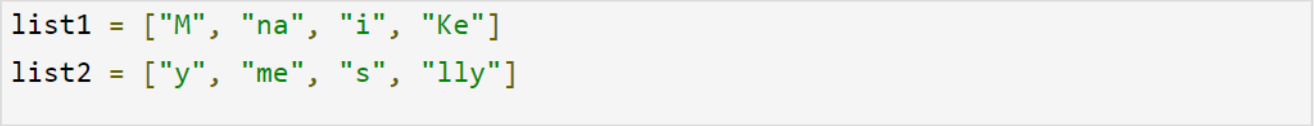
Python. Practice 1

NOTE: Don’t forget that each topic block should be written in a single .ipynb (.py) script with description of input, output and a small explanation of what is happening!

General Hint: Build-in Methods

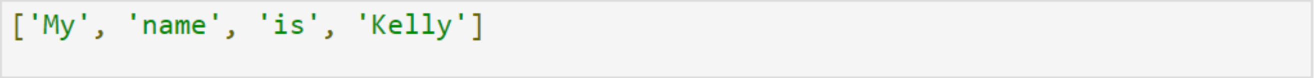
Topics: Lists

1. Given a Python list. Turn every item of a list into its square root
2. Concatenate two lists with string items index-wise



```
list1 = ["M", "na", "i", "Ke"]
list2 = ["y", "me", "s", "lly"]
```

Expected output:

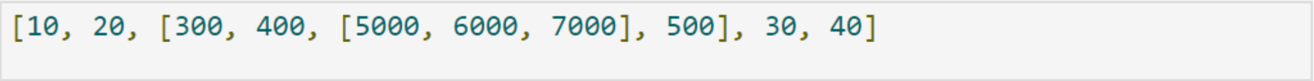


```
['My', 'name', 'is', 'Kelly']
```

3. Given a two Python list. Iterate both lists simultaneously such that list1 should display item in original order and list2 in reverse order
4. Remove empty strings from the list of strings
5. Add item 7000 after 6000 in the following Python List

```
list1 = [10, 20, [300, 400, [5000, 6000], 500], 30, 40]
```

Expected output:

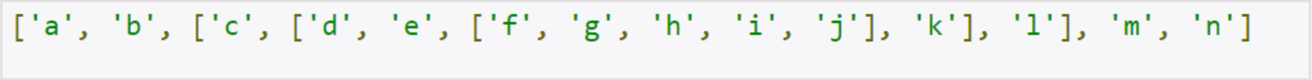


```
[10, 20, [300, 400, [5000, 6000, 7000], 500], 30, 40]
```

6. Given a nested list extend it with adding sub list ["h", "i", "j"] in a such a way that it will look like the following list

```
list1 = ['a', 'b', ['c', ['d', 'e', ['f', 'g'], 'k'], 'l'], 'm', 'n']
```

Expected output:

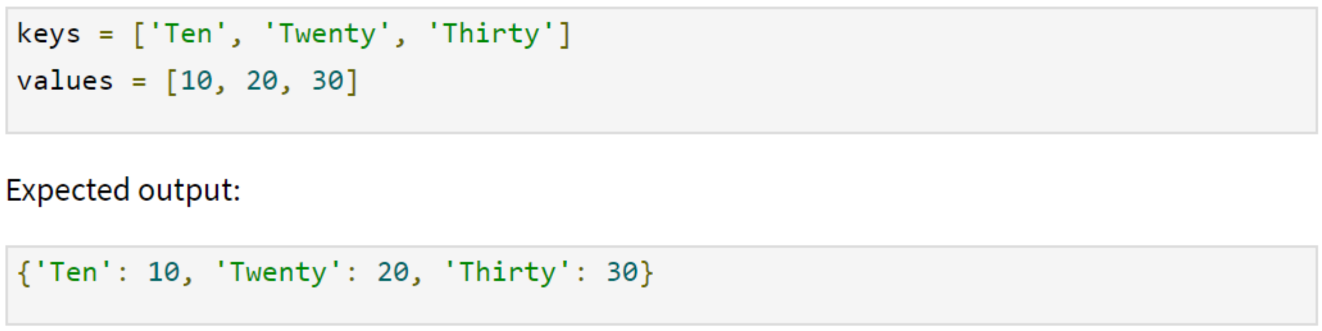


```
['a', 'b', ['c', ['d', 'e', ['f', 'g', 'h', 'i', 'j'], 'k'], 'l'], 'm', 'n']
```

