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

Universidad Andrés Bello®

python powered

Resumen - Semana 2, Sesión 1 (Sesión 3)

Introducción a la Sesión 3

Sintaxis Básica de Python

Refuerzo de Tipos y Variables

Ejemplos Interactivos

Actividad Colaborativa

Conclusiones

Introducción a la Sesión 3

- **Abordar** la sintaxis básica de Python de forma más sistemática.
- **Profundizar** en los tipos de datos, variables y operaciones aritméticas.
- **Realizar** ejemplos interactivos en Google Colab para afianzar la comprensión.
- **Fomentar** la colaboración y el trabajo grupal.

Introducción a la Sesión 3 ∋ Conexión con Sesiones Previas

- Sesión 1 y 2 (Semana 1) se enfocaron en:
  - Introducción al entorno Google Colab.
  - Operaciones básicas y entrada/salida en Python.
  - Ejercicios simples de asignación y aritmética.
- Ahora profundizaremos en la **estructura y sintaxis** de Python.

Sintaxis Básica de Python

- **Indentación**: Bloques de código se definen por la sangría (`4 espacios` usualmente).
- **Uso de dos puntos** (`:`) para iniciar bloques (if, while, for, funciones, etc.).
- **Sensibilidad a mayúsculas/minúsculas**: `var` es distinto de `Var`.
- **Comentarios**: Empiezan con # en una sola línea o con `""" ... """` para bloques.

Ejemplo de comentarios en Python:

```python
# Esto es un comentario de una sola línea

"""
Este es un comentario
que abarca múltiples
líneas de texto
"""

x = 5  # Asigno 5 a x
print(x)
```

- **PEP8**: Recomendaciones oficiales de estilo (espacios, nombres de variables, etc.). https://peps.python.org/pep-0008/
- Facilita la lectura y el mantenimiento de código.

Sintaxis Básica de Python ∋ Palabras Reservadas y Convenciones

- **Palabras Reservadas**: `if`, `elif`, `else`, `for`, `while`, `import`, `class`, `def`, `return`, `...`
- **Nombrado de variables**:
  - Usar *snake_case* para variables y funciones: `mi_variable`, `calcular_area()`.
  - Mayúsculas para constantes: `PI = 3.14159`.
- **Longitud de línea**: Idealmente menor a 79 caracteres (conforme a PEP8).

Refuerzo de Tipos y Variables

Refuerzo de Tipos y Variables ∋ Recordatorio: Tipos de Datos en Python

- **int**: números enteros (`10, -3, 0, 9999`).
- **float**: números con punto decimal (`3.1415, 1.0, -2.5`).
- **complex**: números complejos (`3+4j`).
- **bool**: `True` / `False`.
- **str**: cadenas de texto (`"Hola"`, `'Mundo'`).

```python
# De string a entero
edad_str = input("Ingresa tu edad: ")
edad = int(edad_str)

# De string a float
altura_str = input("Ingresa tu altura (m): ")
altura = float(altura_str)

# De int a float
numero_entero = 5
numero_flotante = float(numero_entero)  # 5.0
```

- **ValueError** si la conversión es inválida (ej.: `int("hola")`).

```
# Orden de precedencia:
# 1. Paréntesis
# 2. Exponente **
# 3. Multiplicación, División, Módulo
# 4. Suma y Resta

a = 2 + 3 * 4       # 2 + 12 = 14
b = (2 + 3) * 4     # 5 * 4 = 20
c = 2**3 * 3        # 2^3 * 3 = 24
```

- **Evitar ambigüedades**: usar paréntesis cuando sea necesario.
- **Operadores compuestos**: `+=`, `-=`, `*=`, `/=`.

Refuerzo de Tipos y Variables ∋ Actividad 1: Ejercicios Rápidos en Colab

- Crea un nuevo **notebook** para “Semana 2, Sesión 1”.
- Implementa ejercicios:
  1. Calcular $(3 + 4) * 2^2$ y mostrar resultado.
  2. Ingresar una cadena y convertirla en `int` o `float`.
  3. Usar un **print** para concatenar texto con variables numéricas.
- **Tip**: Observa qué pasa si ingresas valores no válidos.

Ejemplos Interactivos

Ejemplos Interactivos ∋ Ejemplo 1: Manejo de Variables

```python
nombre = input("¿Cuál es tu nombre? ")
edad_str = input("¿Cuál es tu edad? ")
edad = int(edad_str)

# Jugando con los datos
edad_futura = edad + 5
print("Hola,", nombre)
print("Hoy tienes", edad, "años.")
print("Dentro de 5 años tendrás", edad_futura, "años.")
```

Discusión: Manejo de `int` vs. `str`, impresiones múltiples.

```python
# Aritmética directa en una celda Colab
result1 = (2 + 3) * (5 - 1)
result2 = 10 / 2 + 6 // 3 - 4
print("Result1 =", result1)
print("Result2 =", result2)
```

Punto a destacar:

- `//` es **división entera** (trunca el resultado).
- `/` es división flotante (da decimal).

