Web Applications History

Anatomy of a Web Application

HTML CSS: Syntax and Semantics

Introduction: The Web, HTML and CSS CPEN 322 - Building Modern Web Applications - Winter 2021-1

Karthik Pattabiraman

The University of British Columbia Department of Electrical and Computer Engineering Vancouver, Canada





September 9, 2021

tory Anato

natomy of a Web Application

HTML CSS: Syntax and Semantics

Web Applications: what are they ?





2 History

3 Anatomy of a Web Application

4 HTML

6 CSS: Syntax and Semantics

・ロト・日本・日本・日本・日本・今日を

Web Applications Anatomy of a Web Application CSS: Syntax and Semantics

Web Application: Definition

0.00



- A client-server software application in which the client (or UI) runs inside a web browser
- What's a client-server application ?
 - Distributed between 2 machines, client and server
- What's a web browser ?
 - Software application to view web content

Web Application: Differences from Desktop Applications

- Web Applications need a browser connected to the web all the time (for the most part)
- Web applications do their processing and storage at both the client and the server
- Web applications do not require cumbersome software installation (on the client)

HTML CSS: Syntax and Semantics

▲ロ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ● ○ ○ ○

Why Web Applications ?



- Ease of installation and use
- Standard UI and interface across platforms
- Service provider retains control of code distribution. No need to patch, upgrade etc.
- Allow compute and data-storage intensive functionality to be offloaded to the server

CSS: Syntax and Semantics

History of the Internet and the Web





History







▲ロ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ● ○ ○ ○

1960's:-1980's The early days



- 1960s: Rise of the ARPAnet and packet switched networks at UCLA and DARPA
- 1970s: Development of TCP/IP protocol stack at Stanford, Univ. College London, BBN Tech.
- Internet in the 1980's:
 - TCP/IP becomes standard protocol on ARPANET
 - Tim Berners Lee invents Hyper Text Markup Language (HTML)

History Anatomy of a Well

Anatomy of a Web Application HTML

HTML CSS: Syntax and Semantics

1985-1990: HTML and Web Browsers



- HTML: Way to embed hyperlinks within Documents to navigate to new documents
 - Invented by Tim Berners Lee at CERN for physicists to share documents in the late 80's
 - Extension of the SGML (Standard Generalized Markup Language) in the mid 1980's
 - Formalized as the HTTP protocol by Lee (1992)
 - Early web browsers were text-based e.g., Lynx
 - Competition: Gopher protocol, Archie client

History Anatomy of a Web

Veb Application HTML

CSS: Syntax and Semantics

Tim Berners Lee's Nexus Browser



• Source: Digital-archeology.org

CSS: Syntax and Semantics

1990-95: Web Browsers

History



- Mosaic: One of the first graphical browsers developed by the Univ. of Illinois/NCSA
 - Led by Marc Anderson and Eric Bina in 1993
 - Marc and Eric left UIUC to start Netscape
 - Netscape Navigator released in 1994
 - UIUC licenses Mosaic code to SpyGlass Inc. which released Spyglass Mosaic
 - Microsoft acquires Spyglass. Releases it as IE 1.0 in 1995. Browser wars start !

Web Applications History

HTML CSS: Syntax and Semantics

Mosaic Web Browser: Screenshot







Web Applications History

HTML CSS: Syntax and Semantics

1995-2000: Browser Wars



- Web browser market was hotting up
 - Netscape Navigator Versus Internet Explorer
 - Netscape had more features and was more robust
 - IE had the marketing power of Microsoft and was bundled with Windows as a free download !
 - JavaScript 1.0 was developed and integrated with Netscape by Brendan Eich (more on this later)
 - In 1997, Microsoft released IE4 which had feature parity with Netscape and integrated with Windows
 - Netscape was never the same again. They were sold to AOL and essentially lost the browser wars by the early 2000s

