SETS!

LOGISTICS

- Congratulations on completing your first exam!
- HW03 has been released

REVIEW: SETS

Sets are an **unordered** container for data.
Unordered means:

- We can still use `for` to loop over every element
- can still use `in` to see if something is in it
- can **not** access into it with an index and `[]` or slice

REVIEW: SETS

Sets look similar to lists, but with `{}` instead of `[]`

- `{"this"}`
- `{"howdy", "partner"}`

Cannot store lists, dicts or other sets within a set

Most important part of a set: it enforces uniquness of its elements. An element can only be in the set once or not at all.

PRACTICE: LIST OR SET? [C1]

For each of the following, decide whether it makes more sense to use a list or a set to store your data:

SITUATION	LIST OR SET?	WHY?
You want to track who entered your party and in what order		
You need to collect all the unique words in an essay		
You want to track which students have turned in permission slips		
You are collecting all zip codes that your store's customers live in		
You are collecting votes cast in a student council election		

PRACTICE: LIST OR SET? [C1]

For each of the following, decide whether it makes more sense to use a list or a set to store your data:

SITUATION	LIST OR SET?	WHY?
You want to track who entered your party and in what order	List	Order
You need to collect all the unique words in an essay	Set	Duplicates
You want to track which students have submitted permission slips	Set	Order
You are collecting all zip codes that your customers live in	Set	Duplicates
You are collecting votes cast in a student council election	List	Duplicates

PRACTICE: COFFEE SHOP [L2]

Imagine you are the owner of the crow-themed coffee shop, Cawfee. You are trying to rebrand into the tea market and need to email everyone on your email list to let them know about your new name, Spill the Tea.

You have a list of every email address connected to a credit card transaction made at your cafe. But oh no! a lot of emails are in that list multiple times since you have regulars and repeat customers.

Help write the below function which gets you a list of all unique email addresses.

```
def get_unique_emails(emails_list : list[str]) -> list[str]:
    # your code here
```

ONE POSSIBLE SOLUTION

```
def get_unique_emails(emails_list : list[str]) -> list[str]:
    return list(set(emails_list))
```

REVIEW: SETS

There are a few common features of a set:

- `.add()` adds an item to the set
- `.remove()` removes an item from a set, if the item is not in the set then crash
- `.discard()` removes an item from a set, ignore if the item is already not in the set

PRACTICE: HARRY'S BIRD SANCTUARY [C2]

Harry's Bird Sanctuary is serious about treats. They keep a `list` of bird names, marking every time a bird receives a treat. They also have a `list` of all birds in the sanctuary.
Using sets and the `.add()` and `.discard()` functions, complete the function below, which finds the list of birds that have not yet recieved a treat.

```
def needs_treat(recieved : list[str], all_birds : list[str]) -> set:
    # Your code here
```

ONE POSSIBLE SOLUTION

```
def needs_treat(recieved : list[str], all_birds : list[str]) -> set:
    in_need = set(all_birds)

    for bird in recieved:
        in_need.discard(bird)
    return in_need
```

REVIEW: SET COMPARISONS

Sets are compared slightly differently from lists and strings because, for sets, *order doesn't matter*

OPERATOR	MEANING	EXAMPLE
X == Y	all values in X are in Y and all values in Y are in X	{1, 3} == {3, 1} is TRUE
X >= Y	all values in Y are in X	{1, 2} >= {1, 2} is TRUE {1, 2} >= {1} is TRUE
X > Y	all values in Y are in X and X is larger than Y	{1, 2} > {1, 2} is FALSE {1, 2} > {1} is TRUE

This is also true in the opposite direction (< rather than >)

Practice: a bird's gotta fly [C3]

At Harry's Bird Sactuary, they keep a log of each time a bird is given free-flight time called `flights_oct8.csv`. Where each line of the file notes each bird and how long they were released. The format looks like the following:

```
Mango, afternoon, 2 hours
Chirpy, afternoon, 1 hour
Mango, evening, 3 hours
... (etc)
```

They also have a file `birds.csv` that lists all of their birds. Each line of the file is the name of a bird. It's possible that some birds have since been adopted, so they exist in `flights_oct8.csv` but not `birds.csv`.

Using sets, write a program to check if all birds currently at the rescue have been given free-flight at least once on Oct8, and prints "good job today!" if they have.

ONE POSSIBLE SOLUTION

```
all_birds = set()
flew_set = set()
bird_file = open('birds.csv', 'r')
flew_file = open('flights_oct8.csv')
for name in bird_file:
    all_birds.add(name)
for line in flew_file:
    name = line.split(',')[0]  # get just the name
    flew_set.add(name)

# <= because there may be some birds in flew_file
# that are no longer at the sanctuary
if all_birds <= flew_set:
    print("good job today!")
```