SWINBURNE UNIVERSITY OF TECHNOLOGY

COS10026
Computing Technology Inquiry Project

Lecture 1

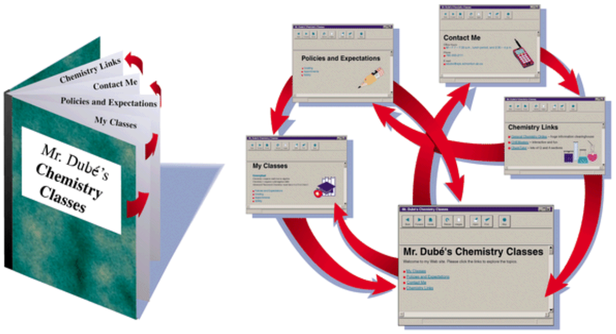
HTML1 - Introduction

Topics

- The Web
- Clients and Servers
- Web Documents
- HTML Document Structure

The Web – Its History

- In late 1990 and early 1991,Tim Berners-Lee created the **World Wide Web** at the European Laboratory for Particle Physics (CERN) in Geneva, Switzerland
- The original purpose of the World Wide Web (WWW) was to provide easy access to cross-referenced documents that existed on the CERN computer network
- Hypertext allows you to quickly link to and open other pages/resources.
- HTML for marking up docs as hypertext
- HTTP for transferring HTML docs over the Internet


Chemistry Links
Contact Me
Policies and Expectations
My Classes
Mr. Dubé's Chemistry Classes
Policies and Expectations
Contact Me
Chemistry Links
My Classes
Mr. Dubé's Chemistry Classes

Reading a linear document

Reading a hypertext document

More on this later…

The Web – What is it now?

- The Web has evolved into much more than a set of hyperlinked passive documents read by humans

→ "Web 2.0"

- □ Dynamic, Location aware, Mobile
- □ Big Data, Searchable, Programmable
- □ User generated content

→ "Web 3.0"

- □ Artificial intelligence
- □ Semantic web
- □ Ubiquitous

The Web – Its Consortium

The **World Wide Web Consortium** (**W3C**), is a group of Web developers, programmers, authors, formed in 1994.

- Purpose of the W3C is to lead, create and recommend **standards** that everyone can use to help bring the web “to its full potential”.
- The W3C has no enforcement power, however the recommendations of the W3C are usually followed since a uniform approach is in the best interest of everyone.
- The Web is based on the HTTP internet application protocol
- The standards they recommend cover many web areas and include: HTML, CSS, XML, RDF, SVG, SMIL, PNG and more!

See http://www.w3.org *(Note: there is no “c” like w3c)*

Topics

- The Web
- Clients and Servers
- Web Documents
- HTML Document Structure

Clients and Servers

- **Web Clients / Browsers** (or user agents)
  - Manage and make HTTP requests
  - Receive HTTP responses
  - Interpret and render/display completed Web Documents
- **Web Servers (HTTP Servers)**
  - Receive HTTP requests
  - Retrieve Web Documents
  - Manage and make HTTP responses

We use
Apache HTTP server
running on a host called
mercury

Web Browsers

- Web Browser software is available for most platforms.
- The appearance of a Web page may differ between browsers.
- Commonly used Web Browsers:

