

03_estructuras_de_control

October 18, 2024

1 Lenguaje de Programación

1.0.1 Universidad Politecnica Salesiana

1.0.2 Ingeniería en Ciencias de la Computación

1.0.3 Programación

1.1 Autores del Material

- **Instructor:** Ing. Pablo Torres
 - **Contacto:** ptorresp@ups.edu.ec
-

1.2 Temas

Estructuras de Decisión en Python Estructuras de Decisión en Python

- If
- If ... else
- If ... elif ... else

1.2.1 Descripción General

En este tema, aprenderemos sobre las estructuras de decisión en Python, las cuales nos permiten controlar el flujo de ejecución de un programa en función de condiciones específicas. Exploraremos las estructuras `if`, `if...else` y `if...elif...else`, comprendiendo su sintaxis y aplicabilidad mediante ejemplos prácticos.

1.2.2 Objetivos de Aprendizaje

- Comprender el uso de la estructura `if` en Python.
 - Implementar estructuras `if...else` para decisiones binarias.
 - Utilizar estructuras `if...elif...else` para manejar múltiples condiciones.
 - Aplicar las estructuras de decisión en la resolución de problemas prácticos.
-

1.3 A. Estructuras de Decisión en Python

Las estructuras de decisión permiten que el programa tome diferentes caminos de ejecución basados en condiciones evaluadas como verdaderas o falsas.

1.3.1 1.1. La Estructura if

Definición y Propósito

- **Definición:** La estructura `if` se utiliza para ejecutar un bloque de código solo si una condición específica se evalúa como verdadera.
- **Propósito:** Permite que el programa tome decisiones basadas en condiciones dinámicas durante la ejecución.

Ejemplo de Uso

```
# Ejemplo de estructura if
numero = 10

if numero > 5:
    print("El número es mayor que 5.")
```

En este ejemplo, el mensaje se imprimirá solo si el valor de `numero` es mayor que 5.

1.3.2 1.2. La Estructura if..else

Definición y Propósito

- **Definición:** La estructura `if..else` permite ejecutar un bloque de código si una condición es verdadera y otro bloque diferente si la condición es falsa.
- **Propósito:** Facilita la toma de decisiones binarias, donde existen dos posibles caminos de ejecución.

Sintaxis

```
if condición:
    # Bloque de código si la condición es verdadera
else:
    # Bloque de código si la condición es falsa
```

Ejemplo de Uso

```
# Ejemplo de estructura if..else
edad = 18

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

En este ejemplo, se imprime un mensaje diferente dependiendo de si la edad es mayor o igual a 18.

1.3.3 1.3. La Estructura if...elif...else

Definición y Propósito

- **Definición:** La estructura if...elif...else se utiliza para manejar múltiples condiciones. Permite evaluar una serie de condiciones de manera secuencial hasta que una de ellas sea verdadera.
- **Propósito:** Facilita la gestión de múltiples escenarios posibles dentro del flujo de ejecución del programa.

Sintaxis

```
if condición1:  
    # Bloque de código si condición1 es verdadera  
elif condición2:  
    # Bloque de código si condición2 es verdadera  
else:  
    # Bloque de código si ninguna condición anterior es verdadera
```

Ejemplo de Uso

```
# Ejemplo de estructura if...elif...else  
nota = 85
```

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

En este ejemplo, el mensaje que se imprime depende del rango en el que se encuentra la nota.

```
[1]: # Ejemplo de estructura if...elif...else  
nota = 85  
  
if nota >= 90:  
    print("Excelente")  
elif nota >= 80:  
    print("Muy bien")  
elif nota >= 70:  
    print("Bien")  
else:  
    print("Necesitas mejorar")
```

Muy bien

1.4 B. Ejercicios Prácticos

1.4.1 Ejercicio 1: Determinar si un Número es Positivo, Negativo o Cero

Descripción: Crea un programa que solicite al usuario que ingrese un número y determine si es positivo, negativo o cero.

Instrucciones: 1. Utiliza la función `input()` para solicitar al usuario que ingrese un número. 2. Convierte la entrada a tipo `float` usando la función `float()`. 3. Utiliza estructuras `if..elif..else` para determinar si el número es positivo, negativo o cero. 4. Imprime el resultado utilizando `print()` con una f-string.

Código Base:

```
# Solicitar al usuario que ingrese un número
numero_str = input("Ingresa un número: ")
numero = float(numero_str)

# Determinar si el número es positivo, negativo o cero
if numero > 0:
    print(f"{numero} es un número positivo.")
elif numero < 0:
    print(f"{numero} es un número negativo.")
else:
    print(f"{numero} es cero.")
```

```
[2]: # Solicitar al usuario que ingrese un número
numero_str = input("Ingresa un número: ")
numero = float(numero_str)

