

HTML+ CSS PRINCIPLES

Getting started with web
design the right way

HTML : a brief history

1

1960s : ARPANET is developed... It is the first “packet-switching” network using TCP/IP protocol and is a precursor to World Wide Web.

<http://en.wikipedia.org/wiki/ARPANET>

HTML : a brief history

2

HTML language is invented in
early 1980s

<http://www.w3.org/History.html>

HTML : a brief history

3

In the 1980s desktop computing also became a much more common tool for businesses and consumers.

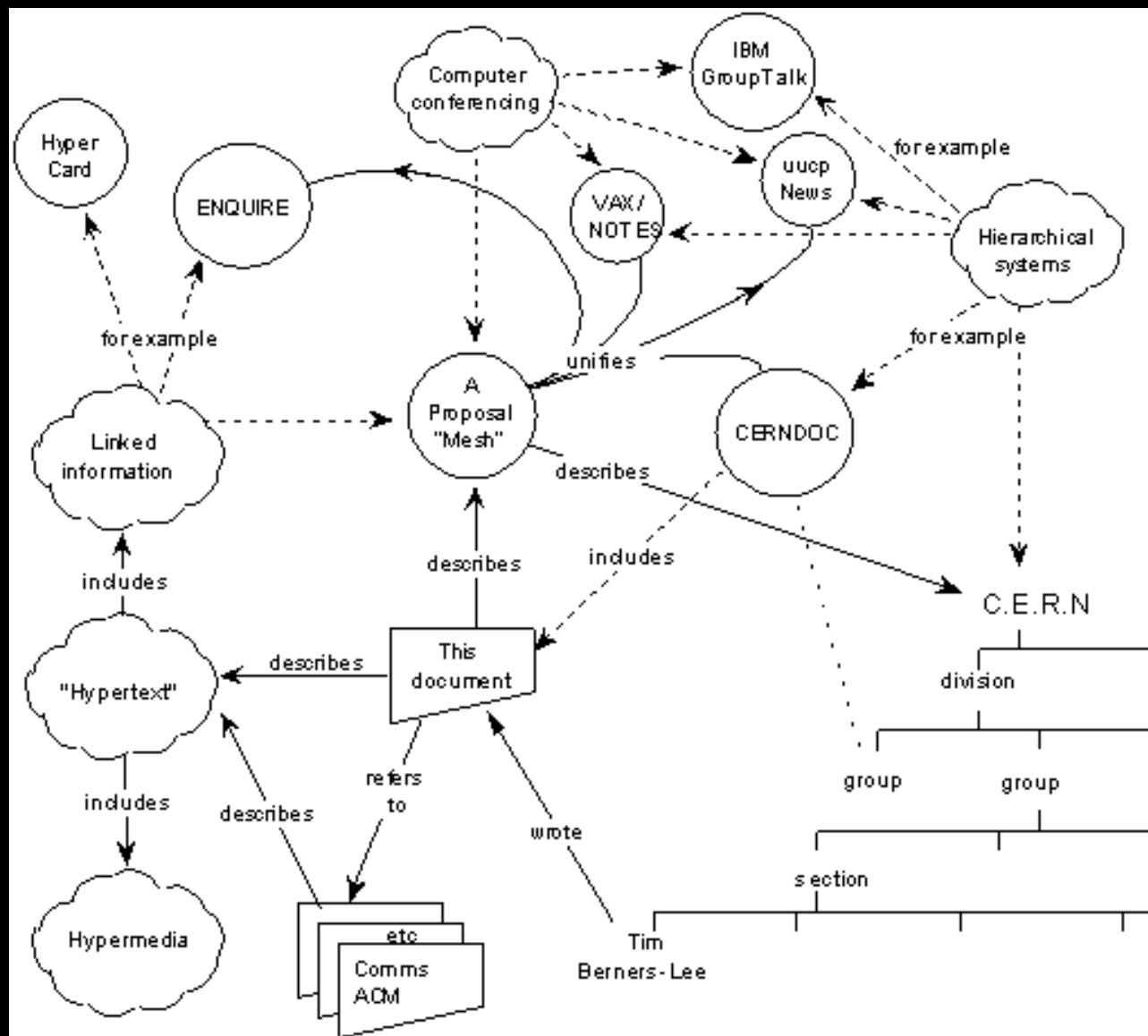
HTML : a brief history

4

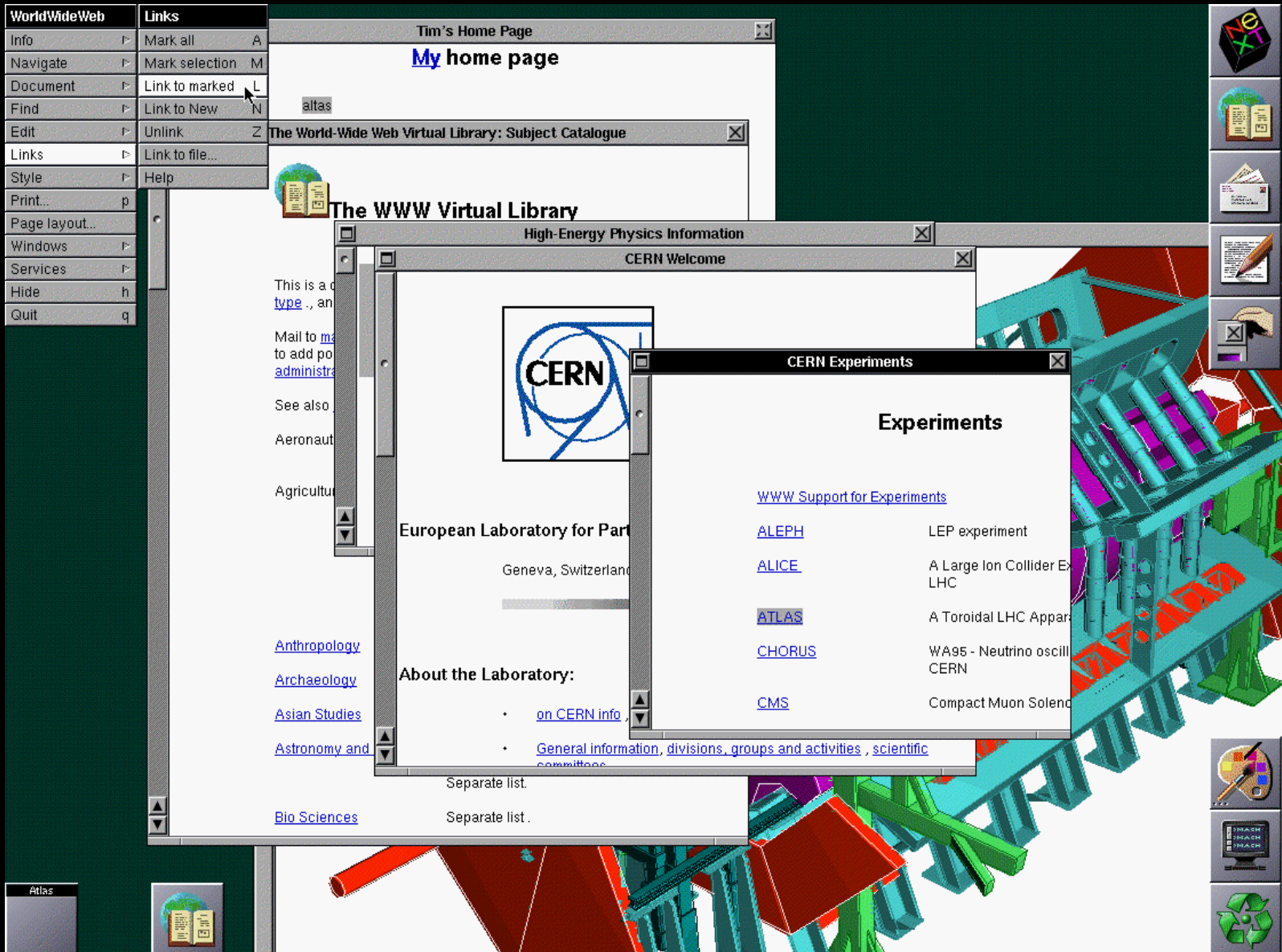
1989-90 the “World Wide Web” is developed by Tim Berners-Lee along with the first visual web browser...

<http://www.w3.org/History/1989/proposal.html>

The idea:



The browser:



HTML : a brief history

5

More users + new addition of commercial uses created more complexities and greater need for standards.

HTML : a brief history

⑥

2002: XHTML was developed for use with stricter standards and a more flexible framework to incorporate XML data files.

HTML : a brief history

7

Then this started happening:



HTML : a brief history

⑧

2008: HTML5 is introduced. It moves away from xml but offers database integration, semantic markup, javascript integration, and and better video capabilities.