Google **Chrome**

Apple **Safari**

Mozilla **Firefox**

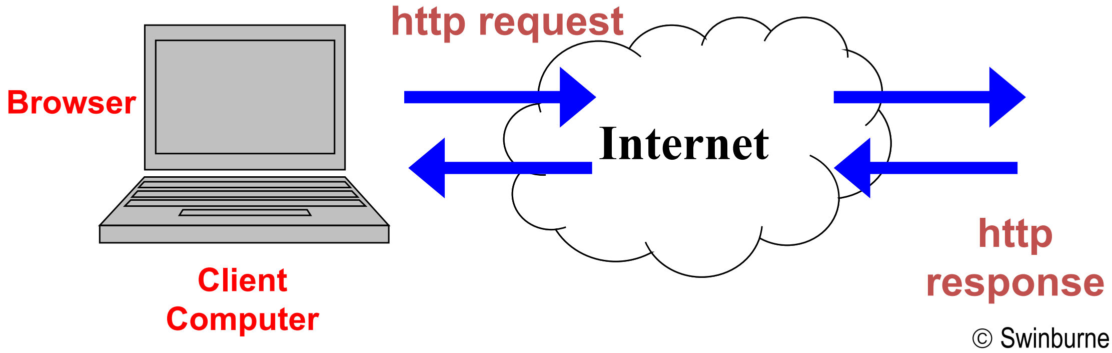
Opera

… and many others …
http://en.wikipedia.org/wiki/List_of_web_browsers

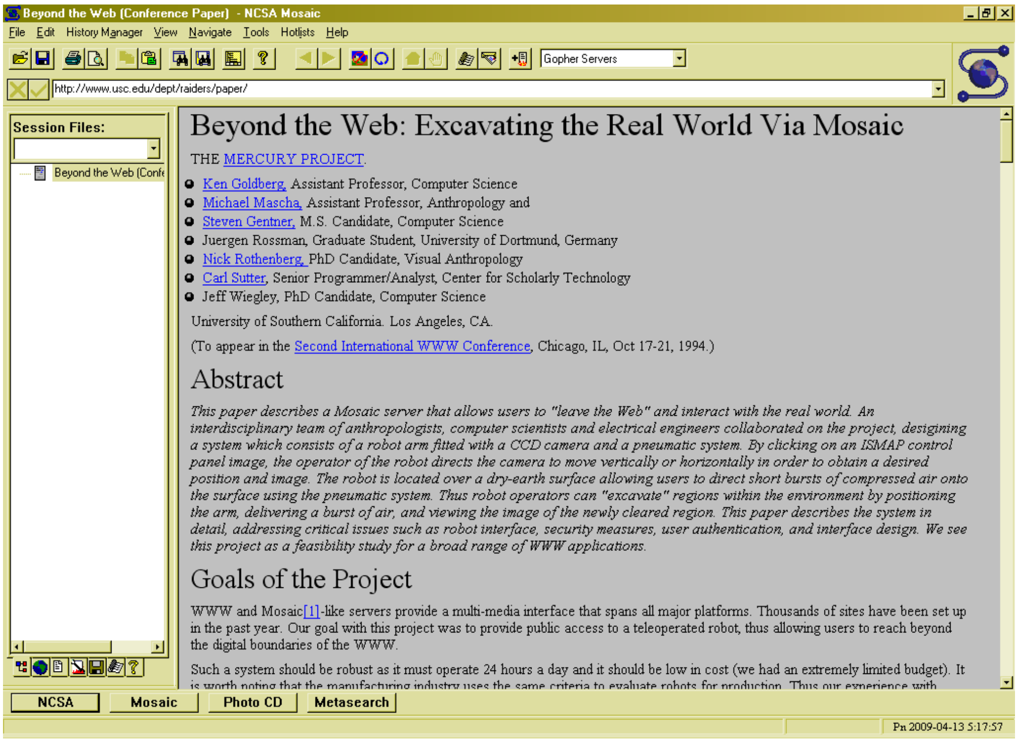
Comparison of Web Browsers : See
http://en.wikipedia.org/wiki/Comparison_of_web_browsers

Web Browsers

- Web browser:
  - □ **Software application** that **lays out** or **renders mark-up**: displaying text, images, and other information typically located in a Web page.
  - □ **Users interact** with the Web browser, *requesting web pages by URL*, clicking on hyperlinks or *submitting forms* within the Web page.
  - □ Web pages are usually located on a **Web Server** on the Internet, but can be located on the local computer, or on a local area network.
  - □ Web browsers format and send HTTP requests, and receive, analyse and layout or render HTTP responses.



Web Browsers - 1994



http://www.usc.edu/dept/raiders/paper/

Beyond the Web: Excavating the Real World Via Mosaic

THE MERCURY PROJECT.

