



Lab Work – Dictionary Comprehension Tasks

Objective

The objective of this lab is to develop strong skills in Python dictionary comprehensions for efficient data mapping, filtering, and transformation. Through tasks involving power ratings, resistance classification, electrical calculations, temperature conversion, index mapping, and selective filtering, students learn how to create and manipulate dictionaries in a clean, optimized, and Pythonic way. These tasks are essential for handling structured data in engineering and AI programming.

Task 1: Map Device Names to Their Power Ratings

Reads two lists: device names and their power values

Pairs them together using dictionary comprehension

Creates a dictionary with device names as keys and power ratings as values

```
In [2]: devices = ["Fan", "Light", "AC", "Pump"]
power = [60, 15, 1800, 750]

device_power = {d: p for d, p in zip(devices, power)}
print(device_power)

{'Fan': 60, 'Light': 15, 'AC': 1800, 'Pump': 750}
```

Task 2: Categorize Resistances as “Low” or “High”

Reads resistance values

Assigns “Low” if value < 40, else “High”

Stores each result in a dictionary

```
In [3]: R = [5, 33, 47, 82, 10]

resistance_levels = {r: "Low" if r < 40 else "High" for r in R}
print(resistance_levels)

{5: 'Low', 33: 'Low', 47: 'High', 82: 'High', 10: 'Low'}
```

Task 3: Compute Current for Each Load ($I = P / V$)

Reads load values in watts

Uses $V = 200$ V

Computes current by dividing P by V

Stores load: current pairs in a dictionary

```
In [4]: loads = [60, 100, 150, 200]
voltage = 200

current = {load: load / voltage for load in loads}
print(current)

{60: 0.3, 100: 0.5, 150: 0.75, 200: 1.0}
```

Task 4: Convert Dictionary Values From °C to °F

Reads temperature values in Celsius

Converts each value to Fahrenheit using formula $F = C \times 9/5 + 32$

Creates a new dictionary with converted values

```
In [1]: temps_c = {"motor": 40, "transformer": 60, "room": 28}
temps_f = {k: (v * 9/5) + 32 for k, v in temps_c.items()}
print(temps_f)

{'motor': 104.0, 'transformer': 140.0, 'room': 82.4}
```

Task 5: Create Dict of Components With Index Keys

Reads component names

Uses enumerate() to assign indexes

Creates a dictionary {index: component}

```
In [2]: components = ["Resistor", "Capacitor", "Inductor", "Diode"]
comp_dict = {i: name for i, name in enumerate(components)}
print(comp_dict)

{0: 'Resistor', 1: 'Capacitor', 2: 'Inductor', 3: 'Diode'}
```

Task 6: Filter Only Devices With Power > 100W

Reads a dictionary of device: power

Selects only devices with power greater than 100 W

Creates a filtered dictionary

```
In [3]: device_power = {"Fan": 60, "Light": 15, "Heater": 1200, "Iron": 1000, "Router": 500}
high_power = {k: v for k, v in device_power.items() if v > 100}
print(high_power)

{'Heater': 1200, 'Iron': 1000}
```