

# WYSIWYG CONSIDERED HARMFUL\*

## You Don't Always Get WYS (What You See)

John J. Cupak\*\*

Lecturer, Department of Mathematics & Computer Sciences, Rivier University

Today, most people who want to create a website for themselves typically have not studied web development and are unaware of the correct way to write HTML, CSS, or JavaScript. Instead, they use a "What You See Is What You Get" (WYSIWYG — "wizzy-wig") website editor which allows them to create beautiful web pages from templates, stock images, and content blocks. Having used application programs such as Microsoft® Word to write documents, they make the intuitive leap to creating websites with a WYSIWYG website editor.

I consider these WYSIWYG website editors harmful for many reasons. Unlike a Word document which is typically written and printed once on a standard-sized paper, a website is dynamic. It is viewed on different browsers, screen sizes, and hardware platforms ranging from a full-sized desktop to a small wrist watch. Furthermore, websites live for decades and require frequent updates to its content and style.

WYSIWYG website editors fail to separate the content and the style, which is an important design principle. Styles are embedded in HTML tags everywhere, making it difficult to maintain. WYSIWYG website editors which generate websites do not always use the latest semantic HTML styles, instead they create a plethora of `divs` and `spans` with embedded class, id, and style attributes. With all the bloat generated by the WYSIWYG website editors, the resultant web pages are slow to load and often hit a capability ceiling for pages with complex content.

To illustrate the difference between creating a web page using a text editor as compared to using a WYSIWYG website editor and its associated problems, I will present a brief comparison of the two methodologies and their results.

The Rivier's CS597 Multimedia and Web Development course that covered the three components of good web pages: the content written in HTML (Hypertext Markup Language), the design written in CSS (Cascading Style Sheets), and the behavior written in JavaScript. Each of these three components reside in separate files, due to the difference in language syntax. However, both the style files and behavior code files are dependent upon the content markup.

The web page content HTML syntax has evolved to include semantic<sup>1</sup> tags; for example, rather than writing `<div class="navigation">`, HTML5 now includes the `<nav>` tag which clearly identifies its purpose without requiring an additional attribute. Web pages now are clearly written with `<header>`, `<nav>`, `<main>`, `<section>`, `<article>`, `<aside>`, and `<footer>` semantic tags.

In addition, where earlier web pages included repeated, embedded styles such as `<font color="#FFCC99" face="Arial, Helvetica, sans-serif" size="1">` to denote links to another page, all styles are removed and placed in a separate site style sheet which describes the style for all links: a `{font: 12pt/1.5 Arial, Helvetica, sans-serif; color: "#FFCC99";}`.

---

\* The title is similar to Edsger Dijkstra's letter "Go To Statement Considered Harmful", published in the *Communications of the ACM* (CACM) in March 1968.

<sup>1</sup> Semantic HTML or semantic markup tag is HTML that introduces meaning to the web page rather than just presentation.

Changing any of the styles need only be done once in the common style sheet instead of embedding the style everywhere it appears in the HTML content, simplifying the website style and maintenance.

The behavior of web pages in response to user interaction, such as highlighting links or button rollovers, are now provided in CSS styles rather than requiring embedded or separate JavaScript code — further simplifying the understandability and maintenance of web site pages.

All of the web pages — the content HTML, style CSS files, and JavaScript — may be written with a simple text editor. Students used Notepad++ (<https://notepad-plus-plus.org>) on classroom or personal PCs, or Atom (<https://atom.io>) on Apple computers to write their HTML, CSS, and JavaScript files. What they wrote was directly reflected on their web pages.

For example, the Rivier's CS597 Multimedia and Web Development final exam (summer 2018) required students to create a single web page on the Indus Valley Civilization. Students were provided a requirements document which described the colors, dimensions, contents, and behavior of the web page. In addition, students were provided the header graphic, a map, an MP3 audio file, and a file of the text to include in the web page.

When completed, the final exam HTML file was approximately 210 lines long (including blank lines for readability), with a simple structure of a head, body, header, a main section containing an article with two internal sections and an aside with two internal sections, and a footer. Each article, section, and aside contained an id attribute, as shown in the Figure 1 code snippet (*right*).

