Week 7, Day 1

☰ Tags

Practice Problem

List Comprehensions

—> Python docs: https://docs.python.org/3/tutorial/datastructures.html

Read more: https://realpython.com/list-comprehension-python/

https://builtin.com/data-science/pythonic-code

https://docs.python-guide.org/writing/style/ (Hitchiker's Guide to Python0

BENEFITS:
1. One main benefit of using a list comprehension in Python is that it's a single tool that you can use in many different situations. In addition to standard list creation
, list comprehensions can also be used for mapping and filtering.

2. List comprehensions are also more **declarative** than loops, which means they're easier to read and understand. With a list comprehension in Python, you can instead focus on *what* you want to go in the list and trust that Python will take care of *how* the list construction takes place.

Slides:

- writing Pythonic code
    - Pythonic code is just a description of a code that's easy to read, follow and understand. (Zen of Python) ("import this")
    - **In order to be fluent in Python you'll need to use idioms**
    -

PEP 8: The Style Guide for Python Code

This document gives coding conventions for the Python code comprising the standard library in the main Python distribution. Please see the companion informational PEP describing style guidelines for the C code in the C implementation of Python . This document and PEP 257 (Docstring Conventions) were

https://pep8.org/#a-foolish-consistency-is-the-hobgoblin-of-little-minds

- list comprehension — basic, parts of a list comprehension
    - Interactive example in Jupyter Notebook
        - Transform different types in different ways: dictionary
- Jupyter notebook: questions 1-5
    - (or directly in Learn — clean out answers, practice running)