Quick review on Web-1

By: Ashiqullah Alizai
Herat University
Computer Science Faculty
0795642400
Alizai.csf@hotmail.com

Applications

- Definition
  - Application is a term for a set of instructions, which makes a computer to perform a task.
  - The set of instructions are commonly known as a program.
  - Many programs are not applications ,A program with a user-interface is an application .
- Different view
  - Web Engineer view: Related configuration files to run the program with suitable documentation
  - End-user view: The program that is useful for him /her and perform specific task for them.

Application System

- A group of related applications programs designed to perform a specific function
- An application system normally consists of a user interface and a database.

Type of Application Softwares

- There are many different types of Application Software but according usage scope we divide them in two category:
  - Desktop Application
  - Web Application

Desktop Applications

- A desktop application means any software that can be installed on a single computer (laptop or a desktop) and used to perform specific tasks.

Web Applications

- A Web application is a system that utilizes **W3C** standards & technologies to deliver Web-specific resources to clients through a browser.
- A web application is any application that uses a web browser as a client.
- The application can be as simple as a HTML page or as complex as a word processor.
  - An online shopping website can be considered as a web application

Categories of Web Applications

- Based on development history and degree of complexity
  - Web applications can be categorized in many ways there is no unique or widely accepted way.
  - Below you can look through different categories of Web applications depending on their development history and their degree of complexity and gives examples
    - Document Centered(Informational ,Download)
    - Interactive
    - Transactional
    - Workflow Based
    - Collaborative work environments
    - Semantic Web

Web crisis

Many organizations are heading toward a Web crisis in which they are unable to keep the system updated and/or grow their system at the rate that is needed. (Dart, 2000)

Engineering

- To successfully build large-scale, complex Web-based systems , Web developers need to adopt a disciplined development process and a sound methodology, use better development tools, and follow a set of good guidelines.
- Engineering approaches are needed to meet it.
  - Engineering is widely taken as a disciplined application of scientific knowledge for the solution of practical problems.

Sulotion

- This new discipline that has been established during the previous years, is therefore seen as :Web Engineering
- So Web engineering was introduced as solution for Web crisis

Web Engineering

- Web Engineering uses:
  - Scientific
  - Engineering
  - Management Principles
- To successfully and cost-efectively Develop Deploy and Maintain high-quality Web systems and applications (Murugesan et al., 1999)
- The outcome of the Web Engineering process are high quality Web applications that provide Web pages that can be displayed in a Web browser.

Goals of Web Engineering

- Solving customers problems
- Minimize risks
- Improve quality
- Maintainability

Stakeholders in Web Engineering

- Major categories of stakeholders are:
  1. Users
  2. Customers
  3. Web developers
  4. Development Managers

Indicators for quality of web applications

- Important Indicators for quality of software product:
  1. Correctness
  2. Maintainability
  3. Efficiency
  4. Usability
  5. Reusability

Web development process

- Before you Start - Remember!
  - Attempting to build a website without a plan is like trying to construct a house without an architectural blueprint

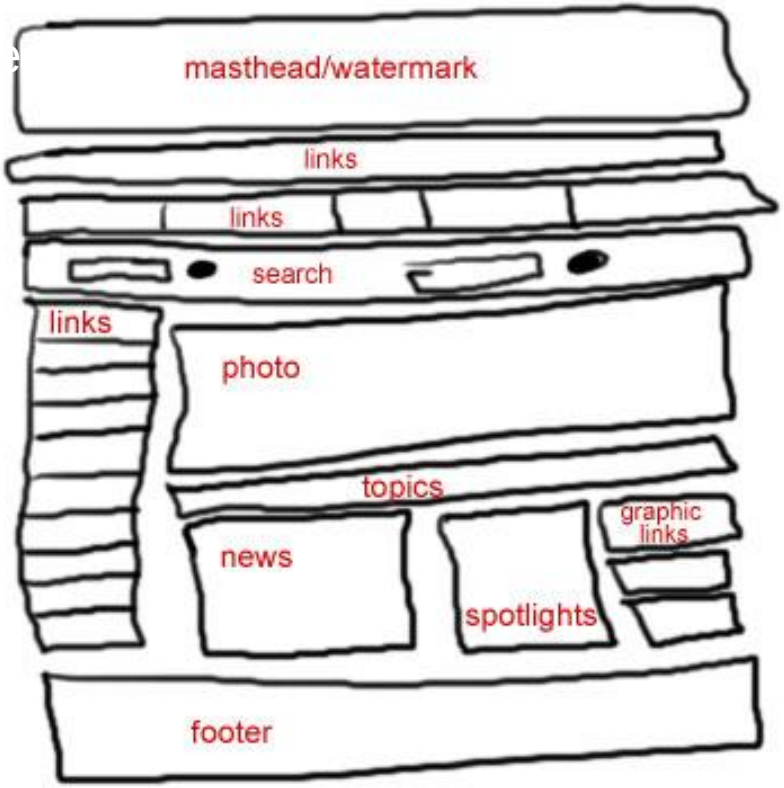
Web development process

- There are numerous steps in the web site design and development process.
  1. Information Gathering
  2. Planning
  3. Design
  4. Development
  5. Testing and Delivery
  6. Maintenance

Phase Two :Planning

- Content Outline:
- Working closely with your clients, create a categories list of all existing content with final feedback.

Phase Three :Design



Wire Framing U.I. homepage

Phase Three :Design

Wireframes:

Using different shapes such as boxes, ovals, and diamonds) to represent content.

Web System Design Challenges

- The Internet is an open platform that provides unparalleled opportunities.
- It has virtually no control over visitor volume, or when and how they access a Web system
- Satisfying the expectations of different types of users with varying skills is not easy.
- Satisfy many different stakeholders besides the diverse range of users, including:
  - The organization that needs the system and fund the system development
  - Persons who maintain the system
- Poor design and infrastructure have caused many Web applications to be unable to support the demands placed on them, so they have therefore failed.

