

CS50 Section 9

Somewhere in Between

Annaleah Ernst, TF

Agenda

- ▶ Loose ends in Python
 - ▶ Decorators
- ▶ Flask
- ▶ SQL
- ▶ Jinja
- ▶ HTML
- ▶ MVC
 - ▶ C\$50 Finance!

Python: Decorators

- ▶ A decorator is a function that modifies behavior of other functions
 - ▶ adds additional functionality
- ▶ You can use this to change the behavior of functions
- ▶ EX:

```
def override(func):  
    def incr():  
        return func() + 1  
    return incr
```

```
@override  
def one():  
    return 1
```

```
print(one())
```

Python: Decorators (cont)

- ▶ In the course, we will be using the `@app.route()` decorator
 - ▶ Typically to specify the URL associated with a function and what method applies
- ▶ May also use the `@login-required` decorator
 - ▶ Use this to ensure that certain functions are only available if the user logged in
 - ▶ eg, `buy()`

Flask

- ▶ Python-based microframework
- ▶ Automates the process of building simple web-apps
- ▶ Applications typically written in file called `application.py`
 - ▶ Behavior of whole site springs from here
- ▶ For C\$50 Finance
 - ▶ Vast majority of your code inside of the `application.py` file
 - ▶ Use this to define routes and behavior

A simple Flask application.py...

- ▶ To run, from within the directory containing application.py:

- ▶ `flask run`

- ▶ Click on **CS50 IDE** in the top left of the IDE and select **Web Server**

- ▶ Make sure you've run `update50` recently!

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def index():
    return "You are at the index!"
```

- ▶ If you make changes and save, no need to relaunch flask! (though you may want to reload your webpage)

Jinja

- ▶ Python inspired templating language
- ▶ Allows us to interweave HTML markup and Python
- ▶ Simple layout.html page
 - ▶ Some parts will be “fill-ins” using Jinja syntax
 - ▶ This is the template for your other pages
 - ▶ Other html files will fill in the blanks you left with Jinja
 - ▶ Like code based Mad-Libs!
- ▶ <https://realpython.com/blog/python/primer-on-jinja-templating/>

Passing info with HTML

- ▶ Two ways to pass info between Python and html
- ▶ GET
 - ▶ Information passed via URL
 - ▶ Ex: youtube uses GET to pass information since you want to copy and share URL
- ▶ POST
 - ▶ Passes data in the HTTP message body
 - ▶ Consider data to be “hidden” compared to GET requests
- ▶ FYI, neither of these are considered secure

SQL

- ▶ Four basic commands:
- ▶ UPDATE
 - ▶ Update data in database table
- ▶ INSERT INTO
 - ▶ Insert certain values into a table
- ▶ SELECT
 - ▶ Select values to view
- ▶ DELETE
 - ▶ Delete from table
- ▶ Go to <http://www.w3schools.com/sql/> for more tutorials

SQL - UPDATE syntax

- ▶ To update existing records in the table:

```
UPDATE table_name
```

```
SET column1=value1,column2=value2,...
```

```
WHERE some_column=some_value;
```

- ▶ Notice the WHERE clause; this is SUPER IMPORTANT
- ▶ The WHERE clause specifies which record or records should be updated
- ▶ Without the WHERE clause, all records will be updated!
- ▶ This is usually not what you want

SQL - INSERT INTO syntax

- ▶ Two ways to insert into

- ▶ Option 1: specify both the column names and the values to be inserted:

```
INSERT INTO table_name (column1,column2,column3,...)
VALUES (value1,value2,value3,...);
```

- ▶ Option 2: don't specify the column names where the data will be inserted, only their values:

```
INSERT INTO table_name
VALUES (value1,value2,value3,...);
```

SQL - SELECT syntax

- ▶ When selecting particular columns in a database:

```
SELECT column_name,column_name  
FROM table_name;
```

- ▶ When selecting all columns in a table:

```
SELECT * FROM table_name;
```

SQL - DELETE syntax

- ▶ Delete rows in a table:

```
DELETE FROM table_name  
WHERE some_column=some_value;
```

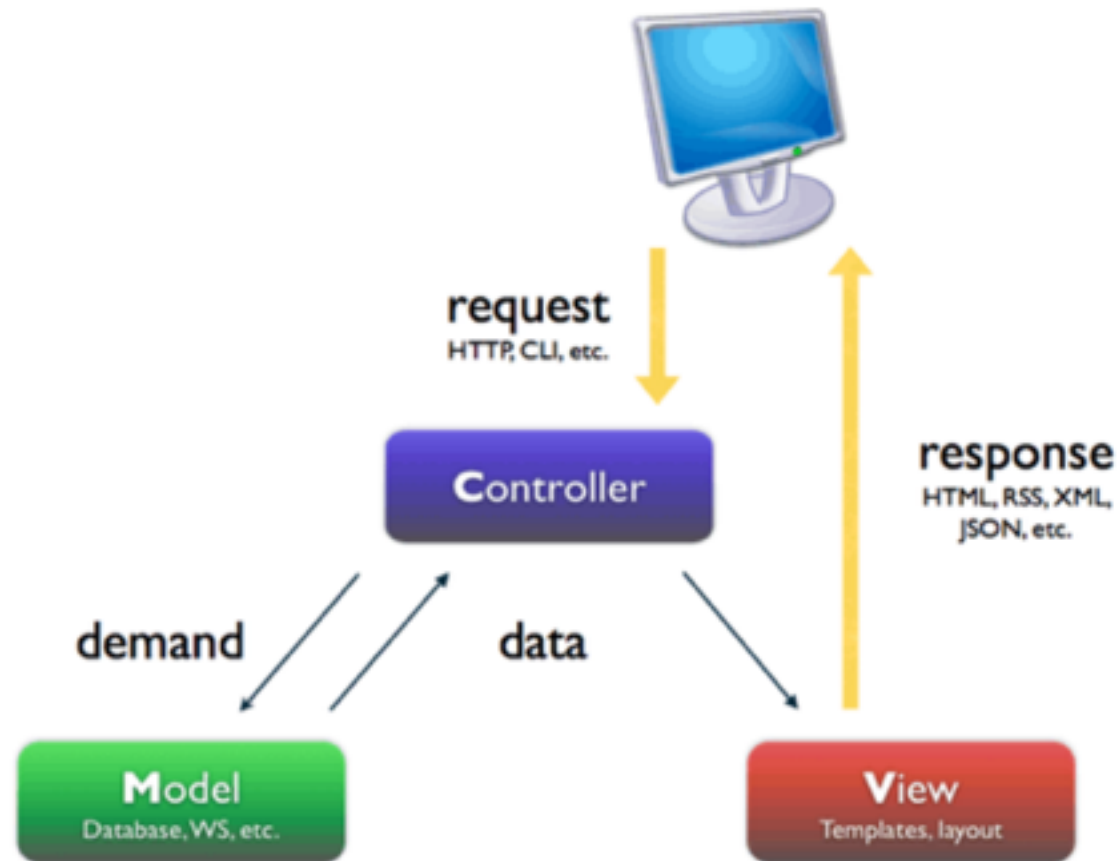
- ▶ Notice the WHERE clause; this is SUPER IMPORTANT
- ▶ It specifies which record/records should be deleted
- ▶ Without it, like with UPDATE, all records will be deleted

Your turn! SQL stufffff

- ▶ How would I insert a person whose name was Al and hometown was Boston into a database called CoolCats that included columns titled Name, City, and Pet. Which do you think is going to be this database's primary key?
 - ▶ `INSERT INTO CoolCats (Name, City)
VALUES ('Al', 'Boston');`
- ▶ How could I update Al's row include that he has a pet dog?
 - ▶ `UPDATE CoolCats
SET Pet='dog',
WHERE Name='Al';`
- ▶ Turns out Al isn't really a cool cat. How do I delete him from my database?
 - ▶ `DELETE FROM CoolCats
WHERE Name='Al';`

MVC

- ▶ Model View Controller
- ▶ design paradigm for creating software applications



- ▶ Image credit:
- ▶ http://symfony.com/legacy/doc/jobee/1_2/en/04?orm=Propel

MVC (in web design)

Component	Function	Example
Model	<i>-persistent storage of information -managing and organizing data</i>	<i>-MySQL database -data files</i>
View	<i>-presentation of information to user -user interface</i>	<i>-HTML -CSS -Jinja</i>
Controller	<i>-handles user requests -gets information from the model</i>	<i>-Python</i>

HTML & CSS

- ▶ HTML: HyperText Markup Language
- ▶ You get to practice and experiment :}
 - ▶ Check out <http://www.w3schools.com/> for useful tutorials!
- ▶ Best practices
 - ▶ Close all tags!
 - ▶ Note: real websites often don't, so if you're looking at a page's source for inspiration, be sure not to copy bad habits!
 - ▶ Validate your page with W3 Validator
 - ▶ <https://validator.w3.org/>
 - ▶ Separate markup (HTML) and style (CSS)
 - ▶ MCV (Model View Controller) paradigm to come!

HTML & CSS

- ▶ CSS: Cascading Style Sheets
- ▶ Instead of tags, CSS uses attributes
 - ▶ Selectors will be used to match tags with attributes
 - ▶ Map to tags
- ▶ Selectors can be
 - ▶ Id: unique
 - ▶ `#<idname>` in CSS file
 - ▶ Class: can refer to multiple blocks
 - ▶ `.<classname>` In CSS file

HTML & CSS: selection code

- ▶ reference a tag directly, such as <p>

```
p {  
    text-align: center;  
}
```

- ▶ reference the class, such as <p class="example">

```
.example {  
    color: blue;  
}
```

- ▶ reference its id, such as <p id="main">

```
#main {  
    background-image: url("rob_gym.jpg");  
}
```

HTML & CSS: Resources

- ▶ HTML Cheat Sheet
 - ▶ <http://lesliefranke.com/files/reference/htmlcheatsheet.html>
- ▶ CSS Cheat Sheet
 - ▶ <http://www.lesliefranke.com/files/reference/csscheatsheet.html>
- ▶ W3schools (for HTML and CSS tutorials)
 - ▶ <http://www.w3schools.com/>
- ▶ W3validator (run this like you would check50 and style50!)
 - ▶ <https://validator.w3.org/>
- ▶ Looking at the source code of pretty webpages, as with Google Chrome's "Inspect Element" tool (just right click on the interesting part of the webpage)

Together...

- ▶ Let's code up a webpage! Grab your computers and get ready to Google - syntax is the name of the game.
 - ▶ Check out the attached files for our class website!