- Ken Goldberg, Assistant Professor, Computer Science
- Michael Mascha, Assistant Professor, Anthropology and
- Steven Gentner, M.S. Candidate, Computer Science
- Juergen Rossman, Graduate Student, University of Dortmund, Germany
- Nick Rothenberg, PhD Candidate, Visual Anthropology
- Carl Sutter, Senior Programmer/Analyst, Center for Scholarly Technology
- Jeff Wiegley, PhD Candidate, Computer Science

University of Southern California. Los Angeles, CA.

(To appear in the Second International WWW Conference, Chicago, IL, Oct 17-21, 1994.)

Abstract

This paper describes a Mosaic server that allows users to "leave the Web" and interact with the real world. An interdisciplinary team of anthropologists, computer scientists and electrical engineers collaborated on the project, desigining a system which consists of a robot arm fitted with a CCD camera and a pneumatic system. By clicking on an ISMAP control panel image, the operator of the robot directs the camera to move vertically or horizontally in order to obtain a desired position and image. The robot is located over a dry-earth surface allowing users to direct short bursts of compressed air onto the surface using the pneumatic system. Thus robot operators can "excavate" regions within the environment by positioning the arm, delivering a burst of air, and viewing the image of the newly cleared region. This paper describes the system in detail, addressing critical issues such as robot interface, security measures, user authentication, and interface design. We see this project as a feasibility study for a broad range of WWW applications.