The Internet Versus the Web

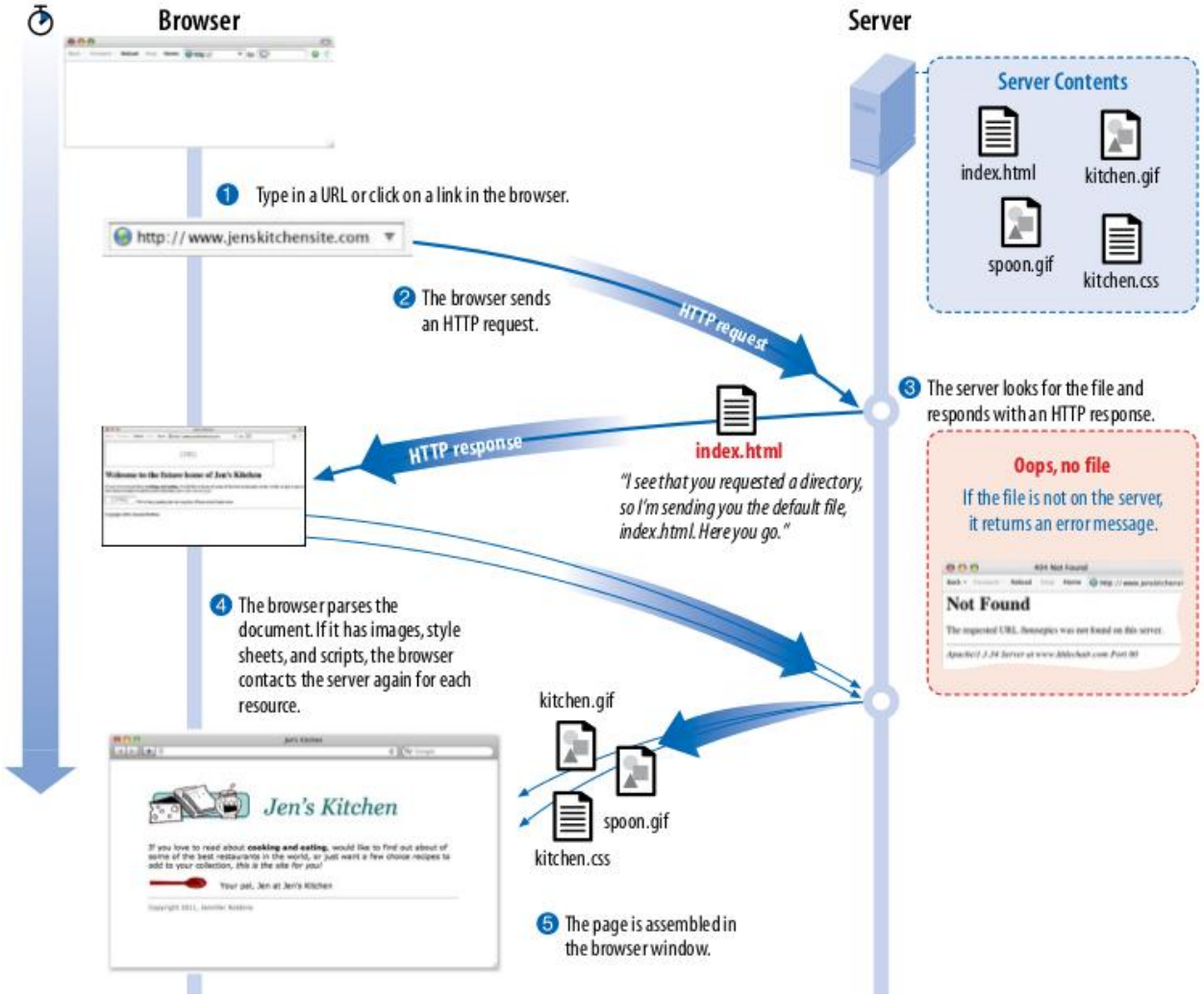
The Internet Versus the Web: What is Internet

- The Internet is network of connected computers that electronically connects millions of people in the world.
- No company owns the Internet, it is a cooperative effort governed by a system of standards and rules.
- The purpose of connecting computers together, of course, is to communicate and share information

The Internet Versus the Web
What is Web?

- The Web (originally called the World Wide Web, thus the www in site addresses) is part of the Internet which itself is "a network of interconnected computers" to transfer data (for example, emails, web documents etc.) between computers It is just one of the ways information can be shared over the Internet.
- The World Wide Web was created in 1989 by Sir Tim Berners-Lee
- The Web is a subset of the Internet. It is just one of many ways information can be transferred over networked computers.

How Browsers Display Webpage



HTML

- A language for **describing web pages** Developed by World Wide Web Consortium(W3C)headed by Tim Lee
- HTML is not a programming language, it is a **Markup Language.**
- A markup language is a set of **markup tags**.
- The markup tags describe **how text should be displayed**
- HTML is **not case sensitive** language.
- HTML documents contain **HTML tags and plain text**.
- Must be saved as text files with extension **html** or **htm**.
- The extension tells the browser that file is to be interpreted according to HTML standards

HTML Elements(tag)

- There are two different types of tags:
  - **Paired tags or Container Element or two-sided tag :**
    - Container Tags contains start tag & end tag
    - Example: <HTML>… </HTML>
  - **Unpaired tags or Empty Element or one-sided tags:**
    - HTML elements with **no content**
    - Empty Tags contains start tag
    - Example :
 or <IMG>

