

1. WAP in Python to sort (ascending and descending) a dictionary by value.

hint: import operator

```
d = {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
```

```
print('Original dictionary : ',d)
```

```
sorted_d = sorted(d.items(), key=operator.itemgetter(1))
```

```
print('Dictionary in ascending order by value : ',sorted_d)
```

2. WAP in Python to add a key to a dictionary.

Sample Dictionary : {0: 10, 1: 20}

Expected Result : {0: 10, 1: 20, 2: 30}

3. WAP in Python to concatenate following dictionaries to create a new one.

Sample Dictionary :

```
dic1={1:10, 2:20}
```

```
dic2={3:30, 4:40}
```

```
dic3={5:50,6:60}
```

Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

4. WAP in Python to check whether a given key already exists in a dictionary.

5. WAP in Python to iterate over dictionaries using for loops.

6. WAP in Python to generate and print a dictionary that contains a number (between 1 and n) in the form (x, x*x).

Sample Dictionary (n = 5) :

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

7. WAP in Python to print a dictionary where the keys are numbers between 1 and 15 (both included) and the values are square of keys.

Sample Dictionary

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144, 13: 169, 14: 196, 15: 225}
```

8. WAP in Python to merge two Python dictionaries.

9. WAP in Python to iterate over dictionaries using for loops.

10. WAP in Python to sum all the items in a dictionary.

11. WAP in Python to multiply all the items in a dictionary.

12. WAP in Python to remove a key from a dictionary.

13. WAP in Python to map two lists into a dictionary.

14. WAP in Python to sort a given dictionary by key.

15. WAP in Python to get the maximum and minimum value in a dictionary.

16. WAP in Python to get a dictionary from an object's fields.

17. WAP in Python to remove duplicates from Dictionary.

18. WAP in Python to check a dictionary is empty or not.

19. WAP in Python to combine two dictionary adding values for common keys.

```
d1 = {'a': 100, 'b': 200, 'c': 300}
```

```
d2 = {'a': 300, 'b': 200, 'd': 400}
```

Sample output: Counter({'a': 400, 'b': 400, 'd': 400, 'c': 300})

20. WAP in Python to print all unique values in a dictionary.

Sample Data : [{"V": "S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"}, {"VII": "S005"}, {"V": "S009"}, {"VIII": "S007"}]

Expected Output : Unique Values: {'S005', 'S002', 'S007', 'S001', 'S009'}

21. WAP in Python to create and display all combinations of letters, selecting each letter from a different key in a dictionary.

Sample data : {'1':['a','b'], '2':['c','d']}

Expected Output:

ac

ad

bc

bd

22. WAP in Python to find the highest 3 values of corresponding keys in a dictionary.

23. WAP in Python to combine values in python list of dictionaries.

Sample data: [{'item': 'item1', 'amount': 400}, {'item': 'item2', 'amount': 300}, {'item': 'item1', 'amount': 750}]

Expected Output: Counter({'item1': 1150, 'item2': 300})

24. WAP in Python to create a dictionary from a string.

Note: Track the count of the letters from the string.

Sample string : 'w3resource'

Expected output: {'w': 1, '3': 1, 'r': 2, 'e': 2, 's': 1, 'o': 1, 'u': 1, 'c': 1}

25. WAP in Python to print a dictionary in table format.

26. WAP in Python to count the values associated with key in a dictionary.

Expected Output:

6

2

27. WAP in Python to convert a list into a nested dictionary of keys.

28. WAP in Python to sort a list alphabetically in a dictionary.

29. WAP in Python to remove spaces from dictionary keys.

30. WAP in Python to get the top three items in a shop.

Sample data: {'item1': 45.50, 'item2':35, 'item3': 41.30, 'item4':55, 'item5': 24}

Expected Output:

item4 55

item1 45.5

item3 41.3

31. WAP in Python to get the key, value and item in a dictionary.

32. WAP in Python to print a dictionary line by line.

33. WAP in Python to check multiple keys exists in a dictionary.

34. WAP in Python to count number of items in a dictionary value that is a list.

35. WAP in Python to sort Counter by value.

Sample data : {'Math':81, 'Physics':83, 'Chemistry':87}

Expected data: [('Chemistry', 87), ('Physics', 83), ('Math', 81)]

36. WAP in Python to create a dictionary from two lists without losing duplicate values.

Sample lists: ['Class-V', 'Class-VI', 'Class-VII', 'Class-VIII'], [1, 2, 2, 3]

Expected Output: defaultdict(<class 'set'>, {'Class-V': {1}, 'Class-VI': {2}, 'Class-VII': {2}, 'Class-VIII': {3}})

37. WAP in Python to replace dictionary values with their average.

38. WAP in Python to match key values in two dictionaries.

Sample dictionary: {'key1': 1, 'key2': 3, 'key3': 2}, {'key1': 1, 'key2': 2}

Expected output: key1: 1 is present in both x and y

39. WAP in Python to store a given dictionary in a json file.

Original dictionary:

```
{'students': [{'firstName': 'Nikki', 'lastName': 'Roysden'}, {'firstName': 'Mervin', 'lastName': 'Friedland'},  
{'firstName': 'Aron ', 'lastName': 'Wilkins'}], 'teachers': [{'firstName': 'Amberly', 'lastName': 'Calico'},  
{'firstName': 'Regine', 'lastName': 'Agtarap'}]}
```

<class 'dict'>

Json file to dictionary:

```
{'students': [{'firstName': 'Nikki', 'lastName': 'Roysden'}, {'firstName': 'Mervin', 'lastName': 'Friedland'},  
{'firstName': 'Aron ', 'lastName': 'Wilkins'}], 'teachers': [{'firstName': 'Amberly', 'lastName': 'Calico'},  
{'firstName': 'Regine', 'lastName': 'Agtarap'}]}
```

40. WAP in Python to create a dictionary of keys x, y, and z where each key has as value a list from 11-20, 21-30, and 31-40 respectively. Access the fifth value of each key from the dictionary.

```
{'x': [11, 12, 13, 14, 15, 16, 17, 18, 19],
```

```
'y': [21, 22, 23, 24, 25, 26, 27, 28, 29],
```

```
'z': [31, 32, 33, 34, 35, 36, 37, 38, 39]}
```

15

25

35

x has value [11, 12, 13, 14, 15, 16, 17, 18, 19]

y has value [21, 22, 23, 24, 25, 26, 27, 28, 29]

z has value [31, 32, 33, 34, 35, 36, 37, 38, 39]

41. WAP in Python to drop empty Items from a given Dictionary.

Original Dictionary:

```
{'c1': 'Red', 'c2': 'Green', 'c3': None}
```

New Dictionary after dropping empty items:

```
{'c1': 'Red', 'c2': 'Green'}
```

42. WAP in Python to filter a dictionary based on values.

Original Dictionary:

```
{'Cierra Vega': 175, 'Alden Cantrell': 180, 'Kierra Gentry': 165, 'Pierre Cox': 190}
```

Marks greater than 170:

```
{'Cierra Vega': 175, 'Alden Cantrell': 180, 'Pierre Cox': 190}
```

43. WAP in Python to convert more than one list to nested dictionary.

Original strings:

```
['S001', 'S002', 'S003', 'S004']
```

```
['Adina Park', 'Leyton Marsh', 'Duncan Boyle', 'Saim Richards']
```

```
[85, 98, 89, 92]
```

Nested dictionary:

```
{'S001': {'Adina Park': 85}}, {'S002': {'Leyton Marsh': 98}}, {'S003': {'Duncan Boyle': 89}}, {'S004': {'Saim Richards': 92}}
```

44. WAP in Python to filter the height and width of students, which are stored in a dictionary.

Original Dictionary:

```
{'Cierra Vega': (6.2, 70), 'Alden Cantrell': (5.9, 65), 'Kierra Gentry': (6.0, 68), 'Pierre Cox': (5.8, 66)}
```

Height > 6ft and Weight > 70kg:

