**Python for Data Science**

**10-week**

**Batch 3**

June 26 - August (→ September) 26, 2020

**======================================================================**

**CURRICULUM**

Weeks:

1. Introduction to
   1. Course
   2. Logistics
   3. Jupyter Notebook - Integrated Development environment (IDE)
      1. Installation
      2. Basic navigation
      3. Help, Documentation
      4. Shell commands
      5. Magic Commands
   4. Python
2. Python Fundamentals - Essentials
   1. Basic syntax
   2. Basic data types
      1. Numerics
      2. Bool
      3. String
   3. Containers
      1. Lists
      2. Dictionaries
      3. Sets
      4. Tuples
   4. Variables
   5. Nomenclature - PEP 8
   6. Keywords
   7. Operators
      1. Arithmetic
      2. Assignment
      3. Relational
      4. Boolean (logical)
      5. Membership
      6. Identity
      7. Callables
      8. Bitwise
3. Python Intermediate
   1. Conditionals - control flow - branching
   2. Loops
      1. Definite
      2. Indefinite
   3. Functions
      1. Passing arguments to functions
      2. Function return value
      3. Passing functions as arguments
         1. Positional
         2. Default keyname
         3. \*args, \*\*kwargs
      4. Built-in functions
      5. Functional Programming
         1. lambda
         2. map, filter, reduce
         3. generators
4. Python - Advanced
   1. Python Standard Library
   2. Modules/Packages
   3. Classes
      1. Attributes
      2. Methods
      3. Inheritance
      4. Polymorphism
   4. Exception handling
   5. File I/O
5. Numpy - scientific computing
   1. Multidimensional array objects
   2. Array indexing/slicing
   3. Vectorization - array math
   4. Routines for processing arrays
   5. Broadcasting
   6. Comparisons, masks, and boolean logic
   7. Basic statistics
   8. random
   9. Linear algebra
6. Pandas - data manipulation and analysis
   1. High performance data objects
   2. Data Indexing and Selection
   3. Data Wrangling
   4. Handling Missing Data
   5. Hierarchical Indexing
   6. Combining Datasets
      1. Concat and Append
      2. Merge and Join
   7. Aggregation and Grouping
   8. Pivot Tables
   9. Vectorized String Operations
   10. Working with Time Series
   11. High-Performance pandas: eval() and query()
   12. Visualisation
7. Matplotlib - data visualization
   1. Two Interfaces
   2. Line Plots
   3. Scatter Plots
   4. Visualizing Errors
   5. Density and Contour Plots
   6. Histograms, Binnings, and Density
   7. Customizing Plot Legends
   8. Customizing Colorbars
   9. Multiple Subplots
   10. Text and Annotation
   11. Customizing Ticks
   12. Customizing Matplotlib: Configurations and Stylesheets
   13. Three-Dimensional Plotting
   14. Geographic Data
   15. Images
8. Seaborn - high-level data visualization interface
   1. higher level visualization
   2. Improved styling over matplotlib
   3. Production ready graphics
   4. Works well with numpy and pandas
9. Scikit-Learn - collection of algorithms and tools for machine learning
   1. Types of Machine Learning
      1. Supervised
      2. Unsupervised
      3. Reinforcement
      4. Artificial Neural Networks
   2. Model Validation
   3. Feature Engineering
   4. Hyperparameters
   5. Models:
      1. Decision Trees and Random Forests
      2. Logistic Regression
      3. SVM - Support Vector Machines
      4. KNN - K-Nearest Neighbor
      5. Naive Bayes Classification
      6. Linear Regression
      7. PCA - Principal Component Analysis
      8. Manifold Learning
      9. k-Means Clustering
      10. Gaussian Mixture Models
      11. KDE - Kernel Density Estimation
10. Advanced topics

**======================================================================**

**Title**

Python for Data Science - 14-Week Program

Batch #3

**Description**

Businesses increasingly create, track, and rely on data. Your ability to manipulate data with your programming skills to solve business requirements is extremely valuable.

