Thutong Coding Standards Document

This document contains the coding conventions created by theDigital Blacksmiths team, based on the:
html,
php,
css,
javascript (Ajax, etc)
-programming languages as well as SQL database setup.

File Names

As in the html, php, Javascript, sql language coding standards, source files have the either the:
.php extension
.js extension
.html extension
.sql extension
.css extension

Source Files

Each source file that contains html, php should be able to run html, php, css and javascript accompanying languages.

Html, php, javascript source files have the following ordering:

1. Beginning comments
2. Package and Import statements
3. Class Header and Declaration
4. Method Headers and Declarations

Comments

These should describe what the file should do within all file types as well as to describe functions and elements of the coding of the website.

Package and Import Statements

The first non-comment line of most html/php source files is a package or import description and functionality statement as follows:

1. <!DOCTYPE html>
2. <html> // describes type
3. <head> // defines head of document
4. <body> // actual website coding is written here.

After that, import statements can follow. For example:

1. Externally - <link rel="stylesheet" href="styles.css"> imports the styles.css documents styles.
2. Internally - by using a <style> element in the <head> section.
3. Inline - by using the style attribute in HTML elements.

Headers and Declarations

Html, php

All source files should contain a comment that lists the file name, attributes, functionality, visibility and the statement which declares its description/module of which it is a part of.

```
<!—**********************************************************
        * File name:
    * Attributes:
    * Functionality:
    * Visibility:
    * Module:
    **********************************************************—>
```

.js Methods and files

Same as above:

```
<!—**********************************************************
        *  File name:
    *  Attributes:
    *  Functionality:
    *  Visibility:
    *  Module:
    **********************************************************—>
```

Lines and Indentation

Four spaces should be used as the unit of indentation.

Line Length

Avoid lines longer than 150 characters in length, enter the rest into a new line.

Wrapping Lines

When an expression will not fit on a single line, break it according to these general principles:

1. Break after a comma.
2. Break before an operator.
3. Align all new line

Comments

Implementation Comment Formats

Programs can have four styles of implementation comments:

Block Comments

1. Block comments are used to provide descriptions of files, methods, data structures and algorithms.
2. A block comment should be preceded by a blank line to set it apart from the rest of the code.

Single-line comments

1. Short comments can appear on a single line indented to the level of the code that follows.

End-Of-Line Comments

Examples of all three styles follow:

Note: HTML,PHP comment make use of <!— comment here —>

```
        //if  ( bar > 1 ) {
//
//    // Do a triple-flip.
//   ...
        //   ...
//}
//else {
//    return false;
//}

if  ( shoo > 1 ) {
    // Do a double-flip.
                ...
}
else {
                    return false;          // Explain why here.
}
```

Functions and Declarations

Variable names should be sorted by type and in alphabetical order if possible.
Javascript and accompanying languages example:

```
1.    var    level;    // level
2.    var    size;     // size
```

var can describe any variable type.
Php variable names are as follows:

```
1.    var    $level;    // level
2.    var    $size;     // size
```

Braces and functions should be as follows

```
// Opening brace to the left
function foo()
{
	return
	{
	bar: true
	};
}
```

Naming conventions with layout example

1. **Methods**
   drawColor();
2. **Classes**
   class SledgeHammer;
3. **Constants**
   Programs must be written for people to read.

Full layout example in javascript

Note: use of CamelCase

```
var calculateAverageStringLength = function()
{
	var stringLenght = 0;
	var averageLength;

	if ( arguments.length > 0 )
	{
		for ( var i = 0; i < arguments.length; i++ )
		{
			stringLenght += arguments[ i ].length ;
		}

		averageStringLength = stringLenght / arguments.length ;
	}
	else
	{
		averageStringLength = 0 ;
	}

	return averageStringLength ;
}
```

HTML and CSS snippets

Note: utilises the BEM methodology.

```
h4 {
  font-size: 22px;
}

::selection {
  color: #fff;
  background: #17bed2;
  text-shadow: none;
}

::-webkit-selection {
  color: #fff;
  background: #17bed2;
  text-shadow: none;
}

<html lang="en">
    <!-- BEGIN HEAD -->
    <head>
        <meta charset="utf-8"/>
        <title> Freebie </title>
    </head>

<ul class="navbar-nav navbar-nav-right” >
            <li  class=“listItem” > </li>
            <li  class=“listItem” > </li>
            <li  class=“listItem” > </li>
</ul>
```

References:

1. Zakas, Nicholas. Why Coding Style Matters [online]. October 2008.
URL: https://www.smashingmagazine.com/2012/10/why-coding-style-matters/. Accessed 9 April 2016.

2. Burden, Paul. Coding Standard Compliance – Some Facts and Some Fallacies [online]. March 2012.
URL: http://www.programmingresearch.com/resources/seminars/coding- standard-compliance-some-facts-and-some-fallacies/. Accessed 10 April 2016

3. WordPress.org. PHP Coding Standards [online].
URL: https://make.wordpress.org/core/handbook/best-practices/coding- standards/php/. Accessed 2 April 2016.

4. W3schools. JavaScript Functions [online].
URL: http://www.w3schools.com/js/js_functions.asp. Accessed 9 April 2016.

5. StackOverflow. Why are dashes preferred for CSS selectors / HTML attributes [online].
URL: http://stackoverflow.com/questions/7560813/why-are-dashes-preferred-for- css-selectors-html-attributes. Accessed 8 April 2016.

6. StackExchange. Creating a coding standards document [online].
URL: http://programmers.stackexchange.com/questions/196706/creating-a- coding-standards-document. Accessed 10 April 2016.