UNTREF - Ingeniería de Sonido
Introducción a Python para Ingenieros

Alejandro Gronskis

Año 2020

Objetivo del curso

El principal objetivo del mismo reside en proporcionar una introducción al lenguaje de programación Python enfocado principalmente a científicos, ingenieros o cualquier persona interesada en análisis y visualización de datos. En este taller de introducción a Python científico, enseñaremos a utilizar las herramientas básicas que forman su ecosistema.
Este curso pretende, por medio de un entorno contributivo de acceso libre que denominamos cuadernos Jupyter (archivos `.ipynb` que son cargados y procesados por algún entorno de ejecución como Cocalc o Colab) poner a disposición de los alumnos una herramienta web diseñada en base a la practicidad, con ejemplos extraídos de asignaturas comunes en carreras científicas y de ingeniería.

Metodología

El empleo de cuadernos Jupyter provee de un material útil para la enseñanza y el aprendizaje de los fundamentos de **Python**: un lenguaje de programación dinámico, interpretado y fácil de aprender. En concordancia con una multitud de libros de texto disponibles en forma gratuita [1, 2, 3], los cuadernos abordan tareas cuyo desarrollo implica tratar los algoritmos y métodos requeridos. Entrelazando conceptos técnicos, detalles matemáticos, ejemplos de código, ilustraciones y archivos de video en un entorno interactivo **Jupyter Notebooks** permite al alumno acortar la brecha entre clases Teóricas y Prácticas.
A fin de expandir con nuevas capacidades los sistemas tradicionales de gestión del aprendizaje (como Moodle) se pretende introducir a los alumnos en la elaboración

de actividades prácticas y tareas usando la herramienta Github Education en el aula [4].

TEMARIO

- Instalación de paquetes
- Uso del *Notebook* de Jupyter
- Introducción a la sintaxis de Python
- Introducción a NumPy
- Representación gráfica con Matplotlib
- Análisis numérico con SciPy
- Cálculo simbólico con SymPy
- Uso de *Widgets* interactivos
- Introducción a la depuración con pdb, testing y buenas prácticas

Parte 1. Introducción a Python y Jupyter

Presentación. Tipos de datos. Operador de asignación. Operadores de comparación. Estructuras de control: condicionales, bucles `while` y `for`. Sentencias `break` y `pass`. Otros tipos de datos: listas y tuplas. Definición de funciones: `def` y cadena de documentación. Guía de estilo PEP8.

Parte 2. Arrays de NumPy

Introducción a NumPy. Constantes y funciones matemáticas. Definición y utilidad de los *arrays*. Indexación de arrays. Creación de arrays: funciones `empty`, `eye`, `identity`, `like`. Creación de rangos: funciones `linspace`, `logspace`, `meshgrid`. Operaciones con arrays.

Parte 3. Visualización con Matplotlib

Qué es Matplotlib? Interfaz pyplot. Galería de ejemplos de matplotlib.org. Interfaz orientada a objetos de Matplotlib: función `subplots`, propiedades, mapas de colores, tipos de marcadores. Gráficas en dos dimensiones: funciones `meshgrid`, `contour`, `contourf`. Importación de datos.

Parte 4. SciPy

Qué es SciPy? Integración numérica: paquete `integrate`, funciones `quad`. Ecuaciones diferenciales ordinarias: función `odeint`. Transformación de una ecuación diferencial de segundo orden. Ecuaciones algebraicas no lineales: paquete `optimize`. Interpolación: paquete `interpolate`. Ajuste de curvas: función `curve fit`.

Parte 5. SymPy

Introducción y comandos básicos. Creación de expresiones, adición de símbolos e hipótesis. Simplificación y manipulación de expresiones. Derivadas e integrales simbólicas. Series y resolución de ecuaciones algebraicas y diferenciales ordinarias. Algebra lineal, matrices y representación gráfica. Pasaje de simbólico a numérico para ganar eficiencia: uso de `lambdify`.

Parte 6. Widgets interactivos

Ejemplo simple de widget interactivo de Jupyter: Gráficas interactivas con Matplotib y NumPy. Deslizadores (*sliders*), botones y casillas de verificación (*checkboxes*).

Parte 7. Buenas prácticas de programación con Python

Corrección de código y pruebas. Depuración de fallos y optimización de código.

Referencias

[1] Jake VanderPlas, 2016, Python Data Science Handbook

[2] Jake VanderPlas, 2016, A Whirlwind Tour of Python

[3] Al Sweigart, 2019, Automate the Boring Stuff with Python

[4] Integración de la herramienta Github Education en el aula