

HTML

Lecture 1

World Wide Web

World Wide Web the total set of interlinked hypertext documents residing on internet servers all around the world. Documents on the World Wide Web, called pages or Web pages, are written in HTML (Hyper Text Markup Language), identified by URLs.

The World Wide Web was developed by Timothy Berners-Lee in 1990's for the European Laboratory for Particle Physics.

Web page:

A document on the World Wide Web. A Web page consists of an HTML file, with associated files for graphics and scripts, in a particular directory on a particular machine (and thus identifiable by a URL).

Web site:

A group of related webpages and associated files, scripts, and databases that is served up by an internet server on the World Wide Web. The webpage in a Web site generally cover one or more related topics and are interconnected through hyperlinks. Most Web sites have a home page as their starting point, which frequently functions as a table of contents for the site.

Web Browser:

A software which is used to view webpages and access files and software related to those webpages. Originally developed to allow users to view or browse documents on the World Wide Web. Most Web browsers are also capable of downloading and transferring files. e.g. Internet Explorer, Google Chrome, Mozilla Firefox etc.

Web Server

A Web Server is special software, which runs on a computer and responds to requests made by other computers on the network. The web server satisfies the requests of the client (Web Browser). It usually has access to a resource, such as data, that the client wants. When the resource that the client wants becomes available, it sends a message to the client.

3 How web Application can get Execution?

4 Differentiate Client and server side languages

Website scripts run in one of two places – the client side, also called the front-end, or the server side, also called the back-end. The client of a website refers to the web browser that is viewing it. The server of a website is, of course, the server that hosts it.

Most web coding languages are designed to run on either the server side or the client side. This largely defines how they work. Here are some examples.

Client Side Languages

Client side development is done almost exclusively in **JavaScript**. This is, of course, in addition to basic HTML and CSS code. The reason JavaScript is called a client side language is because it runs scripts on your computer after you've loaded a web page.

Server Side Languages

A server side or back-end language runs its scripts before the HTML is loaded, not after. There are a range of server side languages in use on the web today. **PHP** is one of the most popular, as well as **Ruby on Rails**, C# and many others. They are called server side languages because their scripts are run not on your computer, but on the server which hosts the website and sends down the HTML code.

5 which protocol is used by web application?

HTTP and HTTPS

6 what are the types of website?

A website can be of two types:

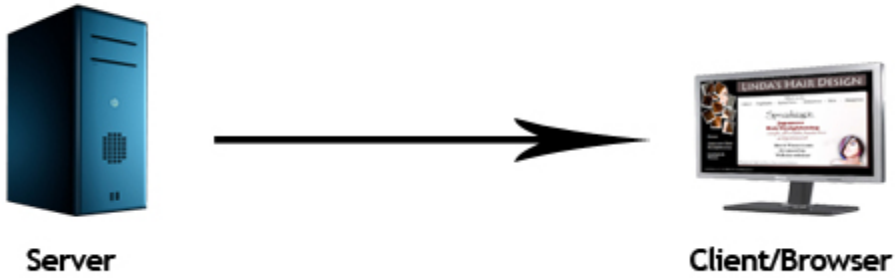
- Static Website
- Dynamic Website

Static website

Static website is the basic type of website that is easy to create. You don't need the knowledge of web programming and database design to create a static website. Its web pages are coded in HTML.

The codes are fixed for each page so the information contained in the page does not change and it looks like a printed page.

Static Website



Dynamic website

Dynamic website is a collection of dynamic web pages whose content changes dynamically. It accesses content from a database or Content Management System (CMS). Therefore, when you alter or update the content of the database, the content of the website is also altered or updated.

Dynamic website uses client-side scripting or server-side scripting, or both to generate dynamic content.

