The Unordinary Story of HTML5

In this lesson, let's unveil the unordinary story of HTML 5!

WE'LL COVER THE FOLLOWING

HTML : A brief history

HTML is an acronym for ***Hypertext Markup Language***. Although it had started its life as a document description language for sharing research papers at **CERN** (European Organization for Nuclear Research), it became the **basic technology of rendering web pages**.

Since its emergence in **1980**, it went through many modifications, while it reached the **HTML 4.0** state at the end of **1997**, when it was published as an ***official W3C (World Wide Web Consortium) Recommendation***.

From HTML to XHTML

Even **W3C** did not believe that HTML would survive the next century; so instead of moving forward with HTML, they stopped working on it.

W3C bet on HTML's renovated successor, ***XHTML (Extensible Hypertext Markup Language)*** that was to be built on XML. The main reason behind this decision was to enforce stricter syntax rules than required by HTML. For example, the following markup that makes the second word boldfaced is totally valid in HTML:

```
<p>Hello <b>HTML!</p></b>
```

Browsers accept it and display exactly what the creator of this markup intended to show, although its perfunctory syntax is not valid. The right and valid version was where the closing `</p>` and `</b>` tags would follow the proper nesting (notice the placement of the `p` and `b` tags in the below snippet as opposed to the one above.)

```
<p>Hello <b>HTML!</b></p>
```

This second style follows the strict XML syntax rules, required by XHTML.

Just think about the advantages an XML-compliant markup such as XHTML could result:

- XML can be easily parsed and processed without dealing with the quirks of HTML (such as the nesting issue in the "Hello HTML!" markup),
- easy integration with XML tools,
- portability to different devices including mobile, and so on.

At the beginning, XHTML seemed to be the right direction, because more and more professional developers and web designers accepted it as a standard. Of course, browsers also parsed it, and displayed XHTML pages as demanded by the standard.

However, browsers did more; they did not enforce the strict rules, and instead of spitting out error messages, they simply rendered the web page *auto correcting the improper markup*, such as exchanging the `</b>` and `</p>` tags in the second sample above.

This kind of deliberate behavior *torpedoed* the XHTML standard, because browsers processed old-fashioned HTML pages called XHTML. W3C thought that tightening up the strict rules and requiring browsers to reject invalid pages in XHTML 2 would solve the issues with standardization.

Well, they were wrong. Not only the too small, actually logical, changes in the standard made it fail, but the extremely slow cadence of getting forward turned developers away from it. After five years, in **2004**, **XHTML 2** seemed to reach a **dead-end**.

The emergence of HTML5

In the very same year, a group of enthusiastic people at **Mozilla Foundation** and **Opera Software**, the creators of Firefox and Opera browsers, tried to lobby for transforming XHTML into a standard that provides more development-oriented features.

They intended to position the markup that was originally created for describing documents into a great tool for developing web applications, but they failed the W3C voting for this change.

With **Apple**, they established the **Web Hypertext Application Technology Working Group (WHATWG)** *within* W3C. The most important goal of this group was to extend the existing HTML standard with developers in mind, while still keeping backwards compatibility.

First, they worked on add-in specifications like **Web Applications 1.0 and Web Forms 2.0**.

In **2007**, W3C indirectly admitted its failure when it disbanded the group

working on XHTML2, and they commenced formalizing a new standard, **HTML5**, based on the work of **WHATWG**.

HTML Forever?

While today **W3C** is responsible for the **official HTML5**, WHATWG have not stopped envisioning new HTML features. They imagine HTML as a *living standard* (visit http://developers.whatwg.org for details), and the markup itself as a *living language*.

According to this vision, HTML pages won't have a version number, and a previously created HTML page will never become obsolete. As time goes on and technology evolves, HTML will introduce new elements and support new features. It's up to browser vendors and developers to choose whether they will use a certain new feature or not.

Although this "living thing" seems weird, just imagine how developers would respect a moving target as new features are added to the language. It's not that different from what we have today with the hodgepodge of HTML features supported by different browsers.

So, will it work? Time will tell. :)