Lecture 1: Internet/WWW/HTML

CS472 Web Programming

Maharishi University of Management

Department of Computer Science

Assistant Professor Asaad Saad

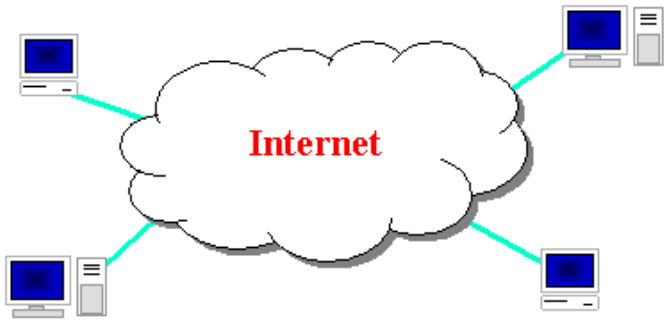
Maharishi University of Management - Fairfield, Iowa © 2016

Lecture: Introduction to Web Programming

In this lecture we introduce the basic technologies that make up the Internet, the World Wide Web and the Hyper Text Markup Language (HTML). We will see that many technologies are built on top of other technologies. *Life is found in layers.*

The Internet

- A connection of computer networks using the Internet Protocol (IP)
- layers of communication protocols: IP → TCP/UDP → HTTP/FTP/POP/SMTP/SSH...



What's the difference between the Internet and the World Wide Web (WWW)?

The Web is the collection of web sites and pages around the world; the Internet is larger and also includes other services such as email, chat, online games, etc.

Brief history

- Began as a US Department of Defense network called ARPANET (1960s-70s)
- Initial services: Electronic mail, File transfer
- Opened to commercial interests in late 80s
- WWW created in 1989-91 by Tim Berners-Lee
- Popular Web browsers released: Netscape 1994, IE 1995
- Amazon.com opens in 1995; Google January 1996

Key aspects of the internet

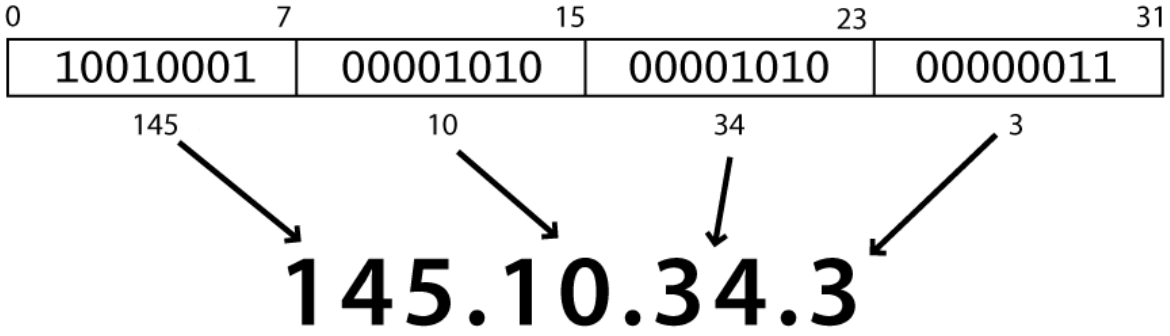
- Sub-networks can stand on their own
- Computers can dynamically join and leave the network
- Built on open standards; anyone can create a new internet device
- Lack of centralized control (mostly)
- Everyone can use it with simple, commonly available software

Organizations

- Internet Engineering Task Force (IETF): internet protocol standards
- Internet Corporation for Assigned Names and Numbers (ICANN): decides top-level domain names
- World Wide Web Consortium (W3C): web standards

Internet Protocol (IP) IPv4

- A simple protocol for attempting to send data between two computers
- Each device has a 32-bit IP address written as four 8-bit numbers (0-255)
- There are two types of IP addresses, servers used to have static IP address while users usually get a dynamic IP address from their ISP.
- Find out your local IP address: in a terminal, type: ipconfig (Windows) or ifconfig (Mac/Linux)



IPv6 addresses are 128-bit IP address written in hexadecimal and separated by colons. An example IPv6 address could be written like this: 3ffe:1900:4545:3:200:f8ff:fe21:67cf

Transmission Control Protocol (TCP)

