Python Commands Cheat Sheet by https://interviewquestions.guru/

1) Core Python Keywords (Statements)

if, elif, else, for, while, break, continue, pass,

def, return, yield, lambda, class,

try, except, finally, raise, assert,

with, import, from, as, global, nonlocal, del, in, is, and, or, not

2) Built-in Functions (Most Used)

print(), input(), len(), type(), isinstance(), id(), dir(), help(),

range(), enumerate(), zip(), map(), filter(), sorted(), reversed(),

sum(), min(), max(), any(), all(), abs(), round(),

list(), tuple(), set(), dict(), frozenset(),

int(), float(), str(), bool(), bytes(), bytearray(),

ord(), chr(), bin(), oct(), hex(),

open(), vars(), getattr(), setattr(), hasattr(), delattr(),

slice(), format(), hash()

Examples

```
print("sum:", sum([1,2,3]))      # sum: 6
for i, v in enumerate(["a","b"]): print(i, v)
names = ["Ada","Alan"]; scores = [99, 95]
for n, s in zip(names, scores): print(n, s)
```

3) Printing & String Formatting

f-strings (recommended)

```
name, score = "Ada", 98.456
print(f"{name}: {score:.1f}")  # Ada: 98.5
```

Python Commands Cheat Sheet by https://interviewquestions.guru/

format()

```
print("{0}: {1:.2f}".format("Pi", 3.14159))
```

printf-style

```
print("%s scored %.1f" % ("Ada", 98.5))
```

4) String Methods (Common)

upper(), lower(), title(), capitalize(), swapcase(),

strip(), lstrip(), rstrip(), replace(old, new[, count]),

split(sep=None, maxsplit=-1), rsplit(), splitlines(),

join(iterable), find(sub), rfind(sub), index(sub), rindex(sub),

count(sub), startswith(prefix), endswith(suffix),

partition(sep), rpartition(sep),

isalnum(), isalpha(), isdigit(), isdecimal(), isnumeric(),

isspace(), islower(), isupper(), istitle(),

zfill(width), center(width[, fillchar]),

ljust(width[, fillchar]), rjust(width[, fillchar])

Examples

```
s = "  hello, world  "
print(s.strip().title())         # Hello, World
print("a,b,c".split(","))        # ['a','b','c']
print(",".join(["x","y"]))       # x,y
print("banana".count("an"))      # 2
```

5) Lists (Methods & Patterns)

Methods

append(x), extend(it), insert(i, x), pop([i]),

remove(x), clear(), sort(key=None, reverse=False),

reverse(), copy(), count(x), index(x[, start[, end]])

Examples

```
nums = [3,1,2]; nums.append(4); nums.extend([5,6])
nums.sort()           # [1,2,3,4,5,6]
nums.reverse()        # [6,5,4,3,2,1]
```

Comprehensions

```
squares = [n*n for n in range(10) if n%2==0]
```

Slicing

```
a = [0,1,2,3,4,5]
print(a[1:4])    # [1,2,3]
print(a[:3])     # [0,1,2]
print(a[::2])    # [0,2,4]
print(a[::-1])   # reversed
```

6) Tuples (Immutable)

Methods

count(x), index(x)

Use cases: fixed records, dict keys, safe return of multiple values.

7) Sets (Unique, Unordered)

Methods

add(x), remove(x), discard(x), pop(), clear(), copy(),

union(*others), intersection(*others), difference(*others),

symmetric_difference(other), issubset(other), issuperset(other), isdisjoint(other)

Examples

A, B = {1,2,3}, {3,4}

print(A | B) # union -> {1,2,3,4}

print(A & B) # intersection -> {3}

print(A - B) # difference -> {1,2}

8) Dictionaries (Key–Value)

Methods

get(k[, default]), setdefault(k[, default]), update(m),

keys(), values(), items(), pop(k[, d]), popitem(),

clear(), copy(), fromkeys(keys[, v])

Examples

user = {"name":"Ada", "age":36}

user.setdefault("role","dev")

user.update({"city":"Oxford"})

for k, v in user.items(): print(k, v)

Dict comprehensions

squares = {n: n*n for n in range(5)}

9) Control Flow Patterns

enumerate + zip

for i, v in enumerate(["a","b"]): print(i, v)

for n, s in zip(names, scores): print(n, s)

Python Commands Cheat Sheet by https://interviewquestions.guru/

```
# early returns
def grade(p):
    if p>=90: return "A"
    if p>=80: return "B"
    return "C"

# ternary expression
status = "adult" if age>=18 else "minor"
```

10) Functions

```
def greet(name, loud=False):
    msg = f"Hello, {name}"
    return msg.upper() if loud else msg
```

Args & kwargs

```
def f(a, b=0, *args, **kwargs): ...
```

Lambdas

```
square = lambda x: x*x
```

Annotations

```
def add(a: int, b: int) -> int: return a+b
```

11) Errors & Exceptions

```
try:
    risky()
except (ValueError, KeyError) as e:
    print("Handled:", e)
```

```
else:
    print("No errors")
finally:
    print("Always runs")


raise ValueError("bad value")
assert x > 0, "x must be positive"
```

