

Python Cheat Sheet (With One-Line Definitions)

1. Variables

Simple definition: Containers used to store data values.

```
x = 10  
y = "Hello"
```

2. Data Types

Simple definition: Categories of data you can store.

```
int, float, str, bool, list, tuple, dict, set
```

3. Type Casting

Simple definition: Converting one data type into another.

```
int("10")  
str(5)
```

4. Strings

Simple definition: Text enclosed in quotes.

```
name = "JP"  
name.upper()
```

5. Lists

Simple definition: Ordered, changeable collections.

```
arr = [1,2,3]  
arr.append(4)
```

6. Tuples

Simple definition: Ordered, unchangeable collections.

```
t = (1,2,3)
```

7. Sets

Simple definition: Unordered collection of unique items.

```
s = {1,2,3}
```

8. Dictionaries

Simple definition: Key-value pairs.

```
d = {"name": "JP", "age": 24}
```

9. If Conditions

Simple definition: Code that runs only if a condition is true.

```
if x > 5: print("OK")
```

10. Loops

Simple definition: Repeating code multiple times.

```
for i in range(5): print(i)
```

11. Functions

Simple definition: Reusable blocks of code.

```
def add(a,b): return a+b
```

12. Lambda Functions

Simple definition: Small one-line anonymous functions.

```
f = lambda x: x*2
```

13. List Comprehension

Simple definition: Short way to create lists.

```
[x*2 for x in range(5)]
```

14. Exceptions

Simple definition: Handling errors without crashing.

```
try: x=1/0  
except: print("Error")
```

15. Classes

Simple definition: Blueprint for creating objects.

```
class Dog: pass
```

16. Objects

Simple definition: Instances of classes.

```
obj = Dog()
```

17. OOP Concepts

Simple definitions:

```
Encapsulation - restricting direct access.  
Abstraction - hiding complex details.  
Inheritance - one class taking features of another.  
Polymorphism - same function, different behavior.
```

18. Modules & Imports

Simple definition: Code files you can reuse.

```
import math
```

19. File Handling

Simple definition: Reading/writing files.

```
open("a.txt").read()
```

20. Virtual Environments

Simple definition: Isolated Python environments.

```
python -m venv env
```

21. Pip

Simple definition: Python package manager.

```
pip install numpy
```

22. Generators

Simple definition: Functions that return values one-by-one.

```
def gen(): yield 1
```

23. Decorators

Simple definition: Functions that modify other functions.

```
@decorator  
def func(): pass
```

24. Map, Filter, Reduce

Simple definition: Functional tools for iterables.

```
list(map(lambda x:x*2, [1,2,3]))
```

25. Unpacking

Simple definition: Splitting items automatically.

```
a,b = (1,2)
```

26. Args & Kwargs

Simple definition: Flexible arguments in functions.

```
def f(*a, **k): pass
```

27. Enumerate

Simple definition: Gives index + value in loops.

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

28. Zip

Simple definition: Combines multiple lists.

```
zip([1,2],["a","b"])
```

29. Comprehensions (All Types)

Simple definition: Compact ways to build collections.

```
{x:x*2 for x in range(3)}
```

30. Time & Date

Simple definition: Working with timestamps.

```
import datetime
```

31. OS Module

Simple definition: Interact with operating system.

```
import os
os.listdir()
```

32. Sys Module

Simple definition: Access Python interpreter internals.

```
import sys
sys.version
```

33. Random Module

Simple definition: Generate random numbers.

```
import random
random.randint(1,10)
```

34. JSON Module

Simple definition: Convert Python ↔ JSON.

```
import json
json.dumps({"a":1})
```

35. Requests (External Lib)

Simple definition: Send HTTP requests.

```
import requests
requests.get("https://api.com")
```