INTRODUCTION TO PYTHON FOR SCIENCE AND ENGINEERING

By Dr. Manel Velasco (Manel.Velasco@upc.edu)
Dr. Alexandre Perera (Alexandre.Perera@upc.edu)

From Jan 28, 2013 to Feb 8, 2013
11.00 to 13.00

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 Squeares
   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