University of Technology
الجامعة التكنولوجية

Computer Science Department
قسم علوم الحاسوب

Static Web Programming
برمجة مواقع ثابتة

Dr. Shatha Habeeb

Dr. Athraa Jasim Mohammed

Dr. Muna Ghazi

cs.uotechnology.edu.iq

The First course 2024-2025

First Lecture

Introduction

- Just as there is a diversity of programming languages available and suitable for conventional programming tasks, there is a diversity of languages available and suitable for Web programming.

- The Internet becomes the main method in exchanging cultures and transferring knowledge between people.

- The Web was originally designed to deliver static Web pages from a Web server connected somewhere on the Internet to a Web browser sitting on a user's desktop computer. Basically, all users can retrieve a new page, read it, and then go on to the next page by clicking on a hot spot or hypertext link to.

- The Web was not designed to support EC (Ecommerce) sites, especially B2C (Business-to-Consumer) sites. In its original state, it was not possible to create pages that would allow consumers to easily determine what products were for sale, to select products as they moved from page to page (i.e., an electronic shopping cart), to place an order, or to verify an order. Similarly, there is no simple way to integrate a Web server with a database system containing product, pricing, The Web was originally designed to deliver and promotional data with transactional systems for processing orders and with payment systems for handling credit card purchases and settlements. Over time, these limitations have been addressed.

  - First, forms were added to HTML. Forms provided a way to produce Web pages from which a consumer could select, order, and pay for products.

- Second, special programming and scripting languages (e.g., Java and JavaScript) were created. These newer languages allowed application developers to produce interactive Web pages whose functionality emulated the rich functionality of standard Windows-based applications.
- Finally, standard application programmings interface (API) called the common gateway interface (CGI) was introduced. Generally speaking, an API provides a way for one software program to communicate with another, here as CGI provides a way for software developers and application programmers to integrate Web servers with various back-end programs and data sources.

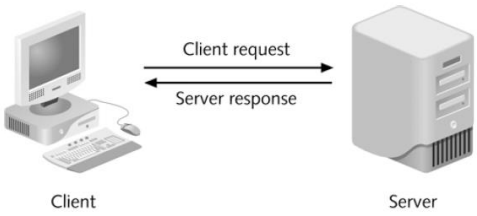
- CGI (Common Gateway Interface): refer to a specification by which program can communicate with a web server.

- Because of CGI‟s inefficiencies, newer APIs and special database gateway programs were also introduced. As a result of these changes, the Web is now well suited for the dynamic world of EC.

Internet

- The Internet is a computer network made up of thousands of networks worldwide.
- All computers on the Internet communicate with one another using the Transmission Control Protocol/Internet Protocol suite, abbreviated to TCP/IP.
- Computers on the Internet use a client/server architecture.
- This means that the remote server machine provides files and services to the user's local client machine.
- Software can be installed on a client computer to take advantage of the latest access technology.

- An Internet user has access to a wide variety of services : electronic mail, file transfer, vast information resources, interest group membership, interactive collaboration, multimedia displays, real-time broadcasting



Web server

- A Web server is a computer that runs special serving software. That software "serves" HTML pages and the files associated with those pages when requested by a client, usually a Web browser.

Client ("front end") :

- Presents an interface to the user gathers information from the user, submits it to a server, then receives, formats, and presents the results returned from the server

Web Browsing

- A web browser is a software application for retrieving, presenting, and traversing information resources on the World Wide Web.
- An information resource is identified by a Uniform Resource Identifier (URI) and may be a web page, image, video, or other piece.

Uniform Resource Locator **URL**

- **It is a four-part addressing scheme.**

Structure of a URL

- A URL is a Web Page's address and identifies where the web page is stored on the Internet.

Example : http://www.icci.org/studies/ips.html .

1. Protocol: http.
2. Host computer name: www.
3. Second-level domain name: icci.
4. Top-level domain name: org.
5. Directory name: studies.
6. File name: ips.html.

Several Top-level domain are common:

mil: للمواقع العسكرية edu: للمؤسسات التعليمية com: commercial enterprise. شركات military entity.

net: network access provider.

org: usually nonprofit organizations

gov: government entity.