HTML

Hypertext Markup Language

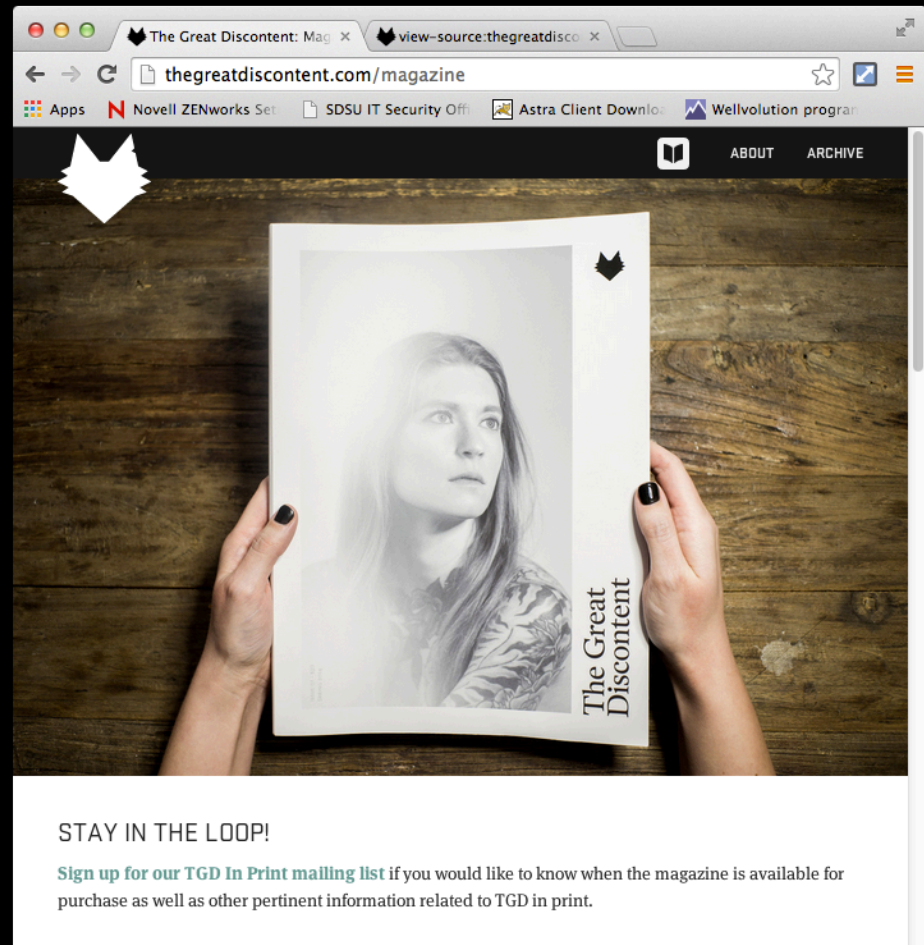
Current version of HTML is HTML5. Prior to HTML5 the standard was XHTML, which was the first to offer strict semantic markup and a focus on separating “*presentation from structure.*”