12) File I/O (Text & Binary) + Pathlib

```
# with open (text)
with open("data.txt","w",encoding="utf-8") as f:
    f.write("hello\n")
with open("data.txt","r",encoding="utf-8") as f:
    print(f.read())

# binary
with open("image.png","rb") as f:
    raw = f.read()

# pathlib (recommended)
from pathlib import Path
p = Path("notes.txt")
p.write_text("Hi", encoding="utf-8")
print(p.read_text(encoding="utf-8"))
print(list(Path(".").glob("*.py")))
```

Python Commands Cheat Sheet by https://interviewquestions.guru/

13) Useful Standard Library Snippets

math / statistics / random

```
import math, statistics as stats, random
math.sqrt(16); math.pi
stats.mean([1,2,3])
random.choice(["red","blue"])
random.sample(range(100), k=5)
```

datetime

```
from datetime import datetime, timedelta, date
now = datetime.now()
print(now.strftime("%Y-%m-%d %H:%M"))
tomorrow = date.today() + timedelta(days=1)
```

collections

```
from collections import Counter, defaultdict, deque
Counter("banana")                # counts
d = defaultdict(int); d["x"] += 1 # auto-init
q = deque([1,2]); q.appendleft(0)
```

itertools (power iterators)

```
from itertools import combinations, permutations, product, groupby
list(combinations([1,2,3], 2))    # [(1,2),(1,3),(2,3)]
```

re (regex)

```
import re
m = re.search(r"\d+", "ID=42"); print(m.group())  # 42
emails = re.findall(r"[a-z.\-]+@[a-z.\-]+\.[a-z]+", text)
```

Python Commands Cheat Sheet by https://interviewquestions.guru/

json / csv

```
import json, csv
# JSON
data = {"a": 1}
s = json.dumps(data)             # to string
obj = json.loads(s)              # from string
Path("x.json").write_text(json.dumps(data, indent=2))

# CSV
rows = [["name","score"],["Ada",99]]
with open("scores.csv","w",newline="",encoding="utf-8") as f:
    csv.writer(f).writerows(rows)
```

subprocess (run shell commands)

```
import subprocess as sp
sp.run(["echo", "Hello"], check=True)
```

14) REPL Helpers

```
help(str.split)     # docs
dir(list)           # attributes/methods
type(obj)           # type info
_                   # last result (REPL only)
quit(), exit()      # leave REPL
```

15) CLI & Environment

Run Python

Python Commands Cheat Sheet by https://interviewquestions.guru/

python --version

python script.py

python - << 'PY'

print("inline")

PY

Virtual Environments (venv)

python -m venv .venv

macOS/Linux

source .venv/bin/activate

Windows

.venv\Scripts\activate

deactivate

pip (always prefer python -m pip)

python -m pip install requests

python -m pip install --upgrade package

python -m pip uninstall package

python -m pip list

python -m pip freeze > requirements.txt

16) Debugging & Testing

Debugging (built-in)

breakpoint() # enters debugger (python -Xfrozen_modules=off sometimes helps)

pytest (very common)

test_example.py

```
def add(a,b): return a+b
def test_add(): assert add(2,3)==5
# run:
# pytest -q
```

17) Common Patterns (Copy-Paste)

Sorting by key/value

```
d = {"alice":3, "bob":5, "carl":1}
print(sorted(d.items(), key=lambda kv: kv[1], reverse=True))
```

Safe dictionary access

```
age = user.get("age", 0)
```

Flatten a list of lists

```
flat = [x for row in grid for x in row]
```

Unique while preserving order

```
seen=set(); out=[]
for x in seq:
    if x not in seen:
        seen.add(x); out.append(x)
```

Reading a file safely to list

```
with open("data.txt", encoding="utf-8") as f:
    lines = [line.rstrip("\n") for line in f]
```

18) Performance Hints

- Prefer **comprehensions** and **generator expressions** for transforms.
- Use **set** for membership tests on large collections.

Python Commands Cheat Sheet by https://interviewquestions.guru/

- Use **sorted(iterable)** for a non-destructive sort; list.sort() modifies in place.
- For big loops, consider **itertools** tools (they're C-optimized).
- Avoid building huge strings with + in loops; use ''.join(parts).

19) Pitfalls to Avoid (Quick Reminders)

- is vs ==: use == for value equality.
- **Mutable default args**: use None and set inside the function.
- Strings are **immutable**; methods return new strings.
- Always close files → use with open(...).
- Catch **specific** exceptions, not bare except:.

20) Mini Reference Blocks

Truthiness

Falsey: 0, 0.0, 0j, "", [], {}, set(), range(0), None, False

Everything else is truthy.

Operator Quickies

Arithmetic: + - * / // % **

Comparison: == != < <= > >=

Logic: and or not

Membership: in, not in

Identity: is, is not

Useful Slices

a[start:stop:step] a[::-1] (reverse) a[-n:] (last n)

End note

Python Commands Cheat Sheet by https://interviewquestions.guru/

Keep this handy. Combine it with practice and the **official docs** when you need deeper behavior details.