



Guía docente

240AR064 - 240AR064 - Scientific Python para Ingenieros

Última modificación: 16/04/2024

Unidad responsable: Escuela Técnica Superior de Ingeniería Industrial de Barcelona
Unidad que imparte: 707 - ESAII - Departamento de Ingeniería de Sistemas, Automática e Informática Industrial.

Titulación: MÁSTER UNIVERSITARIO EN AUTOMÁTICA Y ROBÓTICA (Plan 2012). (Asignatura optativa).
MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL (Plan 2014). (Asignatura optativa).
MÁSTER UNIVERSITARIO EN INVESTIGACIÓN EN INGENIERÍA MECÁNICA (Plan 2024). (Asignatura optativa).

Curso: 2024 **Créditos ECTS:** 3.0 **Idiomas:** Inglés

PROFESORADO

Profesorado responsable: Perera Lluna, Alexandre

Otros: Perera Lluna, Alexandre
Velasco Garcia, Manuel

COMPETENCIAS DE LA TITULACIÓN A LAS QUE CONTRIBUYE LA ASIGNATURA

Transversales:

CT3. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo interdisciplinar, ya sea como un miembro más o realizando tareas de dirección, con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

CT4. USO SOLVENTE DE LOS RECURSOS DE INFORMACIÓN: Gestionar la adquisición, la estructuración, el análisis y la visualización de datos e información en el ámbito de especialidad, y valorar de forma crítica los resultados de dicha gestión.

METODOLOGÍAS DOCENTES

OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

El objetivo de la asignatura es desarrollar habilidades para la programación científica, especialmente orientadas en la aplicación de algorítmica de reconocimiento de patrones en robótica en el lenguaje de programación python. La clase desarrollará habilidades orientadas a la resolución de problemas en un entorno competitivo.

HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTADO

Tipo	Horas	Porcentaje
Horas grupo pequeño	27,0	36.00
Horas aprendizaje autónomo	48,0	64.00

Dedicación total: 75 h

CONTENIDOS



Python Científico para Ingeniería

Descripción:

Part I

1. Introduction
 - a. Why python?
 - b. Python History
 - c. Installing Python
 - d. Python resources
2. Working with Python
 - a. Workflow
 - b. ipython vs. CLI
 - c. Text Editors
 - d. IDEs
 - e. Notebook
3. Getting started with Python
 - a. Introduction
 - b. Getting Help
 - c. Basic types
 - d. Mutable and in-mutable
 - e. Assignment operator
 - f. Controlling execution flow
 - g. Exception handling
4. Functions and Object Oriented Programming
 - a. Defining Functions
 - b. Decorators
 - c. Writing Scripts and New Modules
 - d. Input and Output
 - e. Standard Library
 - f. Object-oriented programming
 - g. Magic Functions
5. Iterators and Generators
 - a. Iterators
 - b. Generators
6. Creating Graphic Interfaces (optional)
7. Debugging code
 - a. Avoiding bugs
 - b. Debugging workflow
 - c. Python's debugger
 - d. Debugging segfaults using gdb

Part II

1. Introduction to NumPy
 - a. Overview
 - b. Arrays
 - c. Operations on arrays
 - d. Advanced arrays (ndarrays)
 - e. Notes on Performance (\%timeit in ipython)
2. Matplotlib
 - a. Introduction
 - b. Figures and Subplots
 - c. Axes and Further Control of Figures
 - d. Other Plot Types
 - e. Animations
3. Plotting with Mayavi
 - a. Mlab: the scripting interface
 - b. Interactive work
4. Advanced Numpy
 - a. Life of ndarray
 - b. Universal functions
 - c. Interoperability features



- d. Array siblings: chararray, maskedarray, matrix
- e. Summary
- f. Contributing to Numpy/Scipy

Part III

- 1. Scipy
 - a. Introduction
 - b. Input/Output
 - c. Statistics
 - d. Linear Algebra
 - e. Fast Fourier Transforms
 - f. Optimization
 - g. Interpolation
 - h. Numerical Integration
 - i. Signal Processing
 - j. Image Processing
 - k. Special Functions
- 2. Sparse Matrices in SciPy
 - a. Introduction
 - b. Storage Schemes
 - c. Linear System Solvers
 - d. Others
- 3. Optimizing code
 - a. Optimization workflow
 - b. Profiling your code
 - c. Speeding your code
- 4. Sympy
 - a. First Steps with SymPy
 - b. Algebraic manipulations
 - c. Calculus
 - d. Equation solving
 - e. Linear Algebra

Part IV

- 1. Python scikits
 - a. Introduction
 - b. scikit-timeseries
 - c. scikit-audiolab
- 2. scikit-learn
 - a. Datasets
 - b. Sample generators
 - c. Unsupervised Learning
 - i. Clustering
 - ii. Gaussian Mixture Models
 - iii. Novelty/Outliers Detection
 - d. Supervised Learning
 - i. Linear and Quadratic Discriminant Analysis
 - ii. Nearest Neighbors
 - iii. Support Vector Machines
 - iv. Partial Least Squares
 - e. Feature Selection
- 3. Practical Introduction to Scikit-learn
 - a. Solving an eigenfaces problem
 - i. Goals
 - ii. Data description
 - iii. Initial Classes
 - iv. Importing data
 - b. Unsupervised analysis
 - i. Descriptive Statistics
 - ii. Principal Component Analysis
 - iii. Clustering



- c. Supervised Analysis
 - i. k-Nearest Neighbors
 - ii. Support Vector Classification
 - iii. Cross validation

Dedicación: 30h

Grupo grande/Teoría: 15h

Grupo pequeño/Laboratorio: 3h

Actividades dirigidas: 5h

Aprendizaje autónomo: 7h

SISTEMA DE CALIFICACIÓN

BIBLIOGRAFÍA

Básica:

- Bressert, Eli. SciPy and NumPy: An Overview for Developers. New York: O'Reilly, 2012. ISBN 978-1449305468.

- McKinney, Wes. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython [en línea]. 3rd ed. Sebastopol: O'Reilly, 2022 [Consulta: 30/03/2023]. Disponible a : <https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?pq-origsite=primo&docID=29441847>. ISBN 9781098104009.