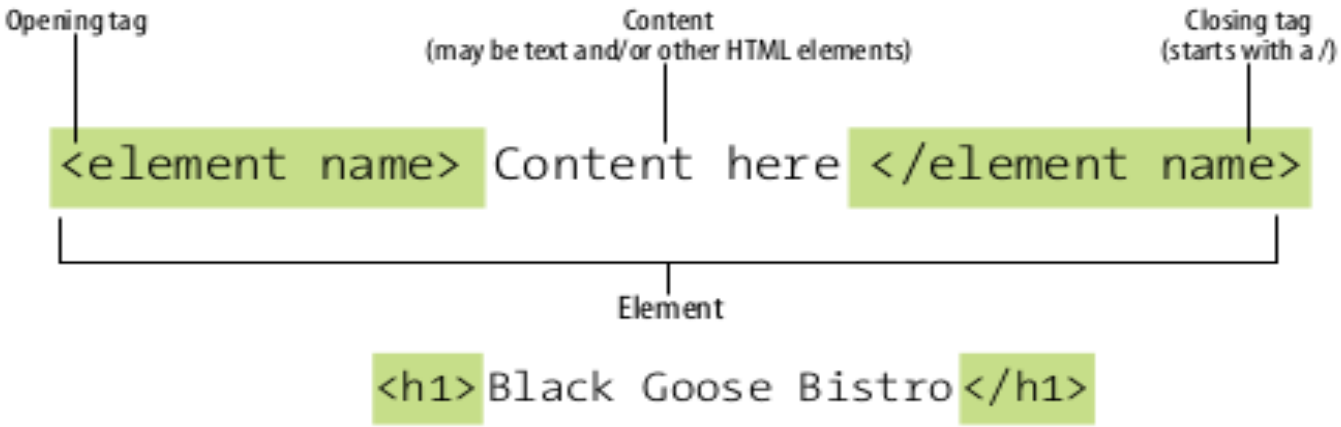
<element-name>

Example: The br element inserts a line break.

```
<p>1005 Gravenstein Highway North<br>Sebastopol, CA 95472</p>
```

HTML Elements

- Generally elements consists of both the :
  - **Content**:everything between the start and the end
  - **Its Markup** :
    - Start tag/opening tag
    - End tag/closing tag



HTML Elements

- In general, HTML elements can be divided into two categories :

- **Block level** elements
  - **Contain other block** level as well as inline elements.
  - By default, block-level elements **begin on new lines**.
  - block level elements create larger Like:p , h1, ol , ul , table, etc
- **Inline** elements
  - It can **contain data and other inline** elements.
  - By default, inline elements do **not begin on new lines**.
  - Inline elements create **shorter structures** (than block level elements Like: b, big, i, small, etc

Document structure elements

- **HTML Element:**
  - The **Root element** of an HTML document; all other elements are contained in this. Structure :<html>...</html>
- **Head Element:**
  - **Container for processing information** and metadata for an HTML document. <head>...</head>
  - **Title Element** - The title of the page <title>...<title>
- **Body Element:**
  - **Container for the displayable** content of an HTML document.
  - Structure : <body>...</body>

Header elements:

- **<head>** - Opening tag for the head of the document.
- **<title>...</title>** -Document title (not part of the text)
- **<link ...>** - Relationships for the document as a whole
- **<base href="url">** - Specifies the base URL of the document.
- **<meta ...>** - Embed meta-information as if given by the server
- **<style type="text/css" href="URL" />** - Specifies a CSS file to be used for the web page.
- **<script type="text/scripttype" href="URL" />** - Specifies a Javascript of VBscript file to be used for the web page.

Body Elements:

- **<body>...</body>** - Encloses the main body of the document.
- **<hn>...</hn>** - Makes the text a heading of various sizes
- **<basefont size="n">** - Sets the default font properties
- **<img src="URL" attributes>** - Places an inline image
- **<map attributes>...</map>** - Specifies a collection of hot spots that define a client-side image map.
- **<area attributes>...</area>** - Specifies the shape and size of a hot spot to be used in the definition of a client-side image map.
- **<marquee attributes>...</marquee>** - Places a scrolling text marquee into the document.
- **<applet attributes>...</applet>** - Inserts a Java applet in the HTML document.

Body Elements:

- **<embed attributes>...</embed>** - Inserts an embedded multimedia object, such as a sound file or video, into the page.
- **...** - When used with the HREF attribute, the enclosed text and/or graphic becomes a link to another document or anchor. When used with the NAME attribute, the enclosed text and/or graphic becomes an anchor.
- **<ol attributes>...
** - Puts the enclosed items marked with <li>, in a numbered list.
- **<ul attributes>...
** - Puts the enclosed items marked with <li>, in a bulleted list.
- **<dl>...</dl>** - Creates a definition list. Within this container, <dt> specifies a definition term and <dd> specifies the definition.

Table Elements:

- **<table attributes>…</table>** -Creates a table that can include any number of rows.
- **<caption attributes>…</caption>** -Specifies the caption.
- **<tr attributes>…</tr>** - Specifies a table row. It can enclose the table heading and table data.
- **<th attributes>…</th>** - Specifies a table heading.
- **<td attributes>…</td>** - Specifies a table data cell.
- **<tbody>…</tbody>** - Encloses the body of your table.
- **<tfoot>…</tfoot>** - Encloses the table rows that are to be used as a footer
- **<thead>…</thead>** - Encloses the table rows that are to be used as a header.

Text Formatting Elements:

- **<address>.....< /address>** - Encloses the signature file of the author of the page. Text is displayed in italics.
- **<acronym>.....< /acronym>** - indicates an acronym in the text.
- **<b>...< /b>** - Boldfaces the enclosed text.
- **<big>...< /big>** - Makes the enclosed text one size larger.
- **<blink>.....< /blink>** - Makes the enclosed text blink continually.
- **<blockquote>.....< /blockquote>** - Encloses a long quote. Both the left and right margins are indented.
- **
** - Inserts a line break.
- ** .....< /center>** - Centers the enclosed elements.
- **<cite>.....< /cite>** - Encloses a citation such as the title of a book or paper.

Text Formatting Elements:

- **<code>…..< /code>** - Encloses a sample of code.
- **<comment>…..< /comment>** - Encloses a comment.
- **<del>…..< /del>** - To mark the document text that has been deleted since a previous version.
- **<dfn>…..< /dfn>** - Encloses a definition.
- ** ..< /div>** - Can be used to divide a document into sections.
- **…< /em>** - Emphasis on the enclosed text (Italics).
- **<font attributes>…< /font>** - Sets the font properties.
- **<fieldset attributes>…< /fieldset>** - Allows you can group related form fields, making your form easier to read and use.
- **<hr attributes />** - Inserts a horizontal line.
- **<i>…< /i>** - The enclosed text is italics.

