***### Exercise 1: Create a Dictionary  
#1. Create a dictionary called `person` with the following key-value pairs:  
# - Name: "Alice"  
# - Age: 25  
#- City: "New York"***person = {  
 "Name": "Alice",  
 "Age": 25,  
 "City": "New York"  
}  
***#2. Print the dictionary.***print(person)  
  
***### Exercise 2: Access Dictionary Elements  
#1. Access the value of the `"Name"` key in the `person` dictionary and print it.***name\_value = person["Name"]  
print(name\_value)  
***#2. Access the value of the `"City"` key and print it.***city\_value = person["City"]  
print(city\_value)  
  
***### Exercise 3: Add and Modify Elements  
#1. Add a new key-value pair to the `person` dictionary: `"email": "alice@example.com"`.***person["email"] = "alice@example.com"  
***#2. Change the value of the `"Age"` key to 26.***person["Age"] = 26  
***#3. Print the modified dictionary.***print(person)  
  
***### Exercise 4: Remove Elements  
#1. Remove the `"City"` key from the `person` dictionary.***person.pop("City")  
***#2. Print the dictionary after removing the key.***print(person)  
  
***### Exercise 5: Check if a Key Exists  
#1. Check if the key `"email"` exists in the `person` dictionary. Print a message based on the result.***if "email" in person:  
 print("The key 'email' is present.")  
else:  
 print("The key 'email' is not present.")  
***#2. Check if the key `"phone"` exists in the dictionary. Print a message based on the result.***if "phone" in person:  
 print("The key 'phone' is present.")  
else:  
 print("The key 'phone' is not present.")  
 ***### Exercise 6: Loop Through a Dictionary  
#1. Iterate over the `person` dictionary and print each key-value pair.***for key, value in person.items():  
 print(key, ":", value)  
***#2. Iterate over the keys of the dictionary and print each key.***for key in person:  
 print(key)  
***#3. Iterate over the values of the dictionary and print each value.***for value in person.values():  
 print(value)  
  
***### Exercise 7: Nested Dictionary  
#1. Create a dictionary called `employees` where the keys are employee IDs (`101`, `102`, `103`) and the values are dictionaries containing employee details (like name and job title). Example structure:  
# ```python  
#employees = {  
#101: {"name": "Bob", "job": "Engineer"},  
#102: {"name": "Sue", "job": "Designer"},  
#103: {"name": "Tom", "job": "Manager"}  
#}  
#```***employees = {  
 101: {"name": "Bob", "job": "Engineer"},  
 102: {"name": "Sue", "job": "Designer"},  
 103: {"name": "Tom", "job": "Manager"}  
}  
***#2. Print the details of employee with ID `102`.***print(employees[102])  
***#3. Add a new employee with ID `104`, name `"Linda"`, and job `"HR"`.***employees[104] = {"name": "Linda", "job": "HR"}  
***#4. Print the updated dictionary.***print(employees)  
  
***### Exercise 8: Dictionary Comprehension  
#1. Create a dictionary comprehension that generates a dictionary where the keys are numbers from 1 to 5 and the values are the squares of the keys.***squares\_dict = {x: x\*\*2 for x in range(1, 5+1)}  
***#2. Print the generated dictionary.***print(squares\_dict)  
  
***### Exercise 9: Merge Two Dictionaries  
#1. Create two dictionaries:  
#```python  
#dict1 = {"a": 1, "b": 2}  
#dict2 = {"c": 3, "d": 4}  
#```***dict1 = {"a": 1, "b": 2}  
dict2 = {"c": 3, "d": 4}  
***#2. Merge `dict2` into `dict1` and print the result.***dict1.update(dict2)  
print(dict1)  
  
***### Exercise 10: Default Dictionary Values  
#1. Create a dictionary that maps letters to numbers: `{"a": 1, "b": 2, "c": 3}`.***letter\_to\_number = {"a": 1, "b": 2, "c": 3}  
***#2. Use the `get()` method to retrieve the value of key `"b"`.***value\_b = letter\_to\_number.get("b")  
print(value\_b)  
***#3. Use the `get()` method to try to retrieve the value of a non-existing key `"d"`, but provide a default value of `0` if the key is not found.***value\_d = letter\_to\_number.get("d", 0)  
print(value\_d)  
  
***### Exercise 11: Dictionary from Two Lists  
#1. Given two lists:  
#```python  
#keys = ["name", "age", "city"]  
#values = ["Eve", 29, "San Francisco"]  
#```***keys = ["name", "age", "city"]  
values = ["Eve", 29, "San Francisco"]  
***#2. Create a dictionary by pairing corresponding elements from the `keys` and `values` lists.***resulting\_dict = dict(zip(keys, values))  
***#3. Print the resulting dictionary.***print(resulting\_dict)  
  
***### Exercise 12: Count Occurrences of Words  
#1. Write a Python program that takes a sentence as input and returns a dictionary that counts the occurrences of each word in the sentence.  
#```python  
#sentence = "the quick brown fox jumps over the lazy dog the fox"  
#```***sentence = "the quick brown fox jumps over the lazy dog the fox"  
words = sentence.split()  
word\_count = {}  
for word in words:  
 if word in word\_count:  
 word\_count[word] += 1  
 else:  
 word\_count[word] = 1  
  
***#2. Print the dictionary showing word counts.***print(word\_count)