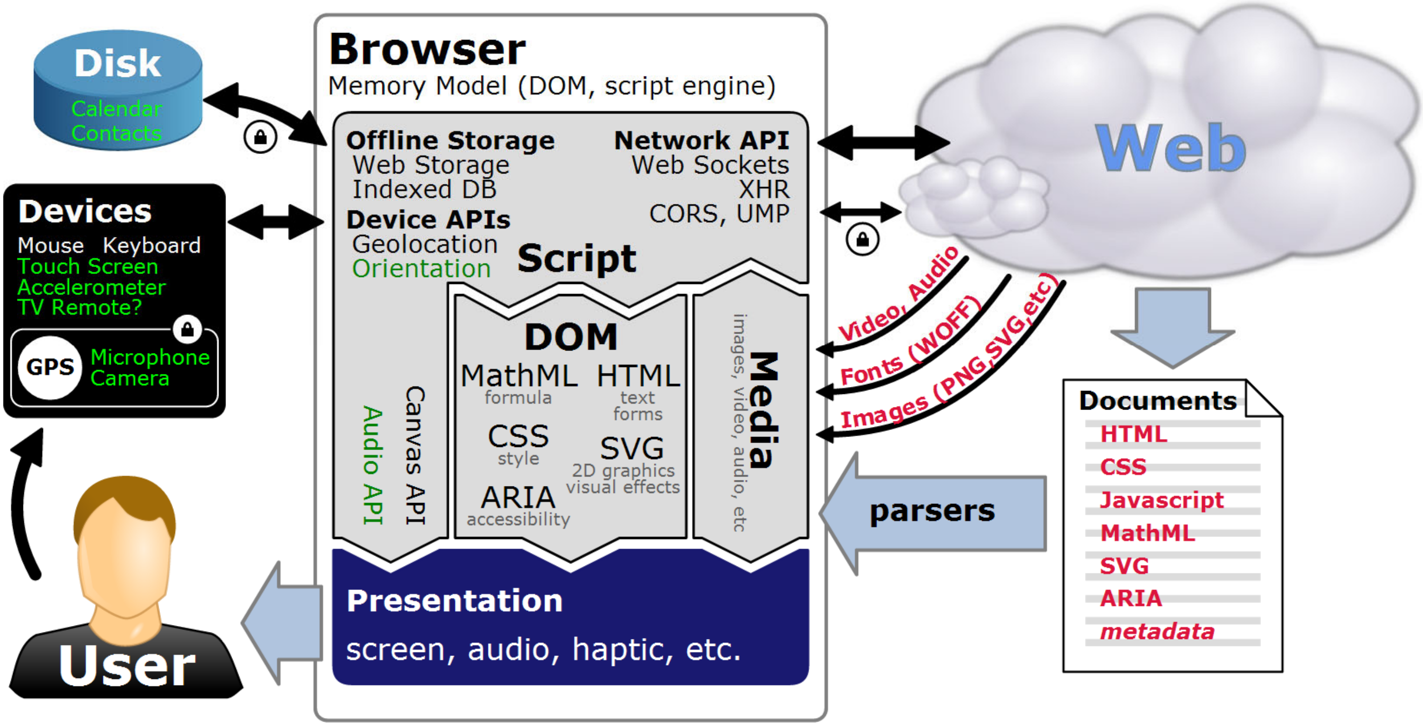
Goals of the Project

WWW and Mosaic[1]-like servers provide a multi-media interface that spans all major platforms. Thousands of sites have been set up in the past year. Our goal with this project was to provide public access to a teleoperated robot, thus allowing users to reach beyond the digital boundaries of the WWW.

Such a system should be robust as it must operate 24 hours a day and it should be low in cost (we had an extremely limited budget). It is worth noting that the manufacturing industry uses the same criteria to evaluate robots for production. Thus our experience with

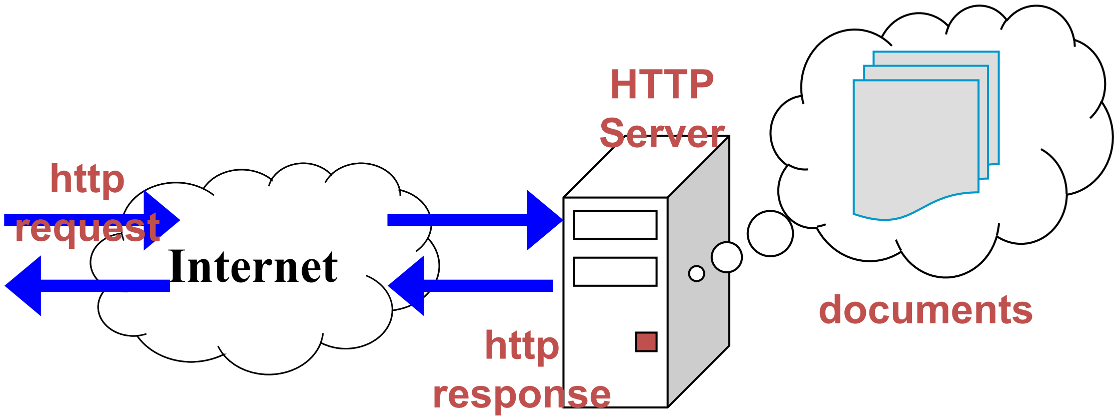
- https://en.wikipedia.org/wiki/Mosaic_(web_browser)

Web Browser Technology – Now



Web Server Features

- A Web server is made up of several components:
  - □ A **computer** with an **Internet connection** and **operating system**.
    - □ The server program usually ***runs continuously***.
  - □ **Web server software** to receive and respond to HTTP requests.
    - □ Handles ***multiple requests***
  - □ **Information**: a collection of documents to be served.
    - □ Careful ***access control*** to server content should be a feature



Web Servers & Scripting

- Servers can support a ***variety of executable scripts*** so that if a particular URL is requested, the server executes the script and then returns its output to the browser.
- Examples of this concept:
  - □ Built-in interpreters for ***embedded scripting*** – ASP, PHP, Perl, etc
  - □ Standard CGI scripts
  - □ Server-side includes (SSI)
  - □ Database interfaces
  - □ Integrated development environments (IDE)

... More about SSI, embedded scripting, later ...

