

PYTHON SYNTAX REFERENCE 2026

Complete quick reference for Python 3.12+ | sudoflare.com

DATA TYPES						FUNCTIONS
# Strings s = "hello world" s.upper() s.lower() s.split(" ") s.strip() f"Hello {name}" # f-string		lst.remove(2) # remove by value lst.pop() # remove last lst.sort() # sort in place sorted(lst) # return sorted copy	remove by value remove last sort in place return sorted copy	{k:v for k,v in d.items()} # dict comp. s = {1, 2, 3} # set s.add(4) s.remove(1) s1 & s2 s1 s2 # intersect, union CONTROL FLOW	dict comp. set intersect, union	def greet(name, age=25): return f"Hi {name}" # Lambda sq = lambda x: x**2 # *args **kwargs def fn(*args, **kw): print(args, kw) # Decorators @staticmethod @property
len(s) s[0] s[1:5] # Numbers x = 42 y = 3.14 int("42") float("3.14") x ** 2 x // 3 x % 3 # power, floor div, mod	f-string power, floor div, mod	lst[0] lst[-1] # first, last lst[1:3] # slice len(lst) # length [x*2 for x in lst] # list comprehension tup = (1, 2, 3) # tuple (immutable)	first, last slice length list compreh ension tuple (immutable)			
# Booleans True False None bool(0) bool("") bool([]) # all False	all False	DICTIONARIES d = {"key": "val"} d["key"] # access d.get("key", "def") # safe access d["new"] = "v" # set value del d["key"] # delete key d.keys() d.values() d.items() # key-value pairs "key" in d # check exists	access safe access set value delete key key-value pairs check exists			
LISTS & TUPLES lst = [1, 2, 3] lst.append(4) # add to end lst.insert(0, 9) # insert at index	add to end insert at index					

PYTHON SYNTAX REFERENCE 2026 — PAGE 2

Classes | Files | Exceptions | Modules | sudoflare.com

<pre>CLASSES & OOP class Dog: def __init__(self, name): self.name = name def bark(self): return "Woof!" # Inheritance class Poodle(Dog): pass</pre>	<pre>d = Dog("Rex") d.bark() isinstance(d, Dog) # True True</pre>	<pre>try: risky_code() except ValueError as e: print(e) except Exception as e: finally:</pre>	<pre>import socket # Networking Networking import subprocess # Run commands Run commands commands import hashlib # Hashing Hashing import base64 # Encode/decode Encode/dec ode ode import requests # HTTP (pip) HTTP (pip) import paramiko # SSH (pip) SSH (pip) from scrapy.all import * # Packets (pip) Packets (pip) import argparse # CLI args CLI args from datetime import datetime</pre>
	<pre>FILE & EXCEPTIONS # Read file with open("f.txt") as f: data = f.read() # Write file with open("f","w") as f: f.write("hello") # Exceptions</pre>	<pre>USEFUL MODULES import os # OS operations OS operations import sys # System/args System/args import json # JSON parse/dump JSON parse/dump import re # Regex Regex</pre>	