

# Referensi Cepat Python 3

Dasar hingga pandas, requests, csv, json

## Dasar

### Variabel

```
name = "Alice" # str
age = 20       # int
gpa = 3.85    # float
active = True  # bool
```

### Tipe Data

```
str    Teks: "hello"
int    Bilangan bulat: 42
float  Desimal: 3.14
bool   True / False
list   Terurut, mutable: [1, 2, 3]
tuple  Terurut, immutable: (1, 2)
dict   Key-value: {"a": 1}
set    Item unik: {1, 2, 3}
```

### Aritmatika

```
+ - *   Tambah, kurang, kali
/       Pembagian (float): 7/2 → 3.5
//      Pembagian bulat: 7//2 → 3
%       Modulo: 7%2 → 1
**      Pangkat: 2**3 → 8
```

### Konversi Tipe

```
int("42") # 42
float("3.14") # 3.14
str(100) # "100"
list("abc") # ['a', 'b', 'c']
```

### Input Pengguna

```
name = input("Your name? ")
age = int(input("Age? "))
```

## String

### Membuat String

```
s1 = 'single quotes'
s2 = "double quotes"
s3 = """triple quotes
for multiline"""
```

### f-String (Python 3.6+)

```
name = "Alice"
f"Hello, {name}!" # Hello, Alice!
f"{2 + 3}" # 5
f"{3.14159:.2f}" # 3.14
f"{1000:,"} # 1,000
```

### Slicing String

```
s = "Python"
# Index: 0 1 2 3 4 5
s[0] # 'P'
s[-1] # 'n'
s[2:5] # 'tho'
s[:2] # 'Py'
s[2:] # 'thon'
s[::-1] # 'nohtyP' (reverse)
```

## Method String

```
len(s)          Panjang string
s.upper()       HURUF BESAR
s.lower()       huruf kecil
s.strip()       Hapus spasi di awal/akhir
s.split(",")    Pecah menjadi list
", ".join(lst)  Gabung list menjadi string
s.replace(a, b) Ganti a dengan b
s.find("x")     Index kemunculan pertama (-1 jika tidak ada)
s.startswith(x) Cek awalan → bool
s.endswith(x)  Cek akhiran → bool
s.count(x)     Hitung kemunculan
"x" in s       Cek keberadaan → bool
```

## List

### Membuat & Mengakses

```
fruits = ["apple", "banana", "cherry"]
fruits[0] # "apple"
fruits[-1] # "cherry"
fruits[1:3] # ["banana", "cherry"]
```

### List Comprehension

```
squares = [x**2 for x in range(5)]
# [0, 1, 4, 9, 16]
evens = [x for x in range(10) if x%2==0]
# [0, 2, 4, 6, 8]
```

### Method List

```
lst.append(x)      Tambah di akhir
lst.extend(lst2)   Tambah semua item dari lst2
lst.insert(i, x)   Sisipkan di index i
lst.pop()          Hapus & kembalikan item terakhir
lst.pop(i)         Hapus & kembalikan item di i
lst.remove(x)      Hapus x pertama
del lst[i]         Hapus berdasarkan index
lst.sort()         Urutkan di tempat
sorted(lst)        Kembalikan salinan terurut
lst.reverse()     Balik urutan di tempat
len(lst)          Jumlah item
x in lst           Cek keanggotaan
lst.index(x)      Index pertama x
lst.count(x)      Jumlah x
```

## Tuple & Set

### Tuple (Immutable)

```
point = (3, 4)
x, y = point # unpacking
point[0] # 3 (read-only)
```

### Set (Item Unik)

```
s = {1, 2, 3}
s.add(4); s.remove(1)
a & b # intersection
a | b # union
a - b # difference
```

## Dictionary

### Membuat & Mengakses

```
student = {"name": "Alice", "age": 20}
student["name"] # "Alice"
student.get("gpa", 0) # 0 (default)
student["gpa"] = 3.85 # tambah/update
```

## Dict Comprehension

```
sq = {x: x**2 for x in range(5)}
# {0:0, 1:1, 2:4, 3:9, 4:16}
```

## Iterasi

```
for k, v in student.items():
    print(f"{k}: {v}")
```

## Method Dict

```
d.keys()          Semua key
d.values()        Semua value
d.items()         Semua pasangan (key, value)
d.get(k, default) Ambil dengan nilai default
d.update(d2)      Gabung d2 ke d
d.pop(k)          Hapus & kembalikan value
del d[k]          Hapus key
"k" in d          Key ada? → bool
len(d)           Jumlah entri
```

