# # Python

## Courses, Collection, Guides, Books

* + <https://github.com/huangsam/ultimate-python>
  + Quantitative economics in python  
    <https://python.quantecon.org/index_toc.html>
  + Etudes for python programming  
    <https://github.com/norvig/pytudes>
  + Best of python libraries  
    <https://github.com/ml-tooling/best-of-web-python>
  + WTF python (Python mysteries)  
    <https://github.com/satwikkansal/wtfpython>
  + Python programming puzzles  
    <https://github.com/microsoft/PythonProgrammingPuzzles>
  + Interactive coding challenge  
    <https://github.com/donnemartin/interactive-coding-challenges>
  + Awesome Python  
    <https://github.com/vinta/awesome-python>

## Rare Syntax and Curious Built-in

* + Trends
    - REST API with Flask/Quart, Async, Dataclass(pydantic)  
      <https://pgjones.dev/blog/rest-api-evolution-2020/>
  + \*a, c = [1,2,3]
  + Float comparison  
    math.isclose(a, b)
  + Useful tips available in python3  
    <https://medium.com/better-programming/9-python-3-features-you-might-not-use-yet-fade13932a3e>
  + Great in python3.10  
    <https://treyhunner.com/2021/10/whats-great-about-python-3-dot-10/>
  + Fraction ( can be useful to keep infinite precision)  
    <https://realpython.com/python-fractions/>
  + Itertools product to iterate over multi dimensional array
  + Force using keyword  
    def func(\*, first, second)
  + Abstract class and method from **abc** module
  + bisect (array bisection algorithm)  
    This is useful to do a numeric table lookup
  + dis (package for disassembling)
  + timeit (measure time for running function)
  + difflib   
    <https://florian-dahlitz.de/blog/create-your-own-diff-tool-using-python>
  + dataclasses (introduced in v3.7, but available in 3.6 as well)
  + enumerate  
    Useful for getting both index and value of iterable.  
    *for count, value in enumerate(values):*
  + functools  
    <https://florian-dahlitz.de/blog/introduction-to-functools>
  + partial (functools.partial)  
    Useful for overriding common method  
    <https://adamj.eu/tech/2021/05/05/3-uses-for-functools-partial-in-django/>
  + for ... else …  
    <https://medium.com/techtofreedom/for-else-a-weird-but-useful-feature-in-python-2076d8dafad1>
  + print  
    <https://adamj.eu/tech/2021/10/08/tips-for-debugging-with-print/>

### Pip

* + pip show
  + pip freeze
  + pip install –upgrade package
  + pip install setuptools –upgrade
  + python manage.py check --deploy

## Interesting Projects

* + Dash (Python framework for building analytical web applications using Flask and Plotly)  
    <https://github.com/plotly/dash>
  + Archivy (self-hosted knowledge repository to save web pages, etc)  
    <https://github.com/archivy/archivy>
  + LibreLingo Language learning (Django, Svelte, CouchDB)  
    <https://github.com/kantord/LibreLingo>
  + Finding usage metrics for python packages  
    <https://pepy.tech/>  
    <https://github.com/psincraian/pepy> (Flask and google big query)
  + Contextualise - Organize information projects and activities  
    <https://github.com/brettkromkamp/contextualise>
  + Driving agent (Openpilot)  
    <https://github.com/commaai/openpilot>
  + Self driving library  
    <https://github.com/autorope/donkeycar>
  + Python projects  
    <https://medium.mybridge.co/36-amazing-python-open-source-projects-v-2019-2fe058d79450>  
    <https://medium.mybridge.co/amazing-machine-learning-open-source-tools-projects-of-the-year-v-2019-95d772e4e985>  
    <https://www.theinsaneapp.com/2021/06/list-of-python-projects-with-source-code-and-tutorials.html>
  + 10 Django projects started on 2016  
    <https://medium.com/@raiderrobert/top-10-new-django-projects-started-in-2016-f87ef043c8bb>
  + Fun project for student learning (Multiplication and division)  
    <https://github.com/JonathanWillitts/mquest>
  + Line-Art Image  
    <https://github.com/lucashadfield/speck>
  + Free python proxy server with AWS Lambda  
    <https://github.com/ian-whitestone/python-proxy-server>  
    <https://ianwhitestone.work/free-python-proxy-server/>
  + PyCon US  
    <https://us.pycon.org/2020/schedule/tutorials/>
  + Python on iOS  
    <https://github.com/joaoventura/pybridge-ios>
  + Django Enhancement proposals  
    <https://github.com/django/deps>
  + Trigger Happy (Bus for your internet service, Django)  
    <https://github.com/push-things/django-th>
  + Find invalid anchors from web page  
    <https://github.com/EndlessTrax/pyanchor>
  + thenewboston digital currency  
    <https://github.com/thenewboston-developers/Bank>
  + Photonix (Photo management application)  
    <https://github.com/photonixapp/photonix>
  + Auto purchase amazon script using selenium  
    <https://github.com/pcomputo/Whole-Foods-Delivery-Slot>
  + Collection of mini projects  
    <https://github.com/Python-World/python-mini-projects>
  + Napari (Interactive, multi-dimensional image viewer using QT)  
    <https://github.com/napari/napari>
  + Slack Machine (extensible slack bot)  
    <https://github.com/DonDebonair/slack-machine>

### Boilerplates

* + FastAPI fullstack boilerplate  
    <https://github.com/tiangolo/full-stack-fastapi-postgresql>
  + Django Boilerplate  
    <https://github.com/pydanny/cookiecutter-django>   
    <https://github.com/nickjj/docker-django-example>

