Python Developer

Master Cheatsheet 2025

Basics | Intermediate | Playwright | PyAutoGUI | Flask | Streamlit

■ Python Basics

Concept	Syntax	Example
Variables	name = value	age = 25; name = "Manoj"
Data Types	int, float, str, bool, list, dict, tuple	x = 10; y = 3.14; s = "hello"
List	[item1, item2, ...]	nums = [1, 2, 3]; nums.append(4)
Dictionary	{key: value}	user = {"name": "Manoj", "age": 25}
Tuple	(item1, item2) - immutable	coords = (10, 20)
Set	{item1, item2} - unique	unique = {1, 2, 3}
If-Else	if condition: ... else:	if x > 0: print("positive")
For Loop	for item in iterable:	for i in range(5): print(i)
While Loop	while condition:	while x > 0: x -= 1
Function	def name(params):	def greet(n): return f"Hi {n}"
Lambda	lambda args: expression	square = lambda x: x**2
List Comp	[expr for item in list]	[x**2 for x in range(5)]

■ Essential String Methods

Method	Description	Example
split()	Split string into list	"a,b,c".split(",") → ["a","b","c"]
join()	Join list into string	",".join(["a","b"]) → "a,b"
strip()	Remove whitespace	" hi ".strip() → "hi"
replace()	Replace substring	"hello".replace("l","x") → "hexxo"
f-string	Format string	f"Name: {name}, Age: {age}"
upper/lower	Change case	"Hi".upper() → "HI"

■ Python Intermediate

Concept	Syntax / Usage	Example
*args	Variable positional args	def func(*args): for a in args
**kwargs	Variable keyword args	def func(**kw): for k,v in kw.items()
Decorator	@decorator_name	@timer def slow(): ...
Context Manager	with statement	with open("f.txt") as f: data=f.read()

Generator	yield keyword	def gen(): yield 1; yield 2
Exception	try/except/finally	try: x=1/0 except: print("error")
Class	class Name:	class Dog: def __init__(self, name):
Inheritance	class Child(Parent):	class Cat(Animal): pass
@property	Getter/Setter	@property def name(self): return self._n
@staticmethod	No self needed	@staticmethod def helper(): ...
@classmethod	Access cls	@classmethod def create(cls): ...

File Operations

Operation	Code
Read file	with open("file.txt", "r") as f: content = f.read()
Write file	with open("file.txt", "w") as f: f.write("hello")
Append	with open("file.txt", "a") as f: f.write("more")
Read lines	with open("f.txt") as f: lines = f.readlines()
Read JSON	import json; data = json.load(open("d.json"))
Write JSON	json.dump(data, open("d.json", "w"), indent=2)

Playwright - Browser Automation

Install: pip install playwright && playwright install

Action	Code
Setup	from playwright.sync_api import sync_playwright
Launch Browser	browser = playwright.chromium.launch(headless=False)
New Page	page = browser.new_page()
Navigate	page.goto("https://example.com")
Click	page.click("button#submit") # or page.locator("btn").click()
Fill Input	page.fill("input#email", "test@example.com")
Type Text	page.type("input#search", "hello", delay=100)
Get Text	text = page.locator("h1").text_content()
Wait Element	page.wait_for_selector("div.loaded", timeout=5000)
Screenshot	page.screenshot(path="screenshot.png")
PDF	page.pdf(path="page.pdf")
Get Attribute	href = page.locator("a").get_attribute("href")
Select Option	page.select_option("select#country", "IN")
Check/Uncheck	page.check("input#agree"); page.uncheck("input#news")
Keyboard	page.keyboard.press("Enter")

Close	browser.close()

```
with sync_playwright() as p:
browser = p.chromium.launch(headless=False)
page = browser.new_page()
page.goto("https://google.com")
page.fill("textarea[name='q']", "Python tutorial")
page.keyboard.press("Enter")
page.wait_for_load_state("networkidle")
page.screenshot(path="results.png")
browser.close()
```