- Adds multiplexing, guaranteed message delivery on top of IP
- **Multiplexing**: multiple programs using the same IP address
  - **port**: a number given to each program or service
  - port 80: web browser (port 443 for secure browsing)
  - port 25: email
  - port 22: ssh
  - port 5190: AOL Instant Messenger
  - more common ports
- Some programs (games, streaming media programs) use simpler UDP protocol instead of TCP

Hypertext Transport Protocol (HTTP)

- The set of commands understood by a web server and sent from a browser
- Some HTTP commands (your browser sends these internally):
  - **GET** *filename* : download
  - **POST** *filename* : send a web form response
  - **PUT** *filename* : upload

Simulating a browser with a terminal window:

Request

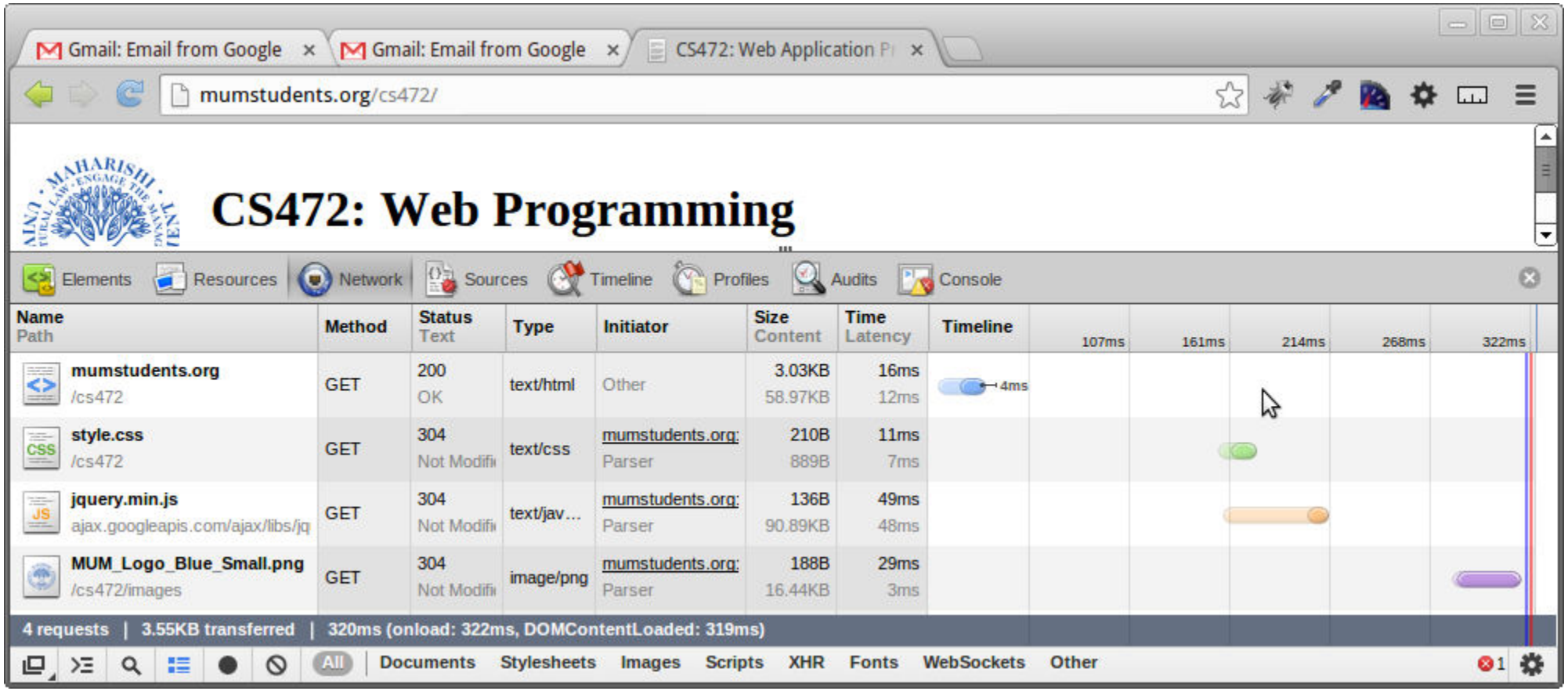
```
GET /index.html HTTP/1.1
HOST: mumstudents.org
```