Text Formatting Elements:

- **<label>…< /label>** - Allows you to lable a tag.
- **<kbd>…< /kbd>** - Specifies text to be entered at the keyboard.
- **<p attributes>…< /p>**- Designates the enclosed text as a plain.
- **<q>…</q>** - acts much the same as the <blockquote> tag.
- **<pre>…..< /pre>** - Displays text in fixed-width type.
- **<s>…..< /s>** - Displays text with a line through it.
- **<small>…< /small>** - Makes the enclosed text one size smaller.
- **<spacer attributes>…< /spacer>** - Inserts blocks of spaces.
- **…< /strong>** - Stronger emphasis on the text.
- **<sub>…< /sub>** - Renders the enclosed text in subscript.
- **<sup>…< /sup>** - Renders the enclosed text in superscript.
- **<tt>…< /tt>** - The enclosed text is typewriter font.
- **<u>…< /u>** - The enclosed text in underlined.

Some Special Characters(**Entities**)

Character	Description	Name	Number
	Character space (nonbreaking space)		
&	Ampersand	&amp;	&#038;
'	Apostrophe	&apos;	&#039;
<	Less-than symbol (useful for displaying markup on a web page)	&lt;	&#060;
>	Greater-than symbol (useful for displaying markup on a web page)	&gt;	&#062;
©	Copyright	&copy;	&#169;
®	Registered trademark	&reg;	&#174;
™	Trademark	&trade;	&#8482;
£	Pound	&pound;	&#163;
¥	Yen	&yen;	&#165;
€	Euro	&euro;	&#8364;
–	En-dash	&ndash;	&#8211;
—	Em-dash	&mdash;	&#8212;
'	Left curly single quote	&lsquo;	&#8216;
'	Right curly single quote	&rsquo;	&#8217;
"	Left curly double quote	&ldquo;	&#8220;
"	Right curly double quote	&rdquo;	&#8221;
•	Bullet	&bull;	&#8226;
…	Horizontal ellipsis	&hellip;	&#8230;

Some characters are reserved in HTML XHTML. For example, you cannot use the greater than and less than signs or angle brackets within your text because the browser could mistake them for markup.

Marking Up Text(Presentational)

- **HTML Standard Attributes**
  - Define the **characteristics of an element** and is placed inside element's opening tag and made up of two parts: **name and value**
  - **Struture:** **<opentag name="value" > ...</closetag>**
  - There are few HTML attributes which are standard and associated to all the HTML tags.
    - Core Attributes
    - Internationalization Attributes(Language Attributes)
    - Generic Attributes

Marking Up Text(Presentational)

- **HTML Attributes(Core Attributes)**
  - The four core attributes that can be used on the majority of HTML elements (although not all) are:
    - id
    - class
    - title
    - style

Marking Up Text(Presentational)

- **Internationalization Attributes: DIR**
  - There are two internationalization attributes:
    - Dir
    - Lang
  - **Dir:**
    - Allows you to indicate to the browser the **direction** in which the text should flow
    - The dir attribute can take one of two values : ltr and rtl
    - Structure: **<opentag dir=" value" > ...</closetag>**

HTML Links

- **Linking to a specific destination within a single web page.**

  1) **Identify the destination :** To create a destination, **use the id attribute** to give the target element in the document a unique name

     Example: `<h1 id="startH">H</h1>`

  2) **Linking to the destination:** use the a element with the href attribute to provide the location of the link

     Example_1 link to a specific pace : `<p>... F | G | <a href="#startH">H</a> | I | J.. </p>`

     Example_2 link to a specific pace an external html document:

     `<a ref="glossary.html#startH">Glossary, letter H</a>`

     Example_3 Link to an email contact

     `<a href="mailto:alklecker@example.com">Contact Al Klecker</a>`

HTML Links

- **Setting Link Colors:**
  - You can set colors of your links, active links and visited links using link, alink and vlink attributes of **<body> tag**.
  - But it is recommended to use CSS to set colors of links, visited links and active links.
  - **Srtucture:** **<body alink=" " link=" " vlink=" ">**
  - **Example:**

```
<body alink="yellow" link="blue" vlink="green">
<a href="test.html" target="_blank">HTML File</a>
```

HTML File

HTML Images

- **Providing the location with src**
  - The value of the src attribute is the **URL of the image file**
  - **Example**:<img src="icon.gif" alt="">

```
<img src="/images/arrow.gif" alt="">
<img src="http://www.example.com/images/smile.gif"
  alt="">
```

- **Providing alternate text with alt**
  - used to provide a brief description of the image for those who are not able to see it,
  - **Example**: If you're <img src="happyface.gif" alt="happy"> and you know it clap your hands.

HTML Images

- **Providing the location with src**
  - The value of the src attribute is the **URL of the image file**
  - **Example**:<img src="icon.gif" alt="">

```
<img src="/images/arrow.gif" alt="">
<img src="http://www.example.com/images/smile.gif"
  alt="">
```

- **Providing alternate text with alt**
  - used to provide a brief description of the image for those who are not able to see it,
  - **Example**: If you're <img src="happyface.gif" alt="happy"> and you know it clap your hands.

