



CENTER FOR TEXTUAL STUDIES AND DIGITAL HUMANITIES

DIGH 402 - Introduction to Digital Humanities Design and Programming

Spring Semester 2014

Week 4

GitHub Setup & Usage

General Setup - Part 1

- [Setup Guide](#)
- [Create a new repository on GitHub](#)
- [GitHub on Mac](#)

GitHub Setup & Usage

General Setup - Part 2

- register for an account on [GitHub](#)
- install [Git](#) on your local machine
- configure git on local machine using the terminal (command line)

```
git config --global user.name "Your name here"  
(**change to your preferred name for GitHub**)  
git config --global user.email "your_email@youremail.com"  
(**add the same email used to register at GitHub**)
```

- create a repository on GitHub (use an obvious project name)
- create a local repository for your Git projects

```
mkdir ~/MyProject  
(**use the same project name as the above new GitHub repository**)  
cd ~/MyProject  
(**change to the new directory**)
```

GitHub Setup & Usage

General Setup - Part 3

- in the same directory, MyProject, create a Readme file

```
touch Readme.txt
```

(**basically creates a blank file called Readme.txt - you can also use markdown etc**)

```
git init
```

(**this tells Git to recognise this directory as a local Git repository**)

```
git status
```

(**shows status at current master branch - 'untracked' = Git currently ignoring file(s)**)

```
git add Readme.txt
```

(**added file**)

```
git commit -m "Add Readme.txt"
```

(**commit files in directory with -m associated commit comment**)

- connect local repository to GitHub repository

```
git remote add origin https://github.com/username/myproject.git
```

(**add a new place where files originate - remote indicates origin**)

```
git remote -v
```

(**shows all remote origins for your current repository**)

GitHub Setup & Usage

General Setup - Part 4

- push initial changes to remote repository

```
git push -u origin master  
(**push changes for master branch of repository**)
```

- push changes made to files in local repository to remote repo

```
git add Readme.txt  
(**propose a single file change**)  
git add *  
(**propose all files in the current directory for change**)  
git commit -m "Update Readme.txt"  
(**commit file/file for local repo**)  
git push origin master  
(**push commit files to master branch of remote repo**)
```

GitHub Setup & Usage

General Setup - Part 5

- clone a remote repository

- cd to local directory for storing remote files eg: ~/github

- ```
git clone https://github.com/ancientlives/digh402.git
```

- (\*\*downloads a complete copy of the remote repository and sets it up as a local directory ready for use with Git\*\*)

- check and update local repository to match cloned remote repository

- cd to local directory for repository eg: ~/github/digh402

- ```
git pull
```

- (**downloads changes from remote repository and merges with local directory**)

[Further information](#)

MVC

Model-View-Controller - Part 1

- avoid chaotic spaghetti programming
- MVC defines a clean separation between critical components of an app
- user interface (UI)
 - present UI elements for the user such as buttons, forms, text fields...
- handling and reacting to the UI
- application needs to store the data eg: a database