Internet Service Provider (ISP):

- Provides access to the Internet along with other types of services such as e-mail.

HyperText Transfer Protocol (HTTP)

- to transmit data Protocols for other Internet applications.
- Client-side:

- HTML / XHTML (Extensible HyperText Markup Language)
- JavaScript / VBScript (client-side scripting)
- Applets / ActiveX controls

- Server-side:
  - JSP (Java Server Pages)
  - ASP (Active Server Pages)
  - ASP.NET (next generation of ASP)
  - PHP
  - Phython

Web application

- An application which is accessed via web browser over a network is called web application or web based application. All the websites are examples of web applications.

- Web application is written in a server side scripting language like ASP (active server pages).

- When user types address of website for example www.programming-web.com, browser transmits request to the web server which hosts the required site.
- On web server, web server software like email receives this request and processes it

The output generated by web server includes only those scripts which can be rendered by web browser.

- Above process repeats when user performs an action which requires communication with server such as submitting entry form or viewing another page.

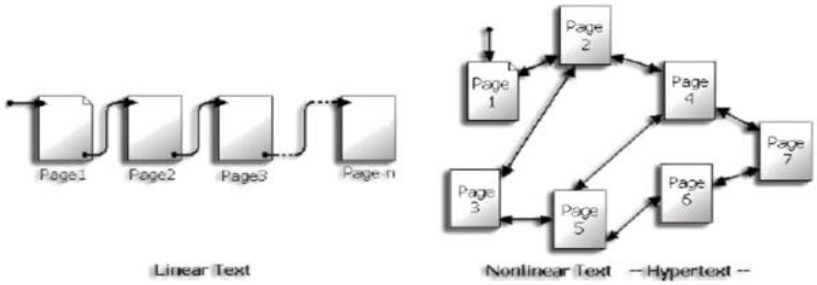
The Web Concepts

- The World-Wide Web (W3) was developed to be a pool of human knowledge, and human culture, which would allow collaborators in remote sites to share their ideas and all aspects of a common project.

- There are many Web concepts as following:

Hypertext:

- Hypertext links allow the reader to jump instantly from one electronic document to another.
- Two type : linear text and nonlinear text



Three basic "rules" of hypertext:

- A large body of information is organized into numerous fragments, or in the case of the Web, into pages.
- The pages relate to each other.
- The user needs only a small fraction of the information at any given moment.

Web Page

- The Web page is a space of information on the Internet, that presents information about a particular person, business, or organization or cause.
- The Web consists of files, called Web pages (documents).
- it is containing links to resources (text, images, audios, videos, and other data), throughout the Internet

web Site

- A Web site is a group of related Web pages
- which presents information about a particular person, business, organization or cause
- A well-designed Web site is a collection of related Web documents (pages) that share a common theme, look, and feel.
- The first thing we can see when „enter" the site is „Home Page", that offers links to more detailed information on the different topics in the same subject covered by the site.

Classifying the Web Sites:

- There are several classify for the early Web sites in many terms, as the follow:
- Environment.
- The General Approach
- Classify in terms of Range of Complexity.

Environment:

- There are three main types of Web sites according to this classify: Internet, Intranet, and Extranet Web Sites.

A- Internet Web Sites:

Internet Web Site is traditional Web sites that are intended for access by the general public.

B- Intranet Web Sites:

Intranet Web Site is intended only for internal (intra-organizational)

C- Extranet Web Sites:

Extranet Web Site is a combination of these. They are typically private and secured areas for the use of an organization and it is designated partners.

The General Approach:

- There are two main types of Web sites according to this classify: Static and Dynamic Web sites.

A- Static Web Sites:

- Static Web Site implies that the Web site will be a flat-file system of HTML files.
- all pages reside on the server and have fixed content that will be served “as is” to the user.

B- Dynamic Web Sites:

- Dynamically site requires that the content be stored in a database, not all sites require complete database functionality
- Part of a site may be dynamic while others are static.

Classify in terms of Range of Complexity

There are five main types of Web sites according to this classify:

- Static Web Sites.
- Static with Form-Based Interactivity Web Sites.
- Static with Dynamic Data Access Web Sites.
- Dynamically Generated Web Sites.
- Web-Based Software Applications Web Sites.