HTML Forms

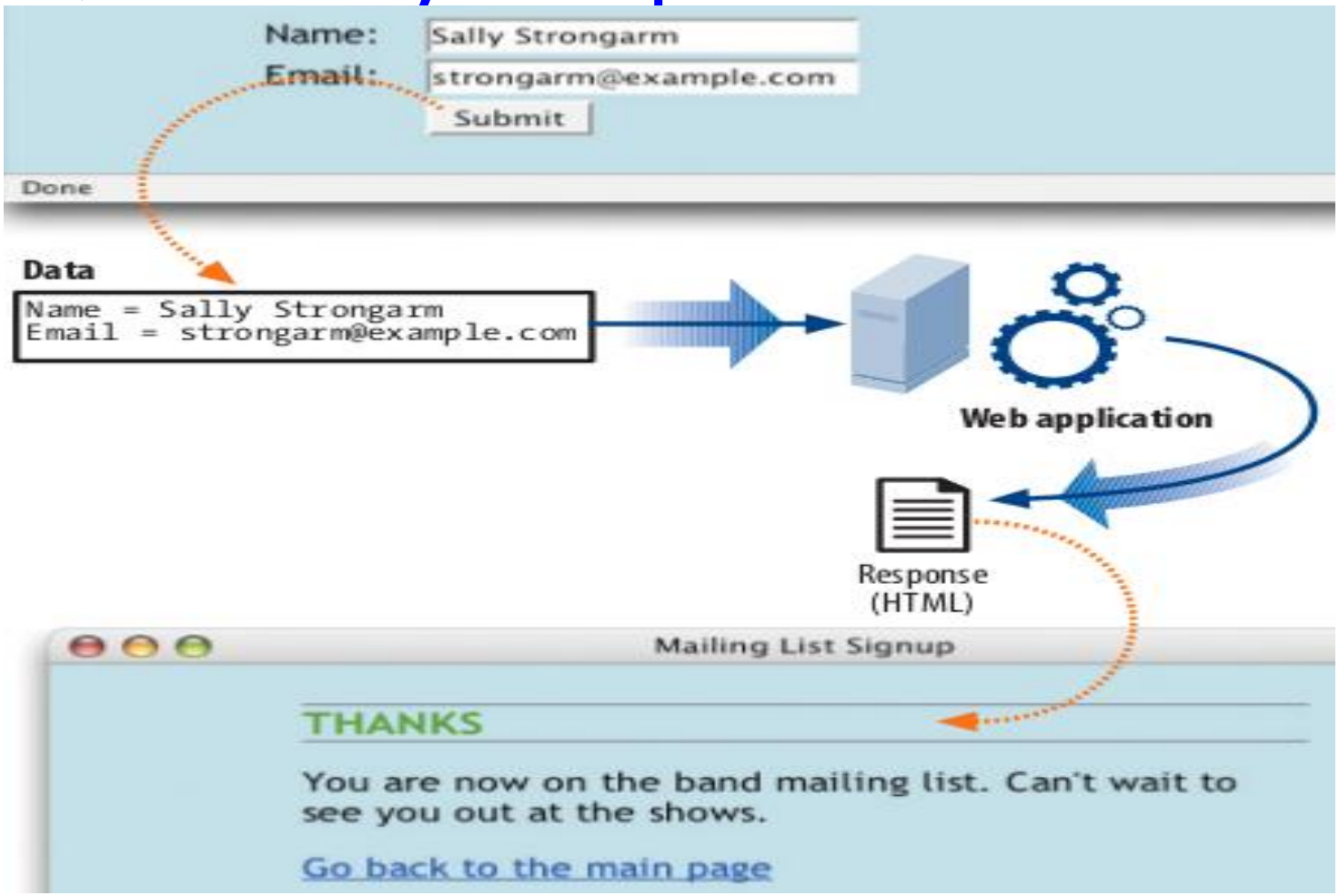
- **HTML Forms**
  - Text entry controls
  - Button entry controls :Submit and reset buttons , Radio and checkbox

HTML Forms

- They Are required when you want to **collect some data from site visitor**.
- For example registration information: name, email address and etc.
- Form elements are like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc. which are used to take information from the user
- **How Forms Work:** There are two parts to a working form
  1. you see on the page itself that is **created using HTML markup**
  2. An application or script on the server that **processes information** collected by the form and returns an appropriate response

HTML Forms

- **From data entry to response**



HTML Forms

- **The form Element**
  - Form element is a **container** for all the content of the form,such as text entry fields,buttons and contain block elements(h1,p and lists, for example),
  - Forms are added to web pages using the form element
  - **Structure:** <form action=" " method=" ">...</form>
  - **Example:**
  - <form action="/mailinglist.php" method="post">...</form>

HTML Forms

The form Element: The action attribute

- The action attribute provides **the location** (URL) of the application or script (sometimes called the **action page**) that will be used to process the form.
  - **Example:** <form action="/mailinglist.php" method="post">...</form>
  - The action attribute in this example sends the data to a script called mailinglist. Php.
  - The .php suffix indicates that this form is processed by a script written in thePHP scripting language,

HTML Forms

- **The Method Attribute:The Get method**
  - With the GET method, the encoded form data gets tacked right **onto the URL** sent to the server
  - A **question mark character** separates the URL from the following data, as shown here :

    get http://www.bandname.com/cgi-bin/mailinglist.pl?name=Sally%20Strongarm&email=strongarm%40example.com
  - The GET method is designed for retrieving information, such as a document, an image, or the results of a database query, from the server

HTML Forms

- **The Method Attribute: The POST method**
  - When the form's method is set to POST, the browser sends a separate server request containing some special headers followed by the data.
  - **Only the server** sees the content of this request, thus it is the best method for sending secure information such as credit card or other personal information.
  - With post method , the data is passed as a separate file.
  - The POST method is meant for posting information, such as a credit card number or information that is to be stored in a database, to the server

HTML Forms

- **Input Element:**
  - The majority of controls are added to a form using the **input element**.
  - The functionality and appearance of the input element changes based on the value of the type attribute in the tag.
  - In HTML5, there are twenty-three different types of input controls
  - **Structure:** <input type=" ." name =" ">
  - **Example:** <input type="text">

HTML Forms

- **Variables and Content:The name attribute**
  - Web forms **use a variety of controls** that allow users to enter information or choose options.
  - The **name attribute provides the variable name** for the control
    - **Example: <input typ="text" name="text1" >**
    - text gathered by a text element is defined as "text1" variable
  - **All form control** elements must include a name attribute
  - Name attribute for submit and reset button elements are **not required**
  - because they have special functions not related to data collection.

Accessing Form Variables

- **Example:**

```
<html>
    <body bgcolor="gray">
    <form action="welcome.php" method="post">
        Name: <input type="text" name="name" />
        Age: <input type="text" name="age" />
        <input type="submit" />
    </form>
</body>
</html>
```

- **Output**

HTML Forms

Accessing Form Variables

- **$_GET, $_POST, $_REQUEST**
  - $_GET[form variable name]
  - $_POST[form variable name],
  - $_REQUEST[form variable name]
- Example:

```
<html>
    <body>
        Welcome <?php echo $_POST["name"]; ?>.<br />
        You are <?php echo $_POST["age"]; ?>years old.
    </body>
</html>
```

HTML Forms

The Great Form Control

- There are different types of form controls:
  - **Text entry controls**
  - **Button entry controls :Submit and reset buttons , Radio and checkbox**
  - Pull-down and scrolling menus
  - File selection and upload control
  - Dates and times (HTML5)
  - Numerical controls (HTML5)
  - Color picker control (HTML5)
  - Boxes : Select boxes , File select boxes
  - Hidden Controls