CSS: Syntax and Semantics

Netscape Navigator 4 vs IE 4



Netscape Navigator 4

History



Internet Explorer 4

Nain Page - Wikipedia, the free encyclopedia - Nicrosoft Internet E	splorer	× D ×
Ele Edit View Go Fgyosiles Help		æ
++ . → . 3 C G G	ales Hatay Dannes Fullcoren Hat Pint	
ddress 🔕 http://en.wikipedia.org/wiki/Main_Page	<u>×</u>	Hypefinks
		-
vlain Page		
from Wikipedia, the free encyclopedia		- 11
um to natioation search		
wrop we merrynerich arteristi		
Welcome to Wikipedia.	Arts History Socie	sv.
the free encyclopedia that	Biography Mathematics Lech Geography Science All pr	ortals
2.253.161 articles in English		_
Overview - Editing - Questions - Help	Contents - Categories - Featured content - A-J	index
Today's featured article	In the news	
	• The	
1933	Academy	
	Awards	
	held Market	-
	likemet	

HTML CSS: Syntax and Semantics

ション ふゆ アメビア メロア しょうくり

2000-2005: Browser Wars Part 2



- But, while Netscape was finished, they released their code to the Mozilla foundation
 - Team of dedicated volunteers that rebuilt Navigator from scratch and ironed out its quirks
 - First release in 2004. Rapid releases in 2005,2006
 - First browser that was standards compatible
 - Microsoft became complacent. Removed most core staff from the IE team and didn't develop it.
 - By 2006, Mozilla (Firefox) was back in the game

Anatomy of a Web Application

CSS: Syntax and Semantics

2005-2009: The Rise of AJAX

History



AJAX (asynchronous JavaScript and HTML)

- Feature introduced by Microsoft IE in their Outlook Web Access (OWA) client in the early 2000s
- Google used it for Google maps and Gmail (2004), and suddenly it was all the rage (2005)
- JavaScript became the new popular kid on the block, and JavaScript performance started to become increasingly important with Mozilla taking the lead
- Google introduced Chrome in 2008 which was primarily about faster JavaScript execution and support - based on Webkit development engine

UBC

AJAX Applications

History

C*				
the second se	mak A TAN	Zammi Outla	al Marah A a	
The n	ISL AJA	x app: Outio	OK WED AC	cess
		\cap		
		()		
Outlook Web Access	Bideos (ressages +)		7.64 T 10250	18 N I I I I
Palari 🔁 🖻	20m · G X C C C	O • CO F D P He		Care CP
a (g Rotatal, Habe	1 0 4 From	Subject	Terred 4	Size V
a diatata	 Arris, Cal 	Str. Nac	PH 20230908 7-02 9H	10
(2 Debted Base	L3 6 Nowle, Water	historial Futurial	AH GOTOKA DIO NA	381-00
Calendo	15 Selana	RC: Hane	DV 702020KK 2:13 PM	6140
S 2220/7	ta jetorea	FC-status	NY 2020086 2112 NH	0-0
C 200	Lip Serie, Gal	Part	P-1 70250385 2104 PM	110
A DUTO	Lá Xésoneo	FC:stiffus	PH 20202065 1:54 PH	110
Aver and	ing rates Capetar	HE's Schuld Mananis and Codes	PH 723(DIKE 1-32 PH	0.0
in the last	La nere capeta	RESOLUTION ALCORN	PR ALQUES LIVER	200
Carboline .	the landstar	RE CENTRES	ALCORDON TO MA	8.40
City Dank Kawley	to the capeta	RE, Schut Makes and Codes	ALCOURS DIS M	208
10 tag fight blocks	T To Later, Later	HE CANNON	ALCORPORT LICENT	100
2 Tabu	10 M 59744	all status	PROCESSION LONG PR	
	FE: Mix			
A 1 5	Service Gal			
1000CT	Co.			
babes				
- Calcodar				() ()
Cantara	That's fine. The 2m	at A112 is may office.		100
7 take				
Tuble Fallers	There is a drop kek	ad the two-drawer file cabinet t	bat coald be ande	
	sective new case 1 is			
Charles .				