Topics

- The Web
- Clients and Servers
- Web Documents
- HTML Document Structure

Web Documents

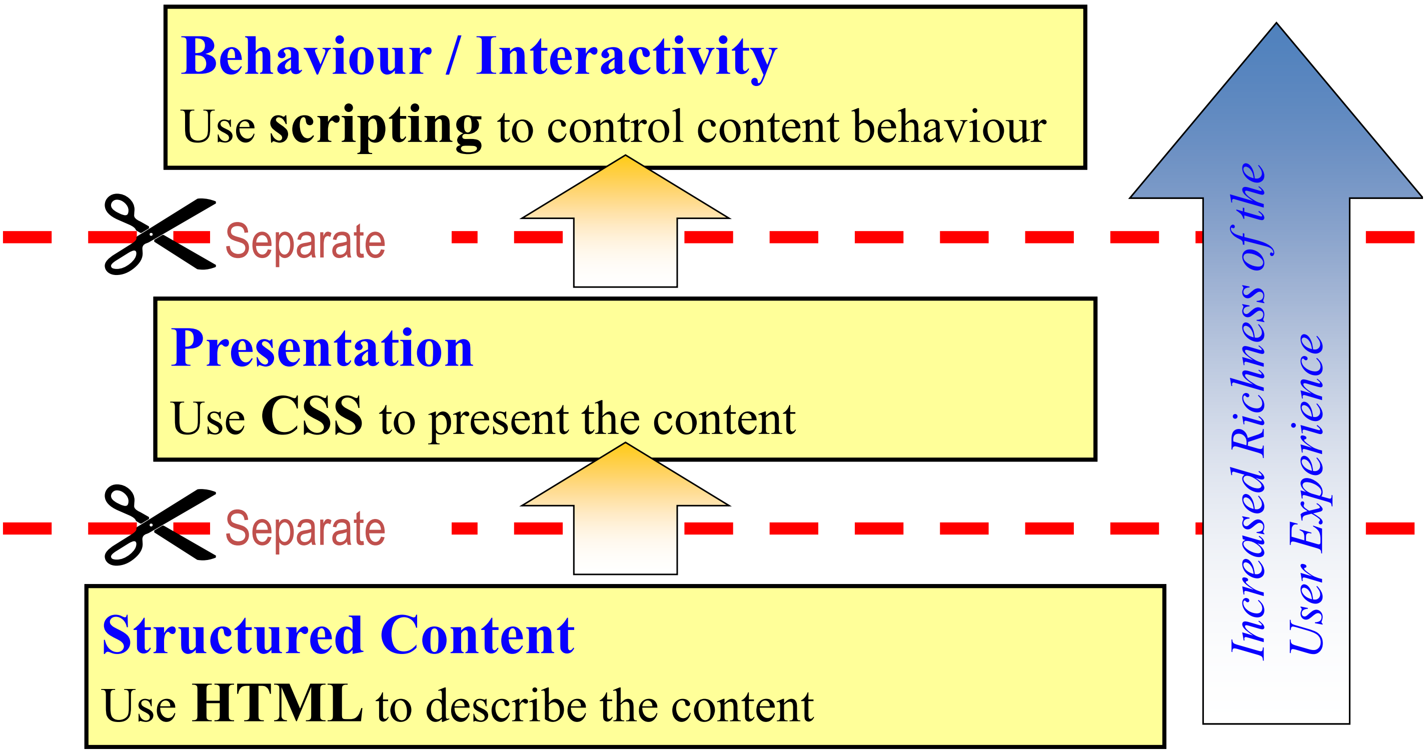
Web Pages (Web documents) are text files with

- ***HyperText Markup Language (HTML)***
  used to mark-up page ***structure*** and ***content***
- ***Cascading Style Sheets (CSS)***
  applied to HTML mark-up page ***presentation***
- ***Images / graphics*** and other ***media***, added to provide ***visual content*** and to ***enrich*** web pages
- ***JavaScript*** (for client-side scripting)
  to ***enhance*** web user ***interaction***.

