Fall 2022

Class 11 - Lists Practice

Announcements

- Required Course Survey: https://bit.ly/comp110-survey-22f
  - Complete tonight!
- Reading 0: Due Monday
- Exercise 4: List Utils - Due Monday
- Quiz 1: Thursday 9/22
  - Practice Problems Posted, Tutoring and Review Session Schedule on Site
- Tuesdays - Continue Async - this Tues 16 of 695 (2 - 3%) attended
  - Have questions on Tuesday content? Come to tutoring and office hours!

Announcements

- Notes on Office Hours / Exercises

1. Be kind and bring positive energy.
   - If I get reports of repeated rudeness, aggression directed at a TA, or otherwise you will lose access to office hours following a conversation with me.

2. Have your instructions open!
   - Be able to point to exactly where it is you think you are not able to translate requirements in the instructions to the code you are working on.

3. Don't post screenshots of code or the autograder feedback!
   - Come see us in office hours. Starting early helps avoid deadline day stress.

Challenge Question #1

```
1    """List diagram example."""
2
3
4    a: list[str] = ["one"]
5    b: list[str] = a
6    a.append("two")
7
8    print(b[1])
```

Challenge Question #2

```
"""Lists and functions."""


def dup(xs: list[int]) -> None:
    """Duplicate a list's values."""
    start_len: int = len(xs)
    i: int = 0
    while i < start_len:
        xs.append(xs[i])
        i += 1


nums: list[int] = [10, 20]
dup(nums)
print(nums)
```

Challenge Question #3

```python
"""Example producing a function."""


def odds(min: int, max: int) -> list[int]:
    """Construct list of odds, inclusive."""
    xs: list[int] = list()
    i: int = (min // 2) * 2 + 1
    while i <= max:
        xs.append(i)
        i += 2
    return xs


ys: list[int] = odds(3, 6)
print(ys)
```

Follow-along: Implementing a `contains` Function

- Let's implement a function where we can call with 2 arguments:
  1. a "**needle**" value we are searching for
  2. a "**haystack**" list of values we are searching in

- The return value of the function should be True iff the **needle** is found in the **haystack** at least once and False otherwise

- The name of the function will be **contains**