### Games

* + PlantVSZombies  
    <https://github.com/marblexu/PythonPlantsVsZombies>

## Interesting blog post

* + f-string tips  
    https://towardsdatascience.com/python-f-strings-are-more-powerful-than-you-might-think-8271d3efbd7d
  + Top python libraries in 2021  
    <https://tryolabs.com/blog/2021/12/21/top-python-libraries-2021>
  + Django vs Flask  
    <https://testdriven.io/blog/django-vs-flask/>
  + Django, GraphQL, Vue  
    <https://realpython.com/python-django-blog/>
  + Data processing and visualization
    - Dash and Vaex  
      <https://medium.com/plotly/interactive-and-scalable-dashboards-with-vaex-and-dash-9b104b2dc9f0>
  + Structure
    - Setting up python project for automation and collaboration  
      <https://eugeneyan.com/writing/setting-up-python-project-for-automation-and-collaboration/>
  + Develop windows service in python  
    <http://thepythoncorner.com/dev/how-to-create-a-windows-service-in-python/>
  + API rate limit implementation with Redis  
    <https://dev.to/astagi/rate-limiting-using-python-and-redis-58gk>
  + Python3.8, assignment expression (a := b)  
    <https://effectivepython.com/2020/02/02/prevent-repetition-with-assignment-expressions>
  + GRPC tutorial (Pros and Cons)  
    <https://martinheinz.dev/blog/23>
  + Pros and Cons of pickle  
    <https://realpython.com/python-pickle-module/>  
    <https://github.com/trailofbits/fickling> (decompile pickle object)
  + Config values in python  
    <https://beepb00p.xyz/configs-suck.html>
  + Celery Distributed task  
    <https://www.ovh.com/blog/doing-big-automation-with-celery/>
  + Common steps to do for all python projects  
    <https://glyph.twistedmatrix.com/2020/02/modules-for-maintenance.html>
  + Reproducible docker image for python apps  
    <https://pythonspeed.com/articles/reproducible-docker-builds-python/>
  + Migrate version 2 to 3  
    <https://gregoryszorc.com/blog/2020/01/13/mercurial's-journey-to-and-reflections-on-python-3/>  
    <https://stxnext.com/blog/2019/04/16/python-3-migration-guide/>
  + Python profiling  
    <https://www.blog.pythonlibrary.org/2020/04/14/an-overview-of-profiling-tools-for-python/>
  + Profiling vs Logging  
    <https://pythonspeed.com/articles/logging-vs-profiling/>
  + Track traffic (Man in the middle)  
    <https://elimernit.com/blog/auto-swipe/>
  + PDF processing in python  
    <https://realpython.com/pdf-python/>
  + Python itertools explanation  
    <https://towardsdatascience.com/tour-of-python-itertools-2af84db18a5e>
  + Generative art in python  
    <https://www.generativehut.com/post/robots-and-generative-art-and-python-oh-my>
  + Python code analysis tools  
    <https://luminousmen.com/post/python-static-analysis-tools>
  + Python interface  
    <https://realpython.com/python-interface/>
  + Pypy or python  
    <https://ao.gl/when-your-python-code-is-much-faster-with-pypy/>
  + Restful Error messages  
    <https://medium.com/@mw123/restful-error-messages-with-django-537047892dff>
  + Python useful functions  
    <https://twitter.com/raymondh/status/1211414561468952577>
  + Avoid Django genericforeignkey (Worth to read)  
    <https://lukeplant.me.uk/blog/posts/avoid-django-genericforeignkey/>
  + Caching  
    <https://engineering.redislabs.com/posts/redis-assisted-client-side-caching-in-python/>
  + Channel (Websocket)  
    <https://www.aeracode.org/2016/6/16/philosophy-channels/>
  + Python formatted string  
    <https://realpython.com/python-formatted-output/>
  + Python command line argument  
    <https://realpython.com/python-command-line-arguments/>
  + Python multi await  
    <http://www.hydrogen18.com/blog/python-await-multiple.html>
  + Pythonic programming  
    <https://access.redhat.com/blogs/766093/posts/2802001>
  + Python to Go conversion  
    <https://gitlab.com/esr/reposurgeon/blob/master/GoNotes.adoc>
  + Kafka, Docker, Python  
    <https://medium.com/big-data-engineering/hello-kafka-world-the-complete-guide-to-kafka-with-docker-and-python-f788e2588cfc>
  + Python equal check “is” and “==”  
    <https://realpython.com/python-is-identity-vs-equality/>
  + Alpine docker image for python is not good  
    <https://pythonspeed.com/articles/alpine-docker-python/>
  + Markdown to email  
    <https://pbpython.com/markdown-email.html>
  + Dependency Injection  
    <https://sobolevn.me/2020/02/typed-functional-dependency-injection>
  + Overloading in python  
    <https://arpitbhayani.me/blogs/function-overloading>
  + Zero downtime deployment (for solo development)  
    <https://pydist.com/blog/zero-downtime-deploys>
  + Atomic lock-free counters (threading)  
    <https://julien.danjou.info/atomic-lock-free-counters-in-python/>
  + NSA python tutorial  
    <https://www.zdnet.com/article/python-programming-language-now-you-can-take-nsas-free-course-for-beginners/>
  + Collection(Dict, List, Set) handling  
    <https://sourcery.ai/blog/effective-collection-handling/>
  + Inheritance and Composition  
    <https://realpython.com/inheritance-composition-python/>
  + Test, Sanic, Websocket  
    <https://tryexceptpass.org/article/manage-test-resources-with-websockets/>
  + Arcade bot  
    <https://mryslab.github.io/bots-in-pieces/python-banyan/arcade/2020/02/21/p2p-arcade-1.html>
  + Requests different use cases  
    <https://findwork.dev/blog/advanced-usage-python-requests-timeouts-retries-hooks/>
  + MacOS application in Python  
    <https://camillovisini.com/create-macos-menu-bar-app-pomodoro/>
  + WebRTC, python   
    <http://pfertyk.me/2020/03/webrtc-a-working-example/>
  + Async API call with treq  
    <https://opensource.com/article/20/3/treq-python>
  + Python game development  
    <https://realpython.com/podcasts/rpp/2/>
  + Python search algorithms  
    <https://realpython.com/binary-search-python/>
  + Scraping github page  
    <https://florian-dahlitz.de/blog/scrape-github-orgs-using-python>
  + Writing python in Rust  
    <https://blog.m-ou.se/writing-python-inside-rust-1/>
  + Context manager tips  
    <https://rednafi.github.io/digressions/python/2020/03/26/python-contextmanager.html>
  + Tracing GIL  
    <https://www.maartenbreddels.com/perf/jupyter/python/tracing/gil/2021/01/14/Tracing-the-Python-GIL.html>
  + Regular Expression  
    <https://realpython.com/regex-python/>
  + uWSGI  
    <https://www.techatbloomberg.com/blog/configuring-uwsgi-production-deployment/>   
    <https://blog.ionelmc.ro/2022/03/14/how-to-run-uwsgi/>

