

# **Lecture 1:**

# **Intro to Web Programming**

Layers of Abstraction

# Wholeness Statement

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.

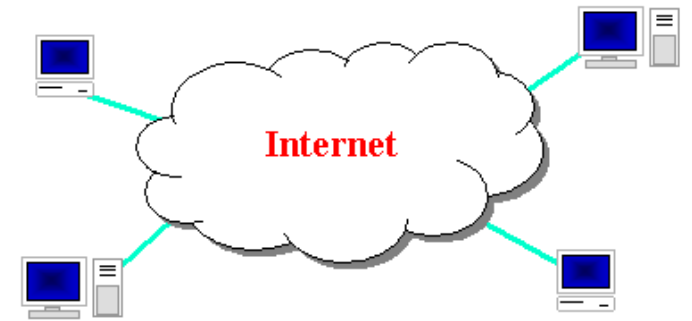
*Life is found in layers and the TM Technique gives us access to the full range of our awareness and thoughts.*

# 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.

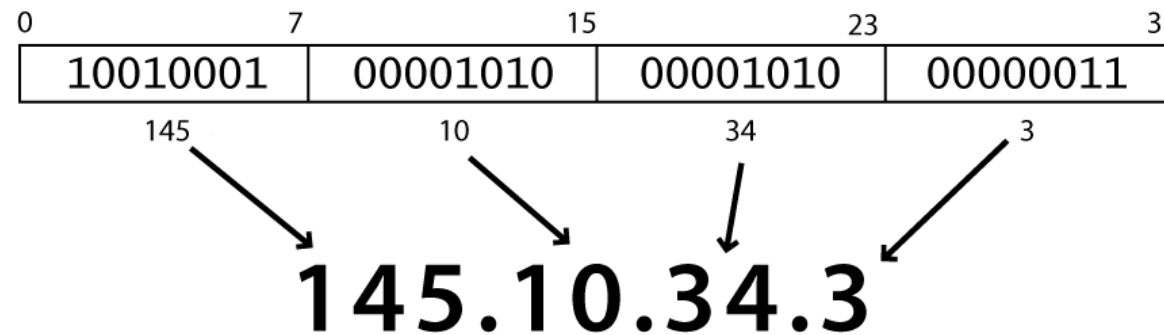


# Brief history

- 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 started in 1995; Google January 1996

# 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**: get resources from the server
  - **POST**: post data to the server for processing (create/update)
  - **PUT** : create/update
  - **DELETE**: delete

