

Parallel Lists

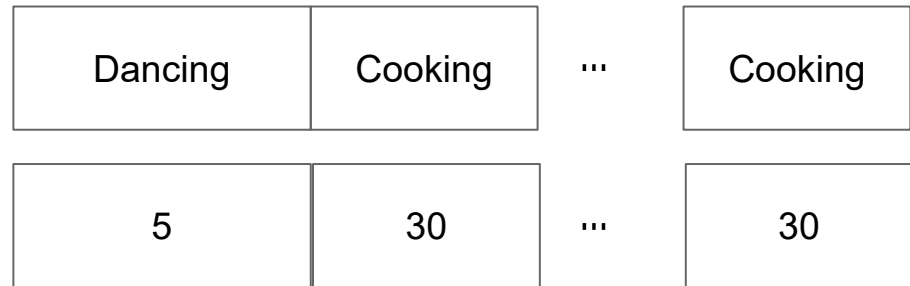
To explore this idea, how could we find the most popular cafe or hobby? With 13 hobbies, we don't want to have a tally variable representing each...

Idea:

Use two parallel lists

- List with hobbies
- List with accumulators

Use the same index to access corresponding data



```

1 # Find most common hobby (using parallel lists)
2
3 # Open the file
4 file = open("ExWeek7/favourites-survey.csv")
5 junk_header = file.readline()
6
7 # Hobbies in the survey
8 hobbies = ["playing video games", "playing an instrument", "dancing",
9            "painting or drawing", "playing a sport", "crafts", "acting",
10           "singing", "yoga", "making videos", "cooking", "learning new languages",
11           "hiking"]
12
13 # Create list of tallies (all 0's)
14 # (accumulator pattern)
15 tallies = [0] * len(hobbies)
16
17 # Tally up the frequency of hobbies in file
18 for line in file:
19     # Get hobby from next line
20     data = line.strip().split(",")
21     hobby_in_file = data[5].lower()
22
23     # Update correct tally in parallel arrays:
24     # - find the index (i) of hobby that matches the one in the file
25     # - add one to tally at matching index
26     for i in range(len(hobbies)):
27         if hobby_in_file == hobbies[i]:
28             tallies[i] += 1
29
30 # Print frequency of each hobby
31 for i in range(len(hobbies)):
32     print(f"{hobbies[i]:<24}{tallies[i]:>4}")

```

Parallel Lists

This is an alternative to having a tally for each hobby and a *bunch* of elifs!



Review Questions

What does this code output?

What does it the output represent?

```
word = "acorn"  
print(len(word) - 1)
```



Review Questions

What does this code output?

```
word = "acorn"
for i in range(len(word)):
    print(i)

for i in range(len(word)):
    print(len(word)-1-i)
```

What does this code output?

```
word = "acorn"
for i in range(len(word)):
    print(word[i])
```



Review Questions

What would this output?

```
word = "acorn"  
for i in range(len(word)):  
    reverse_index = len(word)-1-i  
    print(word[reverse_index])
```