



Practice with Lists
and while Loops

Announcements

- **Re: Quiz 01**
 - Great job!! Median was ~85%
 - *Please review what you missed ASAP*; we will build on the topics covered in Quiz 01 throughout the course, and these foundational concepts are vital!
 - Don't understand a particular question/part of a memory diagram? Please visit us in Office Hours or Tutoring! Full list of hours on the site's [support page](#)
 - Please submit a regrade request by Friday at 11:59pm *if you believe your quiz was not graded correctly according to the rubric*
- **EX03: Wordle due today at 11:59pm**

Warm-up

```
1  def gen(stop: int) -> list[int]:
2      """Generate a list from 0 to stop, not inclusive."""
3      i: int = 0
4      acc: list[int] = []
5      while i < stop:
6          acc.append(i)
7          i = i + 1
8      return acc
9
10
11 print(gen(3))
```

Coding an RSVP tracker!

In this in-class exercise, you will manage a global list of friends who have RSVPed “yes” to a party.

Setup instructions:

1. Create a file named `c116.py` in your workspace.
2. Then, add this global variable to your file:

```
rsvp_list: list[str] = ["Enrico", "Yun"]
```

add_rsvp function

Goal: Record that a friend has RSVPed to the party by adding them to the list.

Instructions:

Write a function with the signature:

```
def add_rsvp(rsvped: list[str], friend: str) -> None:
```

This function should:

1. Add the `friend`'s name (literal value) to `rsvped`.
 - i. *Mutate* the list using a list method.

has_rsvped function

Goal: Check whether a specific friend has already RSVPed.

Instructions:

Write a function with the signature:

```
def has_rsvped(rsvped: list[str], friend: str) -> bool:
```

This function should:

1. Use a `while` loop to search through `rsvped`.
2. Compare each name in the list to the given `friend`'s name (literal value).
3. return `True` as soon as a match is found, OR
return `False` if the loop finishes (after iterating through all elements in `rsvped`)
without finding the friend's name.
4. ***Not mutate*** the list.

not_yet_rsvped function

Goal: Given a list of all invited friends, determine which friends have *not* yet RSVPed.

Instructions:

Write a function with the signature:

```
def not_yet_rsvped(rsvped: list[str], invited: list[str]) -> list[str]:
```

This function should:

1. Create a *new empty list* intended to store names of people who have yet to RSVP.
2. Use a `while` loop to iterate through the `invited` list.

For each invited friend:

- Call your `has_rsvped` function to check if they are already in `rsvped`.
 - If they have **not** RSVPed (i.e, if their name is not in `rsvped`), add their name to the new list.
3. Return the new list of friends who still need to RSVP.
 4. **Not mutate** either `invited` or `rsvped`.