumple una condición es True o False.

Sintaxis

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

Condicionales

Ejemplo:

Determinar si un usuario es mayor de edad o menor de edad.

Solución:

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

Condicionales

Si una condición **if** es **False**, se evalúan las condiciones **elif**.
Si ninguna es **True**, se ejecuta el **else**.

Sintaxis

```
if condicion1:  
    # Código si condicion1 es verdadera  
elif condicion2:  
    # Código si condicion2 es verdadera  
elif condicion3:  
    # Código si condicion3 es verdadera  
else:  
    # Código si ninguna de las condiciones anteriores es verdadera
```

Condicionales

Ejemplo:

Solicitar al usuario ingresar una nota numérica (entero) y, con base en el valor ingresado, muestre un mensaje indicando su desempeño según la siguiente clasificación:

- **Excelente:** si la nota es 90 o superior.
- **Muy bien:** si la nota está entre 80 y 89.
- **Bien:** si la nota está entre 70 y 79.
- **Necesitas mejorar:** si la nota es menor a 70.

Condicionales

Solución:

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

if nota >= 90:
    print("Excelente")
elif nota >= 80:
    print("Muy bien")
elif nota >= 70:
    print("Bien")
else:
    print("Necesitas mejorar")
```

Operadores Lógicos

Operadores Comparación



Se usan para comparar valores y devuelven True o False.

Operador	Descripción	Ejemplo
==	Igual a	5 == 5 → True
!=	Diferente de	5 != 3 → True
>	Mayor que	10 > 5 → True
<	Menor que	5 < 10 → True
>=	Mayor o igual que	5 >= 5 → True
<=	Menor o igual que	4 <= 3 → False

Operadores Lógicos



Se usan para combinar expresiones lógicas.

Operador	Descripción	Ejemplo
and	True si ambas condiciones son True	$(5 > 3) \text{ and } (10 > 5) \rightarrow \text{True}$
or	True si al menos una condición es True	$(5 > 3) \text{ or } (10 < 5) \rightarrow \text{True}$
not	Niega la condición	$\text{not}(5 > 3) \rightarrow \text{False}$

Ejemplo



Ejemplo – Comparación

```
a = 10
b = 5
print(a == b)  # False
print(a > b)   # True
```

Ejemplo – Lógicos

```
edad = 25
es_mayor = edad > 18 and edad < 30
print(es_mayor)  # True
```

Ejemplo



Ejemplo – Lógicos

```
llueve = False

if not llueve:
    print("Puedes salir sin paraguas.")
else:
    print("Lleva un paraguas.")
```

Ejemplo Integrador



Ejemplo:

Determinar si un usuario cumple con los requisitos para solicitar productos bancarios.

Solución:

```
edad = 25
ingresos = 3000
genero = "M"

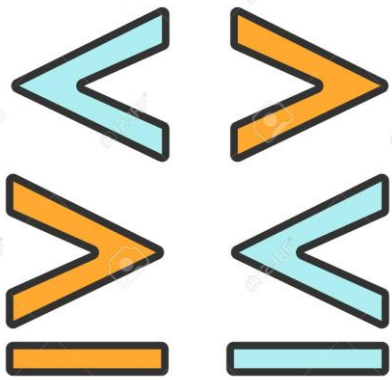
if edad >= 18 and ingresos >= 2500 and genero == "F" or genero == "M":
    print("Puedes solicitar una tarjeta de crédito.")
else:
    print("No cumples los requisitos.")
```

Ejercicios de Aplicación

Ejercicios de Aplicación



- Realizar una aplicación que permita determinar si un numero que digite el usuario es positivo o negativo.



- Realizar una aplicación que permita determinar el numero mayor a partir de tres números enteros que ingresa el usuario por teclado.

Ejercicios de Aplicación



- Realizar una aplicación que permita determinar el tipo de Chile utilizado de acuerdo con las unidades de Scoville que ingrese el usuario por teclado.



Unidades de Scoville

Habanero: 200 mil a 445 mil

Chiltepín: 100 mil a 200 mil

Piquín: 30 mil a 100 mil

Árbol: 23 mil a 30 mil

Serrano: 5 mil a 23 mil

Jalapeño: 1mil a 5 mil

Bell: 0 mil a 1 mil

Condicional en una sola línea



Si el bloque de código tiene solo una instrucción, se puede escribir en una línea.


Sintaxis:

```
valor_si_verdadero if condicion else valor_si_falso
```

Ejemplo:

```
edad = 25
if edad >= 18:
    print("Eres mayor de edad.")
else:
    print("Eres menor de edad.")

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

A large green curved arrow pointing from the multi-line if-else block to the single-line equivalent, indicating a transformation or simplification.

Bucles / Ciclos

Bucles / Ciclos

Se usan para repetir bloques de código varias veces.

for

Recorre una secuencia de elementos

while

Repite mientras una condición sea verdadera



Ciclo For

El bucle for se usa para recorrer elementos en una secuencia (listas, tuplas, cadenas, rangos, etc.).

