Lecture 3: Loops and Functions

Announcements

- Waitlist: We expect 10% to get off, but it's unknown.
- Attend Labs!

Things You Can Do Now

- Write a program that makes a decision.
- Write your own functions
- Use loops so you can process lots of data.

Let's Talk About Python

- Expression
  \[ 3.1 \times 2.6 \]
- Call expression
  \[ \max(0, x) \]
- Variables
  \[ x = <expression> \]
- Define Function:
  \[ \text{def } <\text{function name}>(<\text{parameter list}>): \]
- Control Statements:
  \[ \text{if } ... \]
  \[ \text{while } ... \]

Defining Functions

- Abstracts an expression or set of statements to apply to lots of instances of the problem
- A function should do one thing well
Doctests

- Write the docstring to explain what it does
  - What does the function return? What are corner cases for parameters?

- Write doctest to show what it should do
  - Before you write the implementation.
  - python3 -m doctest [-v] file.py

Returns and Values

- All functions always return SOME value.
- If you don’t specify return, the value is None.

Conditional Statement

- Do some statements, conditional on a predicate expression

  ```
  if <predicate>:
      <true statements>
  else:
      <false statements>
  ```

- Example:

  ```
  if (temperature>98.6):
      print("fever!")
  else:
      print("no fever")
  ```

while Statement – Iteration Control

- Repeat a block of statements until a predicate expression is satisfied

  ```
  <initialization statements>
  while <predicate expression>:
      <body statements>
  <rest of the program>
  ```

Functions and Arguments

```python
def max(x, y):
    return x if x > y else y
def power(x):
    return x**2
```