Ciclos Anidados - Datos bidimensionales

Materia: Algoritmos y Programación

Prof. D.Sc. BARSEKH-ONJI Aboud

Facultad de Ingeniería
Universidad Anáhuac México

20 de octubre de 2025

Agenda

1. Ejemplo 4: Dibujando un Triángulo
2. Estructuras de Datos 2D: Listas de Listas
3. Ejemplo 5: Procesando Datos de Estudiantes
4. Ejemplo 6: Encontrando el Valor Máximo

Ejemplo 4: Dibujando un Triángulo

Objetivo

Escribir un programa que le pida al usuario un número entero (que representará la altura) y que, usando ciclos anidados, dibuje un triángulo rectángulo de asteriscos de esa misma altura.

Salida Esperada

Si el usuario introduce el número **5**, la salida en consola debe ser:

```
*
**
***
****
*****
```

Ejemplo 4: Lógica y Pistas

Nuestra Lógica

Este problema nos obliga a pensar en cómo el número de iteraciones del ciclo interno cambia en cada paso del ciclo externo.

- **Ciclo Externo (Filas):** Controlará el número de filas del triángulo. Si la altura es 5, este ciclo debe ejecutarse 5 veces (una por cada línea).

Ejemplo 4: Lógica y Pistas

Nuestra Lógica

Este problema nos obliga a pensar en cómo el número de iteraciones del ciclo interno cambia en cada paso del ciclo externo.

- **Ciclo Externo (Filas):** Controlará el número de filas del triángulo. Si la altura es 5, este ciclo debe ejecutarse 5 veces (una por cada línea).
- **Ciclo Interno (Columnas):** Controlará cuántos asteriscos imprimir en la fila **actual**.

Ejemplo 4: Lógica y Pistas

Nuestra Lógica

Este problema nos obliga a pensar en cómo el número de iteraciones del ciclo interno cambia en cada paso del ciclo externo.

- **Ciclo Externo (Filas):** Controlará el número de filas del triángulo. Si la altura es 5, este ciclo debe ejecutarse 5 veces (una por cada línea).
- **Ciclo Interno (Columnas):** Controlará cuántos asteriscos imprimir en la fila **actual**.
- **La Clave:** Observa el patrón. En la fila 1, hay 1 asterisco. En la fila 2, hay 2 asteriscos, y así sucesivamente. Por lo tanto, el número de veces que el ciclo interno debe ejecutarse es igual al número de la fila actual del ciclo externo.

Ejemplo 4: Lógica y Pistas

Nuestra Lógica

Este problema nos obliga a pensar en cómo el número de iteraciones del ciclo interno cambia en cada paso del ciclo externo.

- **Ciclo Externo (Filas):** Controlará el número de filas del triángulo. Si la altura es 5, este ciclo debe ejecutarse 5 veces (una por cada línea).
- **Ciclo Interno (Columnas):** Controlará cuántos asteriscos imprimir en la fila **actual**.
- **La Clave:** Observa el patrón. En la fila 1, hay 1 asterisco. En la fila 2, hay 2 asteriscos, y así sucesivamente. Por lo tanto, el número de veces que el ciclo interno debe ejecutarse es igual al número de la fila actual del ciclo externo.
- **Pista de Impresión:** Para imprimir los asteriscos en la misma línea, puedes usar `print(''*'', end='''')`. Después de que el ciclo interno termine, necesitarás un `print()` vacío para pasar a la siguiente línea.

Ejemplo 4: Código y Salida

Código en Python

```
altura = int(input(''Introduce la
    altura del triangulo: ''))
# Ciclo externo para cada fila
for fila in range(1, altura + 1):
    # Se ejecuta 'fila' veces
    for columna in range(fila):
        print(''*'', end='''')
    print() # Salto de línea después
    de cada fila

```

Salida en Consola (para altura = 7)

```
Introduce la altura del triangulo: 7
*
**
***
****
*****
******
*******
```

Agenda

1. Ejemplo 4: Dibujando un Triángulo
2. Estructuras de Datos 2D: Listas de Listas
3. Ejemplo 5: Procesando Datos de Estudiantes
4. Ejemplo 6: Encontrando el Valor Máximo

De una Dimensión a Dos Dimensiones

Recordatorio: Listas Unidimensionales

Hasta ahora, hemos trabajado con listas simples, que son como una fila de casillas. Cada elemento tiene una única posición o índice.

```
1 miLista = [''A'', ''B'', ''C'', ''D''] # Accedemos con miLista[i]
2
```

El Siguiente Nivel: Listas de Listas

Una lista de listas es exactamente lo que su nombre indica: una lista que en lugar de contener números o strings, contiene **otras listas**.