The First course 2024-2025

Second Lecture

HTML

The Lectures leads you through the basics of **Hyper Text Markup Language** (HTML). HTML is the building block for web pages. You will learn to use HTML to author an HTML page to display in a web browser.

You will need a text editor, such as **Notepad** and an **Internet browser**, such as Internet Explorer or Netscape.

- **What is Notepad and where do I get it?**

A Notepad is the default Windows text editor. On most Windows systems, click your Start button and choose Programs then Accessories. It should be a little blue notebook.

What is an html File?

HTML is a format that tells a computer how to display a web page. The documents themselves are plain text files with special "tags" or codes that a web browser uses to interpret and display information on your computer screen.

- HTML stands for Hyper Text Markup Language
- An HTML file must have an htm or html file extension
- Tags are enclosed in brackets **(< >)** and consist of an opening tag and a closing tag.

HTML structure

```
<HTML>
    <HEAD>
        <TITLE> Title of your web page </TITLE>
          (enter here what document is about)
    </head>
    <BODY>
        (here is the content) and ends with
  </BODY>
</HTML>
```

The <TITLE> of your document appears in the very top line of the user's browser. If the user chooses to "Bookmark" your page or save as a "Favorite"; it is the TITLE that is added to the list.

<BODY>...</BODY> Contains the viewed portion of the document.

Try It?

Open your text editor and type the following text:

```
<html>
<head>
<title>My First Webpage</title>
</head>
<body>
This is my first homepage.
</body>
</html>
```

Save the file as **mypage.html**. Start your Internet browser. Select **Open** (or Open Page) in the **File** menu of your browser. A dialog box will appear. Select **Browse** (or Choose File) and locate the html file you just created - **mypage.html** - select it and click **Open**. Now you should see an address in the dialog box, for example **C:\MyDocuments\mypage.html**. Click **OK**, and the browser will display the page.

Example Explained

The first tag in your html document is <html>. This tag tells your browser that this is the start of an html document. The last tag in your document is </html>. This tag tells your browser that this is the end of the html document.

The text between the <head> tag and the </head> tag is header information. Header information is not displayed in the browser window.

The text between the <title> tags is the title of your document. The <title> tag is used to uniquely identify each document and is also displayed in the title bar of the browser window.

The text between the <body> tags is the text that will be displayed in your browser.

Basic HTML Tags HTML Elements

This is an HTML element:

Tag	Description
<html>	Defines an HTML document
<body>	Defines the document's body
<h1> to <h6>	Defines header 1 to header 6
<p>	Defines a paragraph
 	Inserts a single line break
<hr>	Defines a horizontal rule
<!-->	Defines a comment

Tag	Description
	Defines **bold** text
<big>	Defines big text
<i>	Defines *italic* text
<small>	Defines small text
<sup>	Defines superscripted text
<sub>	Defines subscripted text
<u>	Deprecated. Use styles instead

The text between the and tags will be displayed in a bold font.

Note: - HTML tags are not case sensitive
** means the same as </B>**

this text is bold

The HTML element begins with a start tag:
The content of the HTML element is: This text is bold

The HTML element ends with an end tag:
The most important tags in HTML are tags that define headings, paragraphs and line breaks

Body tag and attributes

Tags can have attributes. Attributes can provide additional information about the HTML elements on your page. The <tag> tells the browser to do something, while the

attribute tells the browser how to do it. For instance, if we add the bgcolor attribute, we can tell the browser that the background color of your page should be blue, like this:

<body bgcolor="blue">.

HTML Backgrounds

Backgrounds

The <body> tag has two attributes where you can specify backgrounds. The background can be a color or an image.

Color: Bgcolor

The bgcolor attribute specifies a background-color for an HTML page. The value of this attribute can be a hexadecimal number, an RGB value, or a color name:

<body bgcolor="#00FFFF">

<body bgcolor="rgb(0,0,0)">
<body bgcolor="black">

The lines above all set the background-color to black.

image Background

The background attribute can also specify a background-image for an HTML page. The value of this attribute is the URL of the image you want to use. If the image is smaller than the browser window, the image will repeat itself until it fills the entire browser window.

