Table of Contents

[Python 2](#_Toc110208354)

[a) Design Aphorisms 2](#_Toc110208355)

[b) Coding Conventions 2](#_Toc110208356)

[c) Object Oriented Programming 2](#_Toc110208357)

[d) Report Writing and Documentation 2](#_Toc110208358)

[e) Exercises 2](#_Toc110208359)

[f) Projects 2](#_Toc110208360)

[g) Others 2](#_Toc110208361)

[Statistics 3](#_Toc110208362)

[Machine Learning 3](#_Toc110208363)

[a) Feature Engineering 3](#_Toc110208364)

[b) Algorithms 3](#_Toc110208365)

[c) Metrics 3](#_Toc110208366)

[d) Projects 3](#_Toc110208367)

[Deep Learning 3](#_Toc110208368)

[a) Neural Networks architecture drawing (parametrically) 3](#_Toc110208369)

[SQL 4](#_Toc110208370)

[a) Learn 4](#_Toc110208371)

[Interview Preparation 4](#_Toc110208372)

[Courses 4](#_Toc110208373)

[Other learning 4](#_Toc110208374)

**GitHub Repo: -** [rvs-23/DataScience\_Resources (github.com)](https://github.com/rvs-23/DataScience_Resources)

# Python

## Design Aphorisms

[PEP 20 -- The Zen of Python | Python.org](https://www.python.org/dev/peps/pep-0020/)

## Coding Conventions

[PEP 8 -- Style Guide for Python Code | Python.org](https://www.python.org/dev/peps/pep-0008/)

[Applying PEP 8 [LWN.net]](https://lwn.net/SubscriberLink/868490/cf00c295c514cad0/)

## Object Oriented Programming

[Introduction to object-oriented programming with Python - Learn | Microsoft Docs](https://docs.microsoft.com/en-us/learn/modules/python-object-oriented-programming/)

[Object-Oriented Programming (OOP) in Python 3 – Real Python](https://realpython.com/python3-object-oriented-programming/)

[What is Object Oriented Programming? OOP Explained in Depth (educative.io)](https://www.educative.io/blog/object-oriented-programming)

## Report Writing and Documentation

[Home - How to Write a Good Documentation - Library Guides at UC Berkeley](https://guides.lib.berkeley.edu/how-to-write-good-documentation)

[Process Documentation: Definition, Best Practices & How to Do It (helpjuice.com)](https://helpjuice.com/blog/process-documentation)

[Data Science Reports (udacity.com)](https://career-resource-center.udacity.com/portfolio/data-science-reports)

## Exercises

[The Big Book of Small Python Projects (inventwithpython.com)](https://inventwithpython.com/bigbookpython/)

## Projects

[About - Project Euler](https://projecteuler.net/)

[The Python Challenge](http://www.pythonchallenge.com/)

## Others

<https://pycon2016.regex.training/regex-intro>

[Python Tutor - Visualize Python, Java, C, C++, JavaScript, TypeScript, and Ruby code execution](https://pythontutor.com/)

# Statistics

# Git & GitHub

[Learn Git Branching](https://learngitbranching.js.org/)

# Machine Learning

[ML Wiki](http://mlwiki.org/index.php/Main_Page)

## Feature Engineering

[Feature Engineering and Selection: A Practical Approach for Predictive Models](http://www.feat.engineering/)

## Algorithms

[52.Trees.key (mlvu.github.io)](https://mlvu.github.io/lectures/52.Trees.annotated.pdf)

## Metrics

[Balanced Accuracy: When Should You Use It? - neptune.ai](https://neptune.ai/blog/balanced-accuracy)

## Projects

[CS 229: Machine Learning Final Projects, Autumn 2014 (stanford.edu)](http://cs229.stanford.edu/projects2014.html)

## Papers

[The latest in Machine Learning | Papers With Code](https://paperswithcode.com/)

## Practice

[Interview Query | Interview Questions](https://www.interviewquery.com/questions?searchQuery=&searchQuestionTag=&searchCompany=&ordering=Default&pageSize=20&page=0)

# Deep Learning

## Neural Networks architecture drawing (parametrically)

[alexlenail/NN-SVG: Publication-ready NN-architecture schematics. (github.com)](https://github.com/alexlenail/NN-SVG)

# SQL

## Learn

<https://sqlbolt.com/lesson/>

[Intro to SQL: Querying and managing data | Khan Academy](https://www.khanacademy.org/computing/computer-programming/sql)

# Interview Preparation

[Home - The Data Monk](https://thedatamonk.com/)

[Interview Query | Interview Questions](https://www.interviewquery.com/questions?searchQuery=&searchQuestionTag=&searchCompany=&ordering=Default&pageSize=20&page=0)

# Courses

[Introduction to Computer Science with Python](https://www.youtube.com/watch?v=nykOeWgQcHM&list=PLUl4u3cNGP63WbdFxL8giv4yhgdMGaZNA)

[Introduction to Computational Thinking in Data Science](https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0002-introduction-to-computational-thinking-and-data-science-fall-2016/lecture-videos/)

# Other learning

[Microsoft Learn | Microsoft Docs](https://docs.microsoft.com/en-us/learn/)

[Data School](https://www.dataschool.io/)

[30 days of JavaScript](https://github.com/Asabeneh/30-Days-Of-JavaScript)

[Repository of Machine Learning courses](https://github.com/dair-ai/ML-YouTube-Courses)