## Alur Kontrol

### if / elif / else

```
if score >= 90:
    grade = "A"
elif score >= 80:
    grade = "B"
else:
    grade = "C"
```

### Operator Ternary

```
status = "pass" if score >= 60 else "fail"
```

## Loop

### for Loop

```
for fruit in ["apple", "banana"]:
    print(fruit)
```

### range()

```
range(5) # 0, 1, 2, 3, 4
range(2, 5) # 2, 3, 4
range(0, 10, 2) # 0, 2, 4, 6, 8
```

### while Loop

```
while count < 10:
    count += 1
```

### enumerate() & zip()

```
for i, val in enumerate(["a", "b"]):
    print(i, val) # 0 a, 1 b
```

```
for a, b in zip([1, 2], ["x", "y"]):
    print(a, b) # 1 x, 2 y
```

### break & continue

```
for x in range(10):
    if x == 5: break # stop loop
    if x % 2 == 0: continue # skip
```

# Referensi Cepat Python 3

## Fungsi

### Mendefinisikan & Memanggil

```
def greet(name, greeting="Hi"):
    return f"{greeting}, {name}!"

greet("Alice") # "Hi, Alice!"
greet("Bob", "Hello") # "Hello, Bob!"
```

### Multiple Return Value

```
def min_max(lst):
    return min(lst), max(lst)

lo, hi = min_max([3, 1, 4, 1, 5])
```

### \*args & \*\*kwargs

```
def total(*args): # args is a tuple
    return sum(args)
total(1, 2, 3) # 6

def info(**kwargs): # kwargs is a dict
    print(kwargs)
```

### Fungsi Lambda

```
square = lambda x: x**2
square(5) # 25
sorted(lst, key=lambda x: x["age"])
```

## Class

```
class Dog:
    def __init__(self, name, breed):
        self.name = name
        self.breed = breed

    def bark(self):
        return f"{self.name} says Woof!"

dog = Dog("Rex", "Lab")
dog.bark() # "Rex says Woof!"
```

### Pewarisan

```
class Puppy(Dog):
    def __init__(self, name, breed, toy):
        super().__init__(name, breed)
        self.toy = toy
```

### Penanganan Error

```
try:
    result = 10 / 0
except ZeroDivisionError as e:
    print(f"Error: {e}")
finally:
    print("Always runs")
```

## File I/O

### Membaca File

```
with open("data.txt") as f:
    content = f.read() # full text

with open("data.txt") as f:
    for line in f: # line by line
        print(line.strip())
```

### Menulis File

```
with open("out.txt", "w") as f:
    f.write("Hello\n")
```

"r" = baca "w" = tulis (timpa) "a" = tambah (append)

refmint.com

## CSV

```
import csv

with open("data.csv") as f:
    reader = csv.DictReader(f)
    for row in reader:
        print(row["name"])

with open("out.csv", "w", newline="") as f:
    writer = csv.writer(f)
    writer.writerow(["name", "age"])
```

## JSON

```
import json

data = json.loads('{"name": "Alice"}') # parse
text = json.dumps(data) # serialize

with open("data.json") as f:
    data = json.load(f) # read file
with open("out.json", "w") as f:
    json.dump(data, f, indent=2) # write file
```

## HTTP Request

```
import requests

# GET
r = requests.get("https://api.example.com/data")
r.status_code # 200
data = r.json() # parse JSON

# POST
r = requests.post(url, json={"key": "val"})
```

## Dasar pandas

```
import pandas as pd
df = pd.read_csv("data.csv")
df.head() # first 5 rows
df.shape # (rows, cols)
df["name"] # single column
df[df["age"] > 20] # filter rows
```

## Built-in Berguna

<b>print()</b>	Output ke konsol
<b>len()</b>	Panjang / jumlah
<b>type()</b>	Tipe objek
<b>range()</b>	Urutan angka
<b>enumerate()</b>	Pasangan index + value
<b>zip()</b>	Pasangan item dari beberapa iterable
<b>sorted()</b>	Kembalikan salinan terurut
<b>sum()</b> <b>min()</b> <b>max()</b>	Fungsi agregat

## Modul

```
import math
from math import sqrt, pi
import pandas as pd # alias
```