508ThemeDev

I dont see the point of this page. All 508 related items seem to be going into the main HEAD tpls which means there will be no specific 508 theme. as all tpls and css's under the current commits will attempt to be 508 - TikiGod

We are streamlining a theme that should be an example for beauty and speed. In other words, we're introducing clean XHTML and thought-through CSS. This page summarizes what should be considered while shaping the code. Check out Style508 for more information.

Table of Contents

Table of contents

content

XHTML

Tables

- `<table>` gets some new tags:
  - `colgroup` (for complicated layouts)
  - `thead`
  - `th`
  - `tfoot`
  - `tbody`
  - `summary` (only tables used for data-listings)

Forms

- `<form>` gets:
  - `<fieldset>`

Eliminate

- `class="tableheading"`
- `class="normal"`
- `border="0"` in `<img>` but leave it for tables (tables are taken care of later)
- `hspace="8"`

Change

- `td class="heading"` to `th` (without the class)

And ...

- add comments to make source code easy to understand, but only use Smarty syntax: `{* comment *}`
- *all* links get a `title=""` that start (ideally) with something like "Click here to ..."
- *all* images get `alt=""`, but *no* title attribute
- use `attr="value"` instead of `attr='value'`

...page... *Wiki page pagination has not been enabled.*
Where do we compromise?

||**tag/attr**|**compromise**|**explanation** `<input type="button">`|insert `class="button"`|Despite CSS2's attribute selectors (-+input[type=button] {...})+- add a class to input elements that represent their type. Now it's possible to style different input fields for IE users, too.
Templates

hardcoded link delimeters

Remove **[**
CSS

Class hierarchy

Chunks of template should be grouped by classes. Example: a form is used to let the admin control newsletter settings. The classnames should now follow a certain hierarchy (yet to be found), not based on "how does the chunk look" but based on "what is the chunk there for". Good name: "adminsettings". A wider scope like a surrounding `<div>` might be named "newsletter". An abbreviation like "admset" is not too cool. Now the form has a one or a bunch of fieldsets. Their classes should be automatically set by a cycling Smarty function. Like this, a theme designer can easily overwrite defaults that are global for certain form elements. He or she would just have to use child selectors, which are widely supported, even by Internet Explorer. So the idea is be to start with a specific class (newsletter) and then get more and more unspecific (adminsettings) until when only the tag is left without class (like `<textarea>`, it doesn't need a class if we know it's in newsletter - admin - fieldset 2).

tables vs rows vs cells

td-classes should be removed. tr-classes should be standardized - there's hardly another purpose then to alternate row-colors. This should be acheived with a global concept of letting Smarty cycle through rows and give class names like row1 and row2. Theme-admins can now set colors using child selectors like so: `table.mytable tr.row1 {foo}` Styling of whole columns would be achieved through selecting the col tags in colgroups. Styling of single cells would be achieved through [http://www.w3.org/TR/REC-CSS2/selector.html#adjacent-selectors|adjacent sibling selectors]. Like this, the less common a styling attempts is, the less common is the browser supporting it.