Google Search (AutoComplete)



 Web Applications
 History
 Anatomy of a Web Application
 HT

 000
 00000000000
 00000000000
 000

HTML CSS: Syntax and Semantics

2010-2015: JavaScript Everywhere



- JavaScript becomes mainstream
 - More and more webpages start heavily using JS, often with thousands of lines of minified code
 - New applications (e.g., Google docs, Office live etc.) and frameworks (e.g., Jquery)
 - Academic papers are written about JavaScript in terms of its performance, reliability, security
 - JavaScript is the most popular language on Github, Stackoverflow and is in the top 5 on the Tiobe Index
 - JavaScript is also used to teach introductory CS by Khan Academy – often taught as the first language
 - Many variants of JavaScript: TypeScript (MS), DART (Google) and Flow (Facebook). Also, HTML5
 - Language for programming IoT devices (Samsung)

Web Applications History Ocoococococo Anatomy of a Web Application HTML CSS: Syntax and Semantics

Prevalence of JavaScript (around 2012)

- 97 of Alexa top 100 websites use JavaScript
- Many of them have thousands of lines of code



イロト 不得 トイヨト イヨト

э

Lines of code

tory Ana

Anatomy of a Web Application

HTML CSS: Syntax and Semantics

Anatomy of a Web Application





2 History

3 Anatomy of a Web Application

4 HTML

6 CSS: Syntax and Semantics

CSS: Syntax and Semantics

▲ロ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ● ○ ○ ○

Web Application Components



- Three pre-requisites for a web application
 - Server: To "serve" the web-page and to send content to the client
 - Client: To receive content from the server and display them on the web browser window
 - HTTP connection for client-server interactions
- Everything else is optional.

Traditional Web Applications



- Do not have (much) interactivity on the client
- Getting new content involves navigating to a new webpage or reloading the webpage
 - Typically through a hyper-link
 - Limited information passing from the client (typically in the form of a URL with '?')
- E.g., http://www.google.com?search="CPEN400A"



200

3

ション ふゆ アメビア メロア しょうくり

Modern Web Applications



- Client is a lot more interactive and actively renders content within the browser
- Application logic is split between client and server client can execute (JavaScript) code
- No need to reload the web page for updating the state of the page being displayed (DOM)
- Rich message passing interface with the server through AJAX messages

Anatomy of a Web Application

CSS: Syntax and Semantics

Modern Web Applications (2)





Anatomy of a Web Application

CSS: Syntax and Semantics

▲ロ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ● ○ ○ ○

Components of a Modern Web App



- Client-side components
 - HTML/DOM
 - CSS
 - JavaScript
- AJAX messages (client-server interactions)
- Server code (node.js or any other platform)

▲ロ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ● ○ ○ ○

Client: HTML/DOM



- Hyper-text markup language to describe the structure and contents of the initial page
 - Also has pointers to the JavaScript code (e.g., <script>)
- Is retrieved by the browser and parsed into a tree called the Document Object Model (DOM)
 - Common way for elements to interact with the page
 - Can be read and modified by the JavaScript code
 - Modifications to the DOM are rendered by browser

◆□▶ ◆□▶ ◆三▶ ◆三▶ ・三 ・ つへぐ





- CSS (Cascading style sheets) separate the content of the page from its presentation
- Written in the form of declarative rules with a element on LHS and action to apply on RHS
- Ensure uniformity by applying the rule to all elements of the webpage in the DOM

Client: JavaScript



- Unique to modern web applications and provides active functionality
- Executed when <script> tag is encountered or when events are triggered on DOM elements
- Is a full-fledged programming language with many advanced features (and some bad ones)

Web Applications History

ション ふゆ アメビア メロア しょうくり

AJAX Messages



- Way for client and server to interact with each other without navigating to a new webpage
- Initiated by the client by sending a message asynchronously (activities can happen in parallel after send on the client)
- Request is processed by server, and a response is returned. Response triggers a registered callback at the client.
- Call-back can do whatever it wants with the response, though typically it uses it to update the DOM

 Web Applications
 History
 Anatomy of a Web Application
 HTML
 CSS: Syntax and Semantics

 000
 00000000000
 000000000000
 000000000000
 0000000000000
 0000000000000





- Server can be written in whatever language one wants, as long as it serves the content and responds to AJAX requests from the client
- Typical languages used: Java, RoR, PHP etc.
- Recently, JavaScript is also being used to write server-side applications for uniformity and portability. Example of this is Node.js

HTML CSS: Syntax and Semantics

Other State Elements



- Cookies
- Persistent local storage (HTML5 and variants)
- Server Database (e.g., MongoDB)
- Canvas (HTML5)

y Anatomy of a Web

HTML CSS: Syntax and Semantics

HTML: The Base Substrate of a Web App



- Web Applications
- 2 History
- 3 Anatomy of a Web Application



6 CSS: Syntax and Semantics

HTML: What is it?