Response

```
HTTP/1.1 200 OK
Date: Sun, 30 Aug 2015 16:08:38 GMT
Server: Apache/2.4.7 (Ubuntu)
Last-Modified: Fri, 12 Dec 2014 20:23:01 GMT
Accept-Ranges: bytes
Content-Length: 942
Content-Type: text/html

<!DOCTYPE html>
<html>
...
```

Request and Response



Web servers and Browsers

- **Web server**: software that listens for web page requests
  - Apache
  - Microsoft Internet Information Server (IIS)
- **Web browser**: fetches/displays documents from web servers
  - Mozilla Firefox
  - Microsoft Internet Explorer (IE)
  - Apple Safari
  - Google Chrome
  - Opera

Domain Name System (DNS)

- A set of servers that map written names to IP addresses
  Example: www.cs.mum.edu → 69.18.50.54
- Many systems maintain a local cache called a hosts file
  Windows: `C:\Windows\system32\drivers\etc\hosts`
  Mac: `/private/etc/hosts`
  Linux: `/etc/hosts`

Uniform Resource Locator (URL)

- An identifier for the location of a document on a web site

a basic URL:http://www.abc.com/info/index.html

~~~~ ~~~~~~~ ~~~~~~~~~~

protocol host path

- Upon entering this URL into the browser, it would:
  - Ask the DNS server for the IP address of www.abc.com
  - Connect to that IP address at port 80
  - Ask the server to GET /info/index.html
  - Display the result page on the screen
~~~~

More advanced URLs

- **Anchor**: jumps to a given section of a web page
  - http://www.textpad.com/download/index.html#downloads
  - fetches **`index.html`** then jumps down to part of the page labeled downloads
- **Port**: for web servers on ports other than the default 80
  - http://www.cs.mum.edu:8080/secret/money.txt
- **Query string**: a set of parameters passed to a web program
  - http://www.google.com/search?q=miserable+failure&start=10
  - parameter q is set to "miserable+failure"
  - parameter start is set to 10

HTTP error codes

When something goes wrong, the web server returns a special "error code" number to the browser, possibly followed by an HTML document, common error codes:

Number	Meaning
200	OK
301-303	page has moved (permanently or temporarily)
403	you are forbidden to access this page
404	page not found
500	internal server error

complete list

Web languages / technologies

- Hypertext Markup Language (HTML): used for writing web pages
- Cascading Style Sheets (CSS): stylistic info for web pages
- PHP Hypertext Processor (PHP): dynamically create pages on a web server
- JavaScript: interactive and programmable web pages
- Asynchronous JavaScript and XML (Ajax): accessing data for web applications
- eXtensible Markup Language (XML): metalanguage for organizing data
- Structured Query Language (SQL): interaction with databases

History Of HTTP/1.1

- HTTP is an old protocol, initially defined in 1991, with the last major revision HTTP/1.1 published in 1999.
- Daniel Sternberg notes that the amount of data now required to load the home page of an average website is **1.9 MB**, with over 100 individual resources required to display a page — a “resource” being anything from an image or a font to a JavaScript or CSS file.
- HTTP/1.1 does not perform well when retrieving the large number of resources required to display a modern website.

For more info https://www.smashingmagazine.com/2016/02/getting-ready-for-http2/

SPDY

In 2009, two engineers at Google posted about a research project they had been working on named SPDY. This project addressed some of the issues in HTTP/1.1. SPDY set out to:

- Allow concurrent requests across a single TCP connection, known as multiplexing.
- Allow browsers to prioritize assets so that resources vital to the display of a page could be sent by the server first
- Compress and reduce HTTP headers
- SPDY requires an encrypted (HTTPS) connection between the browser and server
- In addition, SPDY requires an encrypted (HTTPS) connection between the browser and server.
- SPDY doesn't replace HTTP, it's a **tunnel for the protocol** and modifies the way existing HTTP requests and responses are sent.
- It **requires support from both the server and the browser** connecting to that server.

Support for SPDY has been dropped in Edge due to Microsoft implementing support for HTTP/2, the latest version of the HTTP protocol. While other current browsers still maintain support for SPDY, Chrome will remove support in 2016, and other browsers will likely follow.