MVC

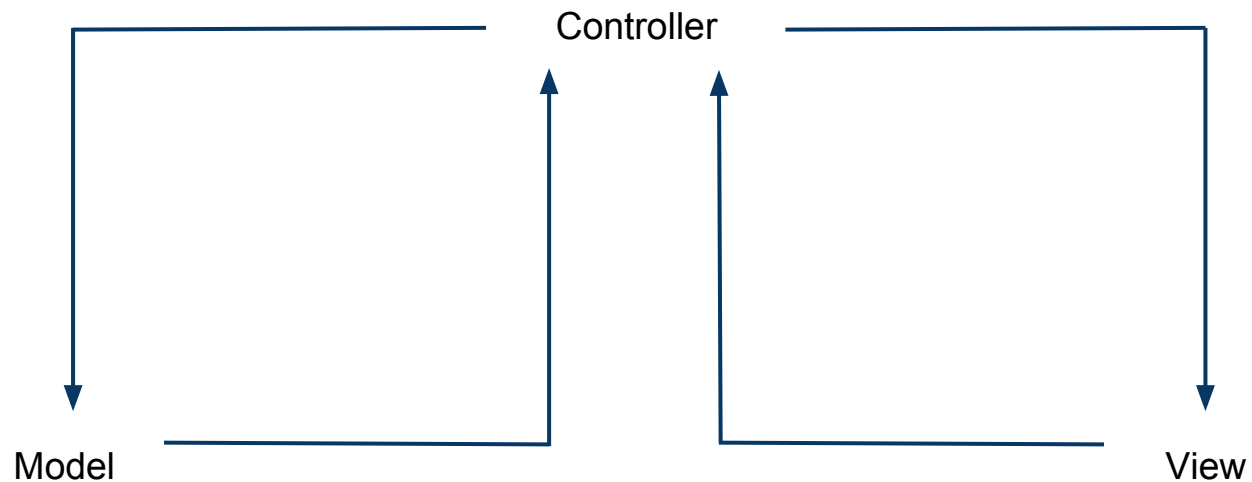
Model-View-Controller - Part 2

- defines three parts of an application called model, view, and controller
- Model
 - provides the underlying data and methods that offer information to the rest of the app
 - does not define how the app will look or how it will act
- View
 - includes different on-screen UI elements such as buttons, fields, switches...
 - multiple views make up the UI for an app
- Controller
 - manages the interaction and flow between the model and the view
 - handles actions such as user input (keyboard, mouse etc) and sends them to the model or view as required

MVC

Model-View-Controller - Part 3

402 Framework



Possible workflow patterns within the framework?

PHP and MySQL

Test basic PHP MySQL connection & SELECT queries - against USERS table

- modify basic connection to query 'users' table
 - SELECT username, usercreated FROM users;
 - SELECT * FROM users WHERE userid=3;
 - SELECT DISTINCT username FROM users;
- output results in a tabular format

[PHP Example](#)

PHP and MySQL

Test basic PHP MySQL connection & SELECT queries - tabular output

```
<?php
$con = mysqli_connect('localhost', 'admin402', 'digh402', '402framework');
if (!$con) {
    die('Could not connect: ' . mysql_error());
}

$result = mysqli_query($con,"SELECT * FROM users");

echo '<table>';
echo '<th>username</th>';
echo '<th>created</th>';

while($row = mysqli_fetch_array($result))
{
    echo '<tr>';
    echo '<td>'.$row['username'].'</td>';
    echo '<td>'.$row['usercreated'].'</td>';
    echo '</tr>';
}

echo '</table>';

mysql_close($con);
?>
```

PHP and MySQL

Test basic PHP MySQL connection & SELECT queries from content_lookup etc

- modify basic connection to query multiple tables

- SELECT content.contentid, content.contentname, users.username FROM content_lookup, content, users WHERE content_lookup.content_id=content.contentid AND users.userid=content_lookup.user_id AND content_lookup.user_id=1;

- output results in an un-ordered list

[PHP Example](#)

PHP and MySQL

Test basic PHP MySQL connection & SELECT queries from content_lookup etc

```
<?php
//currently set to basic user query privileges
$con = mysqli_connect('localhost', '402user', 'celine59', '402framework');
if (!$con) {
    die('Could not connect: ' . mysql_error());
}

$result = mysqli_query($con,"SELECT content.contentid, content.contentname, users.username FROM content_lookup,
content, users WHERE content_lookup.content_id=content.contentid AND users.userid=content_lookup.user_id AND
content_lookup.user_id=1");

echo '<ul>';

while($row = mysqli_fetch_array($result))
{
    echo '<li>contentid = '.$row['contentid'].' & contentname = '.$row['contentname'].' & username = '.$row['username'].'</li>';
}

echo '</ul>';

mysql_close($con);
?>
```

PHP and MySQL

Basic HTML output & queries - Part 1

- output all content currently available in the framework
 - all content
 - by content type (image & text)

[PHP Example](#) | [PHP Code Example](#)

****TO DO**** - ADD EXAMPLES FOR THE FOLLOWING USING FUNCTIONS CREATED...

- output all users currently registered in the framework
- output content per current user

PHP and MySQL

Basic HTML output & queries - Part 2

- output all users currently registered in the framework
- output content per current user

[PHP Example](#) | [PHP Code Example](#)

****TO DO**** - ADD EXAMPLES & REVISE FOR THE FOLLOWING USING LESS REDUNDANCY

- output all users currently registered in the framework
- output content per current user