* Introduction
  + WEB2PY: web development with python
    - Integrate numerous technologies
    - Python
    - Bootstrap
    - SQLite
    - Rocket web server
    - Browser base IDE
  + Full Stack Web Development
    - Front end
    - Server code
    - Database
* Course Overview
  + Download WEB2PY
  + MVW, SQLFORMS
  + Database Admin
* Download and Start WEB2PY
  + Download WEB2PY
  + Start web server
  + Model-View-Controller
  + Go to web2py website
  + In the application folder
    - Actual source code will be in there
  + To start the web server, click on the web2py executable
  + It will open a default welcome app
  + Admin on the welcome app
    - Browser based ide that allows us to access and edit code
* Model-View-Controller
  + controller
    - python code
  + view
    - html files
  + MVC Convention
    - software architecture
    - naming convention also
    - def index(), returns message back to the view
  + view name
    - [controller]/[method name].html
* Build Our First Web Page
  + create a controller
  + inside controller create a method
  + ex) create controller name basics
    - def helloworld():
    - msg = “Hello from the Controller!”
    - return locals()
  + return locals()
    - take all local variables defined in method and return it to view
  + then create a view
    - [controllername]/[methodname]
  + ex)
    - basics/helloworld
  + inside the view, you can mix python with html
  + ex)
    - {{extend ‘layout.html’}}
    - <h1>Our First Web Page</h1>
    - <h2> {{=msg}}</h2>
  + {{}} double curly bracket means python code
  + to go to webpage use url [local address]/[application name]/[controller name]/[view name]
* Request Object
  + web application use http protocol
  + request object and response object
  + request object: anything passed from client to web server
  + arguments: part of url
    - request.args(0), request.args(1), …
  + variables: part of html forms
    - form
    - associated with post
    - request.vars.form\_name
  + in side controller
    - def request\_args():
    - arg1 = float(request.args(0))
    - arg2 = float(request.args(1))
  + have to convert arguments to float because they come across as strings
  + then create a view
    - basics/request\_args