



Day 5

Modules, Classes, & Object-Oriented Programming



Recap

- Functions (& Scope)
- Control-Flow
- Good code
- Documentation
- File input + output



Today's Agenda

- Functions (a bit more advanced),
- Modules and How to use them
- Objects: Objectively Pythonic
- More practice!



Functions: defaults, # arguments

```
def greeter(name, msg = 'Welcome to the Python Course'):
    ....
```

- `**args`
- `**kwargs`

Extra: how to use **enumerate**



Functions: anonymous functions

- Useful when you don't want to pollute namespace

lambda arguments: expression

Square = lambda x : x * x



Functions and Methods

Functions, e.g. `len(x)`

Can have a variety of different inputs (Strings, lists, ...)

Methods, e.g. `text.upper()`

Work on the object they're called on (e.g. String)



Modules

- Play around with module **math**
- Import it
- Try out some methods

```
from <module name> import *
```

```
import <module name> as <alternative_module_name>
```

Module Properties

- `__name__`
- `__doc__`
- `__file__`
- `print(dir(utils))`

Let's try making our own module!

1. import it
2. use some methods from it

What if we want to run it as a standalone program?

If `__name__ == '__main__':`