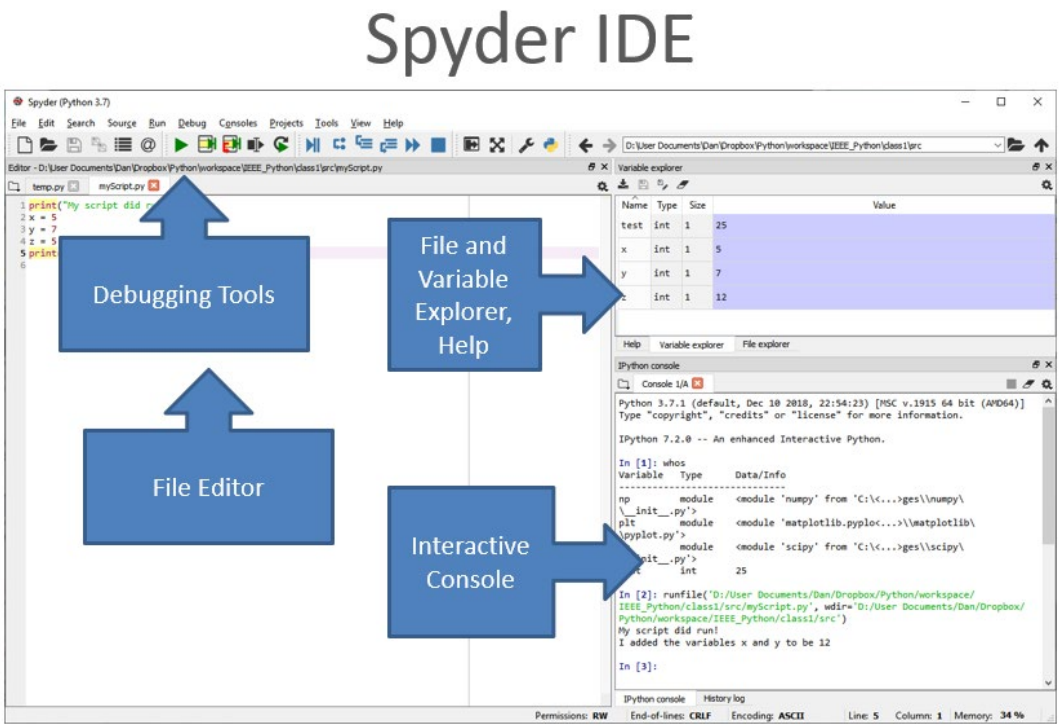
PYTHON APPLICATIONS FOR DIGITAL DESIGN AND SIGNAL PROCESSING

Overview: This is a hands-on course for the popular and powerful open-source Python programming language combining **live Q&A/workshop sessions** together with **pre-recorded lectures** that students can watch **on their own schedule** and an **unlimited number of times**. The videos will be available to the students for viewing up to two months after the conclusion of the course.