- Nos permite organizar la información en una estructura de **filas y columnas**, como una tabla de Excel, un tablero de ajedrez o una matriz matemática.
- Se conocen como estructuras de datos bidimensionales (2D).

Anatomía de una Lista de Listas

Sintaxis y Estructura Visual

Imagina una matriz simple de 2 filas y 3 columnas.

```
matriz = [
    [1,    2,    3],  # Fila 0
    [4,    5,    6]   # Fila 1
]
```

- La **lista exterior** contiene todas las filas.
- Cada **lista interior** representa una fila completa.

Anatomía de una Lista de Listas

Accediendo a los Elementos: Doble Índice

Para llegar a un elemento específico, necesitamos dos coordenadas: primero la fila, y luego la columna. Usamos la notación: `nombre_lista[indice_fila][indice_columna]`

- `matriz[0]` nos daría la primera fila completa: `[1, 2, 3]`
- `matriz[1][2]` nos daría el elemento en la fila 1, columna 2: **6**

Ejemplo Práctico: Un Tablero de Gato (Tic-Tac-Toe)

Objetivo

Vamos a crear un tablero de Gato de 3x3 usando una lista de listas. Luego, colocaremos una "X" en el centro del tablero y lo mostraremos en consola.

Salida en Consola

```
Tablero de Gato:
['-', '-', '-']
['-', 'X', '-']
['-', '-', '-']
```

Punto Clave

La línea `tablero[1][1] = ''X''` es la más importante. Demuestra cómo podemos **leer y escribir** en una posición específica de nuestra estructura 2D sin necesidad de ciclos anidados.

Ejemplo Práctico: Un Tablero de Gato (Tic-Tac-Toe)

Código en Python

```
tablero = [
    ['-', '-', '-'],
    ['-', '-', '-'],
    ['-', '-', '-']
]

tablero[1][1] = 'X'

print('Tablero de Gato:')
for fila in tablero:
    print(fila)
```

Conectando con el Siguiente Tema

¿Y ahora qué?

Ya sabemos cómo acceder a **UN** elemento específico. Pero, ¿qué pasaría si quisiéramos hacer algo con **TODOS** los elementos?

- ¿Cómo sumaríamos todas las calificaciones de todos los estudiantes?
- ¿Cómo buscaríamos un nombre en una lista de invitados sentados en una mesa rectangular?

La Solución: Ciclos Anidados

Para recorrer sistemáticamente cada fila y, dentro de cada fila, cada columna, necesitamos una herramienta que haga precisamente eso. Aquí es donde los **ciclos anidados** se vuelven indispensables.
Ahora sí, veamos el ejemplo de las calificaciones...

Agenda

1. Ejemplo 4: Dibujando un Triángulo
2. Estructuras de Datos 2D: Listas de Listas
3. Ejemplo 5: Procesando Datos de Estudiantes
4. Ejemplo 6: Encontrando el Valor Máximo

Ejemplo 5: Procesando Datos de Estudiantes

Objetivo

Tenemos una "lista de listas" donde cada lista interna representa las calificaciones de un estudiante. El objetivo es escribir un programa que itere a través de esta estructura para calcular e imprimir el promedio de calificaciones de **cada** estudiante.

Ejemplo 5: Procesando Datos de Estudiantes

Estructura de Datos y Salida Esperada

Dada la siguiente lista:

```
calificaciones_grupo = [
    [10, 9, 10],  # Calificaciones del Estudiante 1
    [8, 7, 8],    # Calificaciones del Estudiante 2
    [9, 9, 10]    # Calificaciones del Estudiante 3
]

```

La salida esperada en consola es:

```
Promedio del Estudiante 1: 9.67
Promedio del Estudiante 2: 7.67
Promedio del Estudiante 3: 9.33
```

Ejemplo 5: Lógica y Pistas

Nuestra Lógica

Para resolver esto, el ciclo externo se encargará de un estudiante a la vez, y el ciclo interno se encargará de procesar todas las calificaciones de ESE estudiante.

