Announcements

• No Reading for next time
• No Reading Quiz for next time
• Apt-02 due tonight, Assignment 3 due Thursday

• Exam next Tuesday
• Work the practice exam before Thursday’s class. Will go over next class.
List Comprehension review

• Take advantage of patterns, make a new list based on per element calculations of another list

• Format:

\[
[<\text{expression with variable}> \text{ for } <\text{variable}> \text{ in } <\text{old list}>]
\]

• Format with filtering:

\[
[<\text{expression with variable}> \text{ for } <\text{variable}> \text{ in } <\text{old list}> \text{ if } <\text{filter with variable}>]
\]
List Comprehensions from last time

\[ j+1 \text{ for } j \text{ in } \text{range}(20) \text{ if } j \% 3 == 0 \]
List Comprehensions from last time

$$[j+1 \text{ for } j \text{ in } \text{range}(20) \text{ if } j \% 3 == 0]$$

$$[0, 1, 2, \ldots, 19]$$
List Comprehensions from last time

\[ \{ j+1 \text{ for } j \text{ in } \text{range}(20) \text{ if } j \% 3 == 0 \} \]

\[ \{0, 1, 2, \ldots, 19\} \]

\[ \{1, 2, 3, \ldots, 20\} \quad \text{this is } j+1 \]
List Comprehensions from last time

\[
[j+1 \text{ for } j \text{ in } \text{range}(20) \text{ if } j \% 3 == 0]
\]

\[
[0, 1, 2, \ldots, 19]
\]

\[
[1, 2, 3, \ldots, 20]
\]

this is \(j+1\)

\[
[1, 4, 7, 10, 13, 16, 19]
\]

\(j+1 \text{ if } j \% 3 == 0\)
List Comprehensions from last time

\[ [i \times 2 \text{ for } i \text{ in } [j + 1 \text{ for } j \text{ in } \text{range}(20) \text{ if } j \% 3 == 0] \text{ if } i \times i > 19] \]
List Comprehensions from last time

\[
[i \times 2 \text{ for } i \text{ in } [j + 1 \text{ for } j \text{ in range}(20) \text{ if } j \% 3 == 0] \text{ if } i \times i > 19]
\]

\[
[i \times 2 \text{ for } i \text{ in } [1, 4, 7, 10, \ldots, 19] \text{ if } i \times i > 19]
\]
List Comprehensions from last time

\[ [i \times 2 \text{ for } i \text{ in } [j + 1 \text{ for } j \text{ in range(20)} \text{ if } j \% 3 == 0] \text{ if } i \times i > 19] \]

\[ [i \times 2 \text{ for } i \text{ in } [1, 4, 7, 10, \ldots, 19] \text{ if } i \times i > 19] \]

\[ [[2, 8, 14, 20, 26, 32, 38] \text{ if } i \times i > 19] \]

\[ [14, 20, 26, 32, 38] \text{ <- answer after filtering} \]
List Comprehension examples

- List comprehension creates a new list. Use wherever you need a new list.
- Can use list comprehension to create a list to return
- From Uppity.py – words is list of words

```python
def uppify_list(words):
    return [w.upper() for w in words]
```
- Classwork problem 1
- Show additional examples
Problem: complete program

• Problem:

• Given a file of words, for each line print out only those words that are longer than 4, thus removing all the “short” words.

Example:

Where are all the wild things?

becomes

Where things?
Tasks in solving this problem

- First understand the code given
- Then fill in missing code (TODO)
  - In Eclipse, “Window”, “Show view”, “Tasks”
- Test it with data file you create
  - May need to create data folder
- Where is the fence post problem?
- Debugging – what do you do when it doesn’t work?
Fence post problem

• How many posts, how many supports between all those posts?

• Build fence:

```python
for post in posts:
    fence = fence + post + supports
```
Passing Functions as Parameters

def upperWord(word):
    return word.upper()

def argWord(word):
    return word + "arg"

def transformWord(func, word):
    return func(word)

print transformWord(upperWord, "train")
print transformWord(argWord, "truck")
Assignment 3 Transform

• Look over assignment
• What parts are similar to what we just did?
• Passing functions as parameters
• What are the imports?
• Has file browser for you
  import InputGUI as Input
• To input from Console change line to:
  import InputConsole as Input