Sintaxis

```
for variables in range(inicio, fin, incremento):  
    # Código que se ejecuta en cada iteración
```

Ciclo For



Ejemplos:

```
# Range con Valor Final  
for valor in range(10):  
    print(valor)
```

```
# Range con Valor Inicial y  
Final  
for valor in range(1, 10):  
    print(valor)
```

```
# Range con Valor Inicial,  
Final e Incremento  
for valor in range(1, 10, 3):  
    print(valor)
```

```
# Range con Valor Inicial,  
Final e Incremento Negativo /  
Decremento  
for valor in range(10, 1, -1):  
    print(valor)
```

Ciclo For

Ejercicio:

A partir de un numero que digite el usuario, mostrar la tabla de multiplicar del numero.



$1 \times 0 = 0$ $1 \times 1 = 1$ $1 \times 2 = 2$ $1 \times 3 = 3$ $1 \times 4 = 4$ $1 \times 5 = 5$ $1 \times 6 = 6$ $1 \times 7 = 7$ $1 \times 8 = 8$ $1 \times 9 = 9$ $1 \times 10 = 10$	$2 \times 0 = 0$ $2 \times 1 = 2$ $2 \times 2 = 4$ $2 \times 3 = 6$ $2 \times 4 = 8$ $2 \times 5 = 10$ $2 \times 6 = 12$ $2 \times 7 = 14$ $2 \times 8 = 16$ $2 \times 9 = 18$ $2 \times 10 = 20$	$3 \times 0 = 0$ $3 \times 1 = 3$ $3 \times 2 = 6$ $3 \times 3 = 9$ $3 \times 4 = 12$ $3 \times 5 = 15$ $3 \times 6 = 18$ $3 \times 7 = 21$ $3 \times 8 = 24$ $3 \times 9 = 27$ $3 \times 10 = 30$	$4 \times 0 = 0$ $4 \times 1 = 4$ $4 \times 2 = 8$ $4 \times 3 = 12$ $4 \times 4 = 16$ $4 \times 5 = 20$ $4 \times 6 = 24$ $4 \times 7 = 28$ $4 \times 8 = 32$ $4 \times 9 = 36$ $4 \times 10 = 40$	$5 \times 0 = 0$ $5 \times 1 = 5$ $5 \times 2 = 10$ $5 \times 3 = 15$ $5 \times 4 = 20$ $5 \times 5 = 25$ $5 \times 6 = 30$ $5 \times 7 = 35$ $5 \times 8 = 40$ $5 \times 9 = 45$ $5 \times 10 = 50$
$6 \times 0 = 0$ $6 \times 1 = 6$ $6 \times 2 = 12$ $6 \times 3 = 18$ $6 \times 4 = 24$ $6 \times 5 = 30$ $6 \times 6 = 36$ $6 \times 7 = 42$ $6 \times 8 = 48$ $6 \times 9 = 54$ $6 \times 10 = 60$	$7 \times 0 = 0$ $7 \times 1 = 7$ $7 \times 2 = 14$ $7 \times 3 = 21$ $7 \times 4 = 28$ $7 \times 5 = 35$ $7 \times 6 = 42$ $7 \times 7 = 49$ $7 \times 8 = 56$ $7 \times 9 = 63$ $7 \times 10 = 70$	$8 \times 0 = 0$ $8 \times 1 = 8$ $8 \times 2 = 16$ $8 \times 3 = 24$ $8 \times 4 = 32$ $8 \times 5 = 40$ $8 \times 6 = 48$ $8 \times 7 = 56$ $8 \times 8 = 64$ $8 \times 9 = 72$ $8 \times 10 = 80$	$9 \times 0 = 0$ $9 \times 1 = 9$ $9 \times 2 = 18$ $9 \times 3 = 27$ $9 \times 4 = 36$ $9 \times 5 = 45$ $9 \times 6 = 54$ $9 \times 7 = 63$ $9 \times 8 = 72$ $9 \times 9 = 81$ $9 \times 10 = 90$	$10 \times 0 = 0$ $10 \times 1 = 10$ $10 \times 2 = 20$ $10 \times 3 = 30$ $10 \times 4 = 40$ $10 \times 5 = 50$ $10 \times 6 = 60$ $10 \times 7 = 70$ $10 \times 8 = 80$ $10 \times 9 = 90$ $10 \times 10 = 100$



Ciclo For

Ejercicio:

A partir de un numero que digite el usuario, mostrar la tabla de multiplicar del numero.