- **Ciclo Externo (por Estudiante):** Recorrerá la lista principal `calificaciones_grupo`. En cada iteración, obtendremos una de las listas internas (ej: `[10, 9, 10]`).

Ejemplo 5: Lógica y Pistas

Nuestra Lógica

Para resolver esto, el ciclo externo se encargará de un estudiante a la vez, y el ciclo interno se encargará de procesar todas las calificaciones de ESE estudiante.

- **Ciclo Externo (por Estudiante):** Recorrerá la lista principal `calificaciones_grupo`. En cada iteración, obtendremos una de las listas internas (ej: `[10, 9, 10]`).
- **Inicializar un Acumulador: Antes** de que empiece el ciclo interno, debemos crear una variable (por ejemplo, `suma = 0`). Es crucial que se reinicie a cero por cada nuevo estudiante.

Ejemplo 5: Lógica y Pistas

Nuestra Lógica

Para resolver esto, el ciclo externo se encargará de un estudiante a la vez, y el ciclo interno se encargará de procesar todas las calificaciones de ESE estudiante.

- **Ciclo Externo (por Estudiante):** Recorrerá la lista principal `calificaciones_grupo`. En cada iteración, obtendremos una de las listas internas (ej: `[10, 9, 10]`).
- **Inicializar un Acumulador: Antes** de que empiece el ciclo interno, debemos crear una variable (por ejemplo, `suma = 0`). Es crucial que se reinicie a cero por cada nuevo estudiante.
- **Ciclo Interno (por Calificación):** Para la lista de calificaciones del estudiante actual, este ciclo recorrerá cada número (cada calificación) y lo sumará a nuestra variable `suma`.

Ejemplo 5: Lógica y Pistas

Nuestra Lógica

Para resolver esto, el ciclo externo se encargará de un estudiante a la vez, y el ciclo interno se encargará de procesar todas las calificaciones de ESE estudiante.

- **Ciclo Externo (por Estudiante):** Recorrerá la lista principal `calificaciones_grupo`. En cada iteración, obtendremos una de las listas internas (ej: `[10, 9, 10]`).
- **Inicializar un Acumulador: Antes** de que empiece el ciclo interno, debemos crear una variable (por ejemplo, `suma = 0`). Es crucial que se reinicie a cero por cada nuevo estudiante.
- **Ciclo Interno (por Calificación):** Para la lista de calificaciones del estudiante actual, este ciclo recorrerá cada número (cada calificación) y lo sumará a nuestra variable `suma`.
- **Calcular el Promedio: Después** de que el ciclo interno haya terminado de sumar todas las notas, calculamos el promedio. La fórmula es `suma / cantidad_de_calificaciones`.

Ejemplo 6: Código y Salida

Código en Python

```
calificaciones_grupo = [
    [10, 9, 10],
    [8, 7, 8],
    [9, 9, 10]
]

for i in range (len(calificaciones_grupo)):
    suma_de_notas = 0
    for j in range (len(calificaciones_grupo[i])):
        suma_de_notas += calificaciones_grupo[i][j]
    promedio = suma_de_notas / len(calificaciones_grupo[i])
    print(f"Promedio del Estudiante {i + 1}: {promedio:.2f}")


```

Agenda

1. Ejemplo 4: Dibujando un Triángulo

2. Estructuras de Datos 2D: Listas de Listas

3. Ejemplo 5: Procesando Datos de Estudiantes

4. Ejemplo 6: Encontrando el Valor Máximo

Ejemplo 6: Encontrando el Valor Máximo

Objetivo

Dada una matriz (lista de listas) que representa las ventas diarias de diferentes sucursales durante una semana, escribir un programa que encuentre e imprima cuál fue la venta más alta registrada en general.

Ejemplo 6: Encontrando el Valor Máximo

Estructura de Datos y Salida Esperada

Dada la siguiente matriz de ventas:

```
ventas_semanales = [
  # Lun, Mar, Mie, Jue, Vie
    [150, 200, 180, 220, 250], # Sucursal 1
    [180, 190, 210, 200, 230], # Sucursal 2
    [120, 130, 110, 140, 100], # Sucursal 3
    [260, 240, 280, 270, 290]  # Sucursal 4
]

```

La salida esperada en consola es:

```
La venta mas alta registrada fue: 290
```

Ejemplo 6: Lógica y Pistas

Nuestra Lógica

Necesitamos una variable que recuerde.[el] número más grande que hemos visto hasta ahora. Luego, compararemos cada venta con ese recuerdo.

