

completed_Dict_assignment_ANIL

September 28, 2025

1 Dictionary Assignments – Solved

Solutions to all 14 dictionary-based Python assignments.

1.0.1 Assignment 1: Creating and Accessing Dictionaries

Create a dictionary with the first 10 positive integers as keys and their squares as values.
dictionary.

```
[1]: d = {i: i**2 for i in range(1, 11)}  
     print(d)
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100}
```

1.0.2 Assignment 2: Accessing Dictionary Elements

Print the value of the key 5 and the keys of the dictionary created in Assignment 1.

```
[2]: d = {i: i**2 for i in range(1, 11)}  
     print("Value of key 5:", d[5])  
     print("Keys:", list(d.keys()))
```

```
Value of key 5: 25
```

```
Keys: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

1.0.3 Assignment 3: Dictionary Methods

Add a new key-value pair (11, 121) to the dictionary created in Assignment 1 and then remove the key-value pair with key 1.
Print the modified dictionary.

```
[3]: d = {i: i**2 for i in range(1, 11)}  
     d[11] = 121  
     del d[1]  
     print(d)
```

```
{2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121}
```

1.0.4 Assignment 4: Iterating Over Dictionaries

Iterate over the dictionary created in Assignment 1 and print each key-value pair.

```
[4]: d = {i: i**2 for i in range(1, 11)}
    for key, value in d.items():
        print(f"Key: {key}, Value: {value}")
```

```
Key: 1, Value: 1
Key: 2, Value: 4
Key: 3, Value: 9
Key: 4, Value: 16
Key: 5, Value: 25
Key: 6, Value: 36
Key: 7, Value: 49
Key: 8, Value: 64
Key: 9, Value: 81
Key: 10, Value: 100
```

1.0.5 Assignment 5: Dictionary Comprehensions

Create a new dictionary containing the cubes of the first 10 positive integers using a dictionary comprehension. Print the new dictionary.

```
[5]: cubes = {i: i**3 for i in range(1, 11)}
    print(cubes)
```

```
{1: 1, 2: 8, 3: 27, 4: 64, 5: 125, 6: 216, 7: 343, 8: 512, 9: 729, 10: 1000}
```

1.0.6 Assignment 6: Merging Dictionaries

Create two dictionaries, one with keys as the first 5 positive integers and values as their squares, and another with keys as the next 5 positive integers and values as their squares. Merge these dictionaries into a single dictionary and print it.

```
[6]: dict1 = {i: i**2 for i in range(1, 6)}
    dict2 = {i: i**2 for i in range(6, 11)}
    merged = {**dict1, **dict2}
    print(merged)
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100}
```

1.0.7 Assignment 7: Nested Dictionaries

Create a nested dictionary representing a student with keys 'name', 'age', 'grades', where 'grades' is another dictionary with keys 'math', 'science', and 'english'. Print the nested dictionary.

```
[7]: student = {
    'name': 'Alice',
    'age': 16,
    'grades': {
        'math': 92,
        'science': 88,
        'english': 85
    }
}
```

```

        'english': 95
    }
}
print(student)

```

```
{'name': 'Alice', 'age': 16, 'grades': {'math': 92, 'science': 88, 'english': 95}}
```

1.0.8 Assignment 8: Dictionary of Lists

Create a dictionary where the keys are the first 5 positive integers and the values are lists containing the first 5 multiples of the key. Print the dictionary.

```

[8]: d = {}
    for i in range(1, 6):
        d[i] = [i * j for j in range(1, 6)]
    print(d)

```

```
{1: [1, 2, 3, 4, 5], 2: [2, 4, 6, 8, 10], 3: [3, 6, 9, 12, 15], 4: [4, 8, 12, 16, 20], 5: [5, 10, 15, 20, 25]}
```

1.0.9 Assignment 9: Dictionary of Tuples

Create a dictionary where the keys are the first 5 positive integers and the values are tuples containing the key and its square. Print the dictionary.

```

[9]: d = {i: (i, i**2) for i in range(1, 6)}
    print(d)

```

```
{1: (1, 1), 2: (2, 4), 3: (3, 9), 4: (4, 16), 5: (5, 25)}
```

1.0.10 Assignment 10: Dictionary and List Conversion

Create a dictionary with the first 5 positive integers as keys and their squares as values. Convert the dictionary to a list of tuples and print it.

```

[10]: d = {i: i**2 for i in range(1, 6)}
    lst = list(d.items())
    print(lst)

```

```
[(1, 1), (2, 4), (3, 9), (4, 16), (5, 25)]
```

1.0.11 Assignment 11: Dictionary Filtering

Create a dictionary with the first 10 positive integers as keys and their squares as values. Create a new dictionary containing only the key-value pairs where the key is even. Print the new dictionary.

```

[11]: d = {i: i**2 for i in range(1, 11)}
    even_keys = {k: v for k, v in d.items() if k % 2 == 0}
    print(even_keys)

```

```
{2: 4, 4: 16, 6: 36, 8: 64, 10: 100}
```

1.0.12 Assignment 12: Dictionary Key and Value Transformation

Create a dictionary with the first 5 positive integers as keys and their squares as values. Create a new dictionary with keys and values swapped. Print the new dictionary.

```
[12]: d = {i: i**2 for i in range(1, 6)}  
      swapped = {v: k for k, v in d.items()}  
      print(swapped)
```

```
{1: 1, 4: 2, 9: 3, 16: 4, 25: 5}
```

1.0.13 Assignment 13: Default Dictionary

Create a default dictionary where each key has a default value of an empty list. Add some elements to the lists and print the dictionary.

```
[13]: d = {}  
      d.setdefault('fruits', []).append('apple')  
      d.setdefault('fruits', []).append('banana')  
      d.setdefault('colors', []).append('red')  
      print(d)
```

```
{'fruits': ['apple', 'banana'], 'colors': ['red']}
```

1.0.14 Assignment 14: Counting with Dictionaries

Write a function that takes a string and returns a dictionary with the count of each character in the string. Print the dictionary.

```
[15]: def count_chars(s):  
      d = {}  
      for char in s.lower():  
          d[char] = d.get(char, 0) + 1  
      return d  
  
      print(count_chars("hello"))
```

```
{'h': 1, 'e': 1, 'l': 2, 'o': 1}
```

```
[ ]:
```