<body background="clouds.gif">

<body background="http://profdevtrain.austincc.edu/html/graphics/clouds.gif">

Try It Out!

Open your text editor and type the following text:

```
<html>
<head>
<title>My First Webpage</title>
</head>
<body background="http://html/graphics/clouds.gif" bgcolor="#EDDD9E">
```

</body>
</html>

Save your page as **mypage3.html** and view it in your browser. To view how the page should look, visit this web page: **http:// html/mypage3.html**

HTML Colors

Color Values

Colors are defined using a hexadecimal notation for the combination of red, green, and blue color values (RGB). The lowest value that can be given to one light source is 0 (hex #00). The highest value is 255 (hex #FF). This table shows the result of combining red, green, and blue:

Color	Color HEX	Color RGB
	#000000	rgb(0,0,0)
	#FF0000	rgb(255,0,0)
	#00FF00	rgb(0,255,0)
	#0000FF	rgb(0,0,255)
	#FFFF00	rgb(255,255,0)
	#00FFFF	rgb(0,255,255)
	#FF00FF	rgb(255,0,255)
	#C0C0C0	rgb(192,192,192)
	#FFFFFF	rgb(255,255,255)

Headings

Headings are defined with the <h1> to <h6> tags. <h1> defines the largest heading while <h6> defines the smallest.

<h1>This is a heading</h1>

<h2>This is a heading</h2>

<h3>This is a heading</h3>

<h4>This is a heading</h4>

<h5>This is a heading</h5>

<h6> This is a heading</h6>

HTML automatically adds an extra blank line before and after a heading.

<h5 align="left">I can align headings </h5>

<h5 align="center">This is a centered heading </h5>

<h5 align="right">This is a heading aligned to the right </h5>

align : This attribute specifies the horizontal alignment of element.

Paragraphs

Paragraphs are defined with the tag. Think of a paragraph as a block of text. You can use the align attribute with a paragraph tag as well.

This is a paragraph

this is another paragraph

Important: You must indicate paragraphs with elements. A browser ignores any indentations or blank lines in the source text. Without elements, the document becomes one large paragraph. HTML automatically adds an extra blank line before and after a paragraph.

Line Breaks

The <br> tag is used when you want to start a new line, but don't want to start a new paragraph. The <br> tag forces a line break wherever you place it. It is similar to single spacing in a document.

This Code	Would Display
This is a para graph with line breaks	This is a para graph with line breaks

The <br> tag has no closing tag.

Horizontal Rule

The <hr> element is used for horizontal rules that act as dividers between sections, like this:

The horizontal rule does not have a closing tag. It takes attributes such as align and width. For instance:

This Code	Would Display
<hr width="50%" align="center">	

Comments in HTML

The comment tag is used to insert a comment in the HTML source code. A comment can be placed anywhere in the document and the browser will ignore everything inside the brackets.

This Code	Would Display
This html comment would <!—This is a comment --> be displayed like this.	This HTML comment would be displayed like this.

Note: You need an exclamation point after the opening bracket <!-- but not before the closing bracket -->.

HTML automatically adds an extra blank line before and after some elements, like before and after a paragraph, and before and after a heading. If you want to insert blank lines into your document, use the <br> tag.

Try It Out!

Open your text editor and type the following text:

```
<html>
<head>
<title>My First Webpage</title>
</head>
<body>
<h1 align="center">My First Webpage</h1>
<p>Welcome to my first web page. I am writing this page using a text editor and plain old html.</p>
<p>By learning html, I'll be able to create web pages like a pro....<br>
```

which I am of course.</p>
</body>
</html>

HTML Character Entities

Some characters have a special meaning in HTML, like the less than sign (<) that defines the start of an HTML tag. If we want the browser to actually display these characters we must insert character entities in place of the actual characters themselves.

The Most Common Character Entities:

Result	Description	Entity Name
	non-breaking space	
<	less than	&lt;
>	greater than	&gt;
&	ampersand	&amp;

A character entity has three parts: an ampersand (&), an entity name or an entity number, and finally a semicolon (;). The & means we are beginning a special character, the **;** means ending a special character and the letters in between are sort of an abbreviation for what it's for.

