Python for Engineers

Course Objective

In Python for Engineers, we will learn how to use the programming language python to effectively analyze and assess data. This course will give users hands-on experience in learning python. In addition, we will explore the pythonic way to perform solve and model common numerical methods in thermofluidic motion. There will also be an introduction to machine learning. Ideally, users will use python to solve a new problem of their choosing or an old problem previously written in another language.

I: Introduction to python

Toolbox setup and background

- Begin setup and installation of python tools
- Install and configure python platform Anaconda and explore
    - Spyder
    - Jupyter notebook
- Create a new environment to install and use different versions of python
- Pip install commonly used python packages
- Python pros and cons
- Use the python profiler to understand Spyder backend processes
- Write code to visualize results in the python console

Variables, expressions, and statements

- Topics include:
    - values and types
    - variables
    - keywords
    - statements
    - operations, expressions, string operations
- Exercises

Conditional execution

- Topics include:
    - Boolean expressions
    - logical operators
    - conditional execution
    - chained and nested conditionals
    - exception catching

Functions

- Topics include
    - Function calls
    - Built in functions
    - Math functions
- Exercises

Iteration

- Topics include:
  - Updating variables
  - While statements
  - Infinite loops
  - Using continue, for, and break in loops
- Exercises

Strings

- Topics include:
  - About strings and accessing properties
  - Operators
  - comparison
  - Parsing strings
- Exercises

Files

- Topics include:
  - Opening files
  - Reading files
  - Using try, except, and open
  - Debugging
- Exercises

Lists

- Topics include:
  - About lists
  - operations
  - slices
  - methods
  - aliasing
- Exercises

Tuples and dictionaries

- Topics include:
  - About each
  - Assigning
  - Accessing
  - Looping
- Exercises

Regular Expressions

- Topics include:
  - Character matching
  - Extracting data
  - Searching and combining with extracting
  - Looping
- Exercises

Web scraping and natural language

- Topics include:
  - Parsing and scraping web

    - Parsing using regular expression
    - Parsing using BeautifulSoup
    - Analyze unstructured text data using common python packages
    - Google geocoding
- Exercises

Visualizing data

- Topics include:
    - Building a google map from geocoded data
    - Visualizing networks
- Exercises

Flask tutorial

- creating a basic blog application
- Users will be able to register, log in, create posts, and edit or delete their own posts

II: Using Python as an Engineer

Automating common tasks

- Topics include:
    - File names and paths
    - Command line arguments
    - pipes
- Exercises

Data cleansing

- organizing, cleaning, and managing data
- Methods include: subset, sort, reshape, and merge.

Parallel programming

- How to run parallel processes in python
- Examples
- Work on translating your parallel code

Numpy and Matplotlib

- Using numpy
    - np.linspace, np.pi, np.meshrid np.exp, np.sin/cos np.array np.zeroes
- Plotting data with matplotlib
    - Contour plots
    - 3d plots
    - matplotlib.animation

Solutions of systems of linear algebraic equations

- Conjugated gradient
- SIP – strongly implicit procedure by Stone

Interpolation of grid data

- Algorithms / functions to interpolate data between numerical grids

Conjugate Gradient Descent in Python|

- Steepest descent
- Conjugate directions
- Gram-Schemidt conjugatio
- Conjugate gradients

Additional Materials

Modeling and Simulation in python

- Iterative modeling
- Sweeping parameters
- Modeling growth
- Quadratic growth
- Optimization
- Mixing
- Pharmacokinetics
- Numerical methods
- Projectiles in 2d
- The Manny Ramirez problem
- Rotation
- Torque
- Pressure drop modeling
- Second-order linear diffusion

Introduction to Machine Learning

- General liner models
  - logistic regression
  - binary outcomes
  - Poisson Regression
- Decision trees
  - bagging and boosting
  - dimensional reduction

Neural networks

- artificial neural networks
- fuzzy models
- genetic algorithms
- swarm intelligence.

In class project work

- Students outline a project that they want to solve
- Use class time to

- work through difficulties
- share ideas with peers
- develop more sophisticated methods
- prepare client ready presentation

Prerequisites

There are no formal prerequisites required for Programming in python. While this course spans a large range of complex data science concepts, the curriculum has been designed to accommodate all ranges of data science skills. The expectations are that students attend class regularly, actively participate, and complete assignments.

Grading

Students will likely have different levels of experience coding as well as different uses for python. Grading will not be based on a test, but rather based on each individual's ability to use python for what they want to use it for. This could include but is not limited to:

- Solving math problems with python https://projecteuler.net/archives
- Making a game with python http://inventwithpython.com/chapters/
- Rewriting an old script in python
- Making something brand new in python