- Hierarchical way to organize documents and display them (typically in a web browser)
- Combines semantics (document structure) with presentation (document layout)
- Allows tags to be interspersed with document content e.g., <head> - these are not displayed, but are directives to the layout engine



Web Applications History

HTML: Head



- Is typically NOT displayed by web browser
- Contains metadata to describe the page
 - Title of the webpage: $\langle title \rangle TITLE \langle /title \rangle$
 - Style of the webpage: <style> style rules </style>
 - Link to CSS stylesheets: <link rel="stylesheet" type="text/css" href="">
 - For search engines: <meta name="" content="">
 - Embed JavaScript: <script > Javascript code </script> OR <script src="Javascript file"></script>



HTML: Body



▲ロ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ● ○ ○ ○

- Contains the actual contents of the page with HTML tag descriptors for the structure
- Common tags used in HTML

<div></div>	group elements spanning multiple lines line break before and after
	group elements within a single line
< p >	new paragraph
<br $>$	line break

b Applications History

Anatomy of a Web Application

HTML CSS: Syntax and Semantics



HTML: Body (2)

<h1>,,<h6></h6></h1>	headings
	images
	hyperlink

	tables
<	unordered list
< ol >< li >	ordered list
<form><input/></form>	forms that take in user input

 Web Applications
 History
 Anatomy of a Web Application
 HTML
 CSS: Syntax and Semantics

 The Currique "div"
 cloment
 Image: Control of the curricular set of the curicular set of the curicula

The Curious "div" element



ション ふゆ アメビア メロア しょうくり

- Div's are a way to separate different sections of a page and have no meaning by themselves
 - Used to group together semantically related elements in the same portion of the document
 - Allows semantic attributes such as CSS styling or JavaScript code to be applied to div elements
 - div elements can be nested within each other
- Use of div's allows easier rendering of pages, and adds semantic meaning to webpages (good)

▲ロ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ● ○ ○ ○

Div element: Example



- Div element is used to group menu items
 - id can be used within JavaScript (JS) code to access it must be unique to the element
 - class is used for indicating type of element need not be unique, and is used for multiple elements in JS
 - background is a style to apply to all elements in div



- span is an inline version of 'div', for separating small chunks of the document without a line break. Cannot contain other div elements in it
- Mostly for applying styling rules to small segments of the webpage without line-breaks

```
1 <div id="menu1" class="Menu" style="background: #d3e7dd;">
2 <a href="index.html">Home</a> |
3 <a href="about.html">Home</a> |
4 <a href="faq.html">FAQ</a> |
5 <span class="italics"><a href="contact.html">Contact Us</a> |
6 </div>
```

▲ロ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ● ○ ○ ○

HTML CSS: Syntax and Semantics

▲ロ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ● ○ ○ ○

Why use Divs and Spans?



- Divs and spans are very useful to break a page into semantically related elements
 - Search engines like Google rely on these to find related information
 - Provide hooks to your webpage from CSS, and especially JavaScript code (more on this later)
 - Makes it easier to render across platforms
- However, overuse of these makes webpages hard to read, and also slower (Especially on mobile)

HTML CSS: Syntax and Semantics

Browser's View of HTML: DOM



 HTML is parsed by the browser into a tree structure -Document Object Model (DOM)



◆□▶ ◆□▶ ◆三▶ ◆三▶ ・三里 ・のへで



DOM: Example



• Often one-to-one correspondence between HTML and the DOM rendered by browser



◆ロト ◆御 ト ◆臣 ト ◆臣 ト ○臣 - のへで

Web Applications His

ション ふゆ アメビア メロア しょうくり

DOM: Why is it Important?



