

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 y Repaso

Módulos y Paquetes (Repaso)

Librerías Externas

Tarea Semanal

Introducción y Repaso

Introducción y Repaso \ni Recapitulación de la Sesión Anterior (Sesión 6)

- **Semana 3, Sesión 2 (Sesión 6)** se centró en:
 - **Funciones:** sintaxis (`def`), parámetros, valores por defecto, alcance de variables.
 - **Módulos y Paquetes:** cómo organizar el código en archivos `.py` y carpetas.
 - Ejemplos de proyectos pequeños con `import` y definición de funciones útiles.
- **Objetivo de hoy:** Ampliar la práctica con funciones y módulos, e introducir el uso de librerías externas (vía `pip` o Colab).

- **Profundizar** en el flujo de trabajo al crear y reutilizar módulos en Python.
- **Explorar** la instalación de librerías externas (**pip**, Google Colab).
- **Diseñar** una actividad grupal donde se combine la creación de funciones propias con el uso de librerías de terceros.
- **Fomentar** la colaboración y la discusión sobre buenas prácticas de organización.

Módulos y Paquetes (Repaso)

- Carpetas y `.py` para agrupar funcionalidades.
- Ejemplo:
- `main.py` orquesta la lógica usando `import mis_modulos.fisica` y así sucesivamente.
- Facilita la mantenibilidad y escalabilidad.

- Import completo:

```
1 import mis_modulos.matematicas
2 res = mis_modulos.matematicas.sumar(2, 3)
```

- From / Import:

```
1 from mis_modulos.matematicas import sumar
2 res = sumar(2, 3)
```

- Import renombrado:

```
1 import mis_modulos.matematicas as mm
2 res = mm.sumar(2, 3)
```

Librerías Externas

Librerías Externas \ni ¿Por qué Librerías Externas?

- **Ahorra tiempo:** aprovechas código ya probado por la comunidad.
- **Funcionalidades avanzadas:** Desde manejo de redes hasta machine learning.
- **Ejemplos:** `requests` para peticiones web, `numpy` para cálculo numérico, `pandas` para data frames, etc.
- **Comunidad activa:** librerías mantenidas, actualizaciones frecuentes.

- **pip**: el gestor de paquetes oficial de Python.

- Comando general en terminal:

```
pip install nombre_paquete
```

- Si usas **Google Colab**, puedes instalar temporalmente en una celda:

```
1 !pip install nombre_paquete
```

- La librería quedará disponible para importarse en el resto del entorno (hasta reiniciar).

Librerías Externas \ni Ejemplo: requests en Colab

```
1  # En una celda de Colab:
2  !pip install requests
3
4  import requests
5
6  resp = requests.get("https://api.github.com")
7  print(resp.status_code)
8  print(resp.json())
```

- **Uso real:** Conectarse a APIs, descargar datos, etc.
- **Sugerencia:** Manejar casos de error (`resp.status_code != 200`).

```
# Desde terminal local  
pip install numpy
```

```
1 # Uso en el código  
2 import numpy as np  
3  
4 arr = np.array([1, 2, 3, 4])  
5 print(arr * 2) # [2 4 6 8]
```

- **NumPy** es la base de muchas librerías científicas en Python.
- **Operaciones vectorizadas:** eficiencia y simplicidad.

Tarea Semanal

Manos a la Obra!

- **Sesion 9 (Semana 5):** Repaso integral de Unidades I y II, y Solemne I.
- **Recomendación:**
 - Revisa todos los conceptos vistos: Sintaxis, Estructuras de Control, Funciones, Módulos.
 - Práctica con ejercicios y ejemplos de exámenes pasados (si los hubiera).

¡Prepárate para la evaluación!

- Official Python Packaging Tutorial
- Documentación de la Biblioteca Estándar de Python
- PyPI - Python Package Index
- Numpy Docs
- Matplotlib Docs

¡Muchas gracias y éxito en su práctica!

- Recuerden subir su trabajo a Google Drive o repositorio compartido.
- Próxima sesión: **Solemne I** y repaso integral.
- ¡Sigán explorando librerías externas y creando módulos propios!