### FastAPI

* + Why is FastAPI good?  
    <https://dev.to/tiangolo/the-future-of-fastapi-and-pydantic-is-bright-3pbm>
  + FastAPI with CRUD  
    <https://testdriven.io/blog/fastapi-crud/>

### Map

* + Leafmap on Django admin  
    <https://www.paulox.net/2020/12/08/maps-with-django-part-1-geodjango-spatialite-and-leaflet/>

### WebSocket

* + Message server with websocket  
    <https://mortoray.com/2020/12/06/high-throughput-game-message-server-with-python-websockets/>

## Pyramid

* + Simple Tutorial  
    <https://opensource.com/article/20/1/python-web-api-pyramid-cornice>

## Flask

* + Blueprint  
    <https://realpython.com/flask-blueprint/>

## Django

* + Snippets  
    <https://djangosnippets.org/popular/rated/>
  + Tips
    - Use datetimefield instead of datefield to save timezone info
    - Get language code with “translation.get\_language”
    - Securing Django apps  
      <https://www.mattlayman.com/understand-django/secure-apps/>
    - Advanced tips for ORM  
      <https://www.laac.dev/blog/five-advanced-django-tips/>
    - <https://hakibenita.com/django-group-by-sql> (Group by in ORM)
    - Pagination improvement  
      <https://pganalyze.com/blog/pagination-django-postgres>
    - Pydantic for Django settings  
      <https://www.toptal.com/python/streamline-your-django-settings-with-type-hints-pydantic-tutorial>
  + Admin page tips
    - Get history on admin page   
      <https://djangoweekly.com/blog/post/viewbrowse-all-django-admin-edits-recent-actions-listing>
    - Django admin page tips  
      <https://medium.com/@hakibenita/things-you-must-know-about-django-admin-as-your-app-gets-bigger-6be0b0ee9614>
    - Add user profile in Django admin  
      <https://simpleisbetterthancomplex.com/tutorial/2016/11/23/how-to-add-user-profile-to-django-admin.html>
    - Custom button on django admin page  
      <https://medium.com/@hakibenita/how-to-add-custom-action-buttons-to-django-admin-8d266f5b0d41>
    - CSV export, streaming, Django admin, StreamingHTTPResponse  
      <https://djangosnippets.org/snippets/10595/>
  + Performance  
    <https://blog.mirumee.com/django-fast-part-1-8d068a1b14bc> (Monitor and Comparison)  
    <https://blog.mirumee.com/django-fast-part-2-d73a4ecd61f3>  
    <https://realpython.com/caching-in-django-with-redis/>  
    <https://simpleisbetterthancomplex.com/tips/2016/10/05/django-tip-16-simple-database-access-optimizations.html>  
    <https://medium.com/@hansonkd/performance-problems-in-the-django-orm-1f62b3d04785> (ORM, Memory, Iterator)
  + Beautiful admin dashboard  
    <https://github.com/byashimov/django-controlcenter>   
    <https://github.com/geex-arts/django-jet>   
    <https://github.com/originalankur/awesome-django-admin>
  + Libraries
    - Template render  
      <https://github.com/adamghill/django-unicorn> (reactive component framework)
    - API  
      <https://github.com/jsocol/django-ratelimit>  
      <https://github.com/adamchainz/django-cors-headers>  
      <https://github.com/vitalik/django-ninja> (Fast, Async, OpenAPI, type hints)  
      <https://github.com/robinhood/deux> (DRF MFA)  
      <https://github.com/graphql-python/graphene-django> (Graphene for Django)
    - Auth  
      <https://testdriven.io/blog/django-spa-auth/> (Django session auth with SPA)  
      <https://github.com/django-guardian/django-guardian> (Permission per object)  
      <https://github.com/arteria/django-hijack> (Impersonating user)  
      <https://github.com/python-social-auth/social-app-django> (Social Auth)  
      <https://simpleisbetterthancomplex.com/tutorial/2016/10/24/how-to-add-social-login-to-django.html>
    - Background tasks  
      <https://github.com/arteria/django-background-tasks>
    - Data Anonymization  
      <https://github.com/Tesorio/django-anon>
    - CMS  
      <https://github.com/wagtail/wagtail>  
      <https://github.com/divio/django-cms>  
      <https://github.com/stephenmcd/mezzanine>
    - Ecommerce  
      Saleor  
      Django-Oscar
    - Form  
      <https://github.com/linevych/django-search-admin-autocomplete> (search autocomplete, Django admin)  
      <https://github.com/barseghyanartur/django-fobi>  
      <https://github.com/django-crispy-forms/django-crispy-forms>  
      <https://github.com/caktus/django-jsx> (Django, JSX in template)  
      <https://github.com/django-money/django-money> (Money Field)  
      <https://pypi.org/project/django-dynamic-admin-forms/> (Django Admin dynamic forms)
    - Model  
      <https://github.com/django-polymorphic/django-polymorphic> (Django model inheritance)  
      <https://github.com/rsinger86/django-lifecycle/> (life cycle for model record change)  
      <https://github.com/etianen/django-reversion> (version control for model instances)  
      <https://github.com/jedie/django-reversion-compare>
    - <https://github.com/occipital/django-consistency-model> (Model field consistency check)
    - Logging  
      <https://github.com/JonasKs/django-guid> (Log with GUID)
    - MPTT (a technique for storing hierarchical data in a database)  
      <https://github.com/django-mptt/django-mptt/>
    - Storage  
      <https://github.com/jschneier/django-storages> (Different storages for image and file)
    - Communication  
      <https://github.com/dmonn/django-directmessages>
    - Data science  
      <https://github.com/fridiculous/django-estimators> (retrieve and persist ML model)
    - Dashboard  
      <https://github.com/simonw/django-sql-dashboard> (Read only sql dashboard on Django)
    - Other  
      <https://github.com/Opus10/django-pgtrigger> (Postgres trigger for Django models)  
      <https://github.com/meshy/django-schema-graph/> (Model Schema Graph)  
      <https://github.com/koslibpro/django-traffic> (Middleware to see traffic on Kibana)  
      <https://github.com/llazzaro/django-scheduler> (Django calendar)  
      <https://gitlab.com/rosarior/awesome-django/-/blob/d25bd21d6663dce0facc872896a58a39c4dd006b/README.md> (Awesome Django)
  + Search
    - Haystack, elasticsearch
    - Django postgres full text search in ORM   
      <https://pganalyze.com/blog/full-text-search-django-postgres>   
      <https://testdriven.io/blog/django-search/>
    - SQL Explorer  
      <https://github.com/groveco/django-sql-explorer>
  + Deployment
    - Deploy on AWS with gitlab CI  
      <https://testdriven.io/blog/deploying-django-to-ec2-with-docker-and-gitlab/>
    - <https://cloud.google.com/python/django/kubernetes-engine>  
      <https://dzone.com/articles/battle-of-the-paas-python-apps-in-the-cloud>  
      <https://medium.com/eshares-blog/supercharging-django-productivity-at-eshares-8dbf9042825e>  
      <https://medium.com/evenbit/deploying-django-to-google-app-engine-with-gitlab-ci-541c0dc67a12>  
      <https://www.stavros.io/posts/how-deploy-django-docker/>
  + Blogs
    - Caching  
      <https://testdriven.io/blog/django-caching/>
    - Deployment  
      <https://testdriven.io/blog/django-elastic-beanstalk/> (Elastic beanstalk)
    - Django channel  
      <https://www.freecodecamp.org/news/best-gitter-channels-python-django-41a0a0b1aee6/>  
      <http://masnun.com/2016/09/11/a-brief-introduction-to-django-channels.html>