Actividad Colaborativa

Actividad Colaborativa ∋ Ejercicio Grupal: Mini-Calculadora

Enunciado

- Crear un programa que simule una “mini-calculadora”.
- Se piden dos números (float) y una operación (`+`, `-`, `*`, `/`).
- Se muestra el resultado de la operación.

Extensión: Manejar la división por cero con un mensaje de error.

Actividad Colaborativa ∋ Trabajo en Grupos

- Equipos de 2-3 integrantes.
- Cada equipo edita un notebook compartido en Google Colab.
- Discutir la mejor forma de:
  - Pedir los datos al usuario.
  - Validar la operación.
  - Mostrar los resultados.
- Al final, compararán las soluciones y estrategias.

Actividad Colaborativa ∋ Ejemplo de Solución (Mini-Calculadora)

```python
num1_str = input("Ingresa el primer número: ")
num2_str = input("Ingresa el segundo número: ")
op = input("Ingresa la operación (+, -, *, /): ")

num1 = float(num1_str)
num2 = float(num2_str)

if op == "+":
    res = num1 + num2
elif op == "-":
    res = num1 - num2
elif op == "*":
    res = num1 * num2
elif op == "/":
    if num2 == 0:
        res = "Error: división por cero"
    else:
        res = num1 / num2
else:
    res = "Operación inválida"

print("Resultado:", res)
```

- **Uso de if/elif/else**: Estructura básica de control, veremos más en profundidad pronto.
- **Validaciones**: Cómo manejar entradas no numéricas o el caso de división por cero.
- **Buenas prácticas**: Comentar líneas clave, usar nombres de variables descriptivos.

Actividad Colaborativa ∋ Discusión en Grupo

- ¿Qué método usó cada grupo para manejar la división por cero?
- ¿Cómo presentar un mensaje de error informativo?
- ¿Se te ocurrió alguna manera de extender la mini-calculadora? (e.g. potencia, módulo).

Comparte tus conclusiones con la clase.

Conclusiones

Conclusiones ∋ Resumen de la Sesión

- Reafirmamos la **estructura básica** y reglas de sintaxis en Python (indentación, mayúsculas, etc.).
- Exploramos con mayor detalle los **tipos de datos** y la **aritmética**.
- Realizamos ejercicios prácticos en Colab para afianzar la comprensión.
- Desarrollamos una pequeña **mini-calculadora** colaborativamente.

Conclusiones ∋ Próximos Pasos

- **Sesión 4 (Semana 2)**: Continuar con **estructuras de control** (`if`, `while`), y ejercicios más complejos.
- **Tarea sugerida**:
  - Practicar la creación de scripts con menús (mini-calculadora mejorada).
  - Revisar la documentación de Python sobre tipos numéricos y funciones matemáticas.

- **Python Official Tutorial** (apartados 1-3).
- **Real Python** (guías introductorias).
- **PEP8 Style Guide**: Recomendaciones de estilo y formato (útil para mantener código limpio).

¡Desafío!

- Crea un programa que pida el **radio de un círculo** y elija:
  - '1' para calcular el **diámetro**.
  - '2' para calcular la **circunferencia**.
  - '3' para calcular el **área**.
- Muestra el resultado correspondiente (usa `math.pi`).

Comentario: Requiere condicionales y entradas numéricas.

Conclusiones ∋ Recomendación para Estudio

- Al finalizar la clase, escribe **notas de lo aprendido** en tu cuaderno.
- Identifica *dónde* has tenido más dudas:
  - Manejo de strings vs. números.
  - Operaciones matemáticas.
  - Orden de ejecución en un notebook.
- Prepara preguntas concretas para la siguiente sesión.

Conclusiones ∋ Retroalimentación

- ¿Te sientes más cómodo con la sintaxis de Python?
- ¿Ha quedado clara la importancia de la indentación?
- ¿Tienes inquietudes sobre el uso de Google Colab u otro entorno?

Expón tus dudas e impresiones.

¡Gracias y hasta la próxima sesión!

- Asegúrate de **guardar tu notebook** en Drive.
- Si tienes tiempo, aborda el ejercicio adicional.
- Nos vemos en la **Sesión 4 (Semana 2)** para profundizar en `if` y `while`.