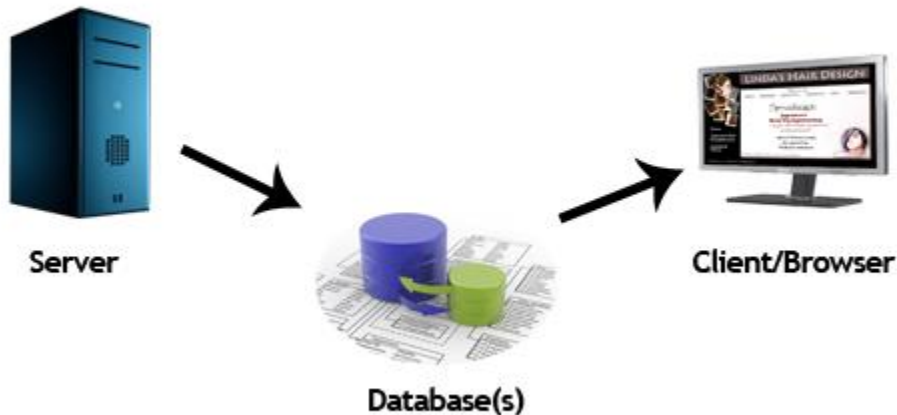
Client side scripting generates content at the client computer on the basis of user input. The web browser downloads the web page from the server and processes the code within the page to render information to the user.

Static Website	Dynamic Website
Prebuilt content is same every time the page is loaded.	Content is generated quickly and changes regularly.
It uses the HTML code for developing a website.	It uses the server side languages such as PHP, SERVLET, JSP, and ASP.NET etc. for developing a website.
It sends exactly the same response for every request.	It may generate different HTML for each of the request.

The content is only changed when someone publishes and updates the file (sends it to the web server).	The page contains "server-side" code which allows the server to generate the unique content when the page is loaded.
Flexibility is the main advantage of static website.	Content Management System (CMS) is the main advantage of dynamic website.

In server side scripting, the software runs on the server and processing is completed in the server then plain pages are sent to the user.

Dynamic Website



Static vs Dynamic website

Web Server

A Web Server is special software, which runs on a computer and responds to requests made by other computers on the network. The web server satisfies the requests of the client (Web Browser). It usually has access to a resource, such as data, that the client wants. When the resource that the client wants becomes available, it sends a message to the client.

WEB Hosting

Web hosting is a service that allows users to post web pages to the server. Web hosting companies provide customers with a variety of services related to building and maintaining websites.

Types of Web Hosting

1. Dedicated hosting

- 2. Shared hosting
- 3. Free Hosting

1. Dedicated hosting

In dedicated hosting the host is responsible for the equipment, dedicating an entire server to the client's websites. Dedicated hosting provides the total control of the server and what goes on the server. It is power full and works the way you want it to.

Dedicated hosting is ideal for websites requires high traffic, a lot of storage space and a high degree of security.

2. Shared Hosting

In this case the server is shared between many different websites. Typically, an individual or small to mid-sized business use shared hosting for their web presence.

3. Free Hosting

Some service providers offer free web hosting. Free web hosting is best suited for small sites with low traffic, like family sites or sites about hobbies.

This type of hosting one cannot use their own domain name at a free site. You have to use a name like `http://www.freesite/users/" yoursite.htm"`. This is hard to type, hard to remember, and not very professional.

Lecture 2

HTML - Elements

An **HTML element** is defined by a starting tag. If the element contains other content, it ends with a closing tag, where the element name is preceded by a forward slash as shown below with few tags –

Start Tag	Content	End Tag
<p>	This is paragraph content.	</p>

<code><h1></code>	This is heading content.	<code></h1></code>
<code><div></code>	This is division content.	<code></div></code>
<code>
</code>		

So here `<p>... </p>` is an HTML element, `<h1>...</h1>` is another HTML element. There are some HTML elements which don't need to be closed, such as `<img.../>`, `<hr />` and `
` elements. These are known as **void elements**.

HTML documents consists of a tree of these elements and they specify how HTML documents should be built, and what kind of content should be placed in what part of an HTML document.

HTML Tag vs. Element

An HTML element is defined by a starting tag. If the element contains other content, it ends with a closing tag. For example, `<p>` is starting tag of a paragraph and `</p>` is closing tag of the same paragraph but `<p>This is paragraph</p>` is a paragraph element.

Nested HTML Elements

It is very much allowed to keep one HTML element inside another HTML element –

Example

```
<! DOCTYPE html>

<html>

  <head>
    <title>Nested Elements Example</title>
  </head>

  <body>
    <h1>This is <i>italic</i> heading</h1>
```

```
<p>This is <u>underlined</u> paragraph</p>
</body>
</html>
```

Lecture 3

HTML - Attributes

We have seen few HTML tags and their usage like heading tags **<h1>**, **<h2>**, paragraph tag **<p>** and other tags. We used them so far in their simplest form, but most of the HTML tags can also have attributes, which are extra bits of information.