<https://adamj.eu/tech/2020/02/10/robots-txt/> (Robot.txt on Django)  
<https://djangostars.com/blog/python-best-programming-language-for-startup/> (Why Django is good for startup)  
<https://simpleisbetterthancomplex.com/tutorial/2016/07/22/how-to-extend-django-user-model.html> (Extend user model)  
<https://czep.net/17/full-text-search.html> (Full text search, PSQL, Django)  
[https://blog.bitlabstudio.com/a-guide-for-using-braintree-in-django-53e99d677f71](https://blog.bitlabstudio.com/a-guide-for-using-braintree-in-django-53e99d677f71%20(braintree) (Braintree, paypal, payment)  
<https://medium.com/@kitsunde/checking-that-django-has-all-migrations-55a1c23c3a59> (Check missing migrations)  
<https://simpleisbetterthancomplex.com/tutorial/2016/10/31/how-to-handle-github-webhooks-using-django.html> (Github hooks on Django)  
<https://simpleisbetterthancomplex.com/tips/2016/11/01/django-tip-19-protecting-sensitive-information.html> (Protect sensitive info)  
<https://geekchick77.dreamwidth.org/5560.html> (Django migration squashing)  
<https://simpleisbetterthancomplex.com/tutorial/2016/10/04/how-to-use-django-flatpages-app.html> (Flat page)  
<https://medium.com/@adandan01/django-inline-formsets-example-mybook-420cc4b6225d> (Inline Form)  
<https://simpleisbetterthancomplex.com/tutorial/2016/10/13/how-to-use-generic-relations.html> (Generic Relation, Content Type)  
<https://openfolder.sh/django-faster-speed-tutorial> (Improvement tips)  
<https://www.b-list.org/weblog/2020/mar/23/still-no-service/> (service layers)

## For performance improvement

### CPython alternative

* + Cinder
  + Pyston
  + PyPy
  + Numba. Ex, machine learning, image processing  
    Numba Para, Numba Pool

### Other

* + Accelerating python with GPU, Numba, etc  
    <https://devblogs.nvidia.com/accelerating-python-for-exotic-option-pricing/>
  + Multiprocessiong, Multithreading, Asyncio (Dask can be used for pandas)  
    <https://realpython.com/python-concurrency/>   
    <https://medium.com/better-programming/building-an-imagedownloader-with-multiprocessing-in-python-44aee36e0424>  
    <https://pythonprogramming.net/threading-tutorial-python/>   
    <https://www.toptal.com/python/beginners-guide-to-concurrency-and-parallelism-in-python>  
    (Tips) : <https://www.cloudcity.io/blog/2019/02/27/things-i-wish-they-told-me-about-multiprocessing-in-python/>
  + Performance improvement with Math functions  
    <https://realpython.com/python-math-module/>
  + Pandas optimization
    - Memory optimization  
      <https://drawingfromdata.com/pandas/concat/memory/exploding-memory-usage-with-concat-and-categories.html>
    - Iteration optimize  
      <https://www.wrighters.io/how-to-iterate-over-dataframe-rows-and-should-you/>
  + Gandiva  
    <http://blog.christianperone.com/2020/01/gandiva-using-llvm-and-arrow-to-jit-and-evaluate-pandas-expressions/>
  + Alternative Dict/JSON handling
    - SimpleJSON
    - UltraJSON
    - Microdict  
      <https://github.com/touqir14/Microdict>
  + Parallel sometimes slower  
    <https://pythonspeed.com/articles/parallelism-slower/>
  + Memory performance  
    <https://pythonspeed.com/memory>   
    <https://pythonspeed.com/articles/measuring-memory-python/>
  + Fast and memory optimized string operation  
    <https://github.com/btwael/superstring.py>
  + Performance comparison in list operation  
    <https://stackoverflow.com/questions/60549865/what-causes-a-to-overallocate>
  + Make python programs fast  
    <https://martinheinz.dev/blog/13>
  + Faster functions
    - itemgetter is faster than item[0]. So use it for sorting  
      <https://realpython.com/sort-python-dictionary/>
  + Ducks (Faster lookup of objects)  
    <https://github.com/manimino/ducks>