WHAT HTML LOOKS LIKE

display view vs. code view

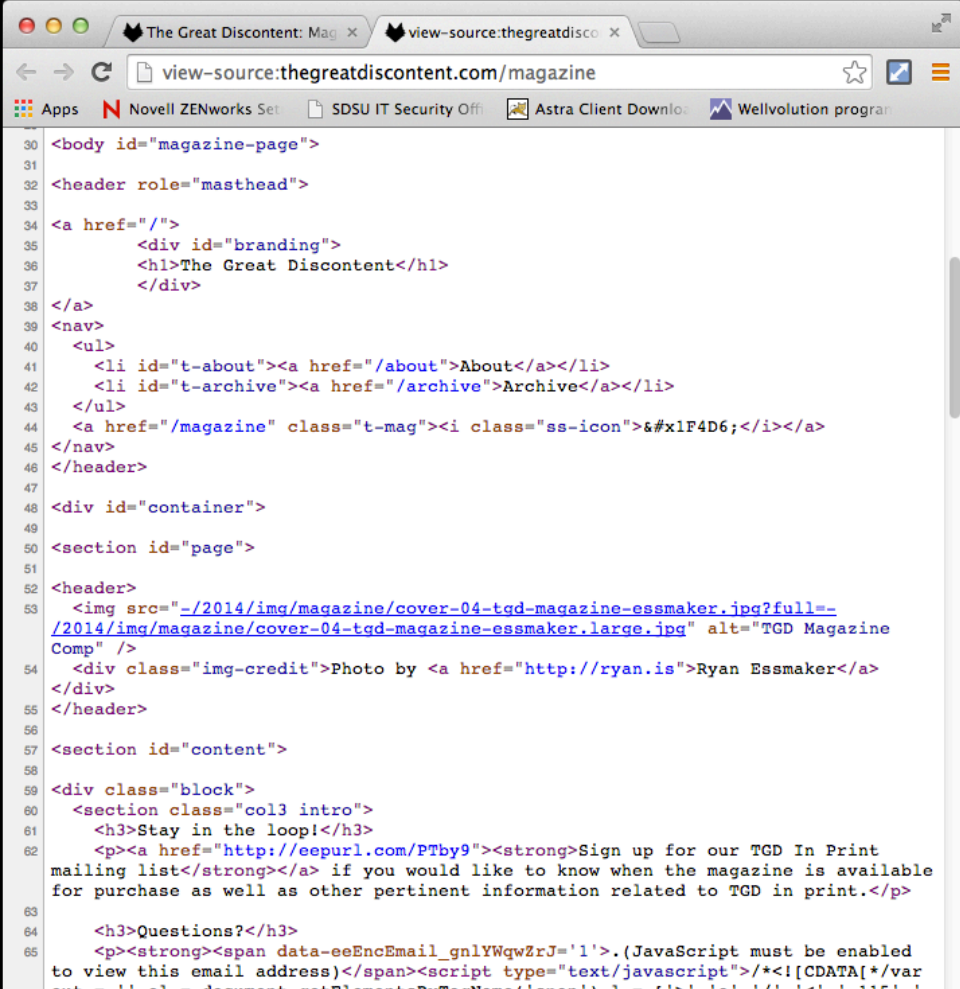
HTML > display

Here's what we see as a viewing audience:



HTML > code

Here's what we do as web designers and developers:



```
30 <body id="magazine-page">
31
32 <header role="masthead">
33
34 <a href="/">
35     <div id="branding">
36         <h1>The Great Discontent</h1>
37     </div>
38 </a>
39 <nav>
40     <ul>
41         <li id="t-about"><a href="/about">About</a></li>
42         <li id="t-archive"><a href="/archive">Archive</a></li>
43     </ul>
44     <a href="/magazine" class="t-mag"><i class="ss-icon">&#x1F4D6;</i></a>
45 </nav>
46 </header>
47
48 <div id="container">
49
50 <section id="page">
51
52 <header>
53     
54     <div class="img-credit">Photo by <a href="http://ryan.is">Ryan Essmaker</a>
55 </div>
56 </header>
57
58 <section id="content">
59
60 <div class="block">
61     <section class="col3 intro">
62         <h3>Stay in the loop!</h3>
63         <p><a href="http://eepurl.com/PTby9"><strong>Sign up for our TGD In Print mailing list</strong></a> if you would like to know when the magazine is available for purchase as well as other pertinent information related to TGD in print.</p>
64
65         <h3>Questions?</h3>
66         <p><strong><span data-eeEncEmail_gnlyWqwZrJ='1'>.(JavaScript must be enabled to view this email address)</span></strong><script type="text/javascript">/*!CDATA[+/var out = 'mailto:' + document.getElementsByTagName('span')[0].getAttribute('data-eeEncEmail_gnlyWqwZrJ'); document.write(out);</script></p>
```

HTML > where to start

HTML is based on the use of TAGs.

TAGs are the beginnings and endings of element declarations. They tell the browser how to process the given data.

HTML > how to start

The basic anatomy of a TAG:

```
<tagname>Blah blah blah</tagname>
```

(BTW, this is a dummy, not-real example)

HTML > how to start

It could also look like this:

```
<tagname>
```

```
    Blah blah blah
```

```
</tagname>
```

(Still not-real)

HTML > how to start

It could also look like this:

```
<tagname>  
    Blah blah blah  
</tagname>
```

Almost all tags have an opening AND closing tag. The only difference is that the closing tag starts with a “/” forward-slash.

HTML > tag examples

```
<p>This is where you type the paragraph  
text you want to show up on the  
screen.</p>
```

(This is a real example of a “paragraph” tag)

HTML > tag examples

```
<h1>I'm the most important heading</h1>
```

```
<h2>I'm important heading level #2</h2>
```

...

```
<h6>Headings go down to a level #6</h6>
```

(These are real examples of “heading level” tags.)

HTML > tag examples

```
<ul>
```

```
  <li>List item one goes here</li>
```

```
  <li>List item two goes here</li>
```

```
</ul>
```

(These are real examples of “unorded list” and “list item” tags.)

