Announcements

• Midterm 1 is on Monday 9/22 from 7pm to 9pm
• 2 review sessions on Saturday 9/20 3pm-4:30pm and 4:30pm-6pm in 1 Pimentel
• No review session on Sunday 9/21 from 12pm to 3pm in 2060 Valley LSB
• No lecture on Monday
• No lab or office hours next week: Tuesday 9/23, or Wednesday 9/24
• Optional Hog strategy contest ends Wednesday 10/1 @ 11:59pm

Abstraction

Functional Abstractions

def square(x):
    return mul(x, x)

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

What does sum_squares need to know about square?

Yes
No
Yes
No

Choosing Names

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

Choosing Names

Reasons to add a new name

More Naming Tips

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 (lab).

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.
Decorators

Function Decorators

(Demo)

```python
@trace1
def triple(x):
    return 3 * x
```

is identical to

```python
def triple(x):
    return 3 * x
triple = trace1(triple)
```

Decorated function

Why not just use this?

Review

```python
def delay(arg):
    print('delayed')
def g():
    return arg
    return g
```

What Would Python Print?

The print function returns None. It also displays its arguments (separated by spaces) when it is called.

```python
from operator import add, mul
def square(x):
    return mul(x, x)
```

Names in nested `def` statements can refer to their enclosing scope

A function that always returns the identity function

```python
def pirate(arggg):
    print('matey')
def plunder(arggg):
    return arggg
    return plunder
```

```
def pirate(pirate)(pirate)(pirate)(5)(7)
```

Error
Matey
Matey
Error
```
def pirate(pirate)(pirate)(pirate)(5)(7)
```

5
5
5
5
```
def pirate(pirate)(pirate)(pirate)(5)(7)
```

Identity function

A name evaluates to the value bound to that name in the earliest frame of the current environment in which that name is found.