Example 1: Signal Processing with Python in Jupyter Notebooks

This example is a Jupyter Notebook (teaching notebook/problem set) that that illustrates how I use a fun but effective narrative to interactively teach simple programming concepts. This is typical of how I used interactive activities in my classroom. In particular, my strategy here is to separate the programming concepts from the science concept, to avoid unnecessary confusion. This can be seen in the following pages, which are created from three files:

1. The README file provided to students explaining the assignment
2. The Programming Assignment, a Jupyter Notebook that introduces concepts like iterables and stem plots
3. The Workshop Assignment, a Jupyter Notebook that uses iterables and stem plots to interpret and evaluate a spectral analysis using the Discrete Fourier Transform.

Note that both notebook files are actually “solution” notebooks: the solutions are included in the code, and the code provided to the students is commented in the code cells (typically with missing parts to fill in).

You can read the assignments in a browser or download the notebook files using the following links:

1. README: **read in a browser**
2. Programming Assignment: read in a browser as **student** or **solution** versions, or download the notebooks as **student** or **solution** versions.
3. Workshop Assignment: read in a browser as **student** or **solution** versions, or download the notebooks as **student** or **solution** versions.

Note that the notebooks are compatible with most versions of Python 3.X and a virtual environment containing numpy and matplotlib.

The documents provided here are explicitly shared here as part of the portfolio of Robert Lanzafame. We are currently working on finalizing the open source licenses, so please contact Robert at R.C.Lanzafame@tudelft.nl if you are interested in using them.