HTML-CSS-JS Guidelines

Article Number: 121 | Rating: 5/5 from 2 votes | Last Updated: Wed, Mar 30, 2016 at 1:07 PM

HTML Guidelines

- Doctype

  A document type declaration, or DOCTYPE, is an instruction that associates a particular SGML or XML document (for example, a webpage) with a document type definition (DTD) (for example, the formal definition of a particular version of HTML). In the serialized form of the document, it manifests as a short string of markup that conforms to a particular syntax.

  - In HTML 4.01, the declaration refers to a DTD, because HTML 4.01 was based on SGML. The DTD specifies the rules for the markup language, so that the browsers render the content correctly.
  - HTML5 is not based on SGML, and therefore does not require a reference to a DTD.
  - For HTML4.x it is compulsory to define a doctype
  - For HTML5.x doctype is not necessary. Read more

- Head

  The element is a container for all the head elements. The element can include a title for the document, scripts, styles, meta information, link, base, noscript.

    - Title

      The <head> tag is required in all HTML documents and it defines the title of the document. The <head> element: defines a title in the browser toolbar. Each page should have unique title value.

    - Metatags
      - keywords

        The meta keywords tag is one of several of meta tags that you can insert into your web pages to provide search engines with information about your pages that isn't visible on the page itself.

      - description

        The meta description tag in HTML is the 160 character snippet used to summarize a web page's content. Search engines sometimes use these snippets in search results to let visitors know what a page is about before they click on it.

      - charset - set to utf8

        HTML5 has a new attribute, charset, which makes it easier to define charset:

        HTML 4.01: <meta http-equiv="content-type" content="text/html; charset=UTF-8">

        HTML5: <meta charset="UTF-8">

      - Social Media Tags

        Social media meta tags are HTML tags that allow you to make the most out of the content you share from your site. You can determine what information is displayed in the post on Twitter, Facebook, Pinterest and beyond. If your posts, tweets and pins show the right information and are neatly organized, they are much more likely to get more engagements and re-shares.

The basics of Facebook Open Graph markup include:

- og:title = title or alternate title of post
- og:url = URL of post
- og:description = two to four sentences describing post
- og:image = URL to unique image at least 1200×630 pixels
- og:type = article (otherwise defaults to “website”)

The basic Meta tags for Twitter are:

- twitter:card = Card type (full list of card types here)
- twitter:title = Title or alternate title of post
- twitter:url = URL of post
- twitter:description = Brief description in less than 200 characters
- twitter:image:src = URL to unique image of at least 280×150 pixels

  - Should have closing tag if html 4.x
  - Scripts

    The HTML <script> Tag. The <script> tag is used to define a client-side script, such as a JavaScript. The <script> element either contains scripting statements or it points to an external script file through the src attribute.

    - Should not be present in head tag unless project specific requirement
    - Google Analytics code should be added at the end of the page code (unless specified by client)
    - Define type in script tag

- Body

  HTML Tag: body. The main content area of an HTML document. You must use this element and it should be used just once. It should start immediately after the closing head tag and end directly before the closing html tag.

    - HTML 5 Semantics

      A semantic element clearly describes its meaning to both the browser and the developer.

      Refer Link Click here

      - Learn more about HTML5 semantics, where they are used and when to use it on MDN HTML5 documentation
    - Div

    HTML - Div Element(s) The

    tag is nothing more than a container unit that encapsulates other page elements and divides the HTML document into sections. Web developers use elements to group together HTML elements and apply CSS styles to many elements at once.
      - Use this tag for page layout
    - Images

The <img> tag defines an image in an HTML page. The <img> tag has two required attributes: src and alt.

Note: Images are not technically inserted into an HTML page, images are linked to HTML pages. The <img> tag creates a holding space for the referenced image.

Tip: To link an image to another document, simply nest the <img> tag inside <a> tags.

    - Default image should be specified as placeholder
    - alt and title tags should be present
    - Width should be present
  - Forms

    The <form> tag is used to create an HTML form for user input. The <form> element can contain one or more of the following form elements:

    <input>, <textarea>, <button>, <select>, <option>, <optgroup>, <fieldset>,

    <label>.

    - Method should be defined
    - Define appropriate action
    - Input ID & Name should follow the naming conventions as follows(except for generated code):
      - Text - txt_
      - Textarea - ta_
      - Radio Buttons - rd_
      - Checkboxes - chk_
      - Select - sel_
      - Option - opt_
      - Button - btn_
      - Submit - sub_
      - Reset - res_
      - Email - email_
      - Phone - phone_
      - Image - img_
      - Number - num_
      - Hidden - hid_
      - Label - lbl_
  - Lists
    - Ordered <ol>
      - Use as data lists
    - Unordered <ul>
      - Use <ul> to aid user navigation elements such as Menus
  - General Guidelines
    - Use '-' separate words to name other elements: Ex: <div id="user-info" class="blue-red"></div>
    - ID should be unique
    - While development: Define tab as "4 spaces" or as defined by the framework. Properly indent the code.
    - All tags should be closed properly
    - Validation should be performed by W3C validator or an equivalent tool
    - Conform Web Content Accessibility AA standards
    - Use tools such as Web Developer Checklist to validate against standard practices
  - Tables (For Grids Only: Do not use table tag for layout purpose)

    The HTML tables allow web authors to arrange data like text, images, links, other tables, etc. into rows and columns of cells. The HTML tables are created using the <table> tag in which the <tr> tag is used to create table rows and <td> tag is used to create data cells.