Example - HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
      <meta charset="utf-8" />
      <meta name="description" content="cat"/>
     <meta name="keywords" content="cat, cute"/>
     <meta name="author"   content="C. DeVille" />
     <title> My Cat </title>
</head>
 <body>
    <h1>
            My Cat
    </h1>
    <img src="cat.png" alt="My Cat" />
</body>
 </html>
```

Build your webpages using the correct tools



Behaviour / Interactivity

Use **scripting** to control content behaviour

Separate

Presentation

Use **CSS** to present the content

Separate

Structured Content

Use **HTML** to describe the content

Increased Richness of the User Experience

Work from the bottom up !

Topics

- The Web
- Clients and Servers
- Web Documents
- HTML Document Structure

HTML

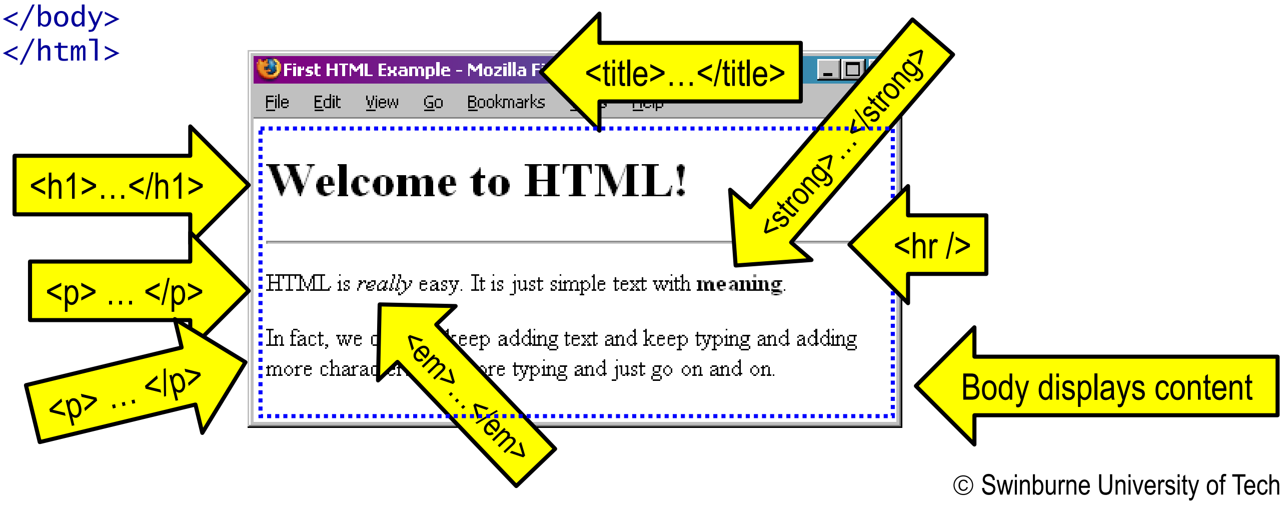
- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- The latest version is HTML 5

First HTML Example

```
<!DOCTYPE html >
<html lang="en" >
<head>
  <meta charset="utf-8" />
  <title>First HTML Example</title>
</head>
<body>
  <h1>Welcome to HTML!</h1>
  <hr />
  <p>HTML is <em>really</em> easy. It is just simple text with
   <strong>meaning</strong>.</p>
  <p>In fact, we can just keep adding text and keep typing and adding
  more characters and more typing and just go on and on.</p>
</body>
</html>
```

doctype standard

Character encoding



Welcome to HTML!

HTML is *really* easy. It is just simple text with **meaning**.

In fact, we c… keep adding text and keep typing and adding more charac… …ore typing and just go on and on.

HTML: Syntax References

http://www.w3schools.com/

HTML Tutorials / References

http://www.w3.org/

The W3C HTML Standards / References

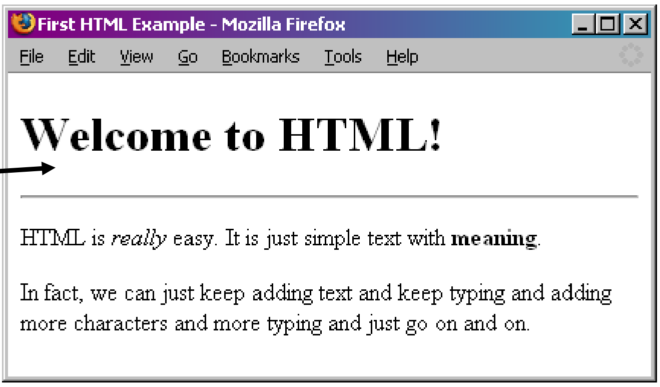
http://reference.sitepoint.com/

HTML Tutorials / References

HTML: Document Structure

- ***The structure of HTML documents is defined by the nesting of HTML elements to form a hierarchy:***

```
<!DOCTYPE html>
<html lang="en">

  <head>
         <meta .....  />

         <title>...</title>

  </head>

  <body>
         ... body content
         goes here ...

