

Programación para Física y Astronomía

Departamento de Física.

Coordinadora: C Loyola

Profesores C Femenías / F Bugini / D Basantes

Primer Semestre 2025

Universidad Andrés Bello

Departamento de Física y Astronomía



Introducción a la Sesión 3

Estructuras de Control: Condicionales

Estructuras de Control: Bucles

Ejercicios Guiados

Actividad Práctica

Consolidación y Retroalimentación

Conclusiones

Introducción a la Sesión 3

- **Introducir** las estructuras de control fundamentales en Python.
- **Dominar** el uso de condicionales (**if**, **elif**, **else**).
- **Comprender** bucles básicos (**for**, **while**) y su aplicación.
- **Aplicar** estas estructuras en problemas físicos y matemáticos.
- **Desarrollar** lógica de programación a través de ejercicios prácticos.

- En la sesión anterior, exploramos la sintaxis básica de Python.
- Aprendimos sobre variables, tipos de datos y operaciones básicas.
- Ahora, construiremos sobre estos conceptos para manejar flujos de control.

Estructuras de Control: Condicionales

Estructuras de Control: Condicionales \ni ¿Qué son las Estructuras Condicionales?

- Hasta ahora, nuestros programas ejecutan todas las líneas en **secuencia**.
- Las **estructuras condicionales** permiten que el programa tome **decisiones**.
- Ejecutan diferentes bloques de código según se cumplan ciertas **condiciones**.

Analogía Física

Como un fotón que toma diferentes caminos según su energía: si $E > E_{umbral} \rightarrow$ efecto fotoeléctrico, sino \rightarrow no hay emisión.

Sintaxis fundamental:

```
1 if condición:  
2     # Código que se ejecuta SI la condición es True  
3     instrucción1  
4     instrucción2
```

Puntos clave:

- La **condición** debe evaluarse como **True** o **False**.
- Los **dos puntos (:)** son obligatorios.
- La **indentación** (4 espacios) define qué código está dentro del **if**.

Estructuras de Control: Condicionales \ni Ejemplo Práctico: Clasificación de Temperatura

```
1 temperatura = float(input("Temperatura del agua (°C): "))
2
3 if temperatura > 100:
4     print("El agua está en estado gaseoso (vapor)")
5
6 if temperatura >= 0 and temperatura <= 100:
7     print("El agua está en estado líquido")
8
9 if temperatura < 0:
10    print("El agua está en estado sólido (hielo)")
```

Operadores de comparación:

- >, <, >=, <=, ==, !=
- and, or, not para combinar condiciones

Estructuras de Control: Condicionales \ni La Estructura `if-elif-else` Completa

Sintaxis mejorada para múltiples condiciones:

```
1  if condición1:  
2      # Se ejecuta si condición1 es True  
3      código1  
4  elif condición2:  
5      # Se ejecuta si condición1 es False y condición2 es True  
6      código2  
7  elif condición3:  
8      # Se ejecuta si las anteriores son False y condición3 es  
9      ↪ True  
9      código3  
10 else:  
11     # Se ejecuta si TODAS las condiciones anteriores son False  
12     código_por_defecto
```

Importante: Solo se ejecuta **UNO** de los bloques (el primero que sea `True`).

Estructuras de Control: Condicionales \ni Ejemplo Mejorado: Clasificación de Temperatura

```
1  temperatura = float(input("Temperatura del agua (°C): "))
2
3  if temperatura > 100:
4      estado = "gaseoso (vapor)"
5      energia = "alta"
6  elif temperatura >= 0:
7      estado = "líquido"
8      energia = "media"
9  else: # temperatura < 0
10     estado = "sólido (hielo)"
11     energia = "baja"
12
13  print(f"A {temperatura}°C, el agua está en estado {estado}")
14  print(f"Nivel de energía cinética molecular: {energia}")
```

Ventaja: Más eficiente y lógico que múltiples `if` independientes.

Estructuras de Control: Bucles

Estructuras de Control: Bucles \ni ¿Qué son los Bucles?

- Los **bucles** permiten repetir un bloque de código múltiples veces.
- Evitan la necesidad de escribir el mismo código una y otra vez.
- Python tiene dos tipos principales: **for** y **while**.