## Libraries

* + ORM/Database
    - SQLAlchemy/Alembic  
      <https://github.com/sqlalchemy/alembic>  
      <https://github.com/olirice/alembic_utils>
    - Sqlite util  
      <https://github.com/simonw/sqlite-utils>
    - SQLModel  
      <https://github.com/tiangolo/sqlmodel>
    - Pymongo
    - Mongita (MongoDB as SQLite)  
      <https://github.com/scottrogowski/mongita>
    - Motor (Async mongo operations)  
      <https://github.com/mongodb/motor>
    - Ormar (Async ORM with FastAPI in mind with pydantic validation)  
      <https://github.com/collerek/ormar>
    - Peewee
    - Tortoise ORM  
      <https://github.com/tortoise/tortoise-orm/> (Asyncio ORM)
  + CLI
    - Ghtop (github real time using “rich” python lib)  
      <https://github.com/nat/ghtop>
    - Stream-unzip (Unzip without loading all file in memory, streamed unzip)  
      <https://github.com/uktrade/stream-unzip>
    - Beautiful terminal rich text  
      <https://github.com/Textualize/rich>   
      <https://github.com/Textualize/rich-cli>
    - HTTPie (Modern, user-friendly CLI HTTP client similar to curl)  
      <https://github.com/httpie/httpie>
  + Task runner
    - Celery
    - Wakaq (Super minimal celery)  
      <https://github.com/wakatime/wakaq>
    - dramatiq  
      <https://github.com/Bogdanp/dramatiq>
    - huey  
      <https://github.com/coleifer/huey>
    - Schedule (Python job scheduling for humans)  
      <https://github.com/dbader/schedule>
    - Chard (Django async/await background task queue)  
      <https://github.com/drpancake/chard>
    - Rocketry (Scheduling library)  
      <https://github.com/Miksus/rocketry>
  + Dataframe and custom data types
    - Pandas  
      <https://github.com/pandas-dev/pandas>   
      <https://github.com/modin-project/modin> (Speed up pandas to read big files, can be used with Ray)  
      <https://github.com/adamerose/pandasgui> (Pandas GUI)  
      <https://github.com/pydata/pandas-datareader> (Access remote data into dataframe, good for financial data)
    - ConnectorX (Fastest library to load data from db to dataframes)  
      <https://github.com/sfu-db/connector-x>
    - Vaex (high performance Python library for lazy Out-of-Core DataFrames)  
      HDF5 and Apache arrow supported, instant opening of Huge data files  
      <https://github.com/vaexio/vaex>
    - Riptable (64bit multithreaded python data analytics tools)  
      <https://github.com/rtosholdings/riptable>
    - Tablib (Module for tabular datasets in xls, csv, json, yaml, etc)  
      <https://github.com/jazzband/tablib>
    - DocArray (array of different document type values, for unstructured data)  
      <https://github.com/jina-ai/docarray>
  + Most used 22 libraries  
    <https://medium.com/better-programming/the-22-most-used-python-packages-in-the-world-7020a904b2e>
  + Easy json parsing  
    <https://github.com/jmespath/jmespath.py>
  + Posthog (Product Analytics)  
    <https://github.com/posthog/posthog>
  + Markdown  
    <https://hakibenita.com/django-markdown>
  + Questionary (multiple types of question prompt)  
    <https://github.com/tmbo/questionary>
  + Dictionary to data classes  
    <https://github.com/konradhalas/dacite>
  + Dependency managers
    - pip-tools: pip-compile requirements.in > requirements.txt
    - poetry
    - pipenv
    - PDM (No virtualenv is required)  
      <https://pdm.fming.dev/>
  + String parse with python formatting  
    <https://github.com/r1chardj0n3s/parse>
  + Convtool (Python object to python object, aggregation, etc)  
    <https://github.com/itechart-almakov/convtools>
  + Shiv - Python package zipping (No virtualenv is needed)  
    <https://github.com/linkedin/shiv>  
    <https://jhermann.github.io/blog/python/deployment/2020/03/08/ship_libs_with_shiv.html>
  + Python3 helpers  
    <https://yakutils.readthedocs.io/en/latest/>
  + Task management (Build-tool)  
    <https://github.com/pydoit/doit>
  + Celery director  
    <https://github.com/ovh/celery-director>
  + Data validation  
    <https://github.com/pyeve/cerberus>
  + Event driven networking engine  
    <https://github.com/twisted/twisted>
  + Http Request
    - Many Request (multiple requests at a time)  
      <https://github.com/joshlk/many_requests>
    - HTTP request retry decorator  
      <https://github.com/channable/opnieuw>
  + Nested containment list  
    <https://github.com/biocore-ntnu/ncls> (High performance)
  + Python stream processing  
    <https://github.com/omegaml/minibatch>
  + Python gmail API helper  
    <https://github.com/abhishekkr/gmail-helper>
  + Data parsing and validation with type hints  
    <https://github.com/samuelcolvin/pydantic>
  + Colorpedia  
    <https://github.com/joowani/colorpedia>
  + Get image format  
    imghdr  
    pillow
  + Google sheet library  
    <https://github.com/socialpoint-labs/sheetfu>
  + Timer  
    <https://github.com/realpython/codetiming>
  + Python for mathematical interval  
    <https://github.com/AlexandreDecan/portion>
  + Inline C in python  
    <https://github.com/georgek42/inlinec>
  + Pandas-vet (Pandas linting plugin for Flake8)  
    <https://github.com/deppen8/pandas-vet>
  + Pystencils (Run blazingly fast stencil codes on numpy arrays)  
    <https://github.com/mabau/pystencils>
  + Sound play and recording
    - spokestack (Audio interface into any python application)  
      <https://github.com/spokestack/spokestack-python>
    - <https://realpython.com/playing-and-recording-sound-python/>
  + Game boy emulator  
    <https://github.com/Baekalfen/PyBoy>
  + Game engine  
    <https://github.com/pokepetter/ursina>
  + Excel to python code  
    <https://github.com/dgorissen/pycel>
  + iCalendar parser  
    <https://github.com/idlesign/icalendar_light>
  + API documentation  
    <https://github.com/hh-h/aiohttp-swagger3> (Swagger, aiohttp)
  + Serverless plugin for python  
    <https://github.com/logandk/serverless-wsgi>
  + Darknet.py (Network traffic anonymize)  
    <https://github.com/multiversecoder/darknet.py>
  + Python switch implementation with Enum  
    <https://github.com/ralsina/enum_switch>
  + Real time stream processing  
    <https://github.com/python-streamz/streamz>
  + File management
    - Parsec (Open source Dropbox-like file sharing)  
      <https://github.com/Scille/parsec-cloud>
  + Deltapy - Tabular Data Augmentation  
    <https://github.com/firmai/deltapy/>
  + Python C++ binding  
    <https://realpython.com/python-bindings-overview/>  
    <https://cppyy.readthedocs.io/en/latest/index.html>
  + SimPy (discrete event simulation framework)  
    <https://gitlab.com/team-simpy/simpy>  
    <https://realpython.com/simpy-simulating-with-python/>
  + OpenGL binding  
    <https://github.com/moderngl/moderngl>
  + Domain-Specific Languages and parsers in Python  
    <https://github.com/textX/textX>   
    <https://tomassetti.me/quick-domain-specific-languages-in-python-with-textx/>
  + Compile Nim in python  
    <https://github.com/Pebaz/nimporter>
  + KILT (Knowledge Intensive Language Tasks)  
    <https://github.com/facebookresearch/KILT>
  + Spylls (Spell checker)  
    <https://github.com/zverok/spylls>
  + Plugin system  
    <https://github.com/pytest-dev/pluggy>
  + COM API
    - comtypes  
      <https://github.com/enthought/comtypes>
    - pywin32  
      <https://github.com/mhammond/pywin32>
  + Decode/Sniffing
    - Packet sniffer(Pure python network packet sniffing)  
      <https://github.com/EONRaider/Packet-Sniffer>
    - Scapy (Interactive packet manipulation)  
      <https://github.com/secdev/scapy>
  + pdoc(Object API documentation)  
    <https://github.com/mitmproxy/pdoc>
  + MovingPandas (GeoPandas related)  
    <https://github.com/anitagraser/movingpandas>
  + Paper ETL (For medical and scientific papers)  
    <https://github.com/neuml/paperetl>
  + Logging
    - Loguru  
      <https://github.com/Delgan/loguru>
    - Icecream (Better than print)  
      <https://github.com/gruns/icecream>
    - Structlog  
      <https://github.com/hynek/structlog>
  + Hot reloading  
    <https://github.com/breuleux/jurigged>
  + Text matching  
    <https://github.com/tfeldmann/simplematch>
  + Datetime library
    - Arrow  
      <https://github.com/arrow-py/arrow>
  + Shell
    - sh (subprocess replacement)  
      <https://github.com/amoffat/sh>
  + Find dead code
    - vulture  
      <https://github.com/jendrikseipp/vulture>
  + Sorting
    - Natsort (Natural sorting library)  
      <https://github.com/SethMMorton/natsort>
  + Cookiecutter (CLI that creates projects from cookiecutters - project templates)  
    <https://github.com/cookiecutter/cookiecutter>
  + Attrs (Python classes without boilerplate)  
    <https://github.com/python-attrs/attrs>
  + Responder (mDNS poisoner and relay)  
    <https://github.com/lgandx/Responder>
  + Semantic Python collection  
    <https://github.com/pysemtec/semantic-python-overview>