  </body>
</html>
```



Welcome to HTML!

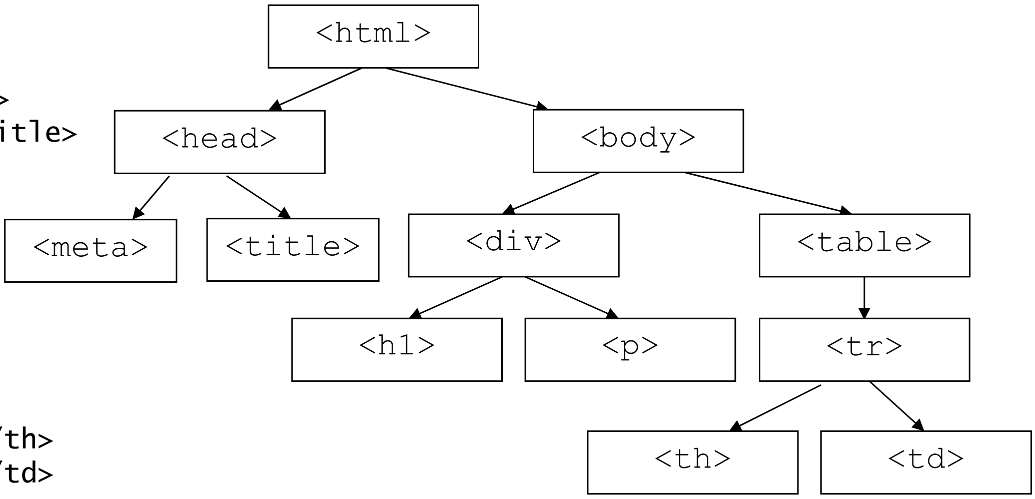
HTML is *really* easy. It is just simple text with **meaning**.

In fact, we can just keep adding text and keep typing and adding more characters and more typing and just go on and on.

HTML: Doc Structure – Tree View

- The “**root**” element of any html document, is the **`html`** element, which usually contains only two children **`head`** and **`body`**
  - □ The **`head`** then contains the **`title`**, and other ‘head’ elements.
  - □ The **`body`** can contain many other elements