```
{'Cierra Vega': (6.2, 70)}
```

45. WAP in Python to check all values are same in a dictionary.

Original Dictionary:

```
{'Cierra Vega': 12, 'Alden Cantrell': 12, 'Kierra Gentry': 12, 'Pierre Cox': 12}
```

Check all are 12 in the dictionary.

True

Check all are 10 in the dictionary.

False

46. WAP in Python to create a dictionary grouping a sequence of key-value pairs into a dictionary of lists.

Original list:

```
[('yellow', 1), ('blue', 2), ('yellow', 3), ('blue', 4), ('red', 1)]
```

Grouping a sequence of key-value pairs into a dictionary of lists:

```
{'yellow': [1, 3], 'blue': [2, 4], 'red': [1]}
```

47. WAP in Python to split a given dictionary of lists into list of dictionaries.

Original dictionary of lists:

```
{'Science': [88, 89, 62, 95], 'Language': [77, 78, 84, 80]}
```

Split said dictionary of lists into list of dictionaries:

```
[{'Science': 88, 'Language': 77}, {'Science': 89, 'Language': 78}, {'Science': 62, 'Language': 84},  
{ 'Science': 95, 'Language': 80}]
```

48. WAP in Python to remove a specified dictionary from a given list.

Original list of dictionary:

```
[{'id': '#FF0000', 'color': 'Red'}, {'id': '#800000', 'color': 'Maroon'}, {'id': '#FFFF00', 'color': 'Yellow'}, {'id':  
'#808000', 'color': 'Olive'}]
```

Remove id #FF0000 from the said list of dictionary:

```
[{'id': '#800000', 'color': 'Maroon'}, {'id': '#FFFF00', 'color': 'Yellow'}, {'id': '#808000', 'color': 'Olive'}]
```

49. WAP in Python to convert string values of a given dictionary, into integer/float datatypes.

Original list:

```
[{'x': '10', 'y': '20', 'z': '30'}, {'p': '40', 'q': '50', 'r': '60'}]
```

String values of a given dictionary, into integer types:

```
[{'x': 10, 'y': 20, 'z': 30}, {'p': 40, 'q': 50, 'r': 60}]
```

Original list:

```
[{'x': '10.12', 'y': '20.23', 'z': '30'}, {'p': '40.00', 'q': '50.19', 'r': '60.99'}]
```

String values of a given dictionary, into float types:

```
[{'x': 10.12, 'y': 20.23, 'z': 30.0}, {'p': 40.0, 'q': 50.19, 'r': 60.99}]
```

50. A Python Dictionary contains List as value. WAP in Python to clear the list values in the said dictionary.

Original Dictionary:

```
{'C1': [10, 20, 30], 'C2': [20, 30, 40], 'C3': [12, 34]}
```

Clear the list values in the said dictionary:

```
{'C1': [], 'C2': [], 'C3': []}
```

51. A Python Dictionary contains List as value. WAP in Python to update the list values in the said dictionary.

Original Dictionary:

```
{'Math': [88, 89, 90], 'Physics': [92, 94, 89], 'Chemistry': [90, 87, 93]}
```

Update the list values of the said dictionary:

```
{'Math': [89, 90, 91], 'Physics': [90, 92, 87], 'Chemistry': [90, 87, 93]}
```

52. WAP in Python to extract a list of values from a given list of dictionaries.

Original Dictionary:

```
[{'Math': 90, 'Science': 92}, {'Math': 89, 'Science': 94}, {'Math': 92, 'Science': 88}]
```

Extract a list of values from said list of dictionaries where subject = Science

```
[92, 94, 88]
```

Original Dictionary:

```
[{'Math': 90, 'Science': 92}, {'Math': 89, 'Science': 94}, {'Math': 92, 'Science': 88}]
```

Extract a list of values from said list of dictionaries where subject = Math

```
[90, 89, 92]
```

53. WAP in Python to find the length of a given dictionary values.

Original Dictionary:

```
{1: 'red', 2: 'green', 3: 'black', 4: 'white', 5: 'black'}
```

Length of dictionary values:

```
{'red': 3, 'green': 5, 'black': 5, 'white': 5}
```

Original Dictionary:

```
{1: 'Austin Little', 2: 'Natasha Howard', 3: 'Alfred Mullins', 4: 'Jamie Rowe'}
```

Length of dictionary values:

```
{'Austin Little': 13, 'Natasha Howard': 14, 'Alfred Mullins': 14, 'Jamie Rowe': 10}
```

54. WAP in Python to get the depth of a dictionary.

Expected Output:

4

55. WAP in Python to access dictionary key's element by index.

Expected Output:

physics

math

chemistry

56. WAP in Python to convert a given dictionary into a list of lists.

Original Dictionary:

```
{1: 'red', 2: 'green', 3: 'black', 4: 'white', 5: 'black'}
```

Convert the said dictionary into a list of lists:

```
[[1, 'red'], [2, 'green'], [3, 'black'], [4, 'white'], [5, 'black']]
```

Original Dictionary:

```
{'1': 'Austin Little', '2': 'Natasha Howard', '3': 'Alfred Mullins', '4': 'Jamie Rowe'}
```

Convert the said dictionary into a list of lists:

```
[['1', 'Austin Little'], ['2', 'Natasha Howard'], ['3', 'Alfred Mullins'], ['4', 'Jamie Rowe']]
```

57. WAP in Python to filter even numbers from a given dictionary values.

Original Dictionary:

```
{'V': [1, 4, 6, 10], 'VI': [1, 4, 12], 'VII': [1, 3, 8]}
```

Filter even numbers from said dictionary values:

```
{'V': [4, 6, 10], 'VI': [4, 12], 'VII': [8]}
```

Original Dictionary:

```
{'V': [1, 3, 5], 'VI': [1, 5], 'VII': [2, 7, 9]}
```

Filter even numbers from said dictionary values:

```
{'V': [], 'VI': [], 'VII': [2]}
```

58. WAP in Python to get all combinations of key-value pairs in a given dictionary.

Original Dictionary:

```
{'V': [1, 4, 6, 10], 'VI': [1, 4, 12], 'VII': [1, 3, 8]}
```

Combinations of key-value pairs of the said dictionary:

```
[{'V': [1, 4, 6, 10], 'VI': [1, 4, 12]}, {'V': [1, 4, 6, 10], 'VII': [1, 3, 8]}, {'VI': [1, 4, 12], 'VII': [1, 3, 8]}]
```

Original Dictionary:

```
{'V': [1, 3, 5], 'VI': [1, 5]}
```

Combinations of key-value pairs of the said dictionary:

```
[{'V': [1, 3, 5], 'VI': [1, 5]}]
```

59. WAP in Python to find the specified number of maximum values in a given dictionary.

Original Dictionary:

```
{'a': 5, 'b': 14, 'c': 32, 'd': 35, 'e': 24, 'f': 100, 'g': 57, 'h': 8, 'i': 100}
```

1 maximum value(s) in the said dictionary:

```
['f']
```

2 maximum value(s) in the said dictionary:

```
['f', 'i']
```

5 maximum value(s) in the said dictionary:

```
['f', 'i', 'g', 'd', 'c']
```

60. WAP in Python to find shortest list of values with the keys in a given dictionary.

Original Dictionary: {'V': [10, 12], 'VI': [10], 'VII': [10, 20, 30, 40], 'VIII': [20], 'IX': [10, 30, 50, 70], 'X': [80]}

Shortest list of values with the keys of the said dictionary: ['VI', 'VIII', 'X']

61. WAP in Python to count the frequency in a given dictionary.

Original Dictionary:

```
{'V': 10, 'VI': 10, 'VII': 40, 'VIII': 20, 'IX': 70, 'X': 80, 'XI': 40, 'XII': 20}
```

Count the frequency of the said dictionary:

```
Counter({10: 2, 40: 2, 20: 2, 70: 1, 80: 1})
```

62. WAP in Python to extract values from a given dictionaries and create a list of lists from those values.

Original Dictionary:

```
{'student_id': 1, 'name': 'Jean Castro', 'class': 'V'}, {'student_id': 2, 'name': 'Lula Powell', 'class': 'V'},  
{ 'student_id': 3, 'name': 'Brian Howell', 'class': 'VI'}, {'student_id': 4, 'name': 'Lynne Foster', 'class': 'VI'},  
{ 'student_id': 5, 'name': 'Zachary Simon', 'class': 'VII'}}
```

Extract values from the said dictionary and create a list of lists using those values:

```
[[1, 'Jean Castro', 'V'], [2, 'Lula Powell', 'V'], [3, 'Brian Howell', 'VI'], [4, 'Lynne Foster', 'VI'], [5, 'Zachary  
Simon', 'VII']]
```

```
[[1, 'Jean Castro'], [2, 'Lula Powell'], [3, 'Brian Howell'], [4, 'Lynne Foster'], [5, 'Zachary Simon']]
```

```
[['Jean Castro', 'V'], ['Lula Powell', 'V'], ['Brian Howell', 'VI'], ['Lynne Foster', 'VI'], ['Zachary Simon', 'VII']]
```

63. WAP in Python to convert a given list of lists to a dictionary.

Original list of lists:

```
[[1, 'Jean Castro', 'V'], [2, 'Lula Powell', 'V'], [3, 'Brian Howell', 'VI'], [4, 'Lynne Foster', 'VI'], [5, 'Zachary  
Simon', 'VII']]
```

Convert the said list of lists to a dictionary:

```
{1: ['Jean Castro', 'V'], 2: ['Lula Powell', 'V'], 3: ['Brian Howell', 'VI'], 4: ['Lynne Foster', 'VI'], 5: ['Zachary  
Simon', 'VII']}
```

64. WAP in Python to create a key-value list pairings in a given dictionary.

Original dictionary:

```
{1: ['Jean Castro'], 2: ['Lula Powell'], 3: ['Brian Howell'], 4: ['Lynne Foster'], 5: ['Zachary Simon']}
```

A key-value list pairings of the said dictionary:

```
[{1: 'Jean Castro', 2: 'Lula Powell', 3: 'Brian Howell', 4: 'Lynne Foster', 5: 'Zachary Simon'}]
```

65. WAP in Python to get the total length of all values of a given dictionary with string values.

Original dictionary:

```
{'#FF0000': 'Red', '#800000': 'Maroon', '#FFFF00': 'Yellow', '#808000': 'Olive'}
```

Total length of all values of the said dictionary with string values:

20

66. WAP in Python to check if a specific Key and a value exist in a dictionary.

Original dictionary:

```
[{'student_id': 1, 'name': 'Jean Castro', 'class': 'V'}, {'student_id': 2, 'name': 'Lula Powell', 'class': 'V'},  
{ 'student_id': 3, 'name': 'Brian Howell', 'class': 'VI'}, {'student_id': 4, 'name': 'Lynne Foster', 'class': 'VI'},  
{ 'student_id': 5, 'name': 'Zachary Simon', 'class': 'VII'}]
```

Check if a specific Key and a value exist in the said dictionary:

True

True

True

False

False

False

67. WAP in Python to invert a given dictionary with non-unique hashable values.

Sample Output:

```
{8: ['Ora Mckinney', 'Mathew Gilbert'], 7: ['Theodore Hollandl', 'Mae Fleming', 'Ivan Little']}
```

68. WAP in Python to combines two or more dictionaries, creating a list of values for each key.

Sample Output:

Original dictionaries:

```
{'w': 50, 'x': 100, 'y': 'Green', 'z': 400}
```

```
{'x': 300, 'y': 'Red', 'z': 600}
```

Combined dictionaries, creating a list of values for each key:

```
{'w': [50], 'x': [100, 300], 'y': ['Green', 'Red'], 'z': [400, 600]}
```

69. WAP in Python to group the elements of a given list based on the given function.

Sample Output:

Original list & function:

