* Python
  + Mocking [Link1](https://www.toptal.com/python/an-introduction-to-mocking-in-python), [Link2](https://realpython.com/python-mock-library/)
    - Mock an item where it is used, not where it came from.
  + Decorators - [Link](https://realpython.com/primer-on-python-decorators/)
  + Use of functool.wraps - [Link1](https://lerner.co.il/2019/05/05/making-your-python-decorators-even-better-with-functool-wraps/), [Link2](https://stackoverflow.com/questions/308999/what-does-functools-wraps-do)
  + Property decorator, Setter/Getters [[Blog1](https://www.programiz.com/python-programming/property)] [[Blog2](https://stackoverflow.com/questions/2627002/whats-the-pythonic-way-to-use-getters-and-setters)]
  + Instance, Class and Static Methods - [Link](https://realpython.com/instance-class-and-static-methods-demystified/)
  + [Deep Copy vs Shallow Copy](https://docs.google.com/document/d/17K8SChkw9oVC2LG9OXe2SrfBFXndTajOsu9HbpLRFUg/edit#bookmark=id.1upjzm92yel3)
  + <https://www.youtube.com/watch?v=naG4uXpmVAU>
  + [Iterator protocol and iterators](https://docs.google.com/document/d/17K8SChkw9oVC2LG9OXe2SrfBFXndTajOsu9HbpLRFUg/edit#bookmark=id.iq40c8yl9muc) [[Blog1](https://www.datacamp.com/community/tutorials/python-iterator-tutorial)]
  + [Difference between Iterable and Iterators](https://docs.google.com/document/d/17K8SChkw9oVC2LG9OXe2SrfBFXndTajOsu9HbpLRFUg/edit#bookmark=id.gn70be68k4ej)
  + [Meaning of underscores in Python](https://dbader.org/blog/meaning-of-underscores-in-python)
  + <https://www.youtube.com/watch?v=ALZmCy2u0jQ>
  + Multithreading
  + Multiprocessing
  + Advance Collections
  + Unit testing
  + Logging
  + GIL
  + [Scope and Use of global & nonlocal](https://docs.google.com/document/d/17K8SChkw9oVC2LG9OXe2SrfBFXndTajOsu9HbpLRFUg/edit#bookmark=id.s3jxvd359ecr)
  + [Global keyword](https://www.programiz.com/python-programming/global-keyword)
  + Mutable & Immutable
  + MRO
  + Super
  + Slots
  + \_\_new\_\_
  + \_\_call\_\_
  + Hasattr
  + getattr
  + Duck typing - [Link](https://www.toptal.com/python/python-design-patterns)
  + Metaclasses
  + Encapsulation
  + Polymorphism
  + Abstraction
  + Inheritance
  + Abstract Classes & Interfaces - <https://stackoverflow.com/a/1913185/1356851>
  + Context managers
* Django / Web Development
  + [Signals](https://www.youtube.com/watch?v=T6PyDm79PFo)
  + [Django Signals vs .save() override](https://books.agiliq.com/projects/django-orm-cookbook/en/latest/signals.html#signals-vs-overriding-save)
  + [WSGI](https://www.agiliq.com/blog/2013/07/basics-wsgi/)
  + DRF
  + [ORM](https://docs.google.com/document/d/17K8SChkw9oVC2LG9OXe2SrfBFXndTajOsu9HbpLRFUg/edit#bookmark=id.t5vr0hb39qwt)
  + Joins in ORM
  + CORS [[Blog1](https://medium.com/@baphemot/understanding-cors-18ad6b478e2b)][[Blog2](https://itnext.io/cors-understanding-it-practically-9c401ed818cd)]
  + CSRF [[CSRF](https://medium.com/davidchia/working-with-csrf-protection-in-django-555b05904ee4)]
  + [Type of Authentication](https://github.com/alex996/presentations/blob/master/auth.md) [[Video](https://youtu.be/2PPSXonhIck)] [[Blog1](https://medium.com/tech-tajawal/microservice-authentication-and-authorization-solutions-e0e5e74b248a)] [[Blog2](https://medium.com/@vivekmadurai/different-ways-to-authenticate-a-web-application-e8f3875c254a)]
    - Basic Auth
    - Session Auth (with Cookies)
    - Token-based Auth (JWT Auth)
    - Single Sign-On (For multiple Microservices) [[Video](https://www.youtube.com/watch?v=kRGPKh3NgHU)]
    - OAuth Authentication [[Blog](https://hackernoon.com/demystifying-oauth-2-0-and-openid-connect-and-saml-12aa4cf9fdba)]
    - Kerberos Authentication
  + Issues with Distributed Session Management
  + Django Management Command [[Blog1](https://docs.djangoproject.com/en/2.2/howto/custom-management-commands/)] [[Blog2](https://simpleisbetterthancomplex.com/tutorial/2018/08/27/how-to-create-custom-django-management-commands.html)]
  + Middleware/Custom Middleware [[Blog1](https://www.agiliq.com/blog/2015/07/understanding-django-middlewares/)] [[Blog2](https://www.agiliq.com/blog/2015/07/profiling-django-middlewares/)]
  + Class-based views [[Blog1](https://simpleisbetterthancomplex.com/article/2017/03/21/class-based-views-vs-function-based-views.html)] [[Blog2](https://medium.com/@ksarthak4ever/django-class-based-views-vs-function-based-view-e74b47b2e41b)]
  + Custom User Model
  + Migrations & Fake migration [[Blog1](https://realpython.com/django-migrations-a-primer/)] [[Blog2](https://realpython.com/digging-deeper-into-migrations/)] [[Blog3](https://realpython.com/data-migrations/)]
    - [Undo a Migration](https://docs.google.com/document/d/17K8SChkw9oVC2LG9OXe2SrfBFXndTajOsu9HbpLRFUg/edit#bookmark=id.ht4qgmdf0ywa)
    - Fake Migration [[Blog1](https://stackoverflow.com/questions/46772762/django-migrate-fake-and-fake-initial-explained)]
  + Serialization

* Databases
  + Database Queries
  + Database Query Optimizations(profiling)
  + Indexing in deep
  + Data Modeling
  + Normalizations
* Some of the Django ORM related topic links may not work. For that refer this website - https://books.agiliq.com/projects/django-orm-cookbook/en/latest/