HTTP/2

- HTTP/2 is backwards-compatible with HTTP/1.1, The protocol change is completely transparent to users.
- All of the browser vendors have decided to only implement HTTP/2 for TLS (https) connections.
- If you have a Gmail account and use Chrome to access it you will have been using SPDY and then HTTP/2 without knowing anything about it.

ADVANCED ONLY – EXCLUDED FROM EXAM

Turning Multiple Image Files Into Sprites

- In HTTP 1.1, retrieving one large image is much more efficient for the browser than making a lot of requests for small ones. This is because multiple requests queue up behind each other. To work around this, we have been advised to turn our little icons into a sprite file.
- With the **multiplexing ability of HTTP/2**, this queuing of resources is no longer an issue. Serving the small images individually will be better in many cases; you will only need to serve what is required for the page that the visitor is on.

ADVANCED ONLY – EXCLUDED FROM EXAM

Inlining Images Using Data URIs

- Another workaround for the problem of multiple HTTP requests in HTTP/1.1 was to inline images in CSS using data URIs. Embedding images in this way will make a style sheet much larger. If you have combined this with another optimization technique for concatenating assets, then a visitor will likely download all of this code even if they never visit the pages where the images are being used.
- With HTTP requests being very cheap in HTTP/2, this “best practice” will create difficulties rather than help performance.

ADVANCED ONLY – EXCLUDED FROM EXAM

Splitting Resources Between Hosts: Sharding

- With HTTP/1.1, you are restricted to the number of open connections. If loading a large number of resources is unavoidable, one method to get around this restriction is to retrieve them from multiple domains. This is known as domain sharding. This can achieve better loading times, yet can cause problems itself, not to mention the development overhead of preparing this for your website.
- **HTTP/2 removes this need for domain sharding** because you can request as many resources as you need. In fact, this technique will likely hurt performance because it creates additional TCP connections and hinders HTTP/2 from prioritizing resources.

How browsers display a webpage

- User machines have IP on the Internet
- Server machines have IP and Domain Name
- Domain names and IP addresses are registered at global DNS Server
- When the user opens a browser window and ask for www.test.com
- First, the browser will check the local DNS (host file) for the IP address of that domain
- If not found, it will connect to ISP and ask it for the DNS
- Once retrieved, the browser will send another request to that server
- Requests are carried by TCP protocol, and standardized by HTTP or HTTPS protocol
- The server will send the browser a response with HTML code.
- The browser will interpret the HTML code line by line and start building the web page.
- For every resource not found in the browser cache, the browser will send a new request to the server again asking for that resource and so on.

Main Point

The internet is a global computer network that uses the IP protocol to uniquely identify computers on the network. Through the TCP protocol each IP address can work with multiple services at the same time. One of these services is the HTTP protocol which is used by the World Wide Web to transport HTML pages. These are the different layers of the internet. *The benefit of TM is that it settles the mind so that it more easily becomes aware of the deeper layers affecting the relative.*

Hypertext Markup Language (HTML)

- Describes the content and structure of information on a web page
  - Not the same as the presentation (appearance on screen)
- Surrounds text content with opening and closing tags
- Each tag's name is called an element
  - Syntax: **`<element>`** content **`</element>`**
  - Example: **`<p>`**This is a paragraph**`</p>`**
- Most whitespace is insignificant in HTML (ignored or collapsed to a single space)
- We will use a newer version called HTML5

Structure of an HTML5 page

- The **`header`** describes the page and the **`body`** contains the page's contents
  - An HTML page is saved into a file ending with extension **`.html`**
- **`DOCTYPE`** tag tells browser to interpret our page's code as HTML5.

```
<!DOCTYPE html>
<html>
    <head>
    <meta charset="utf-8" />
        information about the page
    </head>
    <body>
        page contents
    </body>
</html>
```

Web page metadata: `<meta>`

Information about your page (for a browser, search engine, etc.)


```
<meta charset="utf-8" />
<meta name="description" content="Learn HTML" />
<meta name="keywords" content="HTML, CSS" />
```