7. Given a Python list, find value 20 in the list, and if it is present, replace it with 200. Only update the first occurrence of a value
8. Given a Python list, remove all occurrence of 20 from the list

Topics: Dictionaries

1. Below are the two lists convert it into the dictionary

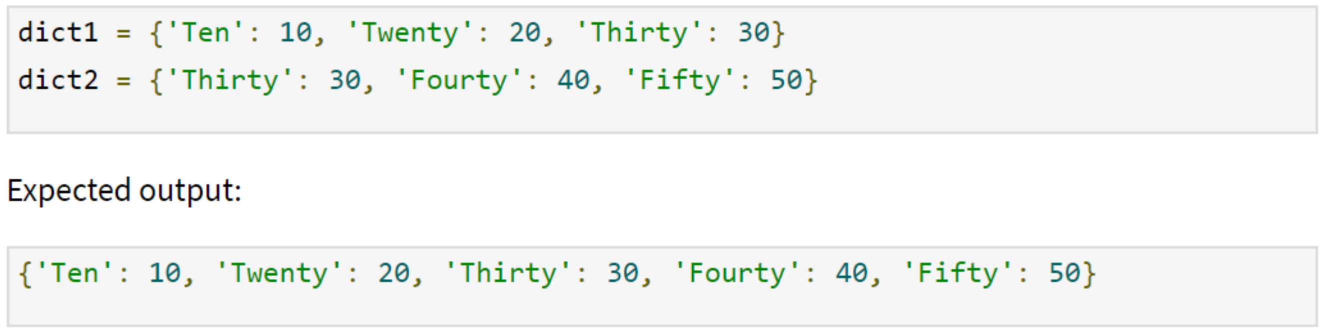


```
keys = ['Ten', 'Twenty', 'Thirty']
values = [10, 20, 30]
```

Expected output:

```
{'Ten': 10, 'Twenty': 20, 'Thirty': 30}
```

2. Merge following two Python dictionaries into one



```
dict1 = {'Ten': 10, 'Twenty': 20, 'Thirty': 30}
dict2 = {'Thirty': 30, 'Fourty': 40, 'Fifty': 50}
```

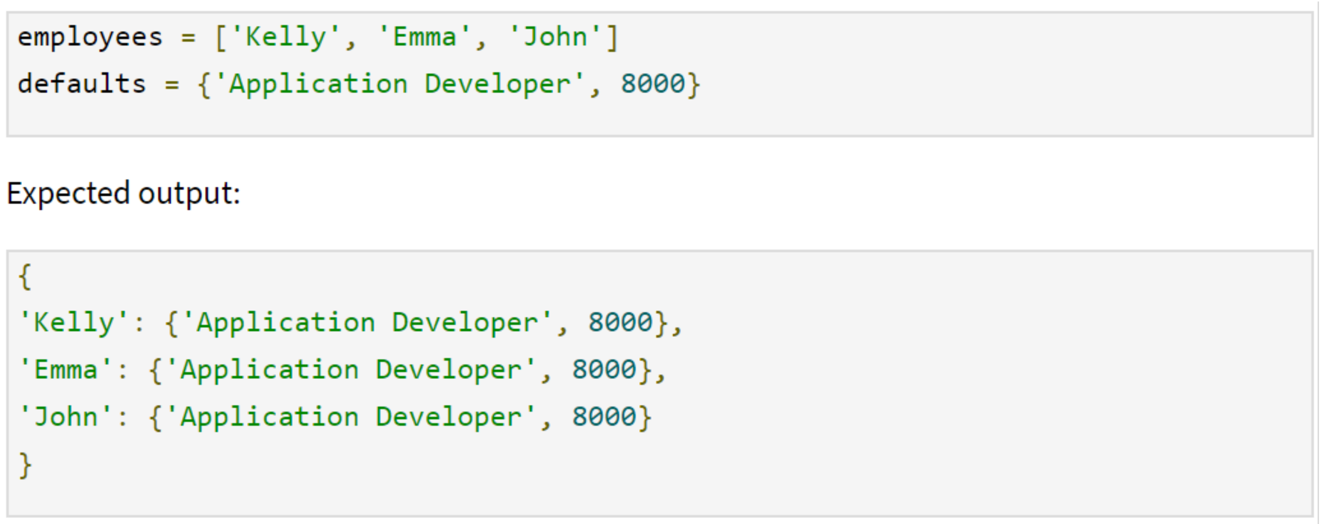
Expected output:

```
{'Ten': 10, 'Twenty': 20, 'Thirty': 30, 'Fourty': 40, 'Fifty': 50}
```