```
<html lang=“en”>
<head>
    <meta ....  />
    <title>...</title>
</head>
<body>
  <div>
    <h1>...</h1>
    <p>...</p>
  </div>
  <table>
    <tr>
         <th>...</th>
         <td>...</td>
    </tr>
  </table>
</body>
</html>
```

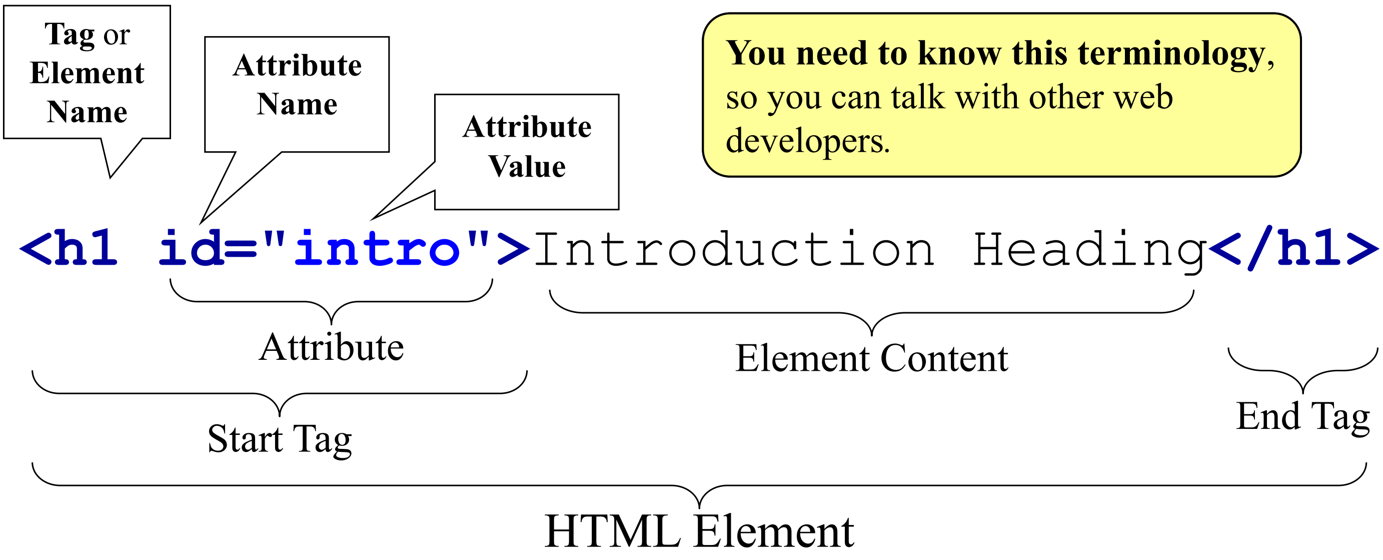


You need to know the basic tree structure of web pages.

HTML Elements

- A HTML **element structure** includes: **start tag**, **tag name**, an attribute name (eg. **id**) with an **attribute value** (eg. **“intro”**), the **element content** (the text affected by the tag meaning), and finally the **end tag** of the element.

You need to know this terminology, so you can talk with other web developers.



```
<h1 id="intro">Introduction Heading</h1>
```

HTML Element Attributes

Start tag can contain **attributes**

attribute *name=“attribute value”*

- **attributes** *modify* the meaning of a tag.

```
<input type="text" … >
<input type="submit" … >
```

Here the attribute “type” provides different “states” for input elements.

- **attribute values *should* be surrounded in a pair of quotes**
  - □ Either double "…" or single '…',
    eg. `id="intro"` or `id='intro'`

HTML Elements – Content and End tags

- **Elements** that can hold *content* begin with a **start tag** and *usually* finish with an **end tag**
- For example:

Element content

```
<h1>This is a major heading</h1>
<p>This is a paragraph</p>
<em>This is text that is emphasised</em>
<strong>This is really important text</strong>
```

- A **tag pair** fully encloses the **element contents**
- **Elements** *might* ***contain*** *other elements*

```
<p>content .. <em> .. Content .. </em> ..content</p>
```

(i.e. elements might be nested)

Some **end tags** are optional in HTML5 when the end tag is implied by the prescience of another tag.

□ e.g. </head> , </li>

However

□ In HTML it is ***good practice*** to: ***'close all tags'***

HTML Elements

Elements are either:

block-level elements or

inline elements.

- **Block-level** elements, like **`<h1>`** headings and **`<p>`** paragraphs, are usually presented by the browser with ***line breaks*** to separate them visually from other elements.
- **Inline** elements, like **`<strong>`** and **`<em>`** occur in the context inside block-level elements, and usually do not introduce any “visual” breaks.

HTML5: Head Template

```
<!DOCTYPE html>

<!-- First HTML5 Example -->
<html lang="en">
<head>
 <meta charset="utf-8" >
  <meta name="description" content="
   [description of what the doc is about] " >
  <meta name="keywords" content="
        [keywords description the document]" >
  <meta name="author"   content="[your name]" >
 <title> [title to show on status bar] </title>
</head>
<body>
   …
</body>
</html>
```

DOCTYPE declaration – not HTML
Must be the first line of the document

HTML root element

Web Page CONTENT

Replace the *[italicized text]* with your code.
Do not forget to validate you code.