Testing and Debugging

- Would be great if our code always worked properly the first time we run it!
- But life ain't perfect, so we need:
  - Testing methods
    - Ways of trying code on examples to determine if running correctly
  - Debugging methods
    - Ways of fixing a program that you know does not work as intended

When should you test and debug?

- Design your code for ease of testing and debugging
  - Break program into components that can be tested and debugged independently
  - Document constraints on modules
    - Expectations on inputs, on outputs
    - Even if code does not enforce constraints, valuable for debugging to have description
  - Document assumptions behind code design

When are you ready to test?

- Ensure that code will actually run
  - Remove syntax errors
  - Remove static semantic errors
  - Both of these are typically handled by Python interpreter
- Have a set of expected results (i.e. input-output pairings) ready