3. Access the value of key ‘history’

```
sampleDict = {
   "class":{
      "student":{
         "name":"Mike",
         "marks":{
            "physics":70,
            "history":80
         }
      }
   }
}
```

4. Initialize dictionary with default values

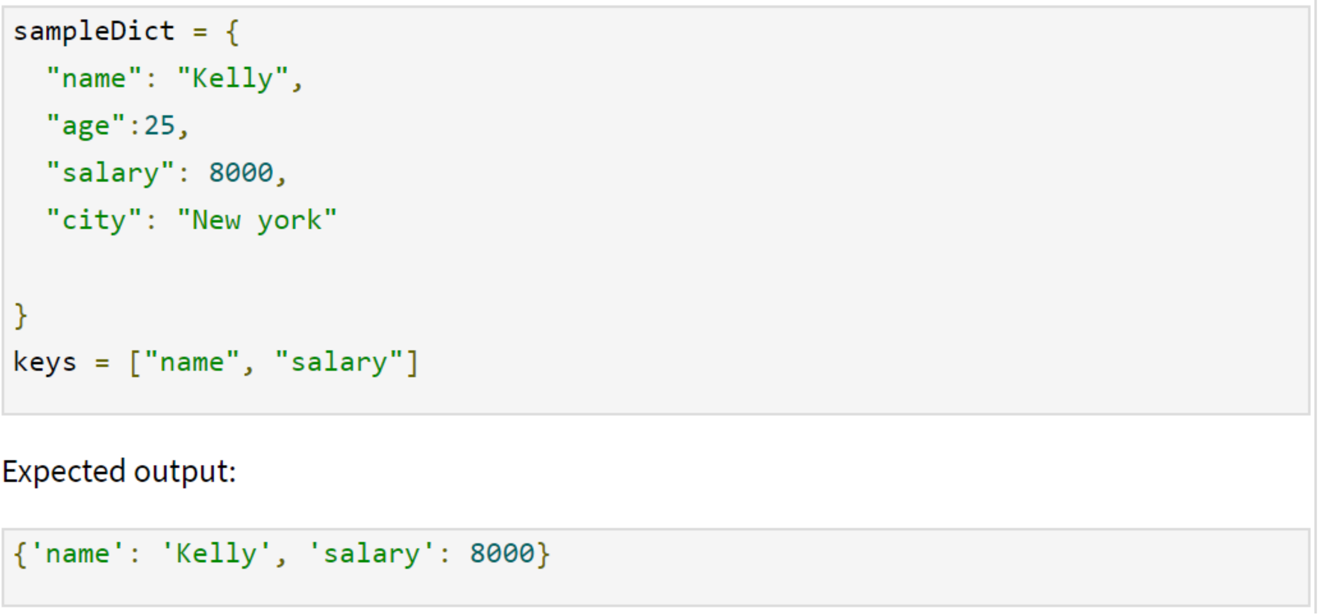


```
employees = ['Kelly', 'Emma', 'John']
defaults = {'Application Developer', 8000}
```

Expected output:

```
{
'Kelly': {'Application Developer', 8000},
'Emma': {'Application Developer', 8000},
'John': {'Application Developer', 8000}
}
```