Analogía Física

Como las órbitas planetarias: el planeta repite su trayectoria alrededor del sol hasta que alguna fuerza externa (condición) cambie el sistema.

Sintaxis:

```
1 while condición:  
2     # Código que se repite MIENTRAS la condición sea True  
3     instrucción1  
4     instrucción2  
5     # IMPORTANTE: modificar algo para que eventualmente sea  
        ↪ False
```

Características:

- Se repite **mientras** la condición sea **True**.
- Si la condición nunca se vuelve **False** \rightarrow **bucle infinito**.
- Útil cuando **no sabemos** cuántas repeticiones necesitamos.

Estructuras de Control: Bucles \ni Ejemplo

Aproximación a la Raíz Cuadrada

Enunciado

- Calcular la raíz cuadrada de un número usando el método de Newton-Raphson.
- Usar la fórmula iterativa: $x_{n+1} = \frac{1}{2}(x_n + \frac{a}{x_n})$
- Usar un bucle **while** que continúe hasta que la precisión sea menor a 0.0001.
- Mostrar cada iteración y el resultado final.
- Verificar comparando con el valor real.

Conceptos: Bucles **while**, precisión numérica, métodos iterativos.

Física relevante: Métodos numéricos en simulaciones físicas.

Estructuras de Control: Bucles \ni Ejemplo: Aproximación a la Raíz Cuadrada (Método de Newton)

```
1  numero = float(input("Número para calcular raíz cuadrada: "))
2  aproximacion = numero / 2  # Estimación inicial
3  tolerancia = 0.0001
4
5  print(f"Calculando raíz cuadrada de {numero}...")
6
7  while abs(aproximacion**2 - numero) > tolerancia:
8      aproximacion = (aproximacion + numero/aproximacion) / 2
9      print(f"Aproximación actual: {aproximacion}")
10
11 print(f"Raíz cuadrada  $\approx$  {aproximacion}")
12 print(f"Verificación: {aproximacion}2 = {aproximacion**2}")
```

Física relevante: Métodos iterativos son fundamentales en simulaciones físicas.

Sintaxis:

```
1 for variable in range(inicio, fin, paso):  
2     # Código que se repite para cada valor de 'variable'  
3     instrucción1  
4     instrucción2
```

Ejemplos de range():

- range(5) \rightarrow 0, 1, 2, 3, 4
- range(1, 6) \rightarrow 1, 2, 3, 4, 5
- range(0, 10, 2) \rightarrow 0, 2, 4, 6, 8
- range(10, 0, -1) \rightarrow 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

Útil cuando: Sabemos exactamente cuántas repeticiones necesitamos.

Estructuras de Control: Bucles \ni Ejemplo: Tabla de Conversión Celsius-Fahrenheit

```
1 print("Tabla de Conversión Celsius  $\rightarrow$  Fahrenheit")
2 print("="*40)
3
4 for celsius in range(0, 101, 10): # De 0°C a 100°C, cada 10°
5     fahrenheit = (9/5) * celsius + 32
6     print(f"{celsius:3d}°C = {fahrenheit:5.1f}°F")
7
8 print("="*40)
```

Salida esperada:

0°C	=	32.0°F
10°C	=	50.0°F
20°C	=	68.0°F
...		
100°C	=	212.0°F

Ejercicios Guiados

Ejercicios Guiados \ni Ejercicio 1:

Clasificador de Velocidades



Enunciado

Clasifique la velocidad de un objeto en función de su magnitud, considerando las siguientes categorías:

- **Movimiento lento:** Velocidad menor a 1 m/s.
- **Movimiento moderado:** Velocidad entre 1 y 10 m/s.
- **Movimiento rápido:** Velocidad entre 10 y 100 m/s.
- **Movimiento muy rápido:** Velocidad mayor a 100 m/s.

Física relevante: Escalas de velocidad en diferentes contextos físicos.

Ejercicios Guiados \ni Ejercicio 2:

Suma de Números Pares



Enunciado

- Calcular la suma de todos los números pares desde 2 hasta un número n dado por el usuario.
- Usar un bucle **for** con **range()**.
- Mostrar cada número par que se suma y el total final.
- Verificar el resultado usando la fórmula: $\sum_{k=1}^{n/2} 2k = n(n/2 + 1)$ para n par.