Solución:

```
numero = int(input("Ingresa un número: "))  
  
for i in range(1, 11):  
    print(f"{numero} x {i} = {numero * i}")
```

Ciclo For (*Secuencia*)

Sintaxis:

```
for variable in secuencia:  
    # Código que se ejecuta en cada iteración
```

Ejemplo:

```
palabra = "Python"  
for letra in palabra:  
    print(letra)
```

```
numeros = 123456789  
for numero in str(numeros):  
    print(numero)
```

Ciclo While

El bucle while se ejecuta mientras una condición sea True.

Sintaxis:

```
condicion = valor_inicio  
  
while condicion:  
    # Código que se ejecuta mientras la condición sea verdadera  
    condicion = valor_actualizacion
```

Operadores Asignación



Se usan para asignar valores a variables.

Operador	Equivalente a	Ejemplo
=	$a = b$	$a = 5$
+=	$a = a + b$	$a += 3$
-=	$a = a - b$	$a -= 2$
*=	$a = a * b$	$a *= 4$
/=	$a = a / b$	$a /= 3$
//=	$a = a // b$	$a //= 2$
%=	$a = a \% b$	$a \% = 3$
**=	$a = a ** b$	$a ** = 2$

Ciclo While

El bucle while se ejecuta mientras una condición sea True.

Ejemplo:

```
condicion = 5

while condicion <= 20:
    print(condicion)
    condicion += 5
```

```
contador = 20

while contador >= 0:
    print(contador)
    contador -= 2
```

Contador / Acumulador

Los acumuladores y contadores son variables utilizadas en los bucles para almacenar valores o contar iteraciones.

- Un **contador** es una variable que se incrementa en una cantidad fija (generalmente +1) en cada iteración de un bucle.
- Un **acumulador** es una variable que se incrementa en valores variables (por ejemplo, sumando los números ingresados por el usuario).

Concepto	Función	Ejemplo
Contador	Cuenta la cantidad de iteraciones.	contador += 1
Acumulador	Suma valores durante el bucle.	acumulador += valor

Contador / Acumulador

Ejemplo:

```
contador = 1  # Para contar las iteraciones
acumulador = 0  # Para almacenar la suma

while contador <= 5:
    num = int(input(f"{contador}. Ingrese el número: "))
    acumulador += num  # Suma los valores ingresados
    contador += 1  # Incrementa el contador

print(f"La suma total de los {contador-1} números ingresados es:
{acumulador}")
```

Ejercicio While



Trabajas en una empresa y debes calcular el sueldo semanal de los empleados basado en las horas trabajadas. Un sistema solicita la cantidad de horas trabajadas por cada empleado hasta que se introduzca un número negativo para finalizar el proceso.



Contador / Acumulador

Ejercicio:

Desarrolla un programa en Python que simule el funcionamiento de un cajero automático.

El usuario tendrá un saldo inicial de \$1000 y podrá realizar retiros de dinero hasta que su saldo llegue a \$0 o decida salir ingresando 0.



Contador / Acumulador

Solución:

```
saldo = 10
cantidad = -1
while cantidad != 0 and saldo > 0:
    cantidad = float(input("Ingrese la cantidad a retirar (o 0 para salir): "))
    if cantidad > saldo:
        print("No tienes suficiente saldo.")
    elif cantidad < 0:
        print("No puedes retirar cantidades negativas.")
    else:
        saldo -= cantidad
        print(f"Tu saldo actual es de ${saldo}")
```

Control de Flujo

Declaraciones



Python proporciona tres instrucciones para modificar el flujo de ejecución dentro de los bucles (for y while):

Break

Permite detener el ciclo incluso si la condición es verdadera

Continue

Permite detener la iteración actual y continuar con la siguiente

Pass

No hace nada; se usa como marcador de posición.

Declaración Break



WHILE ←

```
# Break en una estructura While
i = 1
while i < 6:
    #print (i)
    if (i == 4):
        break
    i += 1
print ("Finalizo en la Iteracion: ",i)
```

```
# Break en una estructura For
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    if x == "banana":
        break
    print(x)
```

→ FOR

Declaración Continue



WHILE ←

```
# Break en una estructura While
i = 0
while i < 6:
    i += 1
    if i == 3:
        continue
    print(i)
```

```
# Break en una estructura While
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    if x == "banana":
        continue
    print(x)
```

→ FOR

Ejercicio Break, continue, pass

Se requiere una aplicación para validar la contraseña y permitir el acceso al sistema:

- El usuario tiene 3 intentos para ingresarla correctamente. Si se agotan los intentos, se bloquea el acceso.
- Si la contraseña es incorrecta, se le pide que la ingrese nuevamente.
- Si la contraseña es correcta, se otorga acceso y se finaliza el programa.



Estructura de Casos Switch



GRACIAS

Presentó: Alvaro Pérez Niño
Instructor Técnico

Correo: aperezn@sena.edu.co

<http://centrodeserviciosygestionempresarial.blogspot.com/>

Línea de atención al ciudadano: 01 8000 910270

Línea de atención al empresario: 01 8000 910682



@SENAComunica

www.sena.edu.co