5. Create a new dictionary by extracting following keys from a given dictionary



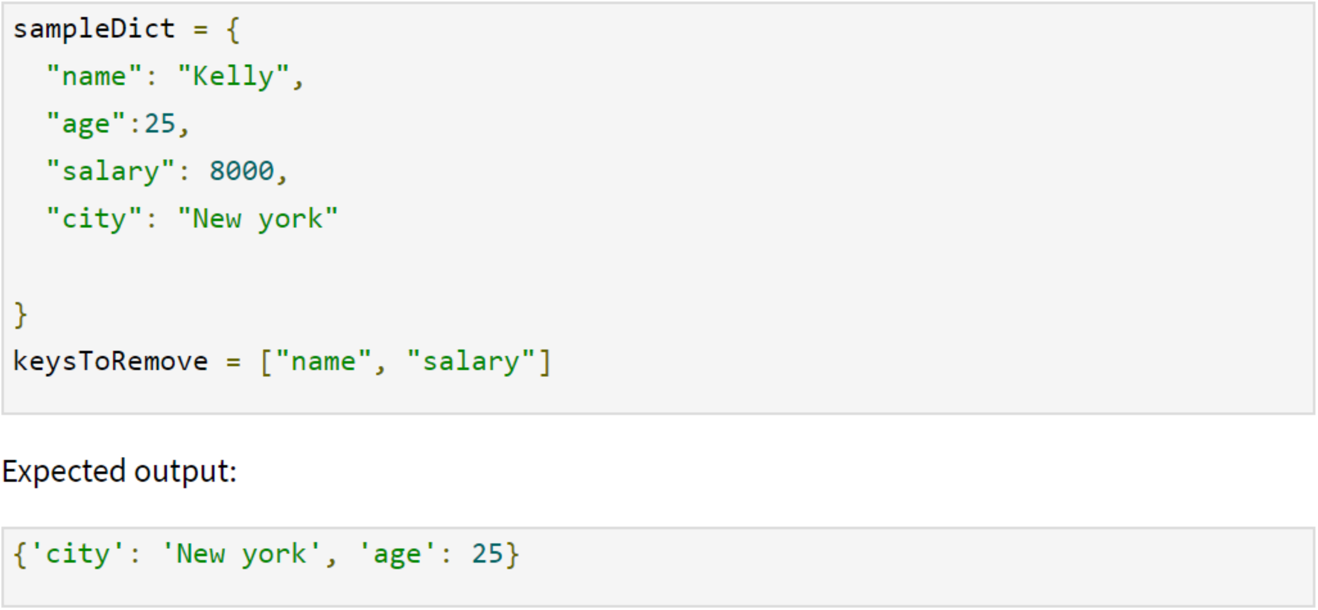
```
sampleDict = {
  "name": "Kelly",
  "age":25,
  "salary": 8000,
  "city": "New york"

}
keys = ["name", "salary"]
```

Expected output:

```
{'name': 'Kelly', 'salary': 8000}
```

6. Delete set of keys from Python Dictionary



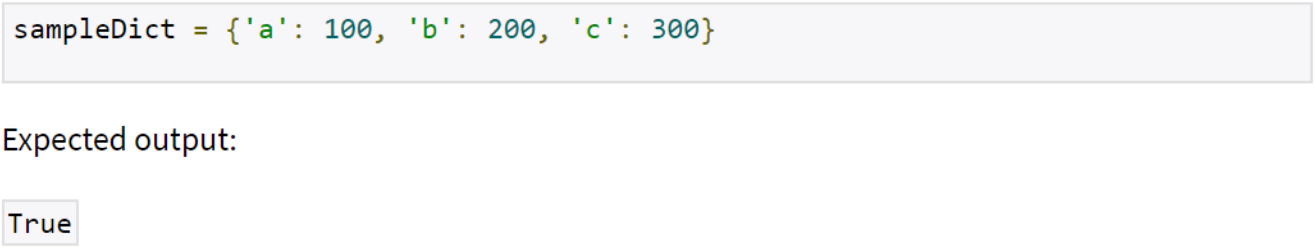
```
sampleDict = {
  "name": "Kelly",
  "age":25,
  "salary": 8000,
  "city": "New york"

}
keysToRemove = ["name", "salary"]
```