An attribute is used to define the characteristics of an HTML element and is placed inside the element's opening tag. All attributes are made up of two parts – a **name** and a **value**

- The **name** is the property you want to set. For example, the paragraph **<p>** element in the example carries an attribute whose name is **align**, which you can use to indicate the alignment of paragraph on the page.
- The **value** is what you want the value of the property to be set and always put within quotations. The below example shows three possible values of align attribute: **left**, **center** and **right**.

Attribute names and attribute values are case-insensitive. However, the World Wide Web Consortium (W3C) recommends lowercase attributes/attribute values in their HTML 4 recommendation.

Example

```
<!DOCTYPE html>
```

```
<html>

  <head>
    <title>Align Attribute Example</title>
  </head>

  <body>
    <p align = "left">This is left aligned</p>
    <p align = "center">This is center aligned</p>
    <p align = "right">This is right aligned</p>
  </body>

</html>
```

Core Attributes

The four core attributes that can be used on the majority of HTML elements (although not all) are –

- Id
- Title
- Class
- Style

The Id Attribute

The **id** attribute of an HTML tag can be used to uniquely identify any element within an HTML page. There are two primary reasons that you might want to use an id attribute on an element –

- If an element carries an id attribute as a unique identifier, it is possible to identify just that element and its content.

- If you have two elements of the same name within a Web page (or style sheet), you can use the id attribute to distinguish between elements that have the same name.

We will discuss style sheet in separate tutorial. For now, let's use the id attribute to distinguish between two paragraph elements as shown below.

Example

```
<p id = "html">This para explains what is HTML</p>
<p id = "css">This para explains what is Cascading Style Sheet</p>
```

The title Attribute

The **title** attribute gives a suggested title for the element. The syntax for the **title** attribute is similar as explained for **id** attribute –

The behavior of this attribute will depend upon the element that carries it, although it is often displayed as a tooltip when cursor comes over the element or while the element is loading.

Example

```
<!DOCTYPE html>

<html>

  <head>
    <title>The title Attribute Example</title>
  </head>

  <body>
    <h3 title = "Hello HTML!">Titled Heading Tag Example</h3>
  </body>

</html>
```

Now try to bring your cursor over "Titled Heading Tag Example" and you will see that whatever title you used in your code is coming out as a tooltip of the cursor.

The class Attribute

The **class** attribute is used to associate an element with a style sheet, and specifies the class of element. You will learn more about the use of the class attribute when you will learn Cascading Style Sheet (CSS). So for now you can avoid it.

The value of the attribute may also be a space-separated list of class names. For example –

```
class = "className1 className2 className3"
```

The style Attribute

The style attribute allows you to specify Cascading Style Sheet (CSS) rules within the element.

```
<!DOCTYPE html>
<html>

  <head>
    <title>The style Attribute</title>
  </head>

  <body>
    <p style = "font-family:arial; color:#FF0000;">Some text...</p>
  </body>

</html>
```

This will produce the following result –

At this point of time, we are not learning CSS, so just let's proceed without bothering much about CSS. Here, you need to understand what are HTML attributes and how they can be used while formatting content.

Internationalization Attributes

which are available for most (although not all) XHTML elements.

- dir
- lang

The dir Attribute

The **dir** attribute allows you to indicate to the browser about the direction in which the text should flow. The dir attribute can take one of two values, as you can see in the table that follows –

Value	Meaning
ltr	Left to right (the default value)
rtl	Right to left (for languages such as Hebrew or Arabic that are read right to left)

Example

```
<!DOCTYPE html>
<html dir = "rtl">

  <head>
    <title>Display Directions</title>
  </head>

  <body>
    This is how IE 5 renders right-to-left directed text.
  </body>
```

```
</html>
```

This will produce the following result –

When *dir* attribute is used within the `<html>` tag, it determines how text will be presented within the entire document. When used within another tag, it controls the text's direction for just the content of that tag.

The lang Attribute

The **lang** attribute allows you to indicate the main language used in a document, but this attribute was kept in HTML only for backwards compatibility with earlier versions of HTML. This attribute has been replaced by the **xml:lang** attribute in new XHTML documents.

The values of the *lang* attribute are ISO-639 standard two-character language codes. Check [**HTML Language Codes: ISO 639**](#) for a complete list of language codes.

Example

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

  <head>
    <title>English Language Page</title>
  </head>

  <body>
    This page is using English Language
  </body>

</html>
```