In this 14-week intensive (rigorous) program you will learn programming skills needed for your career.

Gain a strong knowledge and experience about Python by crafting beautiful, clean, elegant, efficient, maintainable, scalable, robust, and generalized Pythonic programs in multiple areas, such as: numerical processing, probability, statistics, linear algebra, data analysis, machine learning ecosystems, and data visualization. We will practice incremental development as we develop algorithms in our live coding sessions. Time permitting or special optional breakouts may even investigate personal special interests and topics.

Developer tools and techniques will be used throughout the course, such as: lint, unit testing, debugging, performance, profiling, execution visualization.

**Why Python?**

Python is a general purpose, versatile, and popular programming language. Python is great as a first language because it is concise and easy to read, and it is also a good language to have in any programmer’s stack as it can be used for everything from web development to software development, scientific applications, data science, machine learning, deep learning, and more.

**Who should take this course?**

Anyone interested in programming, but more importantly those wanting to learn how to properly program. We will start with basic Python syntax and move to advanced programming concepts with code snippets and applied programs with a data science focus. We will also learn and utilize some of the essential Python libraries used in data science and machine learning.

This course is for you is you:

* want to learn how to program in Python
* are tired of Python courses that are too complicated
* want to learn Python by doing
* like exciting challenges

You will have homework in this course so you have to be prepared to work on it

**Requirements**

* No prior coding or technical knowledge is required.
* A computer with Internet connection.
* No paid tools are required for this course.
* Passion and curiosity to learn.

**Instructor - Edward Bujak**

* 25 years IT experience - developer, consultant, founder
* 6 years of Data Science experience
* 19 years education experience: Computer Science, Mathematics, Engineering
* 2 masters degrees - MS Electrical Engineering & Science, MS Education
* Public speaker - frequently present at Meetup groups, clubs, conferences, on everything programming and computer science, Data Science, Machine Learning,

**Time Commitment**

* In-class (remote [live synchronous], recorded session posted afterwards)
  + 14 weeks
  + 2 sessions/week
  + 2 hours/session
* Out-of-class
  + 5 hours doing the challenge problems, algorithms, exercises using Python
* Optional
  + As-needed help sessions
  + Sessions to explore extracurricular interests and activities
  + All sessions open to entire cohort

**Dates**

June 26, 2020 - August (→ September) 26, 2020

**Cost**

$420 (30% COVID-19 Special Discount, Full Price $600)

**Registration**

<https://forms.gle/gXTdX9q5YR4p1UL16>

**Curriculum**

<https://www.bbds.ma/wp-content/uploads/2020/05/BBDS-14-Week-Python-Programming-Batch-3-Novice-to-Expert.pdf>

====================================================================

**Opening**

1. Become a Pythonista!
2. Coding is pervasive and ubiquitous
3. Why Python?
   1. Simple - easy to write, easy to understand
   2. Powerful
   3. General purpose - flexible - console programs, GUI applications, data
   4. Portable
   5. High-level - has automatic memory management
   6. Interpretive - no compilation
   7. Interactive - terminal for debugging
   8. Multi-paradigm
      1. Object-oriented
      2. Imperative
      3. Functional
   9. Readable - easy to maintain
   10. Huge number of libraries - “batteries included”
   11. Extensible - allows modules to the interpreter for customization
   12. Developed by Guido von Rossum - late 1980, launched 1991 - Netherlands
4. Where Python utilized?
   1. Web applications
   2. Games
   3. Desktop applications
   4. Database programming
   5. Systems programming
   6. AI
5. Who uses Python?
   1. Youtube
   2. BitTorrent
   3. Reddit
   4. Instagram
   5. Google
   6. Microsoft
   7. and many others
6. Why Python 3?
   1. print() is a function instead of a special case statement
   2. Only one integer type
   3. There are different operators / and // for true division and floor division instead of one, type dependent operator
   4. Default str is unicode - UTF-8 - capable of encoding all 1,112,064 valid characters using one to four 8-bit bytes. The first 127 characters of Unicode correspond to ASCII, are encoded as a single octet with same binary value as ASCII. Dominant character encoding for WWW since 2009.
   5. Python 3 continues to evolve. Python 2 will not be maintained pass 2020.
7. Python Environment → <https://xkcd.com/1987/>
8. import antigravity → [www.xkcd.com/353](http://www.xkcd.com/353)
9. Best Programming Language