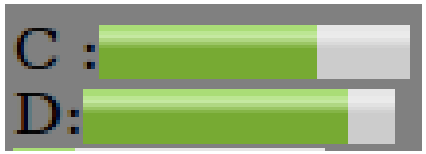
HTML Forms

- **The Great Form Control: Button entry control**
  - There are several different kinds of buttons that can be added to web forms
    - Custom input button
    - Submit and reset buttons
    - Radio and checkbox

- **The Great Form Control:meter**
  - Similar to progress, but it always represents a **measurement within a known range of values** (also known as a gauge)
  - min and max indicate the highest and lowest values for the range
  - **Structure:** <meter min="" max="" name="" >...</meter>
  - **Example:**

```
C:<meter value="70" min="0" max="100"></meter><br/>
D:<meter value="0.85">85%</meter><br>
```

  - **Output**



HTML 5

Contents

- HTML 5
  - A new DOCTYPE
  - Obsolete 4.01 elements
  - New elements and attributes
  - APIs
- Practical Work

- **Obsolete HTML 4.01 Markup**
  - HTML5 also declared a number of elements in HTML 4.01 to be "obsolete" because they are presentational, antiquated, or poorly supported
  - If you use them, browsers will support them but strongly **recommend leaving them** in the dust
  - **Example:**
    - Big,strike,font,dir,frame,frameset,noframe,acronym....

- **Elements and attributes**
  - HTML5 introduced 28 new elements.
    - New **content specific elements**, like article, footer, header,
    - nav, section
    - **New form input types**:color, date, datetime, datetime-local, email, month, number, range, search, tel, time, url, and week
    - The **canvas element** for drawing
    - The **video and audio elements** for media playback
    - **New global attribute**:atribute that can be applied to anyelement

- **Elements and attributes**
  - New **content specific elements**

    Sectionarticle aside nav heade footer figure

  - **HTML4**
  - <body>
  - ...
  - ...
  - ...
  - ...
  - ...
  - </body>

  HTML5

  <body>

  <header>...</header>

  <nav>...</nav>

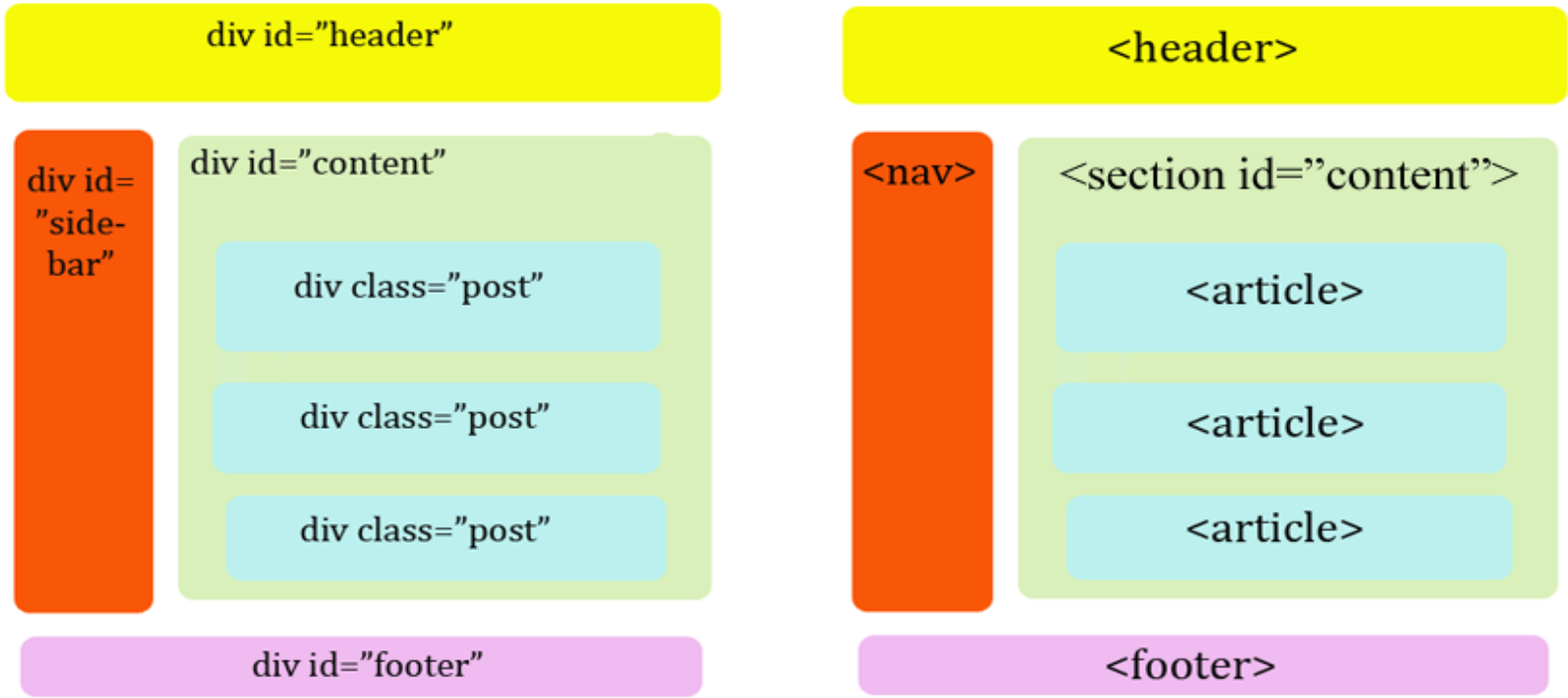
  <section >...</section>

  <aside>...</aside>

  <footer>...</footer>

  </body>

- **Elements and attributes**
  - New **content specific elements**



- **Elements and attributes: Audio & video**
  - HTML5 defines a new element which specifies a standard way to embed an video and audio file on a web page:
  - **Structure:** <video src="" width="" height="" controls autoplay>...</video>
  - **Example:**

```
<video  width="640" height="480" poster="pic.jpg" controls autoplay>
<source src="movie.mp4" type="video/mp4"> browser doesn't support
</video>
```