HTML > tag examples

...and the **TAG** list goes on:

<html>
<head>
<title>
<base>
<link>
<meta>
<style>
<body>
<section>
<nav>
<article>

<aside>
<header>
<footer>
<address>
<p>
<hr>
<pre>
<blockquote>

<dl>
<dt>
<dd>
<figure>
<figcaption>
<div>
<a>

<small>

<cite>
<q>
<abbr>
<data>
<code>

...AND MANY
MANY MORE.

HTML > tag examples

A **TAG** can also be referred to as an **ELEMENT**. Here is a GREAT resource list of HTML5 elements with explanations of each one is used:

[https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5/HTML5 element list](https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5/HTML5_element_list)

HTML > attributes

A **TAG** can also have

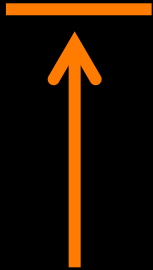
ATTRIBUTES:

```
<a href="http://gmail.com">Go to GMAIL</a>
```

HTML > attributes

ATTRIBUTES are information within the tag, defining the tag. Different tags have different attribute options.

```
<a href="http://gmail.com">Go to GMAIL</a>
```



HTML > values

A **VALUE** is what pinpoints the **ATTRIBUTE** to more specific information.

`Go to GMAIL`



HTML > tag, attribute, value

In this example below, we have an `<a>` tag, which is an “*anchor*” link. But we need to know where the link will go, so we give it an attribute, “`href`”, which stands for *hyper-reference*. And that `href` will take us to the value of “`http://gmail.com`” when clicked. What the viewer sees as a clickable link is inside the `<a>` tags, “Go to GMAIL”.

```
<a href="http://gmail.com">Go to GMAIL</a>
```

HTML > nesting tags

To make pages, we have to “nest” TAGs:

```
<article>
  <header>
    <h1>How Cells Multiply</h1>
  </header>
  <p>Our cells are amazing. They really can
    multiply...</p>
</article>
```

HTML > nesting tags

...and nest them properly:

```
<p>  
  <a href="contact.html">Contact Us</a>  
</p>
```



```
<p>  
  <a href="contact.html">Contact Us</p>  
</a>
```



SPEAKING OF WHICH...

Let's look at the basic, required parts of any HTML page....

Minimum HTML5 requirements

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Title goes here</title>
```

```
  </head>
```

```
  <body>
```

```
  </body>
```

```
</html>
```


What is <!DOCTYPE> ???

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Title goes here</title>
```

```
  </head>
```

```
  <body>
```

```
  </body>
```

```
</html>
```

What is <!DOCTYPE> ???

```
<!DOCTYPE html>
```

The “doctype” announces to a browser which version of HTML it is using. This helps the browser know what to expect and how to behave.

Without a “doctype”, browsers go into “quirks mode” and can interpret code unpredictably.

The <html> tag

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Title goes here</title>
```

```
  </head>
```

```
  <body>
```

```
  </body>
```

```
</html>
```

The <html> tag

```
<html>...</html>
```

The “html” tag represents the root of an HTML or XHTML document. All other elements (except “doctype”) must be descendants of this element.

The <head> tag

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Title goes here</title>
```

```
  </head>
```

```
  <body>
```

```
  </body>
```

```
</html>
```

The <head> tag

```
<head><title>...</title></head>
```

The “head” tag contains a collection of metadata about the document, including links to, or definitions of, scripts and style sheets. At a minimum, it **MUST** contain a “title” tag.

The <title> tag

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Title goes here</title>
```

```
  </head>
```

```
  <body>
```

```
  </body>
```

```
</html>
```

The <title> tag

```
<title>Title goes here</title>
```

Defines the title of the document, shown in a browser's title bar or on the page's tab.

It can only contain text. Nested HTML tags will not be interpreted.

Finally, the <body> tag

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Title goes here</title>  
  </head>  
  
  <body>  
  </body>  
</html>
```

Finally, the <body> tag

```
<body>...</body>
```

Contains the content of an HTML document that is visually displayed in the browser's viewport/window.

There is only one <body> element in a document.

A technical recap

```
{
  <!DOCTYPE html>
  <html>
    {
      <head>
        { <title>Title goes here</title>
        }
      </head>
    }
    {
      <body>
      </body>
    }
  </html>
}
```

A metaphorical recap

HTML tags are fairly intuitive. An easy way to remember the basic parts is to humanize a page:

Doctype = laws governing behavior

<html> = the whole person

<head> = the brain that gathers information

<title> = name / identity a person calls one's self

<body> = the physical representation and
behaviors of the person

SPEAKING OF INTUITIVE...

Let's talk about "SEMANTIC"
html markup.

