Python — Detailed Reference

Generated: 2025-08-19 15:53 UTC

Title

Python — Detailed Reference
Generated: 2025-08-19 15:53 UTC

Contents

Basics

Syntax, indentation (significant), comments (#), docstrings (triple quotes), REPL, script execution (python file.py).

Data Types

Numbers: int, float, complex. Sequences: list, tuple, range. Text: str (unicode). Binary: bytes, bytearray. Sets and dicts. Mutable vs immutable.

Control Flow

if, elif, else. for ... in (iterables). while loops. loop else clause. comprehensions (list, set, dict), generator expressions.

Functions

def, return, positional and keyword args, *args, **kwargs, default values, keyword-only args, positional-only args (/), annotations, lambda expressions, closures, nonlocal.

OOP

class syntax, __init__, self, class vs instance attributes, inheritance (single/multiple), MRO, super(), magic methods (__str__, __repr__, __add__, __len__), dataclasses, slots, properties.

Modules & Packaging

import system, PYTHONPATH, packages, __init__.py, pyproject.toml, pip, virtual environments, pip freeze, requirements.txt, wheel, twine.

I/O & Files

open(...), modes (r,w,rb,wb,a), encoding, pathlib.Path, os and shutil utilities, csv and json modules, context managers for safe resource handling.

Exceptions

try/except/else/finally, raising exceptions, custom exceptions, exception chaining (raise from), best practices to catch specific exceptions.

Iterators & Generators

Iterator protocol: __iter__, __next__. Generators with yield, send(), generator.close(). itertools utilities for combinatorics and streaming.

Decorators & Context Managers

Function and class decorators, functools.wraps, parameterized decorators. Context managers via __enter__/__exit__ and contextlib.contextmanager.

Typing

Type hints (PEP 484): Optional, Union, Any, Callable, Generic, TypedDict, Protocol. Use mypy or pyright for checking.

Concurrency & Asyncio

threading (GIL), multiprocessing, concurrent.futures, asyncio: async/await, event loop, tasks, gather, queues, aiohttp for async HTTP.

Stdlib Highlights

collections, functools, itertools, datetime, logging, subprocess, sqlite3, json, decimal, hashlib, secrets.

Testing & Debugging

unittest, pytest, fixtures, mocking, parametrized tests. pdb, breakpoints, logging configuration, coverage measurement.

Packaging & CI

pyproject.toml, GitHub Actions example, lint-test-build-deploy pipeline, automated releases with twine or poetry.

Performance & C Extensions

profiling (cProfile), line_profiler, memory_profiler, optimizing with numpy, Cython, pybind11, using vectorized operations.

Web & Data Science

Flask/Django/FastAPI overview, WSGI vs ASGI. NumPy, Pandas, Matplotlib, scikit-learn, PyTorch/TensorFlow, Jupyter notebooks.

Best Practices

PEP8, linters (flake8), formatters (black), typing, small functions, tests, CI, avoid mutable default args, manage secrets via env vars.

Snippets

1) Retry decorator
2) Safe file write
3) Simple HTTP server
4) Async fetch pattern
(See full PDF for code examples)

Resources

docs.python.org, Real Python, Fluent Python, Effective Python, PyPI, StackOverflow, community channels