

Technical Overview

Macro Based Development

HTML standards

- Use an XHTML transitional doctype `{<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">}}`
- Validate all pages through the W3C validator <http://validator.w3.org/>
- While avoiding "div-itis" use of wrapper divs is preferable to exotic CSS techniques
- Use HTML layouts that allow for source-order independence when layouts involve up to three content columns
- Use semantically appropriate XHTML tags for elements where appropriate, use `div` or `span` tags when no other XHTML tag is available
- Use anchor tags, buttons, or input tags when defining clickable items (even when using javascript to handle the click handling)

Javascript Standards

- Use the YUI javascript libraries <http://developer.yahoo.com/yui/>
- Never use non-namespaced libraries such as prototype/scriptaculous
- Use object-oriented javascript
- Namespace your classes and all javascript code, never use non-namespaced code.* Use configuration objects as constructor parameters when a large number of parameters are required, especially when those parameters are optional
- Use the `YAHOO.util.Event` handlers to initialize your objects on page load, on dom ready, or on content ready, or use a script block inline in the page immediately following the required XHTML elements to call the object constructor.
- Never explicitly set event handlers in the XHTML code (eg: `onclick="foo();"`) Instead use the `YAHOO.util.Event` utility to assign event listeners
- Use the CSS classes `".none"` and `".hidden"` to set `display:none;` and `visibility:hidden` respectively, remove the classes to turn elements back on
- Use gracefully-degrading javascript

CSS standards

- Use Yahoo User Interface Library's `fonts-min.css` and `reset-min.css` to maintain a consistent base font-size and element attributes across sites <http://developer.yahoo.com/yui/>
- IDs are initial-cap camel case: "MyIdForSomeElement"
- Class names are lower-initial camel case: "myClassNameForABunchOfElements"
- Avoid initialisms or acronyms in class names and IDs
- Choose IDs and Class names semantically based on the content they represent
- Never use IDs or Class names that indicate a certain layout or position, for example, never words like "right" "left" "top" "bottom" or other placement-specific words
- Never use IDs or Class names that indicate a certain visual appearance, for example never use words like "blue" "bold" "underline" or other visually-descriptive words
- Never use sequentially numbered IDs or Class names, for example never use class names such as "box1" "box2" "box3" etc. Use descriptive class names instead.
- Never use IDs or Class names that indicate the type of element, for example, never use class names such as "contentDiv" or "highlightSpan"
- Explicitly set all values for shortcut declarations like `padding`, `margin` or `border`, ie: use `padding:4px 0 4px 0` instead of `padding:4px 0 4px 0 padding:4px 0`

- Always specify the units for padding, margin or border unless the value is zero, in which case do not specify the unit
- Always set non-zero border widths in pixel units
- Always specify font sizes as either percentages or ems (123% == 1.23em), never use pixel values. Use the YUI Font size chart to choose percentages/ems <http://developer.yahoo.com/yui/fonts/>
- Put each style selector on its own line, for grouped selectors, insert a single line break between the comma and the next selector
- In addition to what is defined in the YUI reset and fonts stylesheets, the following classes must be defined: `.none{display:none;}` `.hidden{visibility:hidden;}` as these are used by javascript classes.

CSS guidelines

- Use either self-clearing style rules or clearing div elements `<div class="clear"></div>`, as appropriate
- Do not explicitly set the height on any element which contains text
- Avoid explicitly setting the height of any element
- Avoid explicitly setting the width of an element (other than, say 100% or auto). Especially avoid setting the width of an element which is contained within another element who's width is explicitly set. Use padding or margin instead where appropriate.
- Do not use the `@import` declaration, link all stylesheets normally
- Avoid using `* html` to hide styles from IE 6, use conditional comments instead
- Do not float elements unless necessary to position that element. Especially avoid floating elements and setting their width to 100%
- Avoid positioning elements, especially avoid absolute positioning
- Avoid scoping CSS selectors by element type, for example only use `div.someStyle{...}` when you absolutely need to distinguish between div elements of class "someStyle" and other elements. In all other situations use only `.someStyle`

HTML Guidelines

Html is rendered using velocity as the scripting language and a set of in house macro libraries.

Key to this library is how input fields for forms are generated using the `#drawField` macro.

Example of html form output, all fields are generated by macros, but styles should be present.

FAQs

- Confirm if we need to copy the whole HTML from the form link and not edit the HTML 100%, Instead, we just need to supply them the CSS that implements the design.

Not correct. It is your option - as outlined in the original email what portion of the page you want to reuse. There are three levels of re-usability, of which no re-usability is not an option.

Level 1: Reuse everything between `<content-main>` tags.

Level 2: Reuse everything between `<content-form>` tags.

Level 3: Reuse everything between `<content-fields>` tags.

So with re-use that means not changing a single piece of HTML between those start and end tags.

- Ask if they have sample code for a completed form/design.

<http://www.taxreliefheadquarters.com/>

- Could you please specify which web font we should use for the text at your pages?

STANDARD FONTS

- Can we put 'lorem ipsum' placeholder text for the text at the pages as all the text is rasterized?

YES

- Could you please specify what you mean by "While avoiding "div-itis" use of wrapper divs is preferable to exotic CSS techniques"?

Using wrapper divs to apply a certain formatting vs. more advanced options.

- Referring your note: "Use HTML layouts that allow for source-order independence when layouts involve up to three content columns" could you please specify if you mean that you are going to add columns to your pages?

Does not apply to most of our designs.

- We will investigate your requirements to JavaScript and get back to you with more information.

What is the product here - don't think there will be any javascript needed.

- We will investigate your requirements specified in HTML Guidelines and get back to you with more information.

Do not worry about fields - those are autogenerated and can be added anytime after the product works.