```
[7, 23, 3.2, 3.3, 8.4] Function name: floor:
```

Group the elements of the said list based on the given function:

```
{7: [7], 23: [23], 3: [3.2, 3.3], 8: [8.4]}
```

Original list & function:

```
['Red', 'Green', 'Black', 'White', 'Pink'] Function name: len:
```

Group the elements of the said list based on the given function:

```
{3: ['Red'], 5: ['Green', 'Black', 'White'], 4: ['Pink']}
```

70. WAP in Python to map the values of a given list to a dictionary using a function, where the key-value pairs consist of the original value as the key and the result of the function as the value.

Sample Output:

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

71. WAP in Python to retrieve the value of the nested key indicated by the given selector list from a dictionary or list.

Sample Output:

Russell

2

72. WAP in Python to invert a dictionary with unique hashable values.

Sample Output:

```
{10: 'Theodore', 11: 'Mathew', 9: 'Roxanne'}
```

73. WAP in Python to convert a list of dictionaries into a list of values corresponding to the specified key.

Sample Output:

Original list of dictionaries:

```
[{'name': 'Theodore', 'age': 18}, {'name': 'Mathew', 'age': 22}, {'name': 'Roxanne', 'age': 20}, {'name': 'David', 'age': 18}]
```

Convert a list of dictionaries into a list of values corresponding to the specified key:

```
[18, 22, 20, 18]
```

74. WAP in Python to create a dictionary with the same keys as the given dictionary and values generated by running the given function for each value.

Sample Output:

Original dictionary elements:

```
{'Theodore': {'user': 'Theodore', 'age': 45}, 'Roxanne': {'user': 'Roxanne', 'age': 15}, 'Mathew': {'user': 'Mathew', 'age': 21}}
```

Dictionary with the same keys:

```
{'Theodore': 45, 'Roxanne': 15, 'Mathew': 21}
```

75. WAP in Python to find all keys in the provided dictionary that have the given value.

Sample Output:

Original dictionary elements:

```
{'Theodore': 19, 'Roxanne': 20, 'Mathew': 21, 'Betty': 20}
```

Find all keys in the said dictionary that have the specified value:

```
['Roxanne', 'Betty']
```

76. WAP in Python to combine two lists into a dictionary, where the elements of the first one serve as the keys and the elements of the second one serve as the values. The values of the first list need to be unique and hashable.

Sample Output:

Original lists:

```
['a', 'b', 'c', 'd', 'e', 'f']
```

```
[1, 2, 3, 4, 5]
```

Combine the values of the said two lists into a dictionary:

```
{'a': 1, 'b': 2, 'c': 3, 'd': 4, 'e': 5}
```

77. WAP in Python to convert given a dictionary to a list of tuples.

Sample Output:

Original Dictionary:

```
{'Red': 1, 'Green': 3, 'White': 5, 'Black': 2, 'Pink': 4}
```

Convert the said dictionary to a list of tuples:

```
[('Red', 1), ('Green', 3), ('White', 5), ('Black', 2), ('Pink', 4)]
```

78. WAP in Python to create a flat list of all the keys in a flat dictionary.

Sample Output:

Original dictionary elements:

```
{'Theodore': 19, 'Roxanne': 20, 'Mathew': 21, 'Betty': 20}
```

Create a flat list of all the keys of the said flat dictionary:

```
['Theodore', 'Roxanne', 'Mathew', 'Betty']
```

79. WAP in Python to create a flat list of all the values in a flat dictionary.

Sample Output:

Original dictionary elements:

```
{'Theodore': 19, 'Roxanne': 20, 'Mathew': 21, 'Betty': 20}
```

Create a flat list of all the values of the said flat dictionary:

[19, 20, 21, 20]

80. WAP in Python to find the key of the maximum value in a dictionary.

Sample Output:

Original dictionary elements:

{'Theodore': 19, 'Roxanne': 22, 'Mathew': 21, 'Betty': 20}

Finds the key of the maximum and minimum value of the said dictionary:

('Roxanne', 'Theodore')