    - Use Table Headers when appropriate
    - Use captions when to add table descriptions
    - Numbers should be center aligned and text should be left-aligned
    - Table headers should be bold
  - Links

    The <a> tag defines a hyperlink, which is used to link from one page to another. The most important attribute of the <a> element is the href attribute, which indicates the link's destination. By default, links will appear as follows in all browsers: An unvisited link is underlined and blue.

    - External links should open in new window
    - Always define title tags
    - Use name for in-page links
  - Headers

    The <h1> to <h6> tags are used to define HTML headings. <h1> defines the most important heading. <h6> defines the least important heading.

    - Use h1 tag is compulsory and only to be used once
    - Header tags should appear in ascending order. Example: H2 should appear after H1 and so on.
  - IFRAME

    An inline frame is used to embed another document within the current **HTML** document.

    - Use IFRAMEs sparingly, as it poses security and theming challenges

CSS Guidelines

- Minify custom CSS code whenever appropriate. Check out the tools section for solutions.
- Use CDNs for framework/library stylesheets whenever available and appropriate
- Avoid inline CSS
- Use sprites instead of separate images whenever appropriate
- Use CSS preprocessors such as LESS or SASS for code re-usability
- Avoid styling IDs, prefer classes
- Comment CSS whenever appropriate, remove these before deployment by using compression tool
- Use media-queries to ensure that site remains responsive, frameworks such as twitter bootstrap, have predefined breakpoints use those as base

Javascript Guidelines

  - Avoid using global variables, use closures to avoid exposing variables to global scope

    In JavaScript, objects and functions, are also variables. And any variable declared outside a function has global scope.You probably heard that because JavaScript is a [illegible] dynamically type language global variables are bad and that they easily become source of confusion and bugs. So how can we avoid of creating any global variable, but still have our code executed?

There is such technique, it's called Immediately Invoked Function Expression (IIFE).

Suppose we have this JavaScript code:

```
var func = function(){
console.log("entering func");
var counter = 0;
var innerFunc1 = function(){
counter += 1;
console.log("innerFunc1: " + counter);
}
var innerFunc2 = function(){
counter += 1;
console.log("innerFunc2: " + counter);
}
return {
inner1: innerFunc1,
inner2: innerFunc2,
};
}
var f = func();
f.inner1();
f.inner2();
```

So here we have two global variables - func and f.

Following is the code which we can run without creating ANY global variable by applying ITFE.

```
(function() {
var func = function() {
console.log("entering func");
var counter = 0;
var innerFunc1 = function() {
counter += 1;
console.log("innerFunc1: " + counter);
}
var innerFunc2 = function() {
counter += 1;
console.log("innerFunc2: " + counter);
}
return {
inner1: innerFunc1,
inner2: innerFunc2,
};
}
var f = func();
f.inner1();
f.inner2();
}());
```

- Always define variables before using, not defining them assigns variable to global scope
- Ensure the licenses of the script allows commercial reuse. If commercial/proprietary scripts are to be used, please get approval from appropriate stakeholder.
- Scripts such as jQuery should be only included once, do not include multiple versions of same library, use features such as jQuery’s noConflict to counter negative effects.
- Javascritps should be combined and minified before deployment
- Use CDN or download scripts locally, do not hotlink to other site.
- Do not call asynchronous code within loop without using closures
- Use JSONP for cross domain ajax requests
- In Ajax, define type for the data returned by the server, prefer JSON
- While iterating the through object properties use .hasOwnProperty() to ensure chained properties aren’t iterated
- Do not use eval function, for parsing json use JSON library or browser object.
- Use short hand notations such as a = [] and a = {} instead of new Array() an new Object()
- Use ‘===’ for comparison
- Ensure default values for function parameters
- Use namespaces -- learn more
- Avoid deprecated JavaScript and library functions such as .live() function of jQuery
- Never include Javascript events as inline attributes. This practice should be completely wiped from your mind.

    <a onclick="doSomething()" href="#">Click!</a>
- Never depend on javascript to deliver information
  Example: document.write('Good Afternoon!');
- Don't Pass a String to "SetInterval" or "SetTimeOut"
- Use {} Instead of New Object()
- Use [] Instead of New Array()
- Always, Always Use Semicolons
- For debugging avoid using alert, instead use console.log
- Cache Objects / Memoization
- Native Loops - Avoid using looping provided by libraries e.g. jQuery.forEach, favor native loops instead
- Combine and minify JS files
- Stick to a Strict Coding Style
  - Validate your code: http://www.jslint.com

Tools

All the tools allow you to create a "build system" with common task such as combining and minifying JS, CSS files, performing lint operations.

- Grunt
  - Learn More: http://gruntjs.com/getting-started
- Gulp
  - Get started with Gulp
- Elixir
  - PHP based fluent API built on top of Gulp, Frameworks like Laravel include this.

Posted by: PHP - Administrator - Wed, Mar 30, 2016 at 1:07 PM This article has been viewed 446 times.
Filed Under: Knowledge Sharing

Article Rating (2 Votes)

★★★★★

You've Already Voted.

Bookmark Article (CTRL-D)

Attachments

There are no attachments for this article.

Related Articles

Database Guidelines
Viewed 448 times since Wed, Mar 30, 2016

Hardware - Software Performance Guidelines
Viewed 317 times since Wed, Mar 30, 2016

Hosting Guidelines
Viewed 249 times since Wed, Mar 30, 2016

Neosoft Secure Coding Practices
Viewed 908 times since Wed, Mar 30, 2016

Security Guidelines
Viewed 620 times since Wed, Mar 30, 2016