Font tag and attributes

<FONT>...</FONT>	Changes font attributes for text within the tags
<FONT size=”value”>...</FONT>	Sets the font to a size from 1 to 7, with 1 the smallest and 7 the largest
<FONT face=”name”>...</FONT>	Sets the font face
<FONT color=”color”>...</FONT>	Sets the font color using hexadecimal code

	font	**<font** face="Times New Roman" size="4">Example****	Example
	font	**<font** face="Times New Roman" size="+3" color="#ff0000">Example****	Example

HTML Lists

HTML provides a simple way to show unordered lists or ordered lists.

Unordered Lists

An unordered list is a list of items marked with bullets (typically small black circles). An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

This Code	Would Display
<ul type=”square”> - Coffee - Milk - Tea 	• Coffee • Milk • Tea

```
<UL TYPE="square">
<UL TYPE="circle">
```

Ordered Lists

An ordered list is also a list of items. The list items are marked with numbers. An ordered list starts the <ol> tag. Each list item starts with the <li> tag.

This Code	Would Display
 - Coffee - Milk 	1. Coffee 2. Milk

<OL TYPE="i">	<OL TYPE="I">	<OL TYPE="a">	<OL TYPE="A">
.i .ii .iii .iv .v	.I .II .III .IV .V	.a .b .c .d .e	.A .B .C .D .E

Inside a list item you can put paragraphs, line breaks, images, links, other lists, etc.
Attribute for <OL ...> START = *integer*

```
<ol  start=2>
        <li>apples</li>
        <li>bananas</li>
        <li>grapes</li>
</ol>
```

```
<ol>
<li>List item 1</li>
<li>List item 2</li>
<li>List item 3</li>
<li>List item 4</li>
</ol>

Numbered Special Start

<ol start="5">
<li>List item 1</li>
<li>List item 2</li>
<li>List item 3</li>
<li>List item 4</li>
</ol>

Lowercase Letters

<ol type="a">
<li>List item 1</li>
<li>List item 2</li>
<li>List item 3</li>
<li>List item 4</li>
</ol>

Capital Letters

<ol type="A">
<li>List item 1</li>
<li>List item 2</li>
<li>List item 3</li>
<li>List item 4</li>
</ol>
```

Definition Lists

Definition lists consist of two parts: a **term** and a **description**. To mark up a definition list, you need three HTML elements; a container <dl>, a definition term <dt>, and a definition description <dd>.

This Code	Would Display
<dl> <dt>Cascading Style Sheets</dt> <dd>Style sheets are used to provide presentational suggestions for documents marked up in HTML. </dd> <dt>AI</dt> <dd> hgfddssdf</dd> </dl>	Cascading Style Sheets Style sheets are used to provide presentational suggestions for documents marked up in HTML.

Inside a definition-list definition (the <dd> tag) you can put paragraphs, line breaks, images, links, other lists, etc

Try It Out

<dl>
<dt>Definition Term</dt>
<dd>Definition of the term</dd>
<dt>Definition Term</dt>
<dd>Definition of the term</dd>
</dl>

Try It Out

Open your text editor and type the following:

```
<html>
<head>
<title>My First Webpage</title>
</head>
<body bgcolor="#EDDD9E">
<h1 align="center">My First Webpage</h1>
<p>Welcome to my <strong>first</strong> webpage. I am writing this page using a text editor and plain old html.</p>
<p>By learning html, I'll be able to create web pages like a pro....<br>
which I am of course.</p>
Here's what I've learned:
<ul>
<li>How to use HTML tags</li>
<li>How to use HTML colors</li>
<li>How to create Lists</li>
</ul>
</body>
</html>
```

HTML Links

HTML uses the <a> anchor tag to create a link to another document or web page.

The Anchor Tag and the Href Attribute

An anchor can point to any resource on the Web: an HTML page, an image, a sound file, a movie, etc.

The syntax of creating an anchor:
Text to be displayed

** Visit Our Site****

The <a> tag is used to create an anchor to link from, the href attribute is used to tell the address of the document or page we are linking to, and the words between the open and close of the anchor tag will be displayed as a hyperlink.

This Code	Would Display
Visit ACC!	Visit ACC!

The Target Attribute

With the target attribute, you can define **where** the linked document will be opened. By default, the link will open in the current window. The code below will open the document in a new browser window:

Visit ACC!

