

Announcements

1. [PA 2 World of Elections](#)

- Part A - Reading quiz in canvas - Done.
- Part B due Friday 11/08 (30 pts)
- Part C due Friday 11/15 (70 pts)

2. Quiz 5 next Thursday 11/17, in King248

- Covers chapters 9 and 10: Sequence and File I/O

Today's class

1. (25 mins) Chapter 11(Nested data) preview
2. (30 mins) File IO practice
3. (30 mins) Practice quiz 5

Chapter 11(Nested data) preview

POGIL: Lists of Lists

- Complete the activity in 10 minutes
- Discussion

List comprehensions

- The Python language provides a convenient construct, known as list comprehension, that iterates over a list, modifies each element, and returns a new list of the modified elements.
- Such preference is due to less code and more efficient execution by the interpreter. The table below shows various for loops and equivalent list comprehensions.

Example 1: Add 5 to every element

```
lst = [1,2,3,4,5]
new_lst = [i + 5 for i in lst]
print(new_lst)
```

Example 2: Apply a function to each element

```
line = ["1", "2", "3", "4"]
new_line = [int(i) for i in line]
print(new_line)
```

Example 3: Conditional list comprehensions

```
# create a list of only even numbers
line = [1,2,3,4,5,6]
new_line = [i for i in line if i % 2 == 0 ]
print(new_line)
```

File IO practice

- By default, `split()` considers any whitespace (spaces, tabs, newlines) as a separator.
- The `split()` method takes the delimiter(`sep`)

```
line = "1,2,3,4"
print(line.split(",")) # print ['1', '2', '3', '4']

line = "1 2 3 4"
print(line.split()) # print ['1', '2', '3', '4']

line = "1\n2    3 4"
print(line.split()) # print ['1', '2', '3', '4']
```

Hints for PA2: how to read the ballot data

Practice quiz 25 minutes

- Practice Quiz 5¶
- 1st sheet: "written potion"
- 2nd sheet: "coding potion" (write code on paper)
- 3rd sheet: [Python Reference Sheet](#) – new!
- Turn off monitor during quiz