CMU ONLINE
Carnegie Mellon University

Syllabus Overview Sample: Python for Data Science II

Program: Machine Learning & Data Science Certificate Program

Faculty: Dr. Ralf Brown, Principal Systems Scientist, Language Technologies Institute

Course Number: 11905 **Units:** 6

Zoom Class: Weekday Evening, ET

Course Description

This is the second mini of a two-course sequence on the Python programming language, with a focus on learning concepts, techniques, skills, and tools needed for developing programs in Python. In comparison with the first mini, this course covers more advanced ways to load, format, and store data, including APIs, databases and pandas. This course also provides an introduction to tools that are useful when developing large software projects, including coding standards, version control, unit testing and debugging. Finally, the course covers more abstract building blocks including implementing more complex data structures and using numpy, both of which are the foundation of training and applying modern models and methods. Course projects include working with real datasets from a variety of sources.

Course Learning Objectives

By the end of this course, students will be able to:

- Explain and use fundamental constituents of the Python programming language.
- Use the Python language, Jupyter, and git to write code to solve a variety of computational problems.
- Retrieve, process, analyze and visualize real-world data using Python data science libraries, APIs, RDMBS databases, SQL, and pandas.

Faculty Introduction



Dr. Ralf Brown is a Principal Systems Scientist at Carnegie Mellon University's Language Technologies Institute (LTI). With a Ph.D. in Computer Science from Carnegie Mellon, Dr. Brown has been a pivotal member of the institute since 1993, contributing to groundbreaking advancements in machine translation, language identification, and computational linguistics. His distinguished career has been recognized with awards such as the IJCAI

Distinguished Paper Award and Carnegie Mellon's Allen Newell Award for Research Excellence.

Dr. Brown's research spans the development of innovative tools and techniques in example-based machine translation, corpus linguistics, and low-resource language processing. He is the creator of the CMU-EBMT, an open-source example-based machine translation system, and the LTI LangID Corpus, a resource for training language identifiers in over 2,000 languages. His work combines theoretical insights with practical applications, advancing the frontiers of natural language processing in diverse and multilingual contexts.

A dedicated educator and mentor, Dr. Brown leads courses on algorithm design and computational linguistics, emphasizing hands-on learning and real-world problem-solving. He has repeatedly served as Chair of the PhD Admissions Committee. He also actively contributes to open-source projects, including the widely used darktable image editor. With decades of experience and a passion for linguistic innovation, Dr. Brown continues to inspire the next generation of researchers at LTI while driving progress in language technologies worldwide.

Course Topics

Topic	Learning Objectives
Pandas and Data Cleaning	● Clean up a dataset with missing and noisy data. ● Use Pandas to create, modify and output formatted data, including indexing, joining and subsetting data frames.
API Calls	● Connect to an API and extract meaningful information. ● Use the requests library. ● Diagnose errors (bad requests). ● Register for an API key and use best practices to keep that key secure. ● Make calls to APIs, use them, and format the data. ● Call the OpenAI API and use it to rewrite input.
Debugging	● Recognize different types of bugs - e.g., conceptual, runtime. ● Use the python debugger (pdb) to inspect functions. ● Use pytest to write unit tests for a python library. ● Time different code and types of implementations. ● Be able to predict how code will scale. ● Use %timeit in jupyter and other utilities to time some common operations.
Databases	● Write SQL queries that join databases. ● Explain how to query a database using Python libraries.

Code Style and Version Control	● Identify when code follows good style and conventions. ● Modify and Maintain code using git for version control.
Tries and Advanced Data Structures	● Construct a trie data structure. ● Compare the run time of different data structures.
Numpy	● Vectorize code using numpy ● Recognize issues with floating point, including comparing for equality

---

NOTE: This is a sample designed to provide an overview of topics covered in the course based on previous course syllabi. All students will be provided an updated and final syllabus at the start of the course. Faculty, content, and schedule are all subject to change.