# Determinar si el número es positivo, negativo o cero
if numero > 0:
    print(f"{numero} es un número positivo.")
elif numero < 0:
    print(f"{numero} es un número negativo.")
else:
    print(f"{numero} es cero.")
```

10.0 es un número positivo.

1.4.2 Ejercicio 2: Clasificar una Edad en Categorías

Descripción: Crea un programa que solicite al usuario que ingrese su edad y la clasifique en una de las siguientes categorías: - Niño (0-12 años) - Adolescente (13-17 años) - Adulto (18-64 años) - Senior (65 años o más)

Instrucciones: 1. Utiliza `input()` para solicitar la edad del usuario. 2. Convierte la entrada a tipo `int` usando la función `int()`. 3. Utiliza estructuras `if..elif..else` para clasificar la edad. 4. Imprime la categoría correspondiente utilizando `print()` con una f-string.

Código Base:

```

# Solicitar al usuario que ingrese su edad
edad_str = input("Ingresa tu edad: ")
edad = int(edad_str)

# Clasificar la edad en categorías
if edad >= 0 and edad <= 12:
    categoria = "Niño"
elif edad >= 13 and edad <= 17:
    categoria = "Adolescente"
elif edad >= 18 and edad <= 64:
    categoria = "Adulto"
elif edad >= 65:
    categoria = "Senior"
else:
    categoria = "Edad inválida"

# Mostrar la categoría
print(f"Tienes {edad} años, eres un(a) {categoria}.")

```

```

[3]: # Solicitar al usuario que ingrese su edad
     edad_str = input("Ingresa tu edad: ")
     edad = int(edad_str)

     # Clasificar la edad en categorías
     if edad >= 0 and edad <= 12:
         categoria = "Niño"
     elif edad >= 13 and edad <= 17:
         categoria = "Adolescente"
     elif edad >= 18 and edad <= 64:
         categoria = "Adulto"
     elif edad >= 65:
         categoria = "Senior"
     else:
         categoria = "Edad inválida"

     # Mostrar la categoría
     print(f"Tienes {edad} años, eres un(a) {categoria}.")

```

Tienes 28 años, eres un(a) Adulto.

1.4.3 Ejercicio 3: Calculadora de Notas

Descripción: Crea un programa que solicite al usuario que ingrese una nota (0-100) y determine la calificación correspondiente: - A: 90-100 - B: 80-89 - C: 70-79 - D: 60-69 - F: 0-59

Instrucciones: 1. Utiliza `input()` para solicitar la nota del usuario. 2. Convierte la entrada a tipo `int` usando la función `int()`. 3. Utiliza estructuras `if..elif..else` para asignar la calificación. 4. Imprime la calificación utilizando `print()` con una f-string.

Código Base:

```
# Solicitar al usuario que ingrese una nota
nota_str = input("Ingresa tu nota (0-100): ")
nota = int(nota_str)

# Determinar la calificación
if nota >= 90 and nota <= 100:
    calificacion = "A"
elif nota >= 80 and nota < 90:
    calificacion = "B"
elif nota >= 70 and nota < 80:
    calificacion = "C"
elif nota >= 60 and nota < 70:
    calificacion = "D"
elif nota >= 0 and nota < 60:
    calificacion = "F"
else:
    calificacion = "Nota inválida"

# Mostrar la calificación
print(f"Tu calificación es: {calificacion}")
```

```
[5]: # Solicitar al usuario que ingrese una nota
nota_str = input("Ingresa tu nota (0-100): ")
nota = int(nota_str)
print(nota)
# Determinar la calificación
if nota >= 90 and nota <= 100:
    calificacion = "A"
elif nota >= 80 and nota < 90:
    calificacion = "B"
elif nota >= 70 and nota < 80:
    calificacion = "C"
elif nota >= 60 and nota < 70:
    calificacion = "D"
elif nota >= 0 and nota < 60:
    calificacion = "F"
else:
    calificacion = "Nota inválida"

# Mostrar la calificación
print(f"Tu calificación es: {calificacion}")
```

91

Tu calificación es: A

1.5 C. Resumen de Conceptos Clave

1. **Estructuras de Decisión:** Permiten controlar el flujo de ejecución de un programa basándose en condiciones evaluadas como verdaderas o falsas.
2. **Estructura `if`:** Ejecuta un bloque de código solo si una condición específica es verdadera.
3. **Estructura `if...else`:** Permite ejecutar un bloque de código si una condición es verdadera y otro bloque si es falsa.
4. **Estructura `if...elif...else`:** Maneja múltiples condiciones de manera secuencial, ejecutando el bloque correspondiente a la primera condición verdadera.
5. **f-strings:** Facilitan el formateo de cadenas de texto, permitiendo incorporar variables y expresiones dentro de las cadenas de manera legible y eficiente.
6. **Conversión de Tipos:**
 - `int()`: Convierte una cadena de texto a un entero.
 - `float()`: Convierte una cadena de texto a un número de punto flotante.
 - `str()`: Convierte un valor a una cadena de texto.

1.6 D. Recursos Adicionales

- [Documentación Oficial de Python - Estructuras de Decisión](#)
 - [Tutorial de Estructuras de Decisión en Real Python](#)
 - [Ejemplos de Estructuras de Decisión en W3Schools](#)
 - [Curso de Python en Coursera](#)
-

1.7 E. Referencias

1. Van Rossum, G. (1991). *Python Tutorial*. Python Software Foundation.
 2. Lutz, M. (2013). *Learning Python*. O'Reilly Media.
 3. Sweigart, A. (2015). *Automate the Boring Stuff with Python*. No Starch Press.
-