■■ PyAutoGUI - Desktop Automation

■ ***Install: pip install pyautogui***

Action	Code
Import	import pyautogui
Screen Size	width, height = pyautogui.size()
Mouse Position	x, y = pyautogui.position()
Move Mouse	pyautogui.moveTo(100, 200, duration=0.5)
Move Relative	pyautogui.moveRel(50, 0) # Move right 50px
Click	pyautogui.click(100, 200) # or click()
Double Click	pyautogui.doubleClick()
Right Click	pyautogui.rightClick()
Drag	pyautogui.drag(100, 0, duration=0.5) # Drag right
Scroll	pyautogui.scroll(3) # Scroll up; -3 for down
Type Text	pyautogui.write("Hello World", interval=0.1)
Press Key	pyautogui.press("enter") # or "tab", "esc"
Hotkey	pyautogui.hotkey("ctrl", "c") # Copy
Screenshot	img = pyautogui.screenshot("screen.png")
Locate Image	loc = pyautogui.locateOnScreen("btn.png")
Click Image	pyautogui.click(pyautogui.locateCenterOnScreen("btn.png"))
Alert Box	pyautogui.alert("Done!")
Confirm Box	result = pyautogui.confirm("Continue?")
Prompt	name = pyautogui.prompt("Enter name:")
Failsafe	pyautogui.FAILSAFE = True # Move to corner to stop

■■ Flask - Web API Development

■ ***Install: pip install flask flask-cors***

Concept	Code

Basic App	from flask import Flask; app = Flask(__name__)
Run App	app.run(debug=True, port=5000)
GET Route	@app.route("/users") def get_users(): return jsonify(users)
POST Route	@app.route("/users", methods=["POST"])
Get JSON	data = request.get_json()
Get Form	name = request.form.get("name")
Get Query	page = request.args.get("page", 1, type=int)
URL Param	@app.route("/user/<int:id>") def get_user(id):
Return JSON	return jsonify({"status": "ok", "data": result})
Status Code	return jsonify(error="Not found"), 404
CORS	from flask_cors import CORS; CORS(app)
Before Request	@app.before_request def auth(): ...
Error Handler	@app.errorhandler(404) def not_found(e): ...

```
from flask import Flask, jsonify, request
app = Flask(__name__)

@app.route("/api/hello", methods=["GET"])
def hello():
name = request.args.get("name", "World")
return jsonify({"message": f"Hello {name}!"})

@app.route("/api/users", methods=["POST"])
def create_user():
data = request.get_json()
return jsonify({"created": data}), 201

if __name__ == "__main__":
app.run(debug=True)
```

Streamlit - Data Apps

Install: pip install streamlit | Run: streamlit run app.py

Component	Code
Import	import streamlit as st
Title	st.title("My App")
Header	st.header("Section"); st.subheader("Sub")
Text	st.write("Hello"); st.markdown("**bold**")
Text Input	name = st.text_input("Your name")
Number Input	age = st.number_input("Age", min_value=0)
Text Area	bio = st.text_area("Bio", height=100)
Button	if st.button("Click Me"): st.write("Clicked!")
Checkbox	agree = st.checkbox("I agree")

Radio	choice = st.radio("Pick", ["A", "B", "C"])
Selectbox	opt = st.selectbox("Choose", ["X", "Y"])
Multi Select	items = st.multiselect("Select", [1,2,3])
Slider	val = st.slider("Value", 0, 100, 50)
File Upload	file = st.file_uploader("Upload", type=["csv"])
DataFrame	st.dataframe(df); st.table(df)
Chart	st.line_chart(df); st.bar_chart(df)
Columns	col1, col2 = st.columns(2)
Sidebar	st.sidebar.title("Menu")
Expander	with st.expander("More"): st.write("...")
Spinner	with st.spinner("Loading..."): time.sleep(2)
Success/Error	st.success("Done!"); st.error("Failed!")
Session State	st.session_state["key"] = value

```
import streamlit as st
import pandas as pd

st.title("■ My Data App")
name = st.text_input("Enter your name")
if name:
st.write(f"Hello, {name}!")

uploaded = st.file_uploader("Upload CSV", type="csv")
if uploaded:
df = pd.read_csv(uploaded)
st.dataframe(df)
st.line_chart(df)
```

■ Quick Reference - When to Use What?

Task	Best Tool	Key Feature
Web scraping with JS	Playwright	Handles dynamic content
Simple web scraping	requests + BeautifulSoup	Lightweight, fast
Desktop automation	PyAutoGUI	Mouse, keyboard, screenshots
REST API backend	Flask	Lightweight, flexible
Full web framework	Django	Batteries included
Data dashboard	Streamlit	Fast prototyping, ML-ready
Async web server	FastAPI	Type hints, OpenAPI docs
GUI application	PyQt / Tkinter	Desktop apps

Created by Manoj | The AI Dude Tamil ■

YouTube: The AI Dude Tamil | Master AI, Automation & Prompt Engineering

■ Code smart, automate everything!