SETS

- Think of sets as dictionaries but with only keys.
- All the properties about dictionaries are still true, not index-based, by definition cannot be sorted
- In the book, there are specific operations that work with lists like intersection, differences, union, etc… (exam questions)
- Other CSE classes will talk more in depth about how sets work, and why its so important

SETS

- Good things to put on the notecard are the main set operations (in text book & online)
- “ set_a | set_b“ or “set_a.union(set_b)” # All elements of both sets
- “set_a & set_b” or “set_a.intersection(set_b) ” Elements that exist in both sets
- “set_a - set_b” or “set_a.difference(set_b)” # all elements that in set A but not set b

- Does order matter for these things?

ADDING VALUES TO SETS

```
S = set()
for n,i in enumerate("Cyndy"):
        S.add( i )
        S.add(n)
# after the end of the for loop S prints { "C",0, "y",1 , "n", 2, "d",3 , "y", 4 }
```

ORDERING

- Intersection : creates a new set of elements that are in both sets (order doesn't matter)
- Union: creates a new set of elements containing all the elements in set A & set B (order doesn't matter)
- Difference : new set whose elements are in the first set but NOT in the other set (order matters)
- Symmetric difference: creates a new set of elements are different, opposite of intersection (order doesn't matter)

HOW I SOLVED THIS LAB

Part A

- Add comments to appropriate places
- In the compare_files function, call the build_word_set function for file 1, and file 2
- Find the correct set method that will return all the words in both files
- Find the correct set method that will return the common words between both files

Part B

- In build_word_index function, we are going to build a dictionary where the word is the key, and the value is a set of the line numbers it exists through a file
- That means, iterate through the input_file, for each line in the file,
- Make sure you are keeping track of the line number because that's needed later! (use enumerate or a counter)
- Turn the line to the list (.strip().split()) Now, you have the list of all the words.
- For each word that list, make it lower, remove the punctuation (you can actually use strip for that)
- Check to see if the word is whitespace, if it isn't, check to see if the word is already in our dictionary, if so, add the line number, else create a key in the dictionary and assign the value as the set of the current line number .