JOHNS HOPKINS
UNIVERSITY

Zanvyl Krieger School of Arts and Sciences
Advanced Academic Programs

410.712.81
Advanced Practical Computer Concepts For Bioinformatics

Topic: HTML5 introduction

Instructor: Joshua Orvis

What is HTML?

HyperText Markup Language is the standard by which most web pages are constructed. Originally designed by Tim Berners-Lee as a way to formally describe and share academic documents electronically.

The premise was relatively simple – use nestable *tags* or *elements* to describe certain sections of a document and provide a syntax to allow specification of *attributes* of those elements. For example:

```
<p>
    You are a student in course number <span class='course_id'>410.712.81</span>
</p>
```

The above example has a short paragraph tag (<p>) with a single sentence. That sentence contains a course number wrapped in a . That span has a class attribute with a value of 'course_id'.

Larger pages are, essentially, expansions of this with more elements and attributes employed to better 'mark up' a document. At least that was the theory.

HTML versions

The first versioned release of the HTML specification was 1.1 in 1992. Engineers designing the specification recognized that users wanted to do more than just formally describe their documents using tags, they wanted to use these tags and attributes to display data to their colleagues. Thus, the specification underwent several feature additions and variations of the years until HTML 4.01 was officially recommended in 1999.

For the last decade this new version was how nearly all websites were built were built.

The versions themselves are proposed and documented through the organizing body called the World Wide Web Consortium (W3C).

New versions are developed through a long, arduous, committee-led process that generally yields a well-designed and architected specification.

The release of version 4.01 served developers well for many years, but ultimately needed to be replaced.

What's wrong with HTML 4?

There were a few basic problems with the core design of HTML 4 that needed to be corrected. First, elements had been added over the years to aid in presentation, such a defining fonts, visual lines, etc. These didn't really provide that great of presentation capability, and polluted the purpose of formalized markup, which was to describe a document's contents. The content and presentation needed to be separate.

The second problem was that the elements which did exist weren't adequate for describing modern web documents. Common sections such as navigation panels, sections, footers, etc. were found on most pages but nothing about the language encouraged developers to encode them the same way.

Not going to be limited to the existing element set, developers instead reused simple, generic elements (such as) and added class or id attributes to them all. This resulted in (ugly) code like:

```
<div id='page_header'>
  <div id='section'>
    <div class='story'></div>
  </div>
</div>
```

Known as 'div-itis', pages became overloaded with massively nested divs of different classes. More descriptive elements were needed.

HTML5

Still relatively new, HTML5 aims to correct many of the shortcomings of HTML4.

- Presentational elements have been removed, forcing developers to separate style and content using CSS

- New elements to better describe content, such as 'article', 'footer', 'nav', 'section', etc.

- A new 'canvas' element to provide native vector drawing capabilities.

- Built-in audio/video support

- Custom data attributes and client-side databases

- Improved form controls

- Web Sockets for persistent server connections

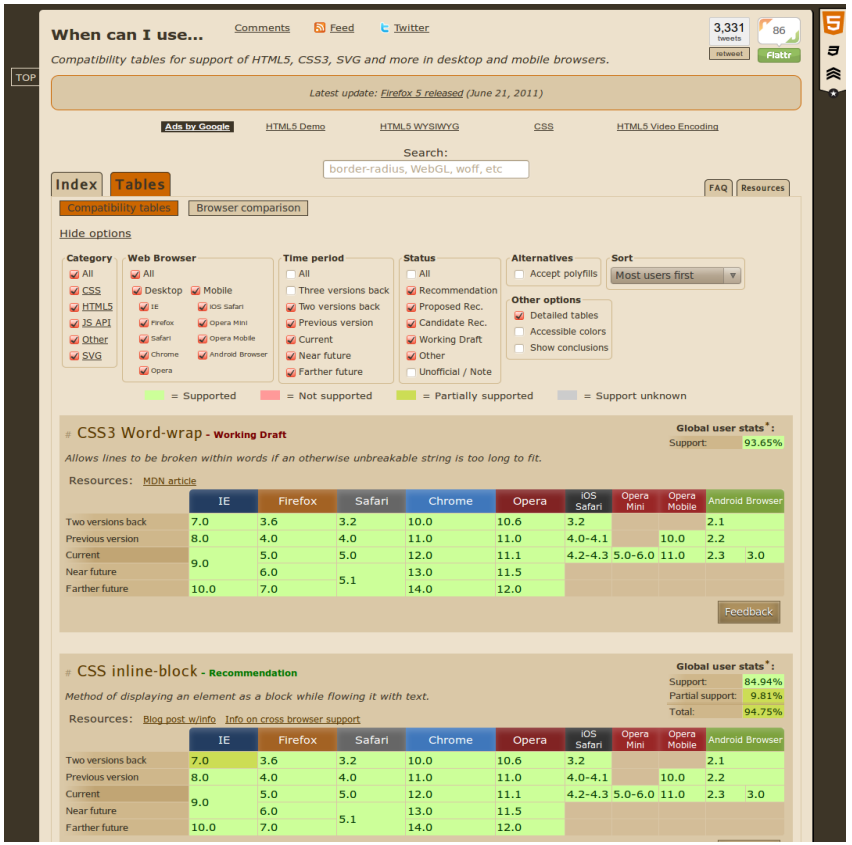
Cascading Style Sheets (CSS)

CSS is the de facto way to apply style rules to your HTML pages. Using this we can transform the layout of content on a page, add background images, control fonts, etc.