The head contained links to external style sheets as well as a small section of JavaScript to show and hide text containing a more detailed description of the subject. Navigation to other web pages in the website were designed using the semantic HTML `<nav></nav>` tags with an internal unnumbered list of links to other pages, as shown in the Figure 2 HTML snippet (*below*).

```

<!DOCTYPE HTML>
<html>
  <head lang="en-US">
  <body id="home">
    <header>
      <div id="hgroup">
        <nav>
      </header>
    <main>
      <article id="indus_valley_civilization">
        <section id="map">
        <section id="about-india">
      </article>
      <aside id="links">
        <section id="references">
        <hr />
        <section id="external_links">
      </aside>
    </main>
    <footer>
  </body>
</html>

```

Figure 1. HTML Structure

```

<nav>
  <ul>
    <li><a href="index.html"    class="home"    >Home    </a></li>
    <li><a href="arts.html"     class="arts"     >Arts     </a></li>
    <li><a href="religion.html"  class="religion">Religion</a></li>
    <li><a href="animals.html"   class="animals" >Animals </a></li>
    <li><a href="food.html"      class="food"     >Food     </a></li>
  </ul>
</nav>

```

Figure 2. HTML Semantic navigation tags

To illustrate the result of using a WYSIWYG website editor, the final exam requirements for a web page were created using the WIX WYSIWYG website editor (<https://www.wix.com>). In contrast to the clear content of the final exam HTML and CSS files, the WIX HTML file alone was 11,021 lines long and 1,325,436 bytes in size. The HTML tags included an over-abundance of attributes, including embedded styles. For example, the navigation section of the WIX website file with only the first link the contents of Figure 3 (*below*).

```

▼<div data-mesh-id="comp-kjqe39d8inlineContent-gridContainer" data-testid="mesh-
container-content">
  ▼<wix-dropdown-menu id="comp-kjqe1xgs" class="_3k4Gu _3xxDQ hidden-during-
prewarmup" dir="ltr" data-stretch-buttons-to-menu-width="true" data-same-width-
buttons="false" data-num-items="5" data-menuborder-y="0" data-menubtn-border="0"
data-ribbon-els="0" data-label-pad="0" data-ribbon-extra="0" data-dropalign=
"center" style="visibility: inherit; overflow-x: visible;" data-dropmode=
"dropDown">
  ▼<nav class="YwHr4" id="comp-kjqe1xgsnavContainer" aria-label="Site">
  ▼<ul class="_22KPg" id="comp-kjqe1xgsitemsContainer" style="text-align:
center" data-marginallchildren="true">
  ▼<li id="comp-kjqe1xgs0" data-direction="ltr" data-listposition="left"
data-data-id="bmi23nr" data-state="menu selected link" data-index="0"
class="_3ZPg7 _3VLfo" data-original-gap-between-text-and-btn="10" aria-
hidden="false" style="width: 141px; height: 28px; position: relative; box-
sizing: border-box; overflow: visible; visibility: inherit;">
  ▶<a data-testid="linkElement" href="._/" target="_self" class="_3NKrl"
aria-haspopup="false">...</a>
  </li>

```

Figure 3. WIX Navigation

With the WIX WYSIWYG website editor, it was not possible to edit the styles of individual components to provide finer tuning of their positions or styles. Each heading and text paragraph must be placed individually on the page. External margins or internal paddings usually specified in CSS styles to align components were unavailable in the editor.

The biggest issue with WYSIWYG website editors, however, is that they embed site tracking code. With privacy a hot topic these days, the websites generated by such editors are able to track where users

have been, what they are looking for, how long they are on the page, and even where they go after they leave the page.

In conclusion, if you want a website that is easy to create, understand, and maintain, stay away from WYSIWYG website editors. They generate bloated and incomprehensible code that is impossible to manage.

## References

Atom. *A hackable text editor for the 21st Century*. Retrieved December 27, 2020, from <https://atom.io>

Garber, Jason. (2009, June 5). *WYSI-dangerous: Why WYSIWYG editors are bad for your website*. [Blog post].

Retrieved December 27, 2020, from <http://redcloth.org/articles/wysi-dangerous-why-wysiwyg-editors-are-bad-for-your-website>

Wix. *Create a Website You're Proud Of*. Retrieved December 27, 2020, from <https://www.wix.com>

Notepad++. *What is Notepad++*. Retrieved December 27, 2020, from <https://notepad-plus-plus.org>

---

\* **PROF. JOHN J. CUPAK** taught Computer Science courses at Rivier University from spring 2014 to summer 2018. He is a member of the Institute of Electrical and Electronic Engineers (IEEE) and the Association for Computing Machinery (ACM), having served as president for the ACM chapter in Rome, NY. He has a B.A. in Biology with a mathematics minor from the Utica College of Syracuse University, and a M.S. in Computer Science from the State University of New York at Albany (SUNYA) that he earned in 1975. Prof. Cupak worked as a software engineer for over 40 years, and has brought his experience in computer-system design, development, documentation, and testing to his courses at local colleges and universities. An avid educator, he researched and implemented the pioneering “Flipped Classroom” methodology in the Department of Mathematics and Computer Science at Rivier University. He notes that “Teaching is a learning experience - for the student and the teacher as well.” Since retiring in 2018, John and his wife have relocated to Florida.