Design

Abstraction

Functional Abstractions

```python
def square(x):
    return pow(x, 2)
def square(x):
    return mul(x, x-1) + x
```

If the name "square" were bound to a built-in function, `sum_squares` would still work identically.

```python
def sum_squares(x, y):
    return square(x) + square(y)
```

Choosing Names

<table>
<thead>
<tr>
<th>From:</th>
<th>True_value</th>
<th>dice</th>
<th>helper</th>
<th>num_rolls</th>
<th>l, 1, 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test:</td>
<td>rolled_a_one</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Names typically don’t matter for correctness but they matter a lot for composition.

Names should convey the meaning or purpose of the values to which they are bound.

The type of value bound to the name is best documented in a function’s docstring.

Function names typically convey their effect (print), their behavior (triple), or the value returned (abs).

Which Values Deserve a Name

Reasons to add a new name

- Repeated compound expressions:
  - `if sqrt(square(a) + square(b)) > 1:`
  - `x = x + sqrt(square(a) + square(b))`
  - `hypotenuse = sqrt(square(a) + square(b))`
  - `if hypotenuse > 1:`
  - `x = x + hypotenuse`
- Meaningful parts of complex expressions:
  - `x1 = [-b + sqrt(square(b) - 4 * a * c)] / (2 * a)`
  - `discriminant = square(b) - 4 * a * c`
  - `x1 = (-b + sqrt(discriminant)) / (2 * a)`

More Naming Tips

- Names can be long if they help document your code:
  - `average_age = average(age, students)`
  - Is preferable to
  - `# Compute average age of students
   aa = avg(a, st)`
- Names can be short if they represent generic quantities: counts, arbitrary functions, arguments to mathematical operations, etc.
  - `n, k, i - Usually integers
   x, y, z - Usually real numbers
   f, g, h - Usually functions`

Test-Driven Development

Write the test of a function before you write the function.

A test will clarify the domain, range, & behavior of a function.

Tests can help identify tricky edge cases.

Develop incrementally and test each piece before moving on.

You can’t depend upon code that hasn’t been tested.

Run your old tests again after you make new changes.

Bonus idea: Run your code interactively.

Don’t be afraid to experiment with a function after you write it.

Interactive sessions can become doctests. Just copy and paste. (Demo)
Function Example: Sounds

The Waveform Audio File Format encodes a sampled sound wave.

A triangle wave is the simple waveform with the most pleasing sound.

https://en.wikipedia.org/wiki/Triangle_wave
https://en.wikipedia.org/wiki/Sampling_(signal_processing)