![](https://ga-dash.s3.amazonaws.com/production/assets/logo-9f88ae6c9c3871690e33280fcf557f33.png) Python Part Time

<!---

Questions? Comments?
1. Log an issue to this repo to alert me of a problem.
2. Suggest an edit yourself by forking this repo, making edits, and submitting a pull request with your changes back to our master branch.
3. Hit me up on Slack @susiremondi
--->

Equal Sets

Overview:

For two sets to be equal, they simply have to contain the same elements - it doesn't matter what order they're in.

Unfortunately in Python, comparing two lists using `==` will only produce `True` if the elements are in the same order, which is not what we want!

There's no built-in function to compare differently-ordered lists, so you'll be writing it here by leveraging your knowledge of sets.

You will practice these programming concepts we've covered in class:
- Functions
- Sets

Deliverables

One `.py` file with code that solves the problem.

Requirements:

Write a function that takes two lists and returns `True` if they have the same elements, even if they aren't in the same order.

Here is an example using lists. Try running this normally:

```python
fruits = ['orange', 'pear', 'kiwi', 'apple', 'banana']

fruits_copy = ['orange', 'pear', 'kiwi', 'apple', 'banana']

fruits_reordered = ['pear', 'apple', 'kiwi', 'orange', 'banana']

print("Copy comparison", fruits == fruits_copy)
print("Reordered comparison", fruits == fruits_reordered)
```

Above, the `Reordered comparison` prints `False`.
Convert the above example from lists to sets. Does the output of the two print statements change? Why?

<!-- Solution
https://repl.it/@sonylnagale/matching-lists

```python
fruits = ['orange', 'pear', 'kiwi', 'apple', 'banana']

fruits_copy = ['orange', 'pear', 'kiwi', 'apple', 'banana']
```

~~~
fruits_reordered = ['pear', 'apple', 'kiwi', 'orange', 'banana']

print("Copy comparison", fruits == fruits_copy)
print("Reordered comparison", fruits == fruits_reordered)

def is_match(list1, list2):
   return set(list1) == set(list2)

print(is_match(fruits, fruits_reordered))
```

-->
~~~