Notes on Web Development Coding Standards

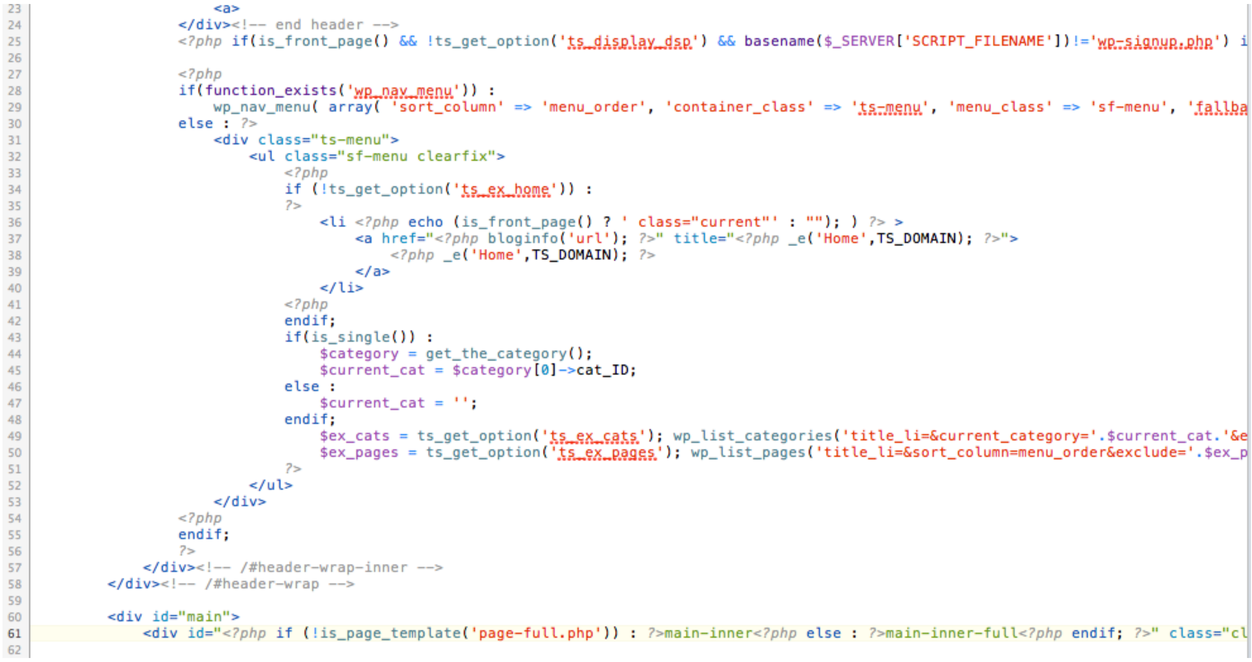
General coding

- No Line wrapping in code views
- Single Line spacing except after large blocks of code (usually if they warrant a closing label [see below])
- Use tabbing for indenting rather than spaces
    - Indent each nested element one tab.
        - User indentation right from opening `<HTML> or <?php>` tag
- Commenting:
    - Label closing tags for key building block elements (eg. Containers / body / header etc.) Use HTML comment: `<!--/` then '.' for class or '`#`' for ID.
    - When commenting around code logic etc., always use PHP commenting. Always favour PHP commenting. HTML comments visible in the source, PHP not.

PHP

- Dont allow extraneous php open and closing tags.
    - eg. `'?><?php First_function_with_no_html_following  ?><?php Second_function_with_no_html_preceeding ?>`
- Semi-colon should *always* mark the start of a new physical line.
    - Eg. Avoid `Line 1; Line 2;` on one line.
- Use variable names on the right side in 'If' cases. Eg. "`(x == $variablename)`" and not "`($variablename == x)`"
- If Statements: Favour colon ahead of curly brackets ("`{}`"). eg.

  ```
  if (is_single()) :
  	Action;
  else:
  	Action;
  endif;
  ```
- Use PHP echo statement rather than hard-coded HTML in order to support multi-lingual etc.
    - eg. `<?php echo__('Test') ?> or <?=__('Test') ?>`.
    - Avoid `<? echo__('Test') ?>`



```
            <a>
        </div><!-- end header -->
        <?php if(is_front_page() && !ts_get_option('ts_display_dsp') && basename($_SERVER['SCRIPT_FILENAME'])!='wp-signup.php') i

        <?php
        if(function_exists('wp_nav_menu')) :
            wp_nav_menu( array( 'sort_column' => 'menu_order', 'container_class' => 'ts-menu', 'menu_class' => 'sf-menu', 'fallba
        else : ?>
            <div class="ts-menu">
                <ul class="sf-menu clearfix">
                    <?php
                    if (!ts_get_option('ts_ex_home')) :
                    ?>
                        <li <?php echo (is_front_page() ? ' class="current"' : ""); ) ?> >
                            <a href="<?php bloginfo('url'); ?>" title="<?php _e('Home',TS_DOMAIN); ?>">
                                <?php _e('Home',TS_DOMAIN); ?>
                            </a>
                        </li>
                    <?php
                    endif;
                    if(is_single()) :
                        $category = get_the_category();
                        $current_cat = $category[0]->cat_ID;
                    else :
                        $current_cat = '';
                    endif;
                        $ex_cats = ts_get_option('ts_ex_cats'); wp_list_categories('title_li=&current_category='.$current_cat.'&e
                        $ex_pages = ts_get_option('ts_ex_pages'); wp_list_pages('title_li=&sort_column=menu_order&exclude='.$ex_p
                    ?>
                </ul>
            </div>
        <?php
        endif;
        ?>
    </div><!-- /#header-wrap-inner -->
</div><!-- /#header-wrap -->

<div id="main">
    <div id="<?php if (!is_page_template('page-full.php')) : ?>main-inner<?php else : ?>main-inner-full<?php endif; ?>" class="cl
```

Naming Standards

- CSS class and ID definitions: lower-case and hyphenated (not underscore)
- PHP variable naming. Lower case and underscored.
- PHP function names. Single word, capitalize new words.
- Never have multiple instances of the same ID in a document. ID's should be exist in a document once.

CSS

- Always check that there isn't a framework (eg Bootstrap) class that will do the required work.
- Always use an over-ride css file rather than the framework's default css file (eg. not Bootstrap.min.css)
- CSS files: indent within every set of curly brackets. So media queries tabbed in.
- Use Bootstrap media queries as default media-query framework.

General

- Every page needs a unique H1. Consider hierarchy of headers based on importance. Use Screaming Frog to check.