<http://www.linuxjournal.com/content/best-programming-language>

1. Python Developers Survey 2017 Results - <https://www.jetbrains.com/research/python-developers-survey-2017/>
2. GitHub visualization of programming languages - <http://githut.info/>
3. Live coding → Not just syntax. We will learn by doing.
4. Python basics to some advanced, then move to applying skills and learning more as we attack various special topics/concepts. All learning how to properly code for elegance, speed, maintainability, intelligibility.
5. syntax → examples → exercises → projects
6. algorithmic/computational thinking
7. GitHub is the new resume.
8. ----
9. Coding challenge projects to extend session topics.

**What you will learn**

1. See Curriculum
2. Fundamentals - Essentials
   1. Data Structures - <https://www.pythoncentral.io/category/python-tutorials/>
      1. stacks, queues, linked lists,
   2. Algorithms - <https://www.pythoncentral.io/category/python-tutorials/>
      1. sort, search
   3. <http://python-reference.readthedocs.io/en/latest/basic_data_types.html>
3. Advanced
   1. Coding for Big Data
      1. More than your available memory
      2. Generator expressions
   2. New Features:
      1. see python3\_new.ipynb
   3. Writing idiomatic (Pythonic) code
   4. SDLC Design: top-down design … bottom-up coding/implementation
   5. OOP - Classes/Objects - data structures
   6. Regex - regular expressions
   7. Sentiment analysis - requests, bs4-Beautiful Soup
   8. Cryptography
      1. Steganography
   9. CGI Programming
   10. Database Access - MySQL
   11. GIS
   12. Networking
   13. Sending Email
   14. Multithreading
   15. SQL - also SQL with pandas (SQLite?)
   16. XML processing
   17. JSON processing
   18. Web scraping - requests module, response, beautiful soup
   19. GUI Programming

<https://trinket.io/>

* 1. Speech Recognition - CMUSphinx4
  2. Markov Chains
  3. Music
  4. AI - eBook
  5. Social Media - eBook
  6. Data Structures and Algorithms - eBook
  7. Games - eBook
  8. Simulation - eBook
  9. Advanced math - Mathematica, Fourier,
  10. Bioinformatics
  11. Financial
  12. Emoji analysis - search it - \_\_emoji.com
  13. Sentiment analysis
  14. AI - Artificial Intelligence
  15. Cyber security
  16. Malware detection
  17. Soundex algorithms
  18. Other algorithms
  19. Advanced math studies
  20. Signal Processing - DSP
  21. Image Processing and Computer Vision
  22. Optimization
  23. NLP
  24. Fractals (Mandelbrot)

**Advanced/complex concepts**

|  |  |
| --- | --- |
| **Module / Package** | **Concepts** |
| math | Additional advanced math functionality |
| random | random |
| datetime |  |
| os | operating system |
| sys | command line arguments  MORE |
| re | regular expressions |
| requests | Web scraping |
| SQL | Structured data |
| SymPy | Symbolic mathematics  CAS - Computer Algebra System |
| functools | functional programming |
| h5py and PyTables | access data stored in HDF5 format |
| dask | parallel programming and big data distributed data with dataframe |
| SciPy | comparative statistics |
| statsmodels | advanced statistics |
| bokeh | interactive visualization |
| altari |  |
| ggplot |  |