By separating the presentation layer from the styling layer, a developer can redesign an entire site simply by writing another style sheet rather than changing the content pages.

Browser compatibility

Any time a new technology comes around developers must keep track of browser support before they rely too heavily on it. There's no use developing with the latest and greatest technology if few of your users can view it yet.



HTML5 support has finally passed the critical point where you can use it with less worry about compatibility issues. This is both because of newer browser versions that support HTML5 and because of fall-back libraries that provide backward compatibility layers for those that don't.

A good site for checking compatibility for new features in HTML, CSS and more is:

http://caniuse.com

Getting started with HTML5

You can open an editor and start typing your code by hand, of course, but a great solution until you're an expert is to use an existing template. One of my favorites is called HTML5 Boilerplate. It's free, and in a single download provides:

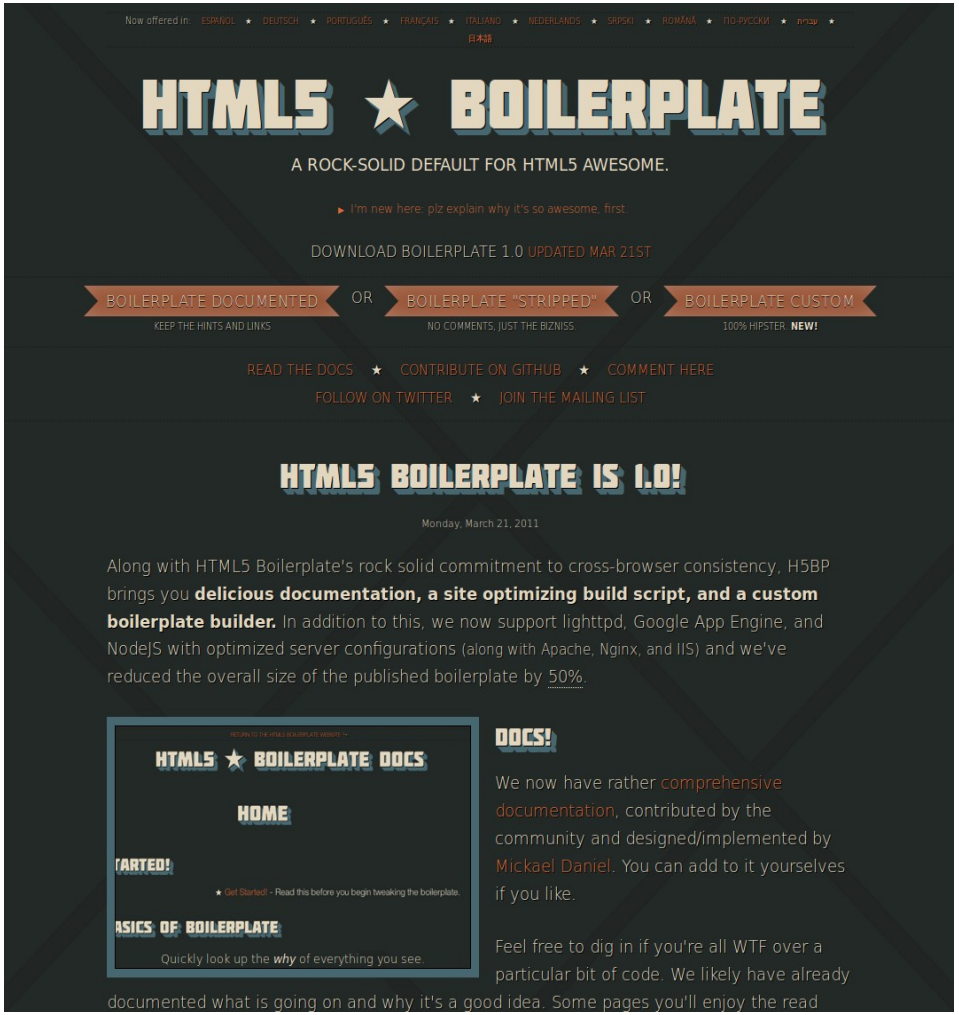
html5boilerplate.com

- An HTML5 page to add your content to that's cross-browser compatible (even IE6)

- Graceful degradation for browsers not supported.

- CSS3 framework ready

- JQuery Javascript framework ready



We will use this in our in-class project.

HTML5 in this class

We could spend an entire semester on the HTML5 / CSS3 / Javascript stack. For this class I want to introduce you to each of these and give you a feel of their roles in bioinformatics application development without getting into them too deeply.

To that end, we'll begin with the HTML5 Boilerplate and I'll introduce the specific elements as we progress through the next sections. You'll learn how we add new content, how we make that content more presentable with CSS, and how we add functional interaction in the browser with Javascript.

Further reading

Dive Into HTML5 – Mark Pilgrim

This great little tutorial is a popular resource for getting started with HTML5.