# HTTP Request and Response messages

## 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

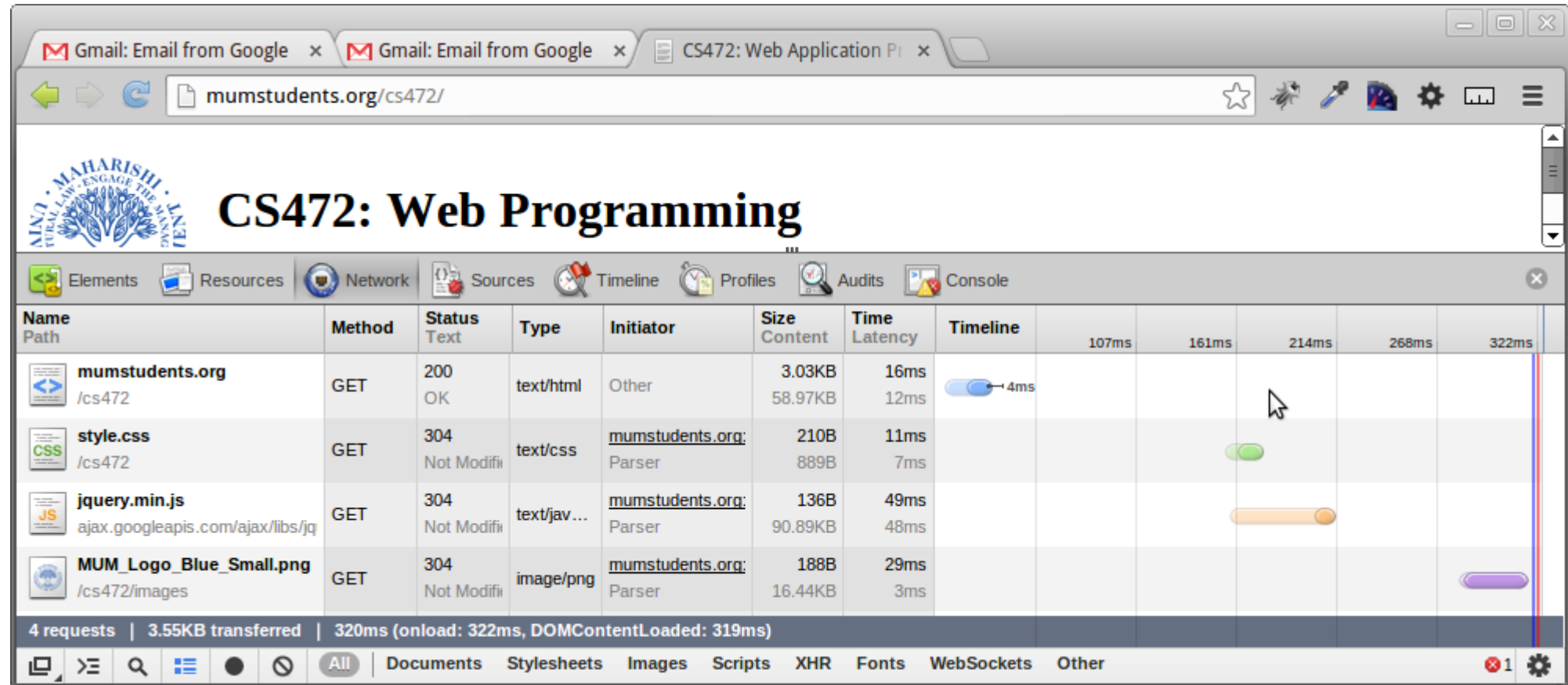
<!DOCTYPE html>

<html>

...



# Request and Response



# Web servers and Browsers

- **Web server:** software that listens for web page requests
  - [Apache](#)
  - Microsoft Internet Information Server (IIS)
  - nginx
  - GWS (Google Web Server)
- **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`

- Type this ip in your browser's url bar.

- 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](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 status 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](#)

# MIME Types (Multi-Purpose Internet Mail Extension)

- A two-part identifier which many web protocols use to categorize each type of data.      broad type/subtype

| MIME Type                        | File extension(s) | Description           |
|----------------------------------|-------------------|-----------------------|
| application/octet-stream         | .exe              | Executable programs   |
| audio/mpeg                       | .mp3, .mpg        | MPEG or MP3 music     |
| image/gif, image/jpeg, image/png | .gif, .jpg, .png  | GIF, JPEG, PNG images |
| text/css                         | .css              | Style sheets          |
| text/html                        | .html, .htm, .php | Web pages             |
| <a href="#">MIME Type list</a>   |                   |                       |

# 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 IP addresses 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 many layers of the internet. *Familiarity with the deep layers is necessary for optimal understanding and use of the surface layers. TM settles our mind so that it is more connected to the deep layers that are the basis of the more expressed layers.*

# 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)

# 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 lang="en">
  <head>
    <meta charset="utf-8" />
    <title></title>
  </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" />
<meta name="viewport" content="width=device-width, initial-scale=1.0"/>
```

- Placed in the **head** section of your HTML page
- **meta** tags often have both the **name** and **content** attributes
- HTML5 introduced a method to give web designers control over the viewport (the user's visible area of a web page)

# 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



# Relative vs Absolute URL

- **Relative URL**

- index.html (path relative)
- graphics/image.png
- ../about.html (directory relative)
- ../../stories/stories.html
- /image.png (root relative)

- **Absolute URL**

- http://www.mysite.com
- C:\website\images\image.png

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

# Common Error on Relative URL

- Placing a leading “/” on a relative URL

<http://www.myexample.com/mysite/files/index.html>

<img src=“images/smiley.gif” alt=“smiley face”/>

<http://www.myexample.com/mysite/files/images/smiley.gif>

<img src=“/images/smiley.gif” alt=“smiley face”/>

<http://www.myexample.com/images/smiley.gif>



# Block-level Elements

- A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).
- Examples of block-level elements:
  - `<div>`
  - `<h1>` - `<h6>`
  - `<p>`
  - `<form>`



# 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

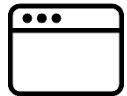
[More paragraph examples](#)

# 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](#)



# Horizontal Rule: `<hr />`

- A line to separate sections (block)

```
<h1>CS472 Web Programming</h1>
```

```
<p>Very good, fun course - HTML, CSS, JavaScript, JSP, Servlets</p>
```

```
<hr />
```

```
<h1>CS421 Modern Programming Practices</h1>
```

```
<p>Profound course - OOAD, Java8 features</p>
```

# Semantic Tags



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

- section – defines a section in a document
- header – specifies a header for a document or a section
- footer – specifies a footer for a document or a section
- nav – defines a set of navigation links
- aside – like a sidebar
- article – specifies independent, self-contained content
- [More Semantic Tags](#)

# 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. This is a foundational concept

*Well begun is half done. Start with a good foundation and build upon that.*



# Inline Elements

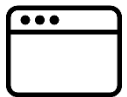
- An inline element does not start on a new line and only takes up as much width as necessary.
- Examples of inline elements:
  - `<span>`
  - `<a>`
  - `<img>`

# 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](http://www.google.com/) or our [Lecture Notes](index.php).

# 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)