Expected output:

```
{'city': 'New york', 'age': 25}
```

7. Check if a value 200 exists in a dictionary

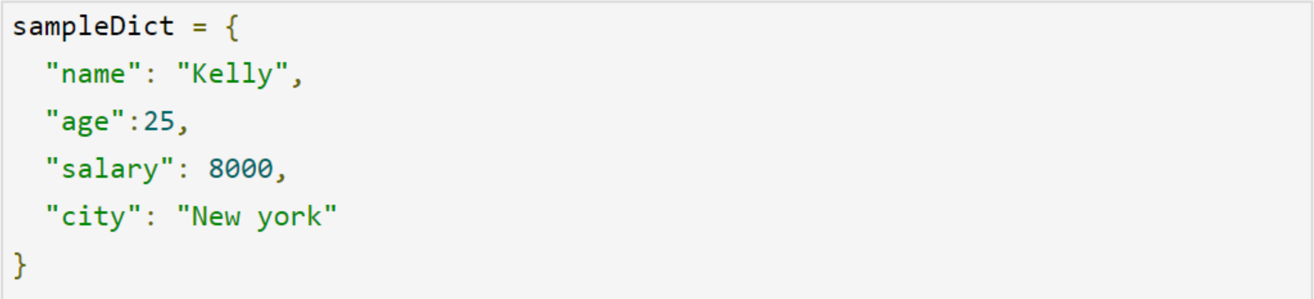


```
sampleDict = {'a': 100, 'b': 200, 'c': 300}
```

Expected output:

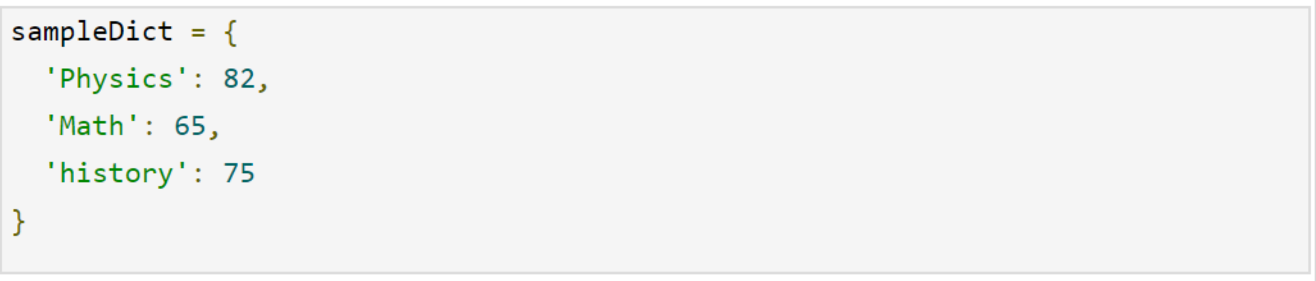
```
True
```

8. Rename key ‘city’ to ‘location’ in the following dictionary



```
sampleDict = {
  "name": "Kelly",
  "age":25,
  "salary": 8000,
  "city": "New york"
}
```

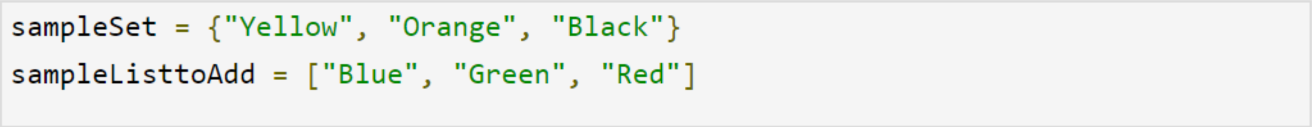
9. Get the key corresponding to the minimum value from the following dictionary



```
sampleDict = {
  'Physics': 82,
  'Math': 65,
  'history': 75
}
```

Topics: Sets

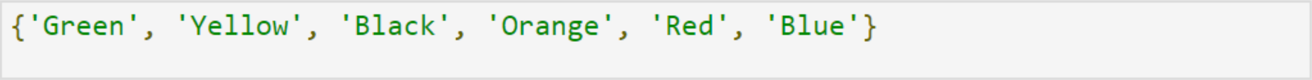
1. Add a list of elements to a given set



```
sampleSet = {"Yellow", "Orange", "Black"}
sampleListtoAdd = ["Blue", "Green", "Red"]
```