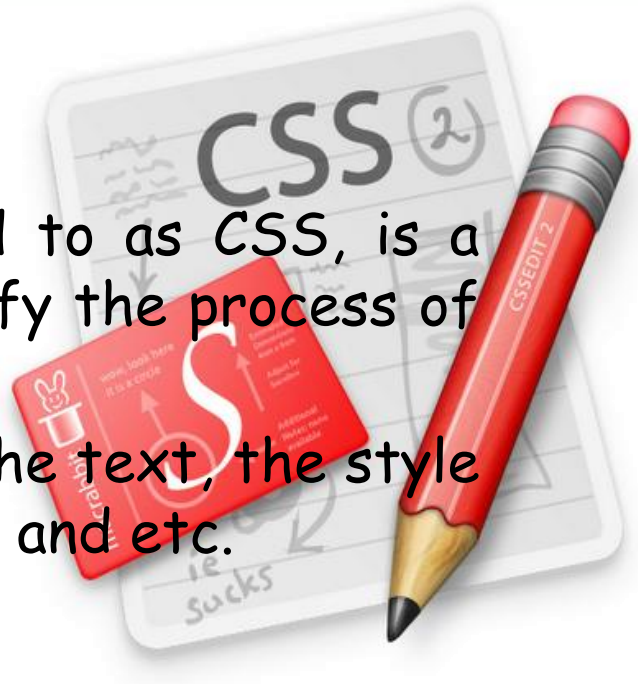
- **Elements and attributes: Audio & video**
  - The audio element uses the same attributes as the video element,with the exception of width, height, and poster
  - Structure: <audio> ….</audio>
  - <audio controls>
  - <source src="soundtrack.mp3" type="audio/mp3">
  - </audio>

- **APIs**
  - HTML5 introduces a number of APIs that **help in creating Web** applications.
  - These can be used together **with the new elements** introduced for applications:
    - **Media elements (video and audio):** have APIs for controlling playback, syncronising multiple media elements and etc
    - **Drag & drop API** : Adds the ability to drag a text selection or file to a target area on the page or another web page.
    - in combination with a draggable attribute

- **APIs**
  - **Editing API** :Provides a set of commands that could be used to create in-browser text editors, allowing users to insert and delete text, format text as bold, italic, or as a hypertext link, and more
  - in combination with a new global contenteditable attribute.
  - **Geolocation API** :Lets users share their geographical location (longitude and latitude) so that it is accessible to scripts in a web application. This allows the app to provide location-aware features such as suggesting a nearby restaurant or finding other users in your area

CSS

What is CSS?



- **C**ascading **S**tyle **S**heets fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.
- Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs and etc.
- CSS separate design from content.

The Benefits of CSS

- Pages load faster
- Easy maintenance
- Superior styles to HTML
- Multiple Device Compatibility
- Global web standard

Writing the rules

- A style sheet is made up of one or more style instructions (called rules or rule sets) that describe how an element or group of elements should be displayed.

  Example: `h1 { color: green; }`

  `p { font-size: small; font-family: sans-serif; }`

- The two main sections of a rule are the selector that identifies the element or elements to be affected, and the declaration that provides the rendering instructions.

Attaching the styles to the document

- External style sheets
- Embedded style sheets

```
<head>
  <title>Required document title here</ title>
  <style>
    /* style rules go here */
  </style>
</head>
```

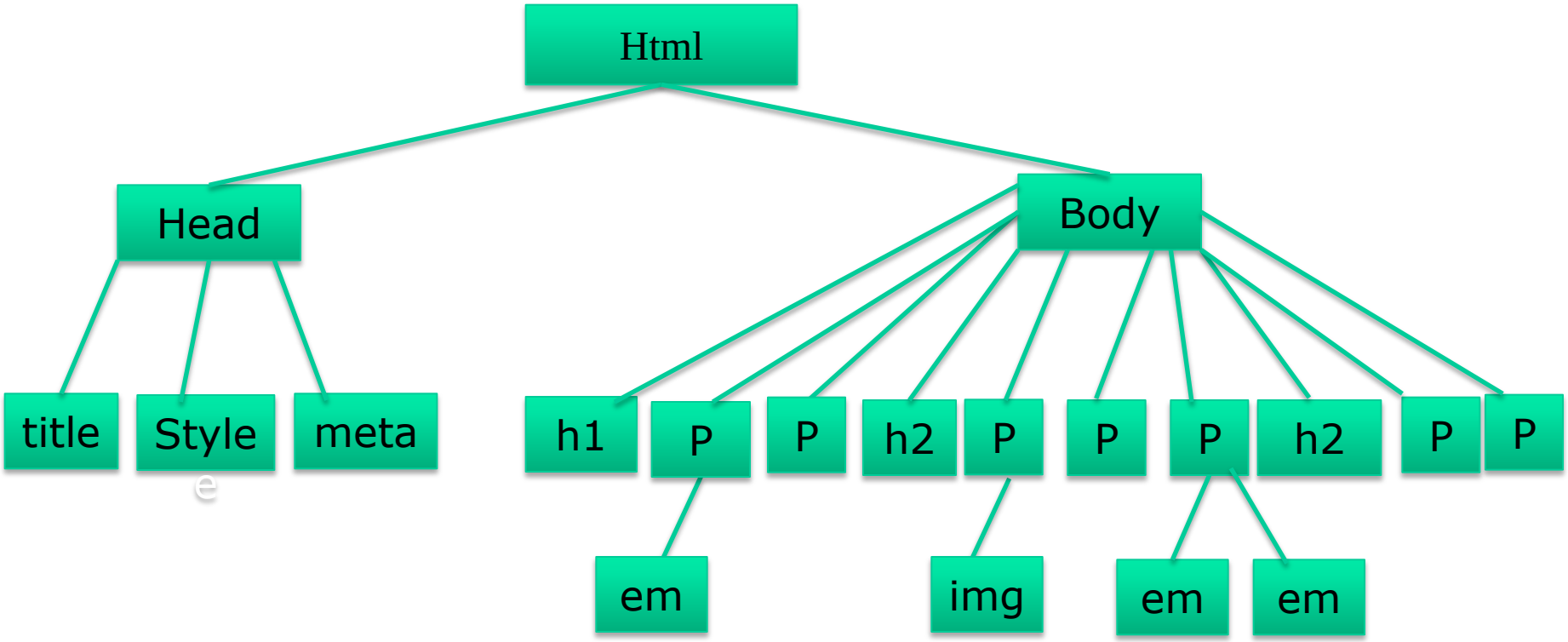
- Inline styles

```
<h1 style="color: red">Introduction</h1>
```

The Big Concepts

- There are a few big ideas that you need to get your head around to be comfortable with how Cascading Style Sheets behave.
- just as parents pass down traits to their children, styled HTML elements pass down certain style properties to the elements they contain.
- when we styled the p elements in a small,sans-serif font, the em element in the second paragraph became small and sans-serif as well, even though we didn't write a rule for it specifically. That is because it inherited the styles from the paragraph it is in.