### Media (Image/Video)

* + Video
    - Videohash (Perceptual video hashing: Near duplicate video detection)  
      <https://github.com/akamhy/videohash>
    - MaryJane (MJPEG server to stream video? With FFMPEG)  
      <https://github.com/bootrino/maryjane>

### Runtime/Compiler

* + Pyscript (Run python on browser html)  
    <https://github.com/pyscript/pyscript>
  + Pyjion (JIT extension for python)  
    <https://github.com/tonybaloney/pyjion>
  + Pypy
  + Pyodide (Python compiled to webassembly, WASM; run python in browser)  
    <https://github.com/pyodide/pyodide>
  + Wasmer  
    <https://github.com/wasmerio/wasmer-python>
  + Brython (Python3 running in the browser)  
    <https://github.com/brython-dev/brython>

### Performance

* + Core
    - fastcore (more core level functions inspired by other languages)  
      <https://github.com/fastai/fastcore>
  + Faster math operations
    - Bolt (10x faster matrix and vector operations)  
      <https://github.com/dblalock/bolt>

### Security

* + TLS certs
    - Trustme (#1 quality TLS certs)  
      <https://github.com/python-trio/trustme>
  + Find security issues  
    <https://github.com/PyCQA/bandit>
  + Secure (Secure header for web frameworks)  
    <https://github.com/TypeError/secure>
  + Pycharm security plugin  
    <https://github.com/tonybaloney/pycharm-security>

### Cache

* + Cache function calls  
    <https://github.com/karlicoss/cachew>
  + Diskcache (Faster than Redis and Memcached)  
    <https://github.com/grantjenks/python-diskcache>

### Debugging & Profiling

* Dependency check
  + Import tracker (tracking third party dependencies within a library)  
    <https://github.com/IBM/import-tracker>
* Debugging
  + Pdb(ipdb)
  + Cyberbrain(Python debugging to see each step in flow)  
    <https://github.com/laike9m/Cyberbrain>
  + Pudb (Console based visual debugger)  
    <https://github.com/inducer/pudb>
* Benchmarking
  + asv(Airspeed Velocity)  
    <https://github.com/airspeed-velocity/asv>
* Profiling
  + Pyinstrument (Call stack profiler)  
    <https://github.com/joerick/pyinstrument>
  + Yappi (Yet another python profiler)  
    <https://github.com/sumerc/yappi>
  + Line profiler  
    <https://github.com/pyutils/line_profiler>
  + Memory profiler  
    <https://github.com/pythonprofilers/memory_profiler>
  + Filprofiler (memory profiler)  
    <https://pypi.org/project/filprofiler/>
  + Pyroscope (Continuous profiling platform)  
    <https://github.com/pyroscope-io/pyroscope>
  + PyLiveUpdate (Runtime python code manipulator)  
    <https://github.com/devopspp/pyliveupdate>
  + Pyheat (Show heatmap of profiling result)  
    <https://github.com/csurfer/pyheat>
  + Memray (Memory profiling)  
    <https://github.com/bloomberg/memray>