SEMANTIC

adjective \si-^lman-tik\
/

: of or relating to the meanings of words and phrases

Semantic tags in HTML

- HTML tags are imbued with meaning.
- They should not be arbitrarily assigned.
- They have meaning outside of visual representation.
- Visually impaired people can still understand the meaning of sections in a page with semantic markup.
- Search engines can too.

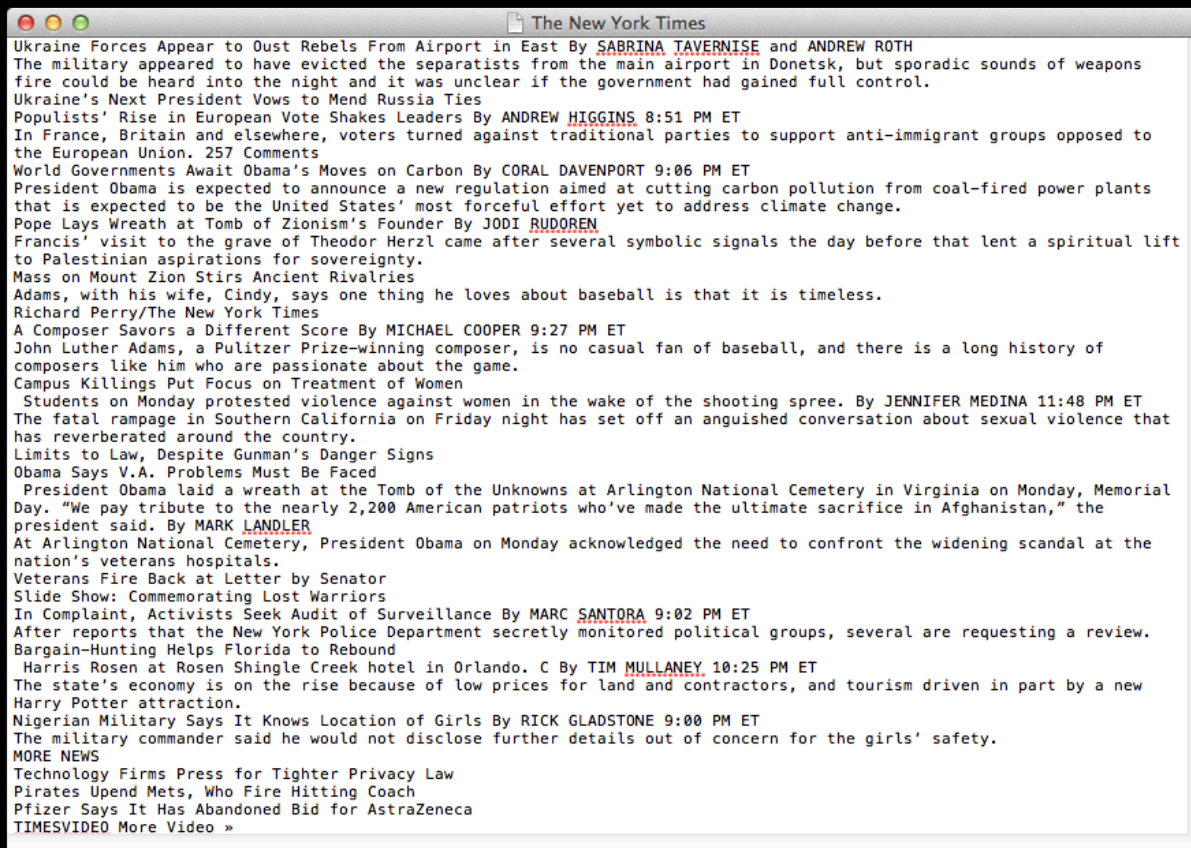
A “semantic” case study

Imagine that you are reading a newspaper but there are no headings, no margins, no indents, no bold, no italics, no paragraph separations, etc.

IT'S JUST TEXT...

A “semantic” case study

LIKE THIS:



A “semantic” case study

Without meaningful hierarchy, no one would want to read through that page because it would be a laborious experience.

That is how search engines and blind people experience pages with poor semantic markup.

A “semantic” case study

SAME SITE, BUT BETTER EXPERIENCE:

The screenshot shows the New York Times website interface. The browser address bar displays 'www.nytimes.com'. The navigation menu includes categories like WORLD, U.S., NEW YORK, OPINION, BUSINESS, TECHNOLOGY, SCIENCE, HEALTH, SPORTS, ARTS, FASHION & STYLE, and VIDEO. The main content area features several articles:

- Ukraine Forces Appear to Oust Rebels From Airport in East** by Sabrina Tavernise and Andrew Roth. The text describes military actions in Donetsk.
- A Composer Savors a Different Score** by Michael Cooper. The article mentions John Luther Adams, a Pulitzer Prize-winning composer.
- Campus Killings Put Focus on Treatment of Women** by Jennifer Medina. The article discusses a fatal rampage in Southern California.
- Populists' Rise in European Vote Shakes Leaders** by Andrew Higgins.