- Placed in the **`head`** section of your HTML page
- **`meta`** tags often have both the **`name`** and **`content`** attributes
- Some meta tags use the **`http-equiv`** attribute instead of name
- The **`meta`** tag with **`charset`** attribute indicates language/character encodings

Favorites icon ("favicon")


```
<link href="filename" type="MIME type" rel="shortcut icon" />
<link href="yahoo.gif" type="image/gif" rel="shortcut icon" />
```

- The link tag, placed in the head section, attaches another file to the page
- In this case, an icon to be placed in the browser title bar and bookmarks
- IE6: Doesn't work; must put a file favicon.ico in the root of the web server (info)

Relative vs Absolute Path

Relative Paths

- index.html
- /graphics/image.png
- ../about.html
- ../../stories/stories.html
- /help/articles/how-do-i-set-up-a-webpage.html

Absolute Paths

- http://www.mysite.com
- http://www.mysite.com/graphics/image.png
- http://www.mysite.com/articles/webpage.html
- C:\website\images\image.png

Why the last example won’t work when we move our code to production?

Paragraph: <p>

Paragraphs of text (block)


```
<p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Nam enim nisl, adipiscing
quis ultrices a, egestas quis lorem.
Pellentesque ultrices nunc id mauris
posuere pulvinar.</p>
```

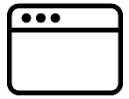
- Placed within the body of the page

More paragraph examples

Headings: <h1>, <h2>, . . . , <h6>

Headings to separate major areas of the page (block)


```
<h1>Maharishi University</h1>
<h2>Department of Computer Science</h2>
<h3>WAP Course</h3>
```



Maharishi University

Department of Computer Science

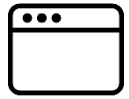
WAP Course

More headings examples

Links: <a>

- Links, or "anchors", to other pages (inline)
- **href** can be absolute or relative
- Anchors are inline elements; must be placed in a block element such as **p** or **h1**


```
<p> Search
        <a href="http://www.google.com/">Google</a> or our
        <a href="index.php">Lecture Notes</a>.
</p>
```

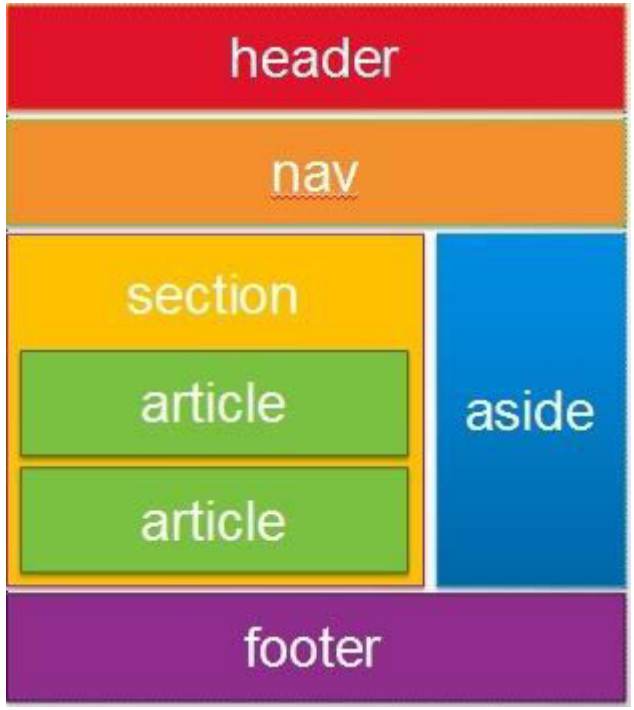


Search Google or our Lecture Notes.

Semantic Tags

They generally have no default outward appearance on the page, instead they give insight into the structure of the page.

- section
- header
- footer
- nav
- aside
- article
- More Semantic Tags



Images: <img>

- Inserts a graphical image into the page (inline)
- The `src` attribute specifies the image URL
- HTML5 also requires an `alt` attribute describing the image
- `title` attribute is an optional tooltip (on ANY element)


```
<img src="../images/WhiteLily.jpg" alt="White Lily" />
```

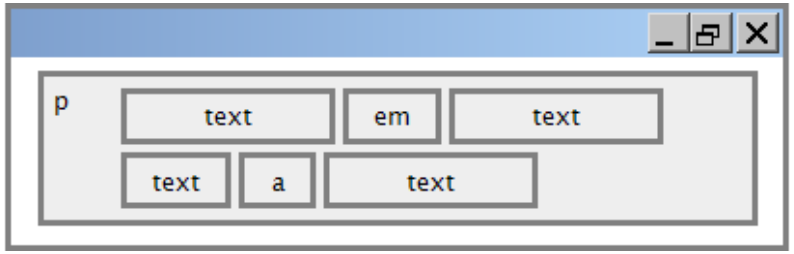
Exercise: Make a Blog

Make a Blog with links to interesting things on the Internet!