Conceptos: Bucles **for**, acumuladores, validación matemática.

Física relevante: Sumas de series son comunes en física estadística.

Ejercicios Guiados \ni Ejercicio 3: Juego de Adivinanza de Números



Enunciado

- El programa "piensa" en un número secreto entre 1 y 50.
- El usuario debe adivinar este número.
- Usar un bucle **while** que continúe hasta que el usuario acierte.
- Dar pistas: "muy cerca", "cerca", "alto", "bajo" según la proximidad.
- Contar el número de intentos y mostrar estadísticas al final.

Conceptos: Bucles **while**, condicionales, contadores.

Física relevante: Lógica de búsqueda y aproximación numérica.

Ejercicios Guiados \ni Solución 1 de Referencia:

Clasificador de Velocidades



```
1  # Entrada de datos
2  velocidad = float(input("Velocidad del objeto (m/s): "))
3  masa = float(input("Masa del objeto (kg, opcional, 0 si no aplica): "))
4
5  # Clasificación de velocidad
6  if velocidad < 1:
7      categoria = "Movimiento lento"
8  elif velocidad < 10:
9      categoria = "Movimiento moderado"
10 elif velocidad < 100:
11     categoria = "Movimiento rápido"
12 else:
13     categoria = "Movimiento muy rápido"
14
15 print(f"Velocidad: {velocidad} m/s  $\rightarrow$  {categoria}")
16
17 # Cálculo de energía cinética si se proporciona masa
18 if masa > 0:
19     energia_cinetica = 0.5 * masa * velocidad**2
20     print(f"Energía cinética: {energia_cinetica} J")
```

Discusión: Uso de condicionales anidadas y validación de entrada opcional.

Ejercicios Guiados \ni Solución 2 de Referencia:

Suma de Números Pares



```
1  # Entrada de datos
2  n = int(input("Ingrese el número límite: "))
3
4  # Inicializar acumulador
5  suma_pares = 0
6  print(f"Números pares desde 2 hasta {n}:")
7
8  # Bucle para sumar números pares
9  for numero in range(2, n + 1, 2): # Desde 2, hasta n+1, de 2 en 2
10     suma_pares += numero
11     print(f"  + {numero}")
12
13  print(f"Suma total de números pares: {suma_pares}")
14
15  # Verificación con fórmula (solo si n es par)
16  if n % 2 == 0:
17     formula_resultado = n * (n // 2 + 1)
18     print(f"Verificación con fórmula: {formula_resultado}")
19     print(f"¿Coinciden? {suma_pares == formula_resultado}")
```

Discusión: Uso eficiente de range() con paso, acumuladores, y verificación matemática.

Ejercicios Guiados \ni Solución 3 de Referencia:

Juego de Adivinanza de Números



```
1  # Número secreto: número entre 1 y 50
2  numero_secreto = 37
3  intentos = 0
4
5  print(";Juego de adivinanza!")
6  print("He pensado en un número entre 1 y 50")
7  print("¿Puedes adivinarlo?")
8
9  while True:
10     intento = int(input("Tu estimación: "))
11     intentos += 1
12
13     diferencia = abs(intento - numero_secreto)
14
15     if intento == numero_secreto:
16         print(f";Correcto! El número era {numero_secreto}")
17         print(f"Lo lograste en {intentos} intentos")
18         break
19     elif diferencia <= 2: # Muy cerca (diferencia de 1 o 2)
20         print(";Muy cerca!")
21     elif diferencia <= 5: # Cerca (diferencia de 3 a 5)
22         print("Cerca...")
23     elif intento < numero_secreto:
24         print("Muy bajo")
25     else:
26         print("Muy alto")
```

Actividad Práctica

Formación de equipos

- Grupos de 2-3 personas
- Cada persona crea su propio notebook
- Comparten pantalla y discuten soluciones

Metodología de trabajo

- **10 min:** Lean y discutan los problemas
- **20 min:** Implementen las soluciones
- **5 min:** Prueben con diferentes valores
- **10 min:** Comparen resultados entre compañeros

Actividad Práctica \ni Actividad Extra 1:

Calculadora de Período Orbital



Enunciado

- Calcular el período orbital de planetas usando la 3ª Ley de Kepler: $T^2 = \frac{4\pi^2}{GM} a^3$
- Pedir el semieje mayor a en metros.
- Usar $G = 6.674 \times 10^{-11} \text{ m}^3/(\text{kg}\cdot\text{s}^2)$ y $M_{\text{sol}} = 1.989 \times 10^{30} \text{ kg}$.
- Mostrar el período en segundos, días y años.
- Usar condicionales para clasificar el tipo de órbita.