### Web Frameworks

* API Frameworks
  + Django
  + Flask
  + DRF
  + FastAPI(rapidly growing)  
    <https://camillovisini.com/article/abstracting-fastapi-services/> (Project structure)  
    <https://github.com/tiangolo/fastapi>
  + Hydra  
    <https://github.com/facebookresearch/hydra> (easy configuration)
  + Pyramid
  + Tornado
  + Ariadne (Python library for implementing GraphQL servers)  
    <https://github.com/mirumee/ariadne>
* Static File serving
  + whitenoise (Static file serving for python web apps)  
    <https://github.com/evansd/whitenoise>
* Microservice
  + Opyrator (Python functions into microservices)  
    <https://github.com/ml-tooling/opyrator>
  + Nameko  
    <https://github.com/nameko/nameko>  
    <https://www.toptal.com/python/introduction-python-microservices-nameko>
* Static site generation
  + Nikola  
    <https://github.com/getnikola/nikola>
  + <https://github.com/getpelican/pelican>
* ASGI
  + Framework
    - Starlette (For high performance asyncio services)  
      <https://www.starlette.io/>
  + Server
    - uvicorn
    - daphne
    - hypercorn
  + libraries
    - magnum (Lambda and API Gateway support)
* Authentication
  + Aioauth (Asynchronous OAuth 2.0)  
    <https://github.com/aliev/aioauth>
* Serverless
  + Zappa
  + Chalice (AWS Lambda, API Gateway, Python)  
    <https://github.com/chalice-dev/awesome-chalice>   
    <https://chalice.readthedocs.io/en/latest/>

### GUI development

* + Gooey (Turn python CLI program into GUI application)  
    <https://github.com/chriskiehl/Gooey>
  + PySimpleGUI
  + DearPyGui  
    <https://github.com/hoffstadt/DearPyGui>
  + Pyside
  + PyQT
  + Tkinter
    - Tkinter designer  
      <https://github.com/ParthJadhav/Tkinter-Designer>

### Python Packaging/Executable

* Python Packaging
  + Sailboat (Distribute your python projects)  
    <https://github.com/cole-wilson/sailboat>
  + <https://github.com/indygreg/PyOxidizer> (Python packaging and distribution tool)  
    <https://github.com/jwodder/check-wheel-contents> (Check wheel content)  
    <https://github.com/pywharf/pywharf> (Private pypi server)  
    <https://github.com/ankur-gupta/rain> (Example for packaging, test and deploy)  
    <https://github.com/tiangolo/typer-cli> (Can be used without creating package)
* Executable
  + Pipx (Run python package in cli)  
    <https://github.com/pypa/pipx>
  + Pyinstaller(Freeze (package) Python programs into stand-alone executables)  
    <https://github.com/pyinstaller/pyinstaller>
  + cx\_freeze

## Visualization

* + SDF (3d mesh with SDF)  
    <https://github.com/fogleman/sdf>
  + Hiplot  
    <https://github.com/facebookresearch/hiplot>  
    <https://ai.facebook.com/blog/hiplot-high-dimensional-interactive-plots-made-easy/>
  + Graphviz  
    <https://github.com/xflr6/graphviz>
  + Visual Automata (python visualization based on automata)  
    <https://github.com/lewiuberg/visual-automata>
  + D3-chord in python  
    <https://github.com/shahinrostami/chord>
  + Handcalcs (Python calculations in Jupyter as though you wrote them by hand)  
    <https://github.com/connorferster/handcalcs>
  + SweetViz  
    <https://github.com/fbdesignpro/sweetviz>
  + Make Diagram in python  
    <https://github.com/mingrammer/diagrams>  
    <https://github.com/pinetr2e/napkin>
  + Visdom (Flexible tool for creating, organizing, and sharing visualizations of live, rich data)  
    <https://github.com/fossasia/visdom>
  + Seaborn
  + Bokeh
  + Pygal(SVG Plot)  
    <https://github.com/Kozea/pygal>   
    <https://towardsdatascience.com/interactive-data-visualization-in-python-with-pygal-4696fccc8c96>
  + Diagrams (Diagram as code for cloud system architecture)  
    <https://github.com/mingrammer/diagrams>
  + Matplot related  
    <https://pbpython.com/effective-matplotlib.html>   
    <https://github.com/dexplo/dexplot/> (Matplotlib with Dataframe)  
    <https://github.com/rougier/matplotlib-3d>  
    <https://github.com/maxhumber/gif>  
    <https://github.com/johnhw/funnelplot> (Funnel plot)  
    <https://github.com/alopezrivera/mpl_plotter> (data visualization using matplotlib)  
    <https://github.com/lgienapp/aquarel> (Matplotlib theme)
  + Map
    - Leafmap (geospatial analysis and interactive mapping)  
      <https://github.com/giswqs/leafmap>
  + Plotting on terminal  
    <https://github.com/piccolomo/plotext>
  + Manim (Python framework for creating mathematical animations)  
    <https://github.com/ManimCommunity/manim>
  + Networkx (Network analysis in python)  
    <https://github.com/networkx/networkx>
  + Vedo (python module for scientific analysis of 3D data)  
    <https://github.com/marcomusy/vedo>
  + Samila (Generative art generator)  
    <https://github.com/sepandhaghighi/samila>

