## Recap of the main points for Week 2:

- For new programmers, focus on the fundamentals of "control of flow" statements.
- For experienced programmers, reflect on how you've coded to focus on "process decomposition."
- **Executing Python:**
  - Build-in IDLE interpreter
  - .py file
  - Jupyter Notebook
  - Spyder and other IDEs
  - Notebook interacts with Operating System (! and %)
- **Expressions**:
  - Assignment and Equivalency
  - Cumulative shortcuts
- **Concept of "Objects"**
  - Everything in python *is* an object
  - Objects as reflecting objects in the physical world (like *cats*!)
    - More on objects in a couple of weeks
  - Casting objects from one type to another
- **Python and Types**
  - Confirm the data type of variables in Python
  - Use python for coding but also for looking at and documenting your code
- **Scope & Visibility of Variables**
  - Namespace versus Object space (in lieu of our using pointers)
- **Strings**
  - Big role in data science, full-text retrieval, data integration
  - Python's use of +, as well as obvious "methods" (.upper())
  - Comment in code - use 'em. Required later as "docstrings"
  - String is a contiguous block of member, read from pointer of the first character
- **If/For/While statements**
  - Each are useful; the logic of your decomposition guides your choices …
    - And python has techniques for even faster processing (next week)
  - Nested if; nested while; nested for loops …
    - We'll revisit this when discussing algorithm efficiency (Big O)