Objetivo: Integrar estructuras de control con cálculos astrofísicos.



Enunciado

- Simular el decaimiento radioactivo usando: $N(t) = N_0 e^{-\lambda t}$
- Pedir: número inicial de átomos N_0 y vida media $t_{1/2}$.
- Calcular $\lambda = \frac{\ln(2)}{t_{1/2}}$.
- Usar un bucle **for** para mostrar $N(t)$ cada 100 años hasta 1000 años.
- Agregar condicionales para alertar cuando quede menos del 10% y 1%.

Objetivo: Bucles con aplicaciones en física nuclear.

Consolidación y Retroalimentación

Consolidación y Retroalimentación \ni Puesta en Común - ¿Cómo les fue?

Cada equipo comparta brevemente:

- ¿Qué estructura de control les resultó más difícil de entender?
- ¿Encontraron errores comunes? ¿Cómo los solucionaron?
- ¿Algún truco o descubrimiento interesante con bucles o condicionales?
- ¿Cómo validaron que sus cálculos físicos fueran correctos?

Objetivo

Aprender de las experiencias de otros equipos y normalizar que los errores son parte del aprendizaje.

Consolidación y Retroalimentación \ni Errores Comunes en Estructuras de Control

Errores frecuentes con IF

- Olvidar los dos puntos (:) después de la condición
- Problemas de indentación (usar tabs y espacios mezclados)
- Usar = (asignación) en lugar de == (comparación)

Errores frecuentes con bucles

- **Bucle infinito:** No modificar la variable de control en **while**
- **Off-by-one:** Confusión con los límites de **range()**
- **Indentación:** No alinear correctamente el código dentro del bucle

Consejo

Usar **print()** para debuggear: imprimir variables dentro de bucles para entender qué está pasando.

Conclusiones

- Dominamos las **estructuras condicionales** (`if`, `elif`, `else`).
- Aprendimos sobre **bucles** (`while`, `for`) y sus aplicaciones.
- Aplicamos estas estructuras a **problemas físicos reales**.
- Desarrollamos **lógica de programación** a través de ejercicios progresivos.
- Practicamos **debugging** y resolución colaborativa de problemas.

Habilidad adquirida

Sus programas ahora pueden tomar decisiones inteligentes y automatizar tareas repetitivas.

- **Sesión 4 (Semana 2):** Introducción a **funciones** en Python.
- **Temas futuros:**
 - Definición de funciones y parámetros
 - Reutilización de código y modularización
 - Funciones en bibliotecas científicas
- **Práctica recomendada:** Crear pequeños programas que combinen condicionales y bucles.

Próxima sesión

Aprenderán a organizar su código en **funciones reutilizables** para resolver problemas más complejos.

- **Python Docs - Control Flow** (documentación oficial).
- **Real Python - Conditionals** (tutoriales avanzados).
- **Real Python - For Loops** (ejemplos prácticos).
- **LearnPython.org - Loops** (ejercicios interactivos).

Práctica adicional

Prueben modificar los ejercicios de hoy agregando más condicionales o cambiando los rangos de los bucles.

Entrega (Canvas)

- Sube tu notebook con los 3 ejercicios principales resueltos
- Incluye comentarios explicando tu razonamiento
- Al menos una de las actividades extra completada
- Fecha límite: antes de la próxima clase

Autoevaluación

- ¿Entiendo cuándo usar **if-elif-else** vs múltiples **if**?
- ¿Sé cuándo elegir **for** vs **while**?
- ¿Puedo debuggear bucles infinitos?

¡Excelente progreso!

- Ahora dominan las **estructuras de control fundamentales**
- Sus programas pueden **tomar decisiones y automatizar tareas**
- Están listos para **funciones y modularización**
- Recuerden: **la práctica constante** es clave

¡Nos vemos en la Sesión 4!