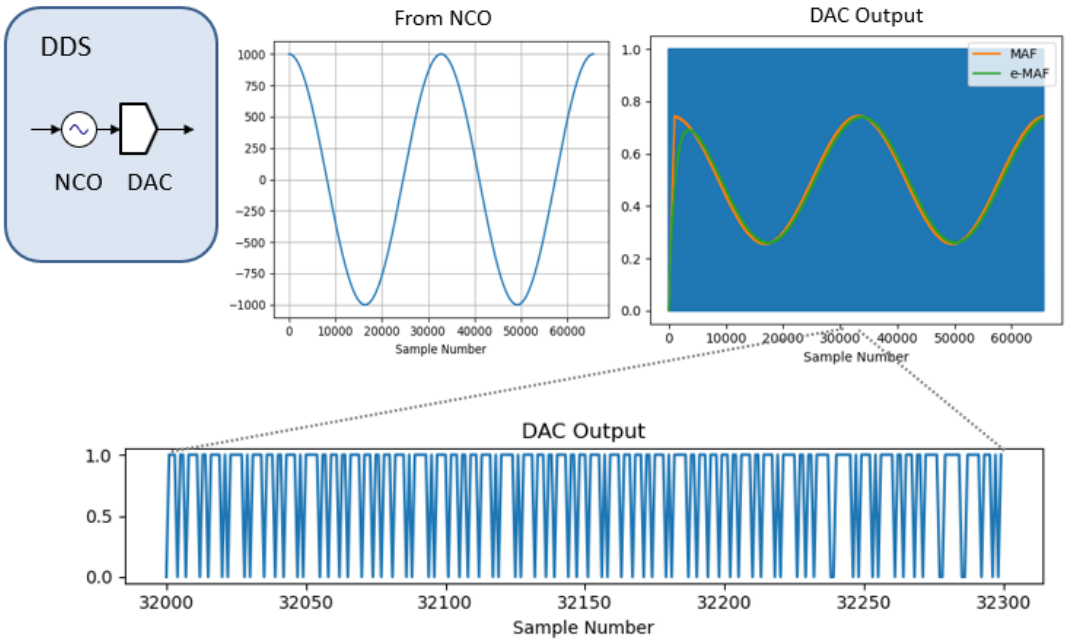
Whether you are new to Python or a new user, Dan provides simple, straight-forward navigation through the multiple configurations and options, providing a best-practices approach for quickly getting up to speed using Python for modelling and analysis for applications in signal processing and digital design verification. Students will be using the Anaconda distribution, which combines Python with the most popular data science applications, and Jupyter Notebooks for a rich, interactive experience.

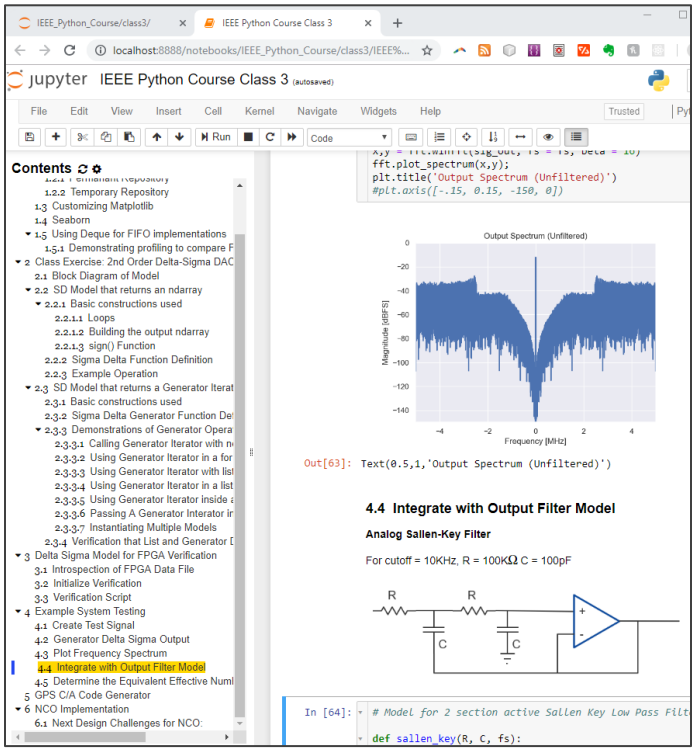
The course begins with basic Python data structures and constructs, including key "Pythonic" concepts, followed by an overview and use of popular packages for scientific computing enabling rapid prototyping for system design.

During the course students will create example designs including a sigma delta converter and direct digital synthesizer both in floating point and fixed point. This will include considerations for cycle and bit accurate models useful for digital design verification (FPGA/ASIC), while bringing forward the signal processing tools for frequency and time domain analysis.



