



## CENTER FOR TEXTUAL STUDIES AND DIGITAL HUMANITIES

### 402 - Introduction to Digital Humanities Design and Programming

Spring Semester 2016

Week 9

## Dev Week - Wednesday 23rd March to Wednesday 30th March

- opportunity to test and work through concepts for 402framework and 402mini
- carefully read and review code for '402mini'
  - [GitHub - 402mini v0.1](#)
  - Review extra notes on [402mini - Part 1](#)
- carefully read and review code for v0.4 '402framework'
  - [GitHub - 402framework v0.4](#)
- create an outline and flow diagram to show routing within each framework (402framework v0.4 & 402mini v0.1)
  - consider how each framework allows a user to view their requested content (from the URL)
  - start with the 'index.php' file, and then follow the path of the app's logic to the rendering of the content in the user's browser
- brief presentation of work to class on Wednesday 30th March
- post updates to class' Trello organisation
  - [Dev week board](#)
- post any questions to the course's group on Slack

## Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

Initial Outline - basic theme templates and views for the framework

- added 'themes' directory to 'design' directory
  - updated directory.php constants for themes
- default template for framework includes both default theme, css, and javascript
- user themes will build on default template and replace default theme where necessary
  - css, javascript etc can be customised as required
- header metadata will be customised per use (eg: project installation)
  - meta settings in settings.php file
  - called in view.php draw\_head() method

[GitHub Code](#)

## Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

Initial Outline - updated /system/library/loader.php file

- we now need to initialise the theme using the `init_theme()` method
  - we require the `view.php` file to set the theme, theme dir, and allow us to check the current user select theme
- `init_theme()` method and setters in `view.php` are used to check and set the user theme
- `draw_theme()` method allows us to call various `view.php` methods to actually build and render our template and theme
- `load_controller()` method now checks selected theme, and then uses the `draw_theme()` method to draw the framework template with the user's selected content

[GitHub Code](#)

## Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

Initial Outline - /frame/view/html\_builder.php file

- parent class to view.php View class
  - allows abstracted HTML building for the framework viewer and template
- currently allows to build any specified HTML start and end element
  - two methods start\_element() and end\_element()
- start\_element() methods accepts two parameters for required HTML element with associated attributes and attribute values
- end\_element() method simply accepts one parameter for the required HTML element
- values are returned for output as required within template rendering, currently within View and Loader classes

[GitHub Code](#)

## Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

Initial Outline - /system/library/view.php file

- View class is currently the main rendering class for the 402 framework
- View class extends BuildHTML class thereby allowing it to use the HTML building methods
- setters for selected user theme
- various draw methods for defined structural components of the framework's HTML template
  - currently output start and end elements for each structural component (eg: head, body, header, footer...)
  - draw\_main() and draw\_middle() methods also accepts parameters for outputting content

[GitHub Code](#)

## Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

[v0.3](#)

## Object Oriented Programming

402framework - Database update for Week 9

- [SQL - Week 9](#)
- Update local database for new weekly import
  - drop existing 402framework
  - import 402framework.sql from week9 on GitHub
  - set collation for DB to utf-8\_unicode
  - check privileges for 402 users and DB
  - test with existing framework



## Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

constants.php file

- updated for default html framework and CSS and JS
- added constants for default HTML elements used within template
  - used with loader, view, and htmlBuilder
  - allows abstraction of element names for HTML markup of framework template
- added constants for default CSS and JS files for framework template
  - frame and grid CSS
  - frame and jQuery js

[GitHub Code](#)

## Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

/system/library/functions.php file

- class Functions stores methods (functions) for general framework wide usage
- required from loader.php file
- first function allows us to use an extended function of implode() to output attributes
  - array\_implode() implodes an array of element attributes to output a string of attributes for use with a specified element
- other general functions will be added

[GitHub Code](#)

## Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

### Initial Outline - default css and js file

- default css stylesheets and javascript functions for the framework
  - design/css/frame.css
  - design/css/grid.css
  - design/js/frame.js
- javascript will be built upon JQuery library
  - design/js/jquery.js
- selected personal theme will supplement and customise the default stylesheets and javascript functions
  - chosen theme added as parameter to init\_theme() method in bootstrap.php
- colours, javascript functions etc can be added in chosen theme directory
  - eg: design/themes/minimal etc

## Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

/design/css/frame.css

- basic css for structural elements of the framework
  - body
  - header
  - centre and node\_content
  - sidebar
  - footer
- basic css styling for framework layout including
  - body
  - headers (h1 to h6)
  - links

and so on...

[GitHub Code](#)

## Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

/design/css/grid.css

- allows framework template to be organised using a grid
- grid can be up to 12 divisions horizontally
  - eg: - a parent container specified as grid 12
    - two child divisions specified as grid 3 and grid 9
    - each grid is calculated as an approximate percentage upto 100%
    - small allowance for borders etc on each grids
    - required size of grid set as a value within the class attribute of the element

[GitHub Code](#)

## Object Oriented Programming

### Initial Outline - framework so far...

- [HTML template output Home Page](#)
- [HTML template output sample content](#)