- Make the blog entries with `section` tags
- Each entry should have a title specified with an `h2`
- Some explanation about the entry in a `p` tag
- And a link to something (funny/interesting) online using an `a` tag.

Block and inline elements

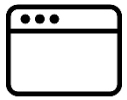
- block elements contain an entire large region of content
  - Examples: paragraphs, lists, table cells
  - The browser places a margin of whitespace between block elements for separation
- inline elements affect a small amount of content
  - Examples: bold text, code fragments, images
  - The browser allows many inline elements to appear on the same line
  - Must be nested inside a block element



Line break: `<br />`

Forces a line break in the middle of a block element (inline)


```
<p>Teddy said it was a hat, <br /> So I put it on.</p>
<p>Now Daddy's saying, <br /> Where the heck's the toilet plunger gone?</p>
```



Teddy said it was a hat,
So I put it on.

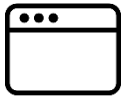
Now Daddy's saying,
Where the heck's the toilet plunger gone?

- Warning: Don't over-use `br` (guideline: >= 2 in a row is bad)
- `br` should not be used to separate paragraphs or used multiple times in a row to create spacing

Phrase elements : ,

- em: emphasized text (usually rendered in italic)
- strong: strongly emphasized text (usually rendered in bold)


```
<p> HTML is <em>really</em>,
<strong>REALLY</strong> fun! </p>
```



HTML is *really*, **REALLY** fun!

- As usual, the tags must be properly nested for a valid page.

Nesting tags

- Tags must be correctly nested
  - (a closing tag must match the most recently opened tag)
- The browser may render it correctly anyway, but it is invalid HTML
  - (how would we get the above effect in a valid way?)


```
<p> HTML is <em>really, <strong>REALLY</em>
lots of</strong> fun! </p>
```

Comments: <!-- -->

- Comments to document your HTML file or "comment out" text
- Many web pages are not thoroughly commented (or at all)
- Useful at the top of page and for disabling code


```
<!-- My web page, by Suzy Student CSE 190 D, Spring 2048 -->
<p>CSE courses are <!-- NOT --> a lot of fun!</p>
```

Main point

The Hyper Text Markup Language uses tags to demarcate different sections of a text. An HTML page always starts with a <html> tag, inside of which it has a <head> tag to describe the page, and a <body> tag of the contents that will actually be displayed. These are the tags you will use for every HTML page. *Seek the highest first, by starting with a good foundation and building upon that.*

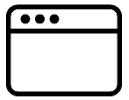
Web Standards

- It is important to write proper HTML code and follow proper syntax.
- Why use valid HTML and web standards?
  - More rigid and structured language
  - More interoperable across different web browsers
  - More likely that our pages will display correctly in the future
  - Can be interchanged with other XML data: SVG (graphics), MathML, MusicML, etc.

W3C HTML Validator

- Checks your HTML code to make sure it follows the official HTML syntax
- More picky than the browser, which may render bad HTML correctly


```
<p>
   <a href="http://validator.w3.org/check/referer">
      <img src="http://mumstudents.org/cs472/2013-09/images/w3c-html.png" alt="Validate" />
   </a>
</p>
```



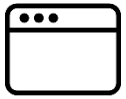
Main Point

By properly adhering to web standards we can assure better results across more platforms, and it is more likely that our page will continue to display properly in the future. By clearly stating what standard we adhere to every browser that supports the standard (now and in the future) will be able to correctly display our page. *The nature of life is to grow.*

Unordered list: `<ul>`, `<li>`

- `ul` represents a bulleted list of items (block)
- `li` represents a single item within the list (block)


```
<ul>
    <li>No shoes</li>
    <li>No shirt</li>
    <li>No problem!</li>
</ul>
```

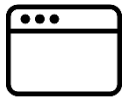


- No shoes
- No shirt
- No problem!