- Common data-structure for holding elements of a web-page (HTML, CSS, JavaScript etc.)
 - No need to worry about parsing HTML, CSS etc.
- Corresponds almost exactly to the browser's rendered view of the document
 - Changes to the DOM are made (almost) immediately to the rendered version of the webpage
 - Heavily used by JavaScript code to make changes to the webpage, and also by CSS to style the page

Web Applications History 000 00000000000

Anatomy of a Web Application 0 0000000000

HTML CSS: Syntax and Semantics

ション ふゆ アメビア メロア しょうくり

DOM Structure: Disadvantages



- No isolation between different parts of the DOM tree for a script as long as its from the same origin
 - All scripts from same origin (i.e., domain) can access the entire DOM tree from that origin
 - Scripts can clobber the DOM and leave it that way
 - Highly dynamic difficult to reason about DOM state
- DOM is also very browser-specific (not standard)
- Can be a significant bottleneck in rendering webpages in parallel as it is a single global structure

Veb Applications History Anato

Anatomy of a Web Application

HTML CSS: Syntax and Semantics

What do Web Browsers do?



- Example from Webkit: Used by Chrome and Safari
- (source:

http://www.html5rocks.com/en/tutorials/internals/howbrowserswork





Class Activity



Draw the DOM tree corresponding to the following HTML code:

```
<html>
  1
  2
      <head>
  3
               <title> ... </title>
  4
                <script> ... </script>
  5
          <script> ... </script>
  6
              <script> ... </script>
  7
        </head>
  8
       <body>
  9
                <div> <A> ... </A> <text> ... </text> <h1> ... </h1>
                          div>
10
              < div >
11
                        <div> <A> ... </A> <text> ... </text> <h1> ... </h1> <</pre>
                                  /div>
12
                        <u>... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... </l
13
                        > ... 
14
             </div>
15
                <div> ... </div>
16
     </body>
17
        </html>
```

 Web Applications
 History
 Anatomy of a Web Application
 HTML
 CSS: Syntax and Sema

 000
 00000000000
 000000000000
 0000000000000
 0000000000000000

Class Activity - Solution



UBC

▲□▶▲圖▶▲≣▶▲≣▶ = 差 - 釣�?

Anatomy of a Web App

HTML CSS: 5

CSS: Syntax and Semantics

CSS: Syntax and Semantics



- Web Applications
- 2 History
- 3 Anatomy of a Web Application
- 4 HTML
- **5** CSS: Syntax and Semantics



y Anatomy of a Web Application

ション ふゆ アメビア メロア しょうくり

CSS: Philosophy and Motivation



- Language for specifying how (HTML) documents are presented to users (Separate from content)
- Declarative set of rules and their actions
 - Makes it easy to modify and maintain the website
- Allows different rules to be specified for different display formats (e.g., printing versus display)

ry Anatomy of a Web Application

tion HTML CSS: Syntax and Semantics

Including CSS in HTML: Example



▲ロ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ● ○ ○ ○



- (Anyone including their CSS code directly in the HTML?
 - I won't judge, but...)

50

Example of CSS stylesheet (style1.cc)



- strong: tag to match
- color: red;: attribute: value

```
1 2 <strong>C</strong>ascading
3 <strong>S</strong>tyle
4 <strong>S</strong>heets
5
```



・ロト・日本・日本・日本・日本・日本



How does CSS work?

51

- Apply styles to the DOM tree of the web page
- CSS rule applies to DOM nodes matching tag, and their descendants (unless overridden)



Here, all STRONG tags will be matched \Rightarrow all descendants of STRONG tags will be *styled*.

