# PHP

Files MUST use only <?php and ?> tags.

Files MUST use only UTF-8 without BOM for PHP code.

Files SHOULD either declare symbols (classes, functions, constants, etc.) or cause side-effects (e.g. generate output, change .ini settings, etc.) but SHOULD NOT do both.

Namespaces and classes MUST follow an "autoloading" PSR: [PSR-0, PSR-4].

Class names MUST be declared in StudlyCaps.

Class constants MUST be declared in all upper case with underscore separators.

Method names MUST be declared in camelCase.

# C#

**Layout Conventions**

Use the default Code Editor settings (smart indenting, four-character indents, tabs saved as spaces). For more information, see Options, Text Editor, C#, Formatting.

Write only one statement per line.

Write only one declaration per line.

If continuation lines are not indented automatically, indent them one tab stop (four spaces).

Add at least one blank line between method definitions and property definitions.

Use parentheses to make clauses in an expression apparent, as shown in the following code.

**Commenting Conventions**

Place the comment on a separate line, not at the end of a line of code.

Begin comment text with an uppercase letter.

End comment text with a period.

Insert one space between the comment delimiter (//) and the comment text, as shown in the following example.

# HTML & CSS

All HTML tags should be in lower case

Eg:

<table></table>

<form></form>

Use meaningful names for ID's and Names of HTML elements

Use meaningful ID's and names for HTML elements Eg:

txtName, txtAge

instead of

text1, text2

Indent HTML code consistently

Indent code consistently. eg:

<body>

<form name="frmActivity" method="post" action="<?php echo $formAction;?>">

<input type="hidden" name="sqlState" value="">

<input type="hidden" name="delState" value="">

<input type="hidden" name="activityId" value="">

<label for="cmbProjectId"></label>

<select name="cmbProjectId">

<option value="value1"></option>

</select>

</form>

</body>

Maximum line length

Limit maximum line length to around 120 characters. Wrap code if longer than that. This will improve readability of code.

Eg: Wrapping code to limit line length

<div id ="addActivityLayer">

<img onClick="<?php echo $saveBtnAction; ?>;"

onMouseOut="this.src='../../themes/beyondT/pictures/btn\_save.gif';"

onMouseOver="this.src='../../themes/beyondT/pictures/btn\_save\_02.gif';"

src="../../themes/beyondT/pictures/btn\_save.gif"/>

</div>

Only use tables for tabular data, use CSS for styling

Do not use tables for layout, use CSS instead.

Do not heavily comment HTML code

Adding too much HTML comments adds to the overhead in bandwidth. Therefore comment sparingly in HTML and use descriptive element names to improve readability.

Do not use inline style attributes

As much as possible, use css classes instead of using inline style attributes. This has the advantages:

Makes it easier to change the look of the UI by changing the stylesheet

Makes it possible to reuse the css class in another place, giving a more consistent UI and making it possible to change the look by changing only one place.

<!-- INSTEAD OF -->

<div style="width:100px;align:center;">

<!-- USE -->

<div class="message">

Use external CSS files instead of style tags in the HTML page

Use external css files, instead of using inline css in the HTML page by using <style> tags. This gives the same advantages as in #Do not Use inline style attributes above.

Refer to images from CSS files when possible

Referring to images from CSS files instead of directly using from the HTML code makes it easier to change the look of the application by changing stylesheets.

All HTML code should validate as XHTML 1.0

Document type

Use the following document type

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

How to validate

You can easily validate html in OrangeHRM using Firefox's "Web developer toolbar". First open just the main frame (without the top level menus etc. - "right click - this frame -> show only this frame" in firefox) and use the 'Tools->Validate Local HTML" option in the web developer toolbar.

Some common issues to look out for

Tables, cells not closed properly

Elements not closed eg:

<input type="text >

Fixed by changing to:

<input type="text" />

Attributes not quoted: eg:

<option value=0>Select</option>

Fixed by changing to:

<option value="0">Select</option>

height attribute in tables:

<table height="100">

If height is needed, can be converted to a style: eg:

<table style="height:100px">

Note that external css is preferred to inline style attribute. So the preferred solution would be:

<table class="abc">

and a css rule

table.abc {height: 100px;}

Tables with empty bodies also do not validate. eg:

<table></table> or

<table><tbody></tbody></table>

Therefore, make sure that the table has some rows or is completely omitted, in the case where there is no data

<script> tag for inline javascript should be as follows:

<script type="text/javascript">

//<![CDATA[

function test() {

}

//]]>

</script>

Note that

<script language="Javascript">

which we have in some places is not valid.

All url's in HTML (not in javscript code) should use &amp; instead of &

eg:

./lib/controllers/CentralController.php?uniqcode=EMP&amp;VIEW=MAIN

All <img tags should have the alt attribute. You can use alt="" for images used just for style.

<textarea> should have cols and rows attributes. eg:

<textarea cols="25" rows="3" name="xx"></textarea>