- **Inicializar una variable de seguimiento:** Antes de empezar los ciclos, crea una variable llamada `venta_maxima`. Una forma segura de inicializarla es con el **primer elemento** de la matriz (`ventas_semanales[0][0]`).

Ejemplo 6: Lógica y Pistas

Nuestra Lógica

Necesitamos una variable que recuerde.[el] número más grande que hemos visto hasta ahora. Luego, compararemos cada venta con ese recuerdo.

- **Inicializar una variable de seguimiento:** Antes de empezar los ciclos, crea una variable llamada `venta_maxima`. Una forma segura de inicializarla es con el **primer elemento** de la matriz (`ventas_semanales[0][0]`).
- **Ciclo Externo (por Fila):** Recorrerá cada lista de sucursal.

Ejemplo 6: Lógica y Pistas

Nuestra Lógica

Necesitamos una variable que recuerde.[el] número más grande que hemos visto hasta ahora. Luego, compararemos cada venta con ese recuerdo.

- **Inicializar una variable de seguimiento:** Antes de empezar los ciclos, crea una variable llamada `venta_maxima`. Una forma segura de inicializarla es con el **primer elemento** de la matriz (`ventas_semanales[0][0]`).
- **Ciclo Externo (por Fila):** Recorrerá cada lista de sucursal.
- **Ciclo Interno (por Venta):** Recorrerá cada venta individual dentro de la lista de la sucursal actual.

Ejemplo 6: Lógica y Pistas

Nuestra Lógica

Necesitamos una variable que recuerde.[el] número más grande que hemos visto hasta ahora. Luego, compararemos cada venta con ese recuerdo.

- **Inicializar una variable de seguimiento:** Antes de empezar los ciclos, crea una variable llamada `venta_maxima`. Una forma segura de inicializarla es con el **primer elemento** de la matriz (`ventas_semanales[0][0]`).
- **Ciclo Externo (por Fila):** Recorrerá cada lista de sucursal.
- **Ciclo Interno (por Venta):** Recorrerá cada venta individual dentro de la lista de la sucursal actual.
- **La Comparación (el 'if'):** Dentro del ciclo interno, compara la venta actual con tu variable `venta_maxima`. Si la venta actual es **mayor**, ¡actualiza `venta_maxima` con este nuevo valor!

Ejemplo 6: Lógica y Pistas

Nuestra Lógica

Necesitamos una variable que recuerde.[el] número más grande que hemos visto hasta ahora. Luego, compararemos cada venta con ese recuerdo.

- **Inicializar una variable de seguimiento:** Antes de empezar los ciclos, crea una variable llamada `venta_maxima`. Una forma segura de inicializarla es con el **primer elemento** de la matriz (`ventas_semanales[0][0]`).
- **Ciclo Externo (por Fila):** Recorrerá cada lista de sucursal.
- **Ciclo Interno (por Venta):** Recorrerá cada venta individual dentro de la lista de la sucursal actual.
- **La Comparación (el 'if'):** Dentro del ciclo interno, compara la venta actual con tu variable `venta_maxima`. Si la venta actual es **mayor**, ¡actualiza `venta_maxima` con este nuevo valor!
- **Imprimir el Resultado Final: Después** de que ambos ciclos hayan terminado.

Ejemplo 6: Código y Salida

Código en Python

```python
ventas_semanales = [
    [150, 200, 180, 220, 250],
    [180, 190, 210, 200, 230],
    [120, 130, 110, 140, 100],
    [260, 240, 280, 270, 290]
]

venta_maxima = ventas_semanales[0][0]

for sucursal in ventas_semanales:
    for venta in sucursal:
        if venta > venta_maxima:
            venta_maxima = venta

print(f"La venta mas alta registrada fue: {venta_maxima}")

```

Enlaces de los ejemplos

Ejemplo 4: Dibujando un Triángulo

https://github.com/AboudOnji/ExamplesAyP/blob/main/Example26.py

Ejemplo 5: Procesando Datos de Estudiantes

https://github.com/AboudOnji/ExamplesAyP/blob/main/Example28.py

Ejemplo 6: Encontrando el Valor Máximo

https://github.com/AboudOnji/ExamplesAyP/blob/main/Example29.py