・ロト・西ト・田・・田・ シック

Web Applications History Anatomy of a Web Application HTML CSS: Syntax and Semantics

CSS Inheritance



▲ロ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ▲ □ ▶ ● ○ ○ ○

- All descendants of a DOM node inherit the CSS styles ascribed to it unless there is a "more-specific" CSS rule that applies to them
- Always apply style rules in top down order from the root of the DOM tree and overriding the rules as and when appropriate
 - Can be implemented with an in-order traversal

```
1 p {color:blue; text-decoration:underline}
2 strong {color:red}
```

```
1 

2 <strong>C</strong>ascading

3 <strong>S</strong>tyle

4 <strong>S</strong>heets

5
```





• CSS rules can also apply to elements of a certain class or an element with a specific ID

1	.key {	1 #principal {
2	<pre>color: green; }</pre>	<pre>2 font-weight: bolder; 3 }</pre>

1

Anatomy of a Web Application

HTML CSS: Syntax and Semantics

CSS Rules and Priority



- What to do when rules conflict with each other ?
 - Always apply the "most specific selector"
- "Most-specific" ('>' represents specificity):
 - ${\ensuremath{\, \circ }}$ Selectors with IDs > Classes > Tags
 - Direct rules get higher precedence over inherited rules (as before)

Anatomy of a Web Application

CSS Class and IDs: Example

55



History Anatomy of a Web Application

CSS Selectors based on Relationships



- Selectors can also be based on relationships between elements in the DOM tree
 - A E : Any element E that is a descendant of A
 - $\bullet~A>E$ Any element E that is a child of A
 - E : first-child: Any element E that is the first child of its parents
 - B + E : Any element E that is the next sibling of B element (i.e., B and E have the same parent)

 Web Applications
 History
 Anatomy of a Web Application
 HTML
 CSS: Syntax and Semantics

 CCCC
 Describe
 Class
 Sales
 Class
 Class
 Class

CSS Pseudo-Class Selectors



- CSS Selectors can also involve actions external to the DOM called pseudo-classes
- Visited: Whether a page was visited in the history
- Hover: Whether the user hovered over a link
- Checked: Whether a check box was checked

```
1 Selector : pseudo-class {
2     property: value
3 }
```

story Anatom

Anatomy of a Web Application

HTML CSS: Syntax and Semantics

CSS Selectors Relationships: Example



```
<div class="menu-bar">
    <11>
      <1i>>
       <a href="example.html">Menu</a>
        <1i>>
           <a href="example.html">Link</a>
         <1i>>
9
           <a class="menu-nav" href="example.html">Submenu</a>
10
           <u1>
             <1i>>
              <a class="menu-nav" href="example.html">Submenu</a>
              <11>
14
                <a href="example.html">Link</a>
                <a href="example.html">Link</a>
16
                <a href="example.html">Link</a>
                <a href="example.html">Link</a>
18
              19
             20
             <a href="example.html">Link</a>
21
           22
         </1i>
       24
      25
    </11>
26
  </div>
27
```



The first rule says that for all 'div' elements of class 'menubar', in which an ul element is a descendant of another ul, do not display the second element

The second rule says that for all 'div' elements of class 'menu-bar', in which an ul element is a child of an li element, display it and the entire block, if the mouse hovers over the second element

・ロト・西ト・西ト・西ト・日下

Class Activity: CSS Rules



• What's the effect of the following CSS spec. on the HTML code in the next two slides ?

```
#news { background-color: silver; font: italic; color: black; }
. sports { color: blue; text-decoration: underline; }
H3, H4 { font-family: sans-serif; }
.latest { color: green;}
#news span { color: red; }
P.select { font-size: medium; }
```

Source: Ali Mesbah and Shabnam Mirshokraie. 2012. Automated analysis of CSS rules to support style maintenance. In Proceedings of the 34th International Conference on Software Engineering (ICSE '12). IEEE Press, Piscataway, NJ, USA, 408-418.





Class Activity: HTML -1

• Use a rich text editor (i.e., Word, LibreOffice Writer, etc.) to draft a preview of the output of this example.

```
<HITML>
<HIML>
<HEAD>
<LINK href="example.css" rel="stylesheet" type="↔
    text/css"/>
</HEAD>
<BODY>
<P id='news' style='font:normal'>World
    <SPAN class='sports '>Sports news</SPAN>
</P>
<DIV class='sport '>Football</DIV>
</BODY>
</HIML>
```



Class Activity: HTML -2



• Use a rich text editor (i.e., Word, LibreOffice Writer, etc.) to draft a preview of the output of this example.

```
<hTML>
<hTML>
<hTML>
<hTML>
<hTML>
<hTML>
<hTML>
<br/>
<br/
```