On the right side, there are sections for 'The Opinion Pages' (including 'A Cable Merger Too Far'), 'Today's Times Insider', and 'MARKETS' with a table of stock indices:

MARKETS »			At 1:17 AM ET	
JAPAN		CHINA		
Nikkei	HangSeng	Shanghai		
14,703.49	22,907.87	2,036.97		
+100.97	-55.31	-4.51		
+0.69%	-0.24%	-0.22%		

Below the market data is a 'SCIENCE TIMES' section with the headline 'Craft Beer, at the Genetic Level' and an 'Eye Pract Game' link.

An argument for semantics

This brings us to the importance of

STRUCTURE

vs.

PRESENTATION

Structure vs. Presentation

The **STRUCTURE** of an HTML document should be a well-crafted page that can be understood semantically by search engines and assistive devices like screen readers (aids for the blind).

Structure vs. Presentation

Additionally, **STRUCTURE** also incorporates the actual **CONTENT** of the document (e.g. headings, article text, links, pictures, video, etc).

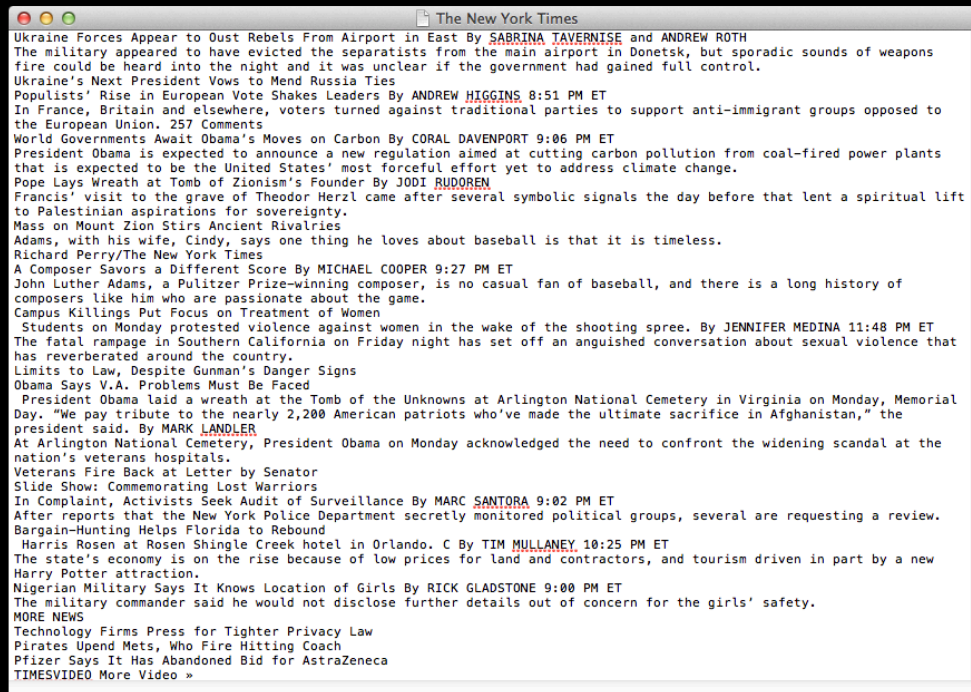
Structure vs. Presentation

The **PRESENTATION** half of the coin uses **VISUAL STYLING** to communicate the semantics to people not using screen readers.

This is accomplished by pairing **CSS** with the structure.

