Everything about Python Dictionary

Creating

```
# Empty dictionary
dict1 = {}

# Dictionary with values
person = {"name": "Rudra",
          "age": 25, "city": "New York"}

# Using dict() constructor
dict2 = dict(name="Rudra", age=30,
             city="London")

# Dictionary with mixed keys
mixed = {1: "one", "two": 2, (3, 4): "tuple key"}
```

Accessing Characters

```
print(person["name"])  # Rudra
print(person.get("age"))  # 25
```

Editing

```
# Update value
person["age"] = 26

# Add new key-value pair
person["gender"] = "Female"

# 2d new dict
person['subjects'] = {'maths', 'Physic'}
```

Operations

```
# Check key existence
print("name" in person)  # True
print("food" not in person) # True

# Dictionary length
print(len(person))  # Number of key-value pairs

# Iterating over keys and values
for key, value in person.items():
    print(key, "->", value)
```

Deleting

```
del person["city"]  # Remove a key
person.pop("age")  # Remove and return value
person.popitem() # last pair deleted
person.clear()  # Empty dictionary
del person # whole dictionary deleted
```

zip()

```
keys = ["name", "age", "city"]
values = ["Alice", 25, "New York"]

dictionary = {k: v for k, v in zip(keys, values)}
print(dictionary)
# {'name': 'Rudra', 'age': 25, 'city': 'New York'}
```

Dictionary comprehension

Basic

```
# Squaring numbers from 1 to 5
squares = {x: x**2 for x in range(1, 6)}
print(squares)  # {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
```

Filtering

```
# Filtering even numbers from a dictionary
numbers = {x: x**2 for x in range(1, 11) if x % 2 == 0}
print(numbers)  # {2: 4, 4: 16, 6: 36, 8: 64, 10: 100}
```

Nested

```
nested = {x: {y: y**2 for y in range(1, 4)} for x in range(1, 3)}
print(nested)
# {1: {1: 1, 2: 4, 3: 9}, 2: {1: 1, 2: 4, 3: 9}}
```

Advanced Dictionary Features

Provides a default value for missing keys instead of throwing an error.

```
from collections import defaultdict

# Default type is list
dd = defaultdict(list)
dd["a"].append(1)
dd["b"].append(2)

print(dd)  # {'a': [1], 'b': [2]}
print(dd["c"])
# [] (returns default empty list instead of error)
```

Dictionary Functions

When/ Why	Function	Input	Output
Get value safely	get(key, default)	person.get("age", 0)	25 or 0 if missing
Get Keys	keys()	person.keys()	dict_keys(['name', ...])
Get Values	values()	person.values()	dict_values(['Rudra', ...])
Get key-values	items()	person.items()	dict_items([('name', ...])
Remove key	pop(key)	person.pop("age")	25 (returns value)
Remove last items	popitem	person.popitem()	('city', 'New York')
update dictionary	updat()	person.update({"age": 27})	Updates age to 27
Copy dictionary	copy()	new_dict = person.copy()	Creates a copy
Clear Dictionary	clear()	person.clear()	{} (empty dictionary)