## Code Format

* + Convention  
    <https://realpython.com/python-return-statement/#using-the-python-return-statement-best-practices> (return usage)
  + Sourcery (Plugin)
  + Best practice python tooling   
    <https://medium.com/@jonas.r.kemper/understanding-best-practice-python-tooling-by-comparing-popular-project-templates-6eba49229106>
  + Type checker  
    <https://github.com/google/pytype>   
    <https://github.com/Microsoft/pyright>   
    <https://github.com/facebook/pyre-check>   
    <https://github.com/python/mypy>  
    <https://blog.meadsteve.dev/programming/2020/02/10/types-at-the-edges-in-python/>  
    <https://hakibenita.com/python-mypy-exhaustive-checking> (exhastiveness checking)
  + Linting  
    <https://github.com/wemake-services/wemake-python-styleguide>  
    Flake8  
    flakehell  
    Pylint  
    Ruff (<https://github.com/charliermarsh/ruff> )
  + Python type in code  
    <https://jaredkhan.com/blog/resist-the-any-type>   
    <https://blog.daftcode.pl/next-steps-with-python-type-system-efc4df5251c9>  
    <https://github.com/instagram/MonkeyType>
  + Code Formatter  
    <https://github.com/google/yapf>  
    <https://github.com/psf/black>  
    <https://github.com/akaihola/darker> (Black reformatting since a given commit)  
    <https://godatadriven.com/blog/partial-python-code-formatting-with-black-pycharm/>
  + Prospector (Analyze python code and output errors, potential issues and convention violations and complexity)  
    <https://github.com/PyCQA/prospector>
  + Clean python code  
    <https://github.com/zedr/clean-code-python>   
    <https://github.com/dosisod/refurb> (refurbishing and modernizing python codebase)
  + Coding guidelines best practice  
    <https://github.com/rednafi/py-sanity>

## Code tracing

* + Hunter  
    <https://github.com/ionelmc/python-hunter/>
  + PyWhat (Identify anything from feed)  
    <https://github.com/bee-san/pyWhat>

## CLI

* + CLI for python functions  
    <https://github.com/kootenpv/cliche>
  + Pyp (Python pipe in terminal)  
    <https://github.com/hauntsaninja/pyp>
  + Typer  
    <https://typer.tiangolo.com/>
  + Pants (Build system)  
    <https://github.com/pantsbuild/pants>
  + Click(Python composable CLI toolkit)  
    <https://github.com/pallets/click/>

## Desktop App development

* + Tkinter  
    <https://realpython.com/python-gui-tkinter/>
  + Simple gui  
    <https://github.com/szsdk/quick>
  + Dear PyGUI  
    <https://github.com/hoffstadt/DearPyGui>
  + PySide
  + Textual (TUI: Text User Interface framework)  
    <https://github.com/willmcgugan/textual>

## TDD

* Posts
  + <https://simonwillison.net/2020/Feb/11/cheating-at-unit-tests-pytest-black/>
  + <https://blog.revolucija.hr/building-testing-and-deploying-django-app-with-bitbucket-pipelines-32868e12a472#.t4k9nv5u8>
  + <https://salmonmode.github.io/2019/03/29/building-good-tests.html> (Good test practices)
  + <https://www.b-list.org/weblog/2020/feb/03/how-im-testing-2020/>
  + <https://testdriven.io/blog/modern-tdd/> (Pytest, Pydantic, Flask)
  + <https://www.vinta.com.br/blog/2020/counting-queries-basic-performance-testing-in-django/>
  + <https://martinheinz.dev/blog/7> (pytest fixtures)
  + <http://blog.vaibhavmishra.com/2016/07/01/speeding-up-django-tests-during-development/>
  + <https://github.com/django-functest/django-functest/>
  + <https://jamescooke.info/django-factory-audit.html> (Django Factory)
  + Test 3rd party apis with mock  
    <https://realpython.com/testing-third-party-apis-with-mocks>
  + <https://sobolevn.me/2020/02/conditional-coverage>
  + <http://dino.codes/posts/mocking-asynchronous-functions-python/>
  + Effective python testing with pytest  
    <https://realpython.com/pytest-python-testing/>
  + Test fixtures for Django models in pytest  
    <https://realpython.com/django-pytest-fixtures/>
  + Python test automation  
    <https://automationpanda.com/2020/11/09/learning-python-test-automation/amp/>
  + S3 testing in python  
    <https://www.sanjaysiddhanti.com/2020/04/08/s3testing/>
* Libraries
  + Coverage
    - interrogate (docstring coverage)  
      <https://github.com/econchick/interrogate>
  + Scanapi (Automated integration testing and live documentation for API)  
    <https://github.com/scanapi/scanapi>
  + Locust (User load testing tool)  
    <https://github.com/locustio/locust>
  + Nose
  + Pytest Django
  + Pytest env
  + Freezegun (Mocking time on python test)  
    <https://github.com/spulec/freezegun>
  + Dynamic fixtures
  + Fixtures  
    <https://github.com/model-bakers/model_bakery>  
    <https://github.com/paulocheque/django-dynamic-fixture>  
    <https://github.com/FactoryBoy/factory_boy>
  + <https://github.com/CFMTech/pytest-monitor> (pytest plugin for monitoring resource usage)
  + <https://github.com/Zac-HD/stdlib-property-tests>
  + <https://github.com/idlesign/pytest-responsemock> (requests calls mocking for pytest)
  + Hypothesis (Property-based testing)  
    <https://github.com/HypothesisWorks/hypothesis>
  + Pynguin (Generate unit tests automatically)  
    <https://github.com/se2p/pynguin>
  + Test case generation
    - Klara  
      <https://github.com/usagitoneko97/klara>

## PDF

* + Sign and stamp on PDF  
    <https://github.com/MatthiasValvekens/pyHanko>
  + PDF to table  
    <https://github.com/socialcopsdev/camelot>
  + PDFx (Extract text, metadata, references from PDF)  
    <https://github.com/metachris/pdfx>
  + PDF in python  
    <https://github.com/mstamy2/PyPDF2>
  + fpdf2 (Simple PDF generation)  
    <https://github.com/PyFPDF/fpdf2>
  + weasyprint (HTML to PDF)
  + wkhtmltopdf wrapper  
    <https://github.com/JazzCore/python-pdfkit>
  + borb (Reading, creating and manipulating PDF files)  
    <https://github.com/jorisschellekens/borb>
  + PyMuPDF (PDF viewer, can be used as text extractor)  
    <https://github.com/pymupdf/PyMuPDF>   
    <https://medium.com/@harald.lieder/text-extraction-using-pymupdf-e455465acb03>