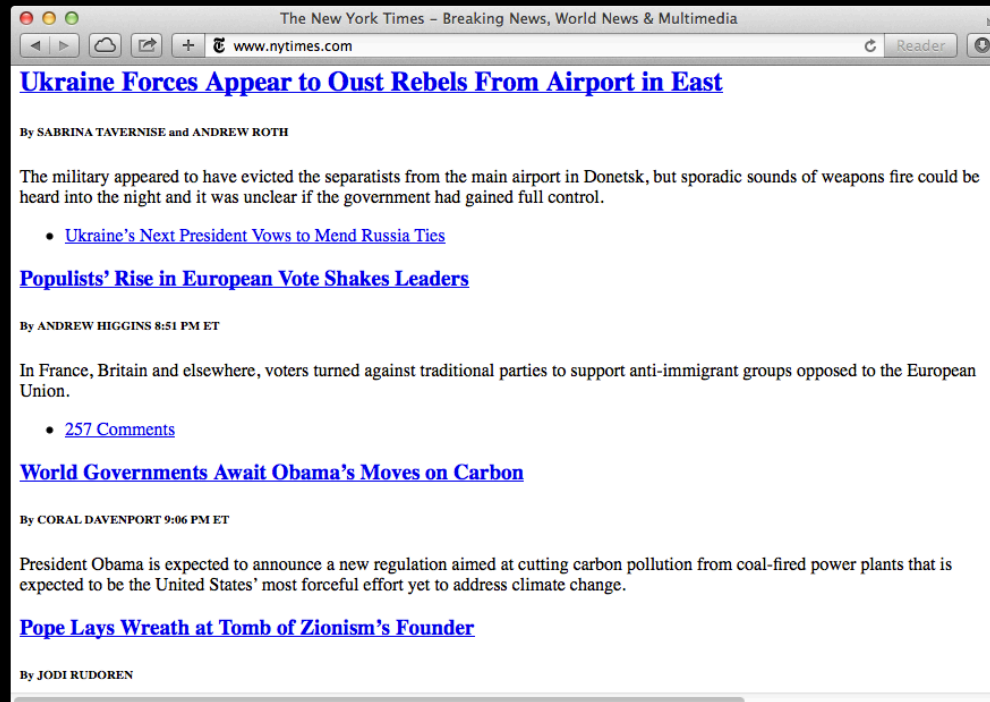
Structure vs. Presentation

To make a point about STRUCTURE vs. PRESENTATION, let's revisit the page with no semantic markup or structure at all:



Structure vs. Presentation

By simply adding semantic markup **STRUCTURE** we can have a meaningful page readable by search engines, screen readers, and even people:



Structure vs. Presentation

Now let's add CSS styling to it for better visual PRESENTATION:

The screenshot shows the New York Times website interface. The browser title is "The New York Times - Breaking News, World News & Multimedia" and the URL is "www.nytimes.com". The navigation bar includes categories like WORLD, U.S., NEW YORK, OPINION, BUSINESS, TECHNOLOGY, SCIENCE, HEALTH, SPORTS, ARTS, FASHION & STYLE, and VIDEO. The main content area features several articles:

- Ukraine Forces Appear to Oust Rebels From Airport in East** by Sabrina Tavernise and Andrew Roth. The article text states: "The military appeared to have evicted the separatists from the main airport in Donetsk, but sporadic sounds of weapons fire could be heard into the night and it was unclear if the government had gained full control." Below the text is a photo of two people in yellow raincoats, one pointing upwards. The caption reads "Richard Perry/The New York Times".
- A Cable Merger Too Far** by The Editorial Board. The text says: "Regulators should block Comcast's acquisition of Time Warner Cable because it would control too much of what consumers watch, read and listen to."
- Today's Times Insider** section with sub-articles: "A Cover With an Artistic Challenge" and "The Ethic of The Ethicist".
- MARKETS** section showing data for Japan (Nikkei: 14,703.49, +100.97, +0.69%), HangSeng (22,907.87, -55.31, -0.24%), and China (Shanghai: 2,036.97, -4.51, -0.22%).
- Science Times** section with sub-articles: "Craft Beer, at the Genetic Level" and "Eye Pract Game".

Other visible articles include "Ukraine's Next President Vows to Mend Russia Ties" by Andrew Higgins and "Populists' Rise in European Vote Shakes Leaders" by Andrew Higgins. The "Campus Killings Put Focus on Treatment of Women" article by Jennifer Medina is also partially visible.

Structure vs. Presentation

So, you might ask:

Is there ever a time to use non-semantic tags?

SURE.

<DIV>

The only two generic html tags that have no semantic meaning at all.

<div> and

There are occasions when you need to use generic tags for PRESENTATION purposes and really don't want a tag container to have any special meaning.

That is when you use either <div> or .

What's the difference between the two?

`<div>`

A “block-level” container tag.

“Block-level” elements take up the entire width of a line unless otherwise specified in the CSS.

``

An “inline” container tag.

“Inline” elements take up only as much space as necessary and can sit side-by-side with other “inline” elements *within a line*.

<div> “block” example

```
<div>div example 1</div>
```

```
<div>div example 2</div>
```

div example 1

div example 2



 “inline” example

span example 1

span example 2

span example 1

span example 2