More about unordered lists

- A list can contain other lists


```
<ul>
    <li>Simpsons:
        <ul>
            <li>Homer</li>
            <li>Marge</li>
        </ul>
    </li>
    <li>Family Guy:
        <ul>
            <li>Peter</li>
            <li>Lois</li>
        </ul>
    </li>
</ul>
```

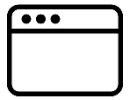


- Simpsons:
  - Homer
  - Marge
- Family Guy:
  - Peter
  - Lois

Ordered list: `<ol>`

- `ol` represents a numbered list of items (block)
- We can make lists with letters or Roman numerals using CSS (later)


```
<p>RIAA business model:</p>
<ol>
    <li>Sue customers</li>
    <li>Profit!</li>
</ol>
```



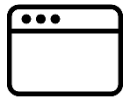
RIAA business model:

1. Sue customers
2. Profit!

Definition list: `<dl>, <dt>, <dd>`

- `dl` represents a list of definitions of terms (block)
- `dt` represents each term, and `dd` its definition


```
<dl>
  <dt>newbie</dt>  <dd>one who does not have mad skills</dd>
  <dt>own</dt>  <dd>to soundly defeat</dd>
  <dt>frag</dt>  <dd>a kill in a shooting game</dd>
 </dl>
```



newbie
: one who does not have mad skills

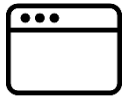
own
: to soundly defeat

frag
: a kill in a shooting game

Quotations: <blockquote>

A lengthy quotation (block)


```
<p>As Lincoln said in his famous Gettysburg Address:</p>
 <blockquote>
 <p>Fourscore and seven years ago, our fathers brought forth
on this continent a new nation, conceived in liberty, and
dedicated to the proposition that all men are created
equal.</p>
 </blockquote>
```



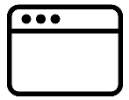
As Lincoln said in his famous Gettysburg Address:

> Fourscore and seven years ago, our fathers brought forth on this continent a new nation, conceived in liberty, and dedicated to the proposition that all men are created equal.

Inline quotations: <q>

A short quotation (inline)


```
<p>Quoth the Raven, <q>Nevermore.</q></p>
```



Quoth the Raven, “Nevermore.”

Why not just write the following? `<p>Quoth the Raven, “Nevermore.”</p>`

- HTML shouldn't contain literal quotation mark characters; they should be written as "
- Using <q> allows us to apply CSS styles to quotations (seen later)

HTML Character Entities

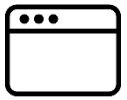
- A way of representing any Unicode character within a web page
- Complete list of HTML entities
- How would you display the text & on a web page?

character(s)	entity
< >	< >
é è ñ	é è ñ
™ ©	™ ©
π δ Δ	π δ Δ
И	И
" &	" &

HTML-encoding text

To display the link text in a web page, its special characters must be encoded as shown below


```
&lt;p&gt;
    &lt;a href=&quot;http://google.com/&quot;&gt; Google&lt;/a&gt;
&lt;/p&gt;
```

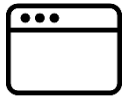


```
<p>
  <a href="http://google.com/"> Google </a>
</p>
```

Abbreviations: <abbr>

- An abbreviation, acronym, or slang term (inline)


```
<p> Safe divers always remember to check their
 <abbr title="Self-Contained Underwater Breathing Apparatus">SCUBA</abbr> gear.
</p>
```



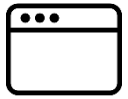
Safe divers always remember to check their SCUBA gear.

Self-Contained Underwater Breathing Apparatus

Computer code: `<code>`

A short section of computer code (usually shown in a fixed-width font)


```
<p> The <code>ul</code> and <code>ol</code> tags make lists. </p>
```

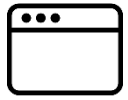


The `ul` and `ol` tags make lists.

