



CENTER FOR TEXTUAL STUDIES AND DIGITAL HUMANITIES

DIGH 402 - Introduction to Digital Humanities Design and Programming

Spring Semester 2014

Week 11

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

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

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

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](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

theme.css and theme.js

- check and load user theme in the load_controller() method
- theme initialised in the init_theme() method in Loader class
- in draw_head() method in View class
 - conditional check for theme other than 'default', eg: 'minimal'
 - if conditional boolean is TRUE then draw_theme() method is called
- draw_theme() method draws selected theme CSS and JS files to HTML head element
 - this method checks for presence of theme.css and theme.js files in theme directory

[GitHub Code](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

theme.css

- override css styles set in frame.css
- modify size, colour, position etc of elements and classes
- customise framework design and layout as required

[GitHub Code](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

updated content loading - loader.php, content.php and new query_builder.php

- new class BuildQuery in frame/model/query_builder.php
 - adds generic sql query methods
- BuildQuery class now inherited by ContentController class in content.php
- load_controller() method in Loader class updated to output dummy content for non-returned content results
 - this allows the draw_theme() method to be loaded correctly without returned DB results

[GitHub Code](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

updated method scope for inheritance

- method scope has now been set to protected for parent class
 - methods in BuildHTML class in html_builder.php
 - methods in BuildQuery class in query_builder.php

[GitHub Code](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

initialise metadata for content (title/description) for selected route - single field query

- use `get_content_field()` method again with `$format` set to required table field
 - eg: `name` (for content title) and `desc` (for content description)
- add to new content metadata array `$content_meta` in Loader class
- pass `$content_meta` array to View class for use in drawing sidebar, content titles...
- output with `div` element and class as `$format` from `$content_meta` array
- style in default `frame.css` and override in `theme.css` if required

[GitHub Code](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

content metadata table and content lookup

- new DB table for start of content metadata
 - record metadata per content record initially based on Dublin Core
 - content and metadata record associated in content_lookup table
- output to sidebar for each content selected in current route
- metadata table can be updated with further columns to match
 - publisher
 - contributor
 - format

and so on...

[DB Example](#)

Object Oriented Programming

Object Oriented Programming - Abstract overview of current framework structure

initial load_meta method

- new load_meta() method in Loader class
 - accepts two parameters (\$controller and \$params)
- called in load-controller() method
 - after conditional check for content and before drawing theme
- method requires new MetadataController class in metadata.php class
- gets metadata results from DB for \$controller.meta table (eg: content_meta) for specified field
- merges returned array of DB results with Loader class property of \$content_meta
- \$content_meta can now be used to update and output metadata results in the sidebar

[GitHub Code](#)