```

```



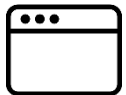


# Line break: `<br />`

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?

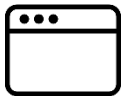
- **Warning:** Don't over-use `<br />` (guideline:  $\geq 2$  in a row is bad)
- `<br />` should not be used to separate paragraphs or used multiple times in a row to create spacing

# Phrase elements : ~~<i>~~<em> , ~~<b>~~<strong>

- 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 effect below in a valid way?)



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



# Displaying List Items

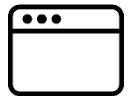
- Unordered List
- Ordered List
- Definition List

# 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>
```



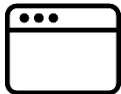
- Simpsons:
  - Homer
  - Marge
- Family Guy:
  - Peter
  - Lois

# 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!
```

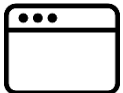


# Definition list: <dl>, <dt>, <dd>

- 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)

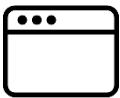


```
<p>As Lincoln said in his famous Gettysburg Address:</p>
```

```
<blockquote>
```

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.

```
</blockquote>
```



As Lincoln said in his famous Gettysburg Address:

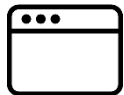
Fourscore and seven years ago, our fathers brought forth on this continent a new nation, conceived in liberty, and dedicated to the proposition that all men are created equal.

# 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 &quot;
- Using <q> allows us to apply CSS styles to quotations (seen later)

# HTML Character Entities

- Reserved characters in HTML must be replaced with character entities.
- Characters that are not present on your keyboard can also be replaced by entities.
- [Complete list of HTML entities](#)

| character(s) | entity                     |
|--------------|----------------------------|
| < >          | &lt; &gt;                  |
| é è ñ        | &eacute; &egrave; &ntilde; |
| ™ ©          | &trade; &copy;             |
| π δ Δ        | &pi; &delta; &Delta;       |
| ∕            | &#1048;                    |
| " &          | &quot; &amp;               |

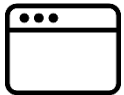
- How would you display the text & on a web page?

# HTML-encoding text

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



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



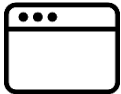
```
<p>  
  <a href="http://google.com/"> Google </a>  
</p>
```

# Computer code: `<code>`

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



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



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

# Preformatted text: `<pre>`

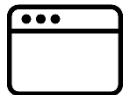
A large section of pre-formatted text (block)



```
<pre>
```

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

```
</pre>
```



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

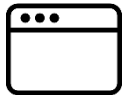
- Displayed with exactly the whitespace / line breaks given in the text
- Shown in a fixed-width font by default
- How would it look if we had instead enclosed it in code tags?

# 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



# Displaying tabular data

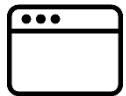
- HTML Table

# 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>
```



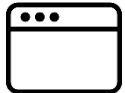
|               |          |
|---------------|----------|
| 1,1           | 1,2 okay |
| 2,1 real wide | 2,2      |

- **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      |

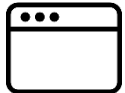
- **th** cells in a row are considered headers; by default, they appear bold
- a **caption** at the start of the table labels its meaning

# 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  |          | 1,3-3,3  |
| 2,1      | 2,2      |          |
| 3,1      | 3,2      |          |

# Don't use tables for layout!

- (borderless) tables appear to be an easy way to achieve grid-like page layouts, specially for form fields.
- 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

# 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



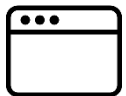
```
<!-- My web page, by Suzy Student CSE 190 D, Spring 2048 -->  
<p>CSE courses are <!-- NOT --> a lot of fun!</p>
```

# W3C HTML Validator

- Checks your HTML code to make sure it follows the official HTML syntax
- Pickier than the browser



```
<p>  
  <a href="https://validator.w3.org/check/referer">  
      
  </a>  
</p>
```



# Main Point

More common tags include inline elements, lists, tables. The most important concept 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 better understand the meaning of your page and use it more effectively.

*Clear and settled awareness allows us to better grasp the meaning of information and take the right actions for success.*



# Web languages / technologies

- Hypertext Markup Language ([HTML](#)): used for writing web pages
- Cascading Style Sheets ([CSS](#)): stylistic info for web pages
- [JavaScript](#): interactive and programmable web pages
- Asynchronous JavaScript and XML ([Ajax](#)): accessing data for web applications
- eXtensible Markup Language ([XML](#)): metalanguage for organizing data
- JavaScript Object Notation (JSON)
- Servlet/ JSP (Java)

# 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 must understand the deeper underlying realities of HTTP, TCP, IP 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.

