THE TECH ACADEMY

COMMON MISTAKES TO AVOID WITH HTML

1. Failing to implement the correct DOCTYPE.

Without the correct document type definition, browsers will have to guess what language your page is written in and will most likely fail to properly apply the instructions contained within the document. The end result may not be exactly what was intended.

2. Forgetting to encode the document with reference to the correct character set.

The Character set meta information tells the browser what language and rules to use when displaying the information.

3. Attempting to employ deprecated or invalid syntax.

The industry is an ever-evolving ecosystem, and many languages like HTML and CSS are not immune to technological improvements and changes. As the industry standards are adjusted, oftentimes their syntax and features adjust as well. For example the Bold tag, <b>, a valid inline styling tag from the past, has since been superseded by the newer style tag.

4. Publishing markup with syntax mistakes.

Though it may seem a minor and easily overlooked mistake, a browser that is interpreting the instructions may not easily discern the intended purpose. Often times, it will only take one errant tag being left open or an improperly applied quotation mark to through the entire meaning of the code out of context, producing some very anomalous results.

5. Parking elements in incorrect locations.

The browser is very specific. It looks for a specific syntax and formatting and will parse syntax which it cannot understand or which doesn't conform to the rules with peculiar results. A <title> tag placed outside of the <head> tag will simply be omitted - or worse yet, it may get mistakenly printed into the body text. Improperly placing body content inside of a <script> tag will also not promote the desired outcome.

6. Improperly applying or failing to close a block-level tag.

These kinds of mistakes will oftentimes produce blank spaces, strange anomalous artifacts, or formatting errors as the browser attempts to apply the markup and CSS instructions.

7. Convoluting or overruling content styling.

Novice developers will oftentimes liberally apply inline styling syntax, but the most common way to manage styling is by controlling the styles with a separate CSS file. Using inline styling may be acceptable to a very small project, but as the project grows in complexity the inline styles may become convoluted or overruled by future CSS styling. Suffice to say that using inline styling is bad practice, and the earlier a novice developer can begin to learn to manage styles with CSS the better off they will be.

8. Using incorrect file or address paths.

When the browser comes across code that has instructions to retrieve a file on the server or redirect to a web address where the path is invalid, the results will be less than professional.

9. Arbitrary replacing or using outdated practices.

This one is more an offense that the senior developers will commit. In the past, it was customary to divide or compartmentalize content into manageable blocks by liberally applying the tag. With the introduction to new standards in HTML this type of practice has been greatly discouraged. HTML5 introduced a suite of newer and better suited structural tags that will not only reproduce the same effect - these newer tags will aptly define the blocks of code they contain. HTML5 has introduced a complement of wrapper tags: <sections>, <article>, <aside> <media>, <canvas>, <nav>, and <footer> to name a few.

10. Not applying enough testing.

Developers sometimes fail to fully check their design across a suite of viewports and resolutions; as a result their code may not appear as intended to the viewers in their audience. The industry standards and the devices the audience may be employing to access content are ever-changing; a developer should remain vigilant by appropriately and thoroughly testing their content.

11. Improperly employing special characters.

The ampersand is just one of these special characters that aspiring developers fail to implement properly - and side-effects may vary. To adeptly prevent these kinds of mistakes, these special characters need to be converted into their HTML entity equivalents. IE: The ‘&’ character should be represented in HTML with ‘&amp’;

12. Failing to encode IMG tags with an alternative text description.

To comply with the standards in HTML5, every image inserted into an HTML5 document must have an ALT attribute. Though the ALT attribute must be present, its content may still be left empty if it is not necessary to present it.

Many of these common mistakes can be avoided by thoroughly checking your code and passing it through W3C's Markup Validation Service or other online resources.

END OF ARTICLE