Preformatted text: <pre>

A large section of pre-formatted text (block)


```
<pre>
        Steve Jobs speaks loudly reality
              distortion Apple fans bow down
</pre>
```



```
        Steve Jobs speaks loudly reality
              distortion Apple fans bow down
```

- Displayed with exactly the whitespace / line breaks given in the text
- Shown in a fixed-width font by default
- How would it look if we had instead enclosed it in code tags?

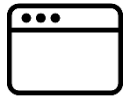
Main Point

We discussed some of the most common tags used on pages, including headings, paragraphs, images, and links. The most important concept to take away is to use tags based on their semantics (meaning), not based on their visual effect (which can easily be changed). By using tags for their meaning, clients (including non visual) will be better able to understand the meaning of your page. *The benefits of TM include a clearer understanding of deeper principles of life, allowing us to more easily take the right actions to achieve a desired result.*

HTML tables: <table>, <tr>, <td>

A 2D table of rows and columns of data (block element)


```
<table>
  <tr><td>1,1</td><td>1,2 okay</td></tr>
  <tr><td>2,1 real wide</td><td>2,2</td></tr>
</table>
```

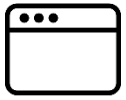


1,1	1,2 okay
2,1 real wide	2,2

- **`table`** defines the overall table, **`tr`** each row, and **`td`** each cell's data
- tables are useful for displaying large row/column data sets

Table headers, captions: `<th>, <caption>`


```
<table>
  <caption>My important data</caption>
  <tr><th>Column 1</th><th>Column 2</th></tr>
  <tr><td>1,1</td><td>1,2 okay</td></tr>
  <tr><td>2,1 real wide</td><td>2,2</td></tr>
</table>
```



My important data

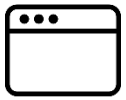
Column 1	Column 2
1,1	1,2 okay
2,1 real wide	2,2

- **`th`** cells in a row are considered headers; by default, they appear bold
- a **`caption`** at the start of the table labels its meaning

The `rowspan` and `colspan` attributes

- **`colspan`** makes a cell occupy multiple columns; rowspan multiple rows
- **`text-align`** and **`vertical-align`** control where the text appears within a cell


```
<table>
  <tr><th>Column 1</th><th>Column 2</th><th>Column 3</th></tr>
  <tr><td colspan="2">1,1-1,2</td> <td rowspan="3">1,3-3,3</td></tr>
  <tr><td>2,1</td><td>2,2</td></tr>
  <tr><td>3,1</td><td>3,2</td></tr>
</table>
```

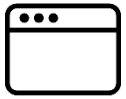


Column 1	Column 2	Column 3
1,1-1,2		
2,1	2,2	1,3-3,3
3,1	3,2	

Column styles: `<col>, <colgroup>`

- **`col`** tag can be used to define styles that apply to an entire column (self-closing)
- **`colgroup`** tag applies a style to a group of columns (NOT self-closing)


```
<table>
    <colgroup>
        <col style="background-color:green" />
        <col span="2" style="background-color:blue" />
    </colgroup>
    <tr><th>Column 1</th><th>Column 2</th><th>Column 3</th></tr>
    <tr><td>1,1</td><td>1,2</td><td>1,3</td></tr>
    <tr><td>2,1</td><td>2,2</td><td>2,3</td></tr>
</table>
```



Column 1	Column 2	Column 3
1,1	1,2	1,3
2,1	2,2	2,3

Don't use tables for layout!

- (borderless) tables appear to be an easy way to achieve grid-like page layouts
- Many "newbie" web pages do this (including many UW CSE web pages...)
- But, a table has semantics; it should be used only to represent an actual table of data
- Instead of tables, use divs, widths/margins, floats, etc. to perform layout

CONNECTING THE PARTS OF KNOWLEDGE WITH THE WHOLENESS OF KNOWLEDGE

The Web Programming Environment

1. HTML is the basis of Web Programming, every web page is made of HTML.
2. To be a proper web programmer you also have to understand the deeper underlying realities of HTTP, TCP, and DNS.

---

3. **Transcendental consciousness** is when our mind is in contact with the deepest underlying reality, the unified field.
4. **Impulses within the Transcendental field:** the infinite dynamism of the unified field constantly expresses itself and becomes all aspects of creation.
5. **Wholeness moving within itself:** In Unity Consciousness, one experiences that this infinite dynamism is nothing but the self.