Email Links

To create an email link, you will use mailto: plus your email address.
<a HREF=”mailto:email”>...</A> Creates a mailto link

Ex. Here is a link to ACC's Help Desk:
Email Help Desk

To add a subject for the email message, you would add ?subject= after the email address. For example:

Email Help Desk

The First course 2024-2025

Third lecture

Tables

Tables are defined with the <table> tag. A table is divided into rows (with the <tr> tag), and each row is divided into data cells (with the <td> tag). The letters td stands for table data, which is the content of a data cell. A data cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables, etc.

This Code	Would Display
<table> <tr> <td>row 1, cell 1</td> <td>row 1, cell 2</td> </tr> <tr> <td>row 2, cell 1</td> <td>row 2, cell 2</td> </tr> </table>	row 1, cell 1 row 1, cell 2 row 2, cell 1 row 2, cell 2

Tables and the Border Attribute

To display a table with borders, you will use the border attribute.

This Code	Would Display
<table border="1"> <tr> <td>Row 1, cell 1</td> <td>Row 1, cell 2</td> </tr> </table>	row 1, cell 1 | row 1, cell 2

and....

This Code	Would Display
<table border="5"> <tr> <td>Row 1, cell 1</td>	row 1, cell 1 | row 1, cell 2

<td>Row 1, cell 2</td> </tr> </table>	

Open up your text editor. Type in your <html>, <head> and <body> tags. From here on I will only be writing what goes between the <body> tags. Type in the following:

<table border="1">

<tr>
<td>Tables can be used to layout information</td>
<td> <img src="http://profdevtrain.austincc.edu/html/graphics/chef.gif">
</td>
</tr>
</table>

Save your page as **mytable1.html** and view it in your browser. To see how your page should look visit this web page: **http://profdevtrain.austincc.edu/html/mytable1.html**

Headings in a Table
Headings in a table are defined with the <th> tag.

This code	Would Display
<table border="1"> <tr> <th>Heading</th> <th>Another Heading</th> </tr> <tr> <td>row 1, cell 1</td> <td>row 1, cell 2</td> </tr> <tr> <td>row 2, cell 1</td> <td>row 2, cell 2</td> </tr>	<table><tr><th>Heading</th><th>Another Heading</th></tr><tr><td>row 1, cell 1</td><td>row 1, cell 2</td></tr><tr><td>row 2, cell 1</td><td>row 2, cell 2</td></tr></table>

</table>	

Cell Padding and Spacing

The <table> tag has two attributes known as cellspacing and cellpadding. Here is a table example without these properties. These properties may be used separately or together.

This Code	Would Display
<table border="1"> <tr> <td>some text</td> <td>some text</td> </tr> <tr> <td>some text</td> <td>some text</td></tr></table>	some text / some text some text / some text

Cellspacing is the pixel width between the individual data cells in the table (The thickness of the lines making the table grid). The default is zero. If the border is set at 0, the cellspacing lines will be invisible.

This Code	Would Display
<table border="1" cellspacing="5"> <tr> <td>some text</td> <td>some text</td> </tr><tr> <td>some text</td> <td>some text</td> </tr> </table>	some text / some text some text / some text

Cellpadding is the pixel space between the cell contents and the cell border. The default for this property is also zero. This feature is not used often, but sometimes comes in handy

when you have your borders turned on and you want the contents to be away from the border a bit for easy viewing. Cellpadding is invisible, even with the border property turned on. Cellpadding can be handled in a style sheet.

This Code	Would Display
<table border="1" cellpadding="10"> <tr> <td>some text</td> <td>some text</td> </tr><tr> <td>some text</td> <td>some text</td> </tr> </table>	some text some text some text some text

Table Tags

Tag	Description
<table>	Defines a table
<th>	Defines a table header
<tr>	Defines a table row
<td>	Defines a table cell
<caption>	Defines a table caption
<TD colspan=”columns”>	Sets a cell to span columns
<TD rowspan=”rows”>	Sets a cell to span rows

Table Data and Table Header Attributes

Ex.

<Table border=1 cellpadding =2>
<tr>
<th> Column 1 Header