Inheritance



Group selectors

- If you ever need to apply the same style property to a number of elements, you can group the selectors into one rule by separating them with commas.
- Grouping them makes future edits more efficient and results in a smaller file size.

```
h1, h2, p, em, img { border: 1px solid blue; }
```

The Font Properties

- In CSS, fonts are specified using a little bundle of font- related properties for typeface, size, weight, and font style.

The font-related properties:

font-family
font-size
font weight
font-style
font-variant font

```
body { font-family: Arial; }
tt { font-family: Courier, monospace; }
p { font-family: “Duru Sans”, Verdana, sans-serif; }
```

A Few More selector Types

- Element selector p { color: navy; }
- Grouped selectors p, ul, p, td, th { color: navy; }
- Descendant selectors : indicated in a list separated by a character space.

  li em { color: olive; }
- Child selector : they are indicated with the greater-than symbol (>).

  p > em {font-weight: bold;}
- Adjacent sibling :an adjacent sibling selector is indicated with a plus (+) sign.

  h1 + p {font-style: italic;}

A Few More selector Types

- ID Selector: the id attribute that gives an element a unique identifying name.

  <li id="catalog1234">Happy Face T-shirt

  li#catalog1234 { color: red; }

- Class selectors: used to classify elements into a conceptual group. Unlike the id attribute, multiple elements may share a class name.

  p.special { color: orange; }

- Universal selectors * {color: gray; }

  #intro * { color: gray; }

Intorducation to Java Script

Web Programming Languages

- There are many programming languages available to Web developers which we divided them in two groups:
  - Client Side Programming
    - **JavaScript**
  - Server Side Programming
    - PHP
    - ASP
    - JSP

Java Script

- **Scripts** make your HTML page more dynamic and interactive.
- JavaScript was designed to add **interactivity** to HTML pages
- It is a **client-side scripting language**, which means it runs on the user's machine and not on the server, as other web programming languages such as PHP and Ruby do
- JavaScript is usually **embedded directly** into HTML pages
- JavaScript is an **interpreted language** (means that scripts execute without preliminary compilation) Everyone can use JavaScript without purchasing a license

Are Java and JavaScript the Same?

- JavaScript was created by **Brendan Eich** at
- Netscape in 1995 and originally named "LiveScript."
- But Java was all the rage around that time, so for
- the sake of marketing, **"LiveScript"** became "JavaScript." Or just "JS,"
- Java and JavaScript are two completely different languages in both **concept** and design!
- Java (developed by Sun Microsystems) is a powerful and much more **complex programming language** – in the same category as *C* and *C++*.
- You can use java in your HTML page with using <**APPLET**>tag

How to Insert a JavaScript

- Like CSS, you can embed a script right in a document or keep it in an external file and link it to the page. Both methods use the **script element**.
  - **Embedded** script:To embed a script on a page, just add the code as the content of a script element
    - Head section
    - Body section
  - **External** Script

Embedded Script

- **Scripts in the head section:**
  - JavaScripts in the head section will be executed when CALLED.

```
<html>
 <head>
    <script Language="javascript" type="text/javascript">
        ....
    </script>
 </head>
 <body>
 </body>
</html>
```

Embedded Script

- **Scripts in the body section:**
  - Scripts to be executed when the page loads go in the body section

```
<html>
 <head>
 </head>
 <body>
    <script Language="javascript" type="text/javascript">
        ....
    </script>
 </body>
</html>
```

Embedded Script

- **Scripts in both body and head section:**
  - You can place an unlimited number of scripts in your documen
- ```
  <html>
   <head>
       <script Language="javascript" type="text/javascript">
           ....
       </script>
   </head>
   <body>
       <script Language="javascript" type="text/javascript">
           ....
       </script>
   </body>
  ```
- ```
  </html>
  ```

External Script

- When you want to run the same JavaScript on several pages so you can write a JavaScript in an external file.
- Save the external JavaScript file with a .js file extension and then call it in your html page
-
```
<html>
 <head>
    <script type="text/javascript" src="xxx.js">
        ....
    </script>
 </head>
 <body>
 </body>
</html>
```

How to Handle Older Browsers

- A browser that does not recognize the <script> tag at all, will display the <script> tag's content as text on the page.
- To prevent the browser from doing this, you should hide the script in comment tags
- <script type="text/javascript">

```
<!--
    document.write("Hello World!")
//-->
```

- </script>
- **<noscript>**Your browser does not supportJavaScript!**</noscript>**

JavaScript Basics

- There are a few common **syntactical rules** that wind their way though all of JavaScript.
- JavaScript is **case-sensitive**.
- JavaScript is a sequence of **statements** to be executed by the browser.
  - syntax
  - <script type="text/javascript">

    document.write("<h1>This is a header</h1>");

    document.write(" This is a paragraph ");
- </script>
- The **semicolon** at the end of the statement tells JavaScript that it's the end of the command, just as a period ends a sentence

JavaScript Comments

- JavaScript comments can be used to make the code more readable.
  - Single line comments start with //
  - Multi line comments start with /* and end with */.
-

```
<script type="text/javascript">
/*
    The code below will write one header and two paragraphs
*/
document.write("<h1>This is a header</h1>");
// This will write two paragraphs:
document.write("<p>This is a paragraph</p>");
document.write("<p>This is another paragraph</p>");
</script>
```

This is the end for this lecture