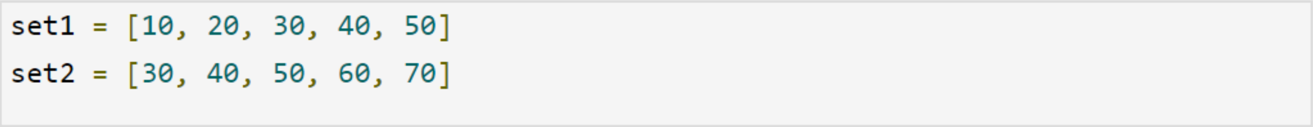
Expected output:

In set item order is not a concern



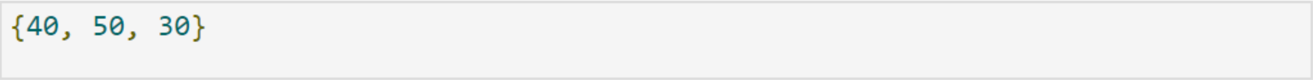
```
{'Green', 'Yellow', 'Black', 'Orange', 'Red', 'Blue'}
```

2. Return a set of identical items from a given two Python set



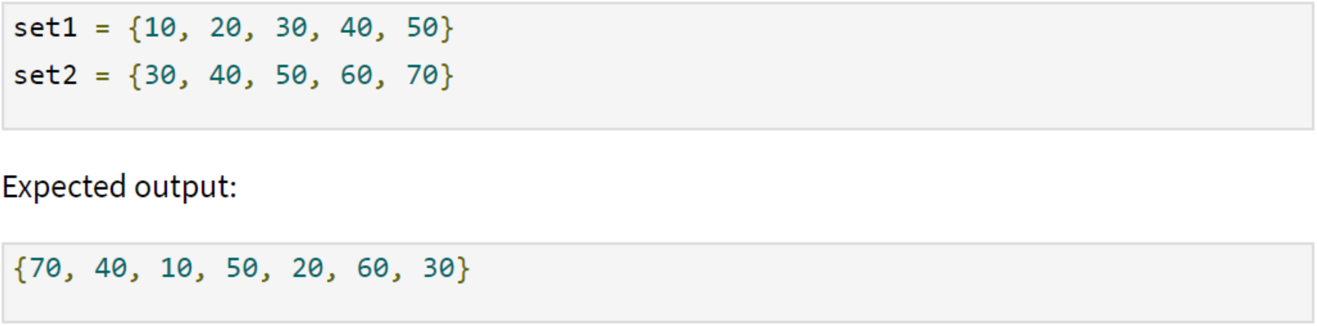
```
set1 = [10, 20, 30, 40, 50]
set2 = [30, 40, 50, 60, 70]
```

Expected output:



```
{40, 50, 30}
```

3. Returns a new set with all items from both sets by removing duplicates

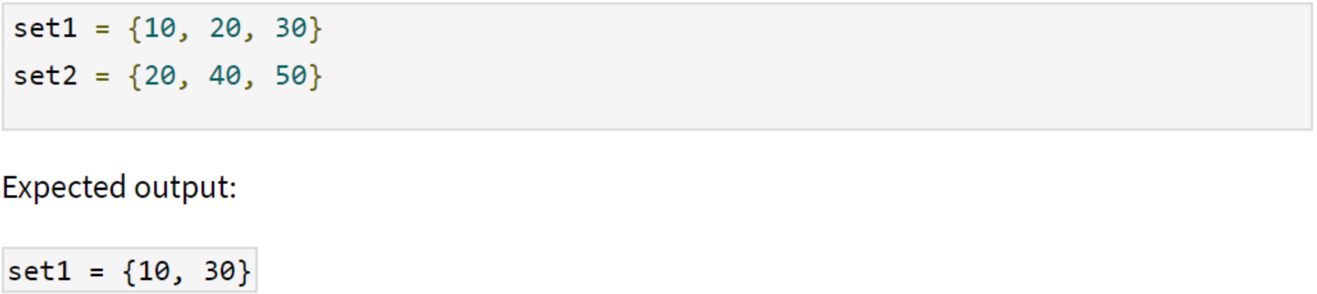


```
set1 = {10, 20, 30, 40, 50}
set2 = {30, 40, 50, 60, 70}
```

Expected output:

```
{70, 40, 10, 50, 20, 60, 30}
```

4. Given a two Python sets, update first set with items that exist only in the first set and not in the second set

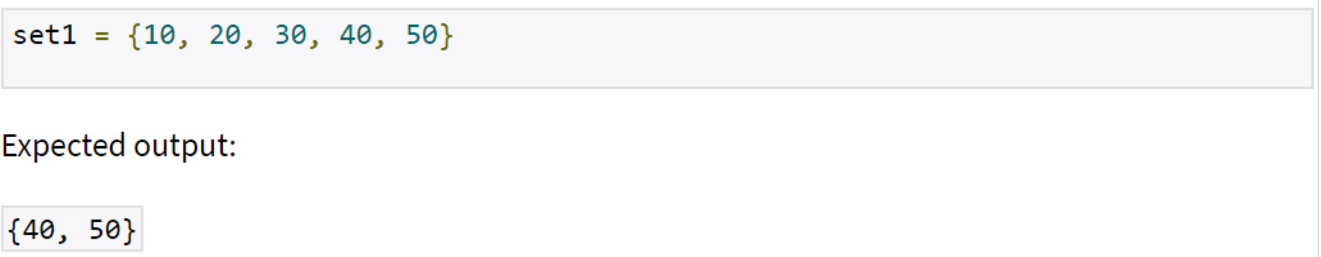


```
set1 = {10, 20, 30}
set2 = {20, 40, 50}
```

Expected output:

```
set1 = {10, 30}
```

5. Remove 10, 20, 30 elements from a following set at once

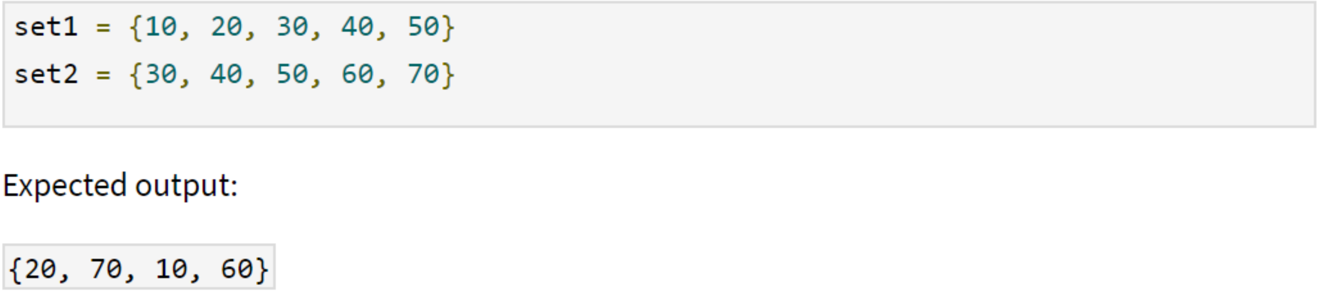


```
set1 = {10, 20, 30, 40, 50}
```

Expected output:

```
{40, 50}
```

6. Return a set of all elements in either A or B, but not both

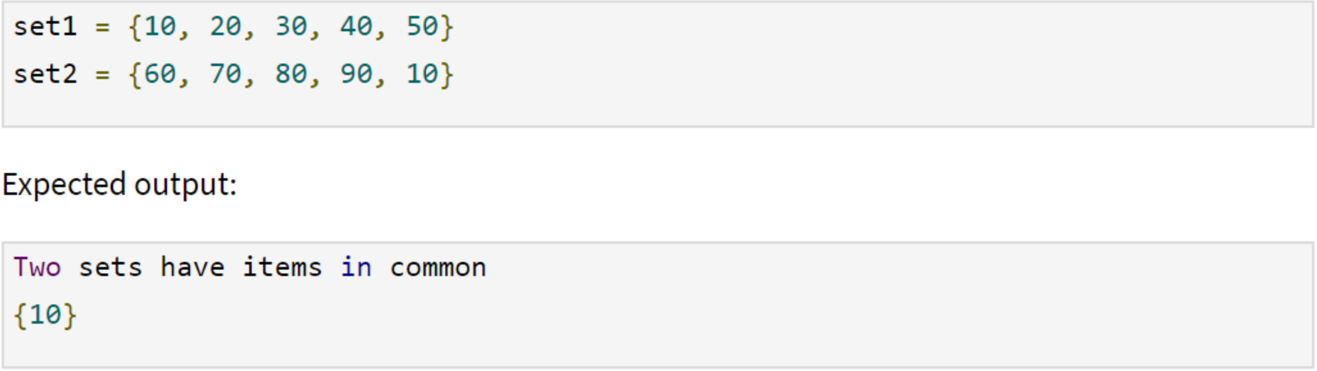


```
set1 = {10, 20, 30, 40, 50}
set2 = {30, 40, 50, 60, 70}
```

Expected output:

```
{20, 70, 10, 60}
```

7. Determines whether or not the following two sets have any elements in common. If yes display the common elements

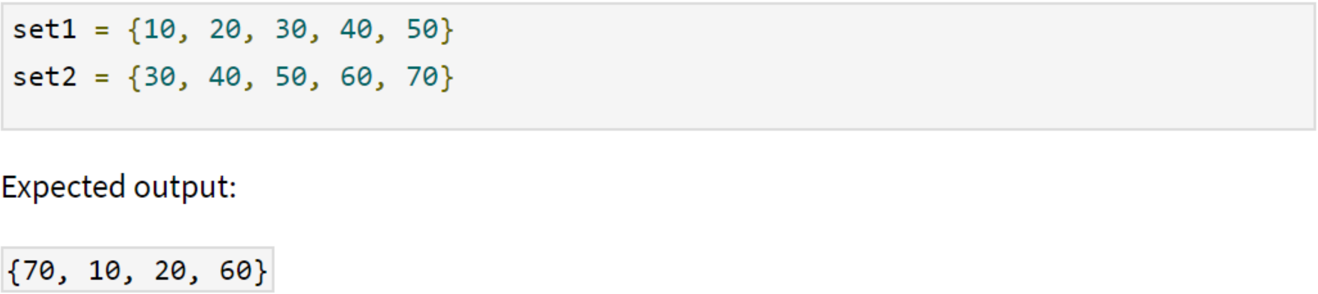


```
set1 = {10, 20, 30, 40, 50}
set2 = {60, 70, 80, 90, 10}
```

Expected output:

```
Two sets have items in common
{10}
```

8. Update set1 by adding items from set2, except common items

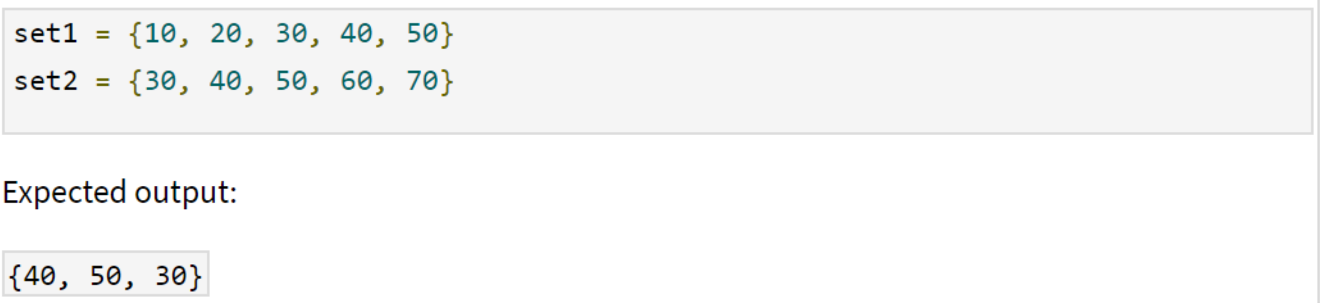


```
set1 = {10, 20, 30, 40, 50}
set2 = {30, 40, 50, 60, 70}
```

Expected output:

```
{70, 10, 20, 60}
```

9. Remove items from set1 that are not common to both set1 and set2



```
set1 = {10, 20, 30, 40, 50}
set2 = {30, 40, 50, 60, 70}
```

Expected output:

```
{40, 50, 30}
```