
🧠 PYTHON DICTIONARY — COMPLETE NOTES (for AI/ML + Interviews)

🟢 1. What is a Dictionary in Python?

A **dictionary** is a **collection of key–value pairs**.

✅ Example:

```
student = {
    "name": "Deepanshu",
    "age": 21,
    "skills": ["Python", "Machine Learning"],
    "is_employed": False
}
```

📘 Here:

- `"name"`, `"age"`, `"skills"` are **keys**
- `"Deepanshu"`, `21`, `["Python", "Machine Learning"]`, `False` are **values**

👉 Keys are **unique** and **immutable** (can't be changed).
👉 Values can be of **any data type** (string, list, int, dict, etc).

🟢 2. Why Dictionary is Important in AI/ML

Dictionaries are **heavily used in AI/ML** for:

- Mapping data (e.g. feature name → value)
- Storing model parameters

- Storing dataset statistics (e.g. mean, std)
- Configuration files (hyperparameters)
- JSON-like data (APIs, ML configs)

Example:

```
model_params = {"learning_rate": 0.01, "epochs": 50, "batch_size": 32}
```

🟢 3. Creating Dictionaries

✅ Different ways:

```
# 1. Using curly braces
person = {"name": "Raj", "age": 25}

# 2. Using dict() function
person = dict(name="Raj", age=25)

# 3. From list of tuples
pairs = [("name", "Raj"), ("age", 25)]
person = dict(pairs)

# 4. Empty dict
person = {}
```

🟢 4. Accessing & Modifying Dictionary

```
student = {"name": "Deepanshu", "age": 21, "skills": ["Python", "ML"]}

# Access value
print(student["name"])       # Output: Deepanshu

# Safer access (no error if key missing)
print(student.get("city", "Not Found"))  # Output: Not Found

# Add new key-value
student["city"] = "Chhindwara"
```

```
# Modify existing
student["age"] = 22

print(student)
```

🟢 5. Important Dictionary Methods (With Short Examples)

💡 (A) dict.keys()

Returns all keys

```
student.keys()  # dict_keys(['name', 'age', 'skills', 'city'])
```

💡 (B) dict.values()

Returns all values

```
student.values()
```

💡 (C) dict.items()

Returns key-value pairs (tuples)

```
for key, value in student.items():
    print(key, ":", value)
```

💡 (D) dict.get(key, default_value)

Safe access without KeyError

```
student.get("marks", 0)  # returns 0 if 'marks' not present
```

💡 (E) dict.update(other_dict)

Adds/updates multiple key-values

```
student.update({"marks": 90, "city": "Nagpur"})
```

💡 (F) dict.pop(key)

Removes key and returns its value

student.pop("age")

💡 (G) dict.popitem()

Removes last inserted key-value

student.popitem()

💡 (H) dict.clear()

Removes all items

student.clear()

💡 (I) dict.copy()

Creates shallow copy

new_student = student.copy()

💡 (J) dict.setdefault(key, default_value)

Adds key with default value if not present

student.setdefault("grade", "A") # adds grade if missing

🟢 6. Useful Tricks (Interview-Favorite Topics)

✅ Dictionary Comprehension

Just like list comprehension.

```
# Square of numbers
squares = {x: x**2 for x in range(5)}
print(squares)  # {0:0, 1:1, 2:4, 3:9, 4:16}
```

✅ Filtering Dictionary

```
scores = {"A": 90, "B": 75, "C": 60, "D": 30}
```

```
passed = {k: v for k, v in scores.items() if v >= 60}
print(passed)  # {'A': 90, 'B': 75, 'C': 60}
```

✅ Merge Two Dictionaries

```
dict1 = {"a": 1, "b": 2}
dict2 = {"b": 3, "c": 4}
merged = {**dict1, **dict2}
# Output: {'a':1, 'b':3, 'c':4}
```

✅ Sorting Dictionary by Value

```
marks = {"Raj": 88, "Ravi": 92, "Amit": 76}
sorted_marks = dict(sorted(marks.items(), key=lambda x: x[1], reverse=True))
print(sorted_marks)  # {'Ravi':92, 'Raj':88, 'Amit':76}
```

🟢 7. Dictionary in AI/ML Practice

💻 Example 1: Store model hyperparameters

```
params = {
    "learning_rate": 0.01,
    "epochs": 100,
    "optimizer": "adam"
}
```

💻 Example 2: Count frequency of labels

```
labels = ["cat", "dog", "dog", "cat", "bird"]
freq = {}
for label in labels:
    freq[label] = freq.get(label, 0) + 1

print(freq)  # {'cat': 2, 'dog': 2, 'bird': 1}
```

💻 Example 3: Convert model metrics into dictionary

```
accuracy = 0.95
loss = 0.12
metrics = {"accuracy": accuracy, "loss": loss}
```

🟢 8. Advanced Dictionary Concepts

✅ Nested Dictionaries

```
students = {
    "S1": {"name": "Raj", "marks": 85},
    "S2": {"name": "Ravi", "marks": 90}
}
print(students["S2"]["name"])  # Ravi
```

✅ Defaultdict (from collections)

Used to avoid key errors and provide default value.

```
from collections import defaultdict

freq = defaultdict(int)
for word in ["AI", "ML", "AI"]:
    freq[word] += 1
print(freq)  # defaultdict(<class 'int'>, {'AI': 2, 'ML': 1})
```

✅ Counter (special type of dict)

Count occurrences easily.

```
from collections import Counter
labels = ["cat", "dog", "dog", "cat"]
print(Counter(labels))  # Counter({'cat':2, 'dog':2})
```

✅ OrderedDict

Maintains insertion order (normal dict does since Python 3.7+).

🟢 9. Common Interview Questions

1. **Difference between list and dictionary?**
   → List: ordered, uses index
   → Dict: unordered (till Python 3.6), uses key-value

2. **What happens if you use a mutable object as a key?**
   → Error (keys must be immutable, e.g. tuple, string, int)

3. **How to remove a key safely from dict?**
   → Use `.pop(key, None)` or `del dict[key]` if exists

4. **How to iterate dictionary in sorted order?**
   → Use `for k in sorted(dict): ...`

5. **What are dictionary comprehensions and why used?**
   → To create or filter dictionaries efficiently in one line.

🟢 10. Quick Revision Summary

Operation	Method	Example
Access	`get()`	`d.get('a')`
Keys	`keys()`	`d.keys()`
Values	`values()`	`d.values()`
Pairs	`items()`	`d.items()`
Add/Update	`update()`	`d.update({'x':1})`
Remove	`pop()`	`d.pop('key')`
Remove last	`popitem()`	`d.popitem()`
Copy	`copy()`	`d.copy()`
Clear all	`clear()`	`d.clear()`
Default value	`setdefault()`	`d.setdefault('x', 0)`

🔥 Pro Tip for AI/ML Interviews

You'll often see dictionary usage in:

- Dataset feature mapping

- Configuration files (YAML → Dict)
- TensorFlow/PyTorch model parameters
- JSON APIs (model metadata, results)

So practice:

```
import json
data = {"accuracy": 0.95, "loss": 0.12}
json_str = json.dumps(data)
print(json_str)  # '{"accuracy": 0.95, "loss": 0.12}'
```