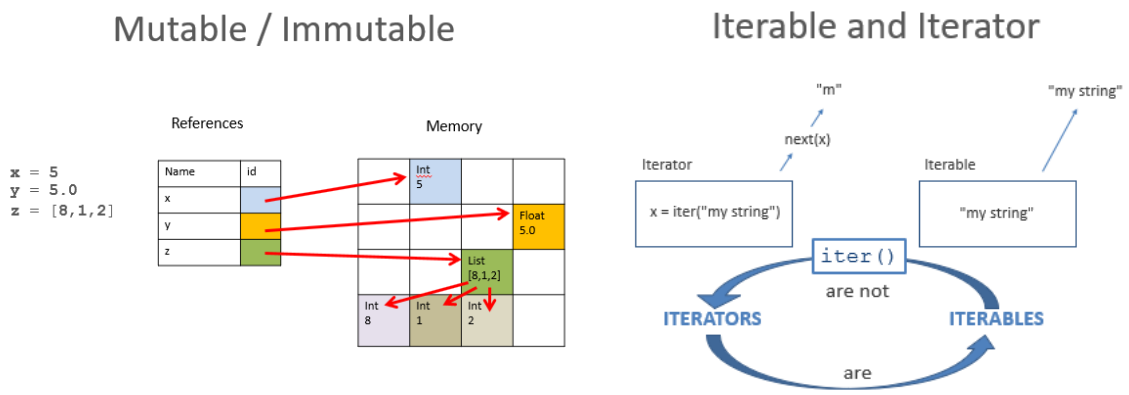
Using Scipy





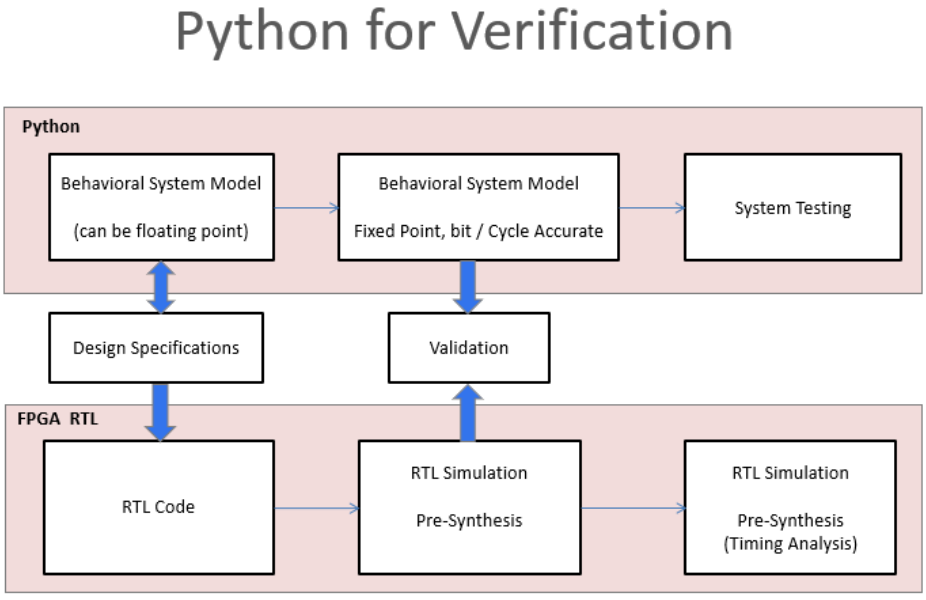
Jupyter Notebooks: This course makes extensive use of Jupyter Notebooks which combines running Python code with interactive plots and graphics for a rich user experience. Jupyter Notebooks is an open-source web-based application (that can be run locally) that allows users to create and share visually appealing documents containing code, graphics, visualizations and interactive plots. Students will be able to interact with the notebook contents and use "take-it-with-you" results for future applications in signal processing.

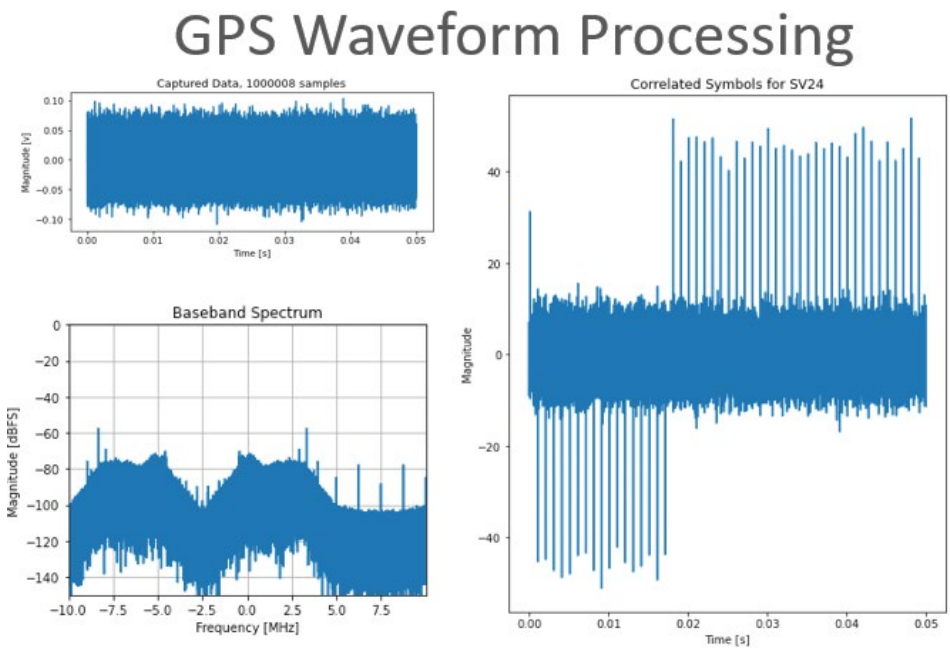
Target Audience: This course is targeted toward users with little to no prior experience in Python, however familiarity with other modern programming languages and an exposure to object-oriented constructs is very helpful. Students should be comfortable with basic signal processing concepts in the frequency and time domain. Familiarity with Matlab or Octave is not required, but the equivalent operations in Python using the NumPy package will be provided for those students that do currently use Matlab and/or Octave for signal processing applications. Those already familiar with Python should also expect to benefit from taking the course covering the latest features in Python 3.



Benefits of Attending / Goals of Course: Attendees will gain an overall appreciation of using Python and quickly get up to speed in best practice use of Python and related tools specific to modeling and simulation for signal processing analysis and design.

A computer (Mac, Linux or Windows OS) preconfigured with Anaconda is highly recommended for hands-on learning; the specific installation instructions will be emailed to students prior to the start of class.





Biography:

Dan Boschen has a MS in Communications and Signal Processing from Northeastern University, with over 25 years of experience in system and hardware design for radio transceivers and modems. He has held various positions at Signal Technologies, MITRE, Airvana and Hittite Microwave designing and developing transceiver hardware from baseband to antenna for wireless communications systems. Dan is currently at Microchip (formerly Microsemi and Symmetricom) leading design efforts for advanced frequency and time solutions.

For more background information, please view Dan's Linked-In page.

Pre-recorded lectures (3 hours each) will be distributed Friday prior to all Workshop dates. Workshop/ Q&A Sessions are 7pm-8pm on the dates listed below:

Tuesday, June 22

Topic 1: Intro to Jupyter Notebooks, the Spyder IDE and the course design examples. Core Python constructs.

Tuesday, June 29

Topic 2: Core Python constructs; iterators, functions, reading writing data files.

Tuesday, July 6

Topic 3: Signal processing simulation with popular packages including NumPy, SciPy, and Matplotlib.

Tuesday, July 13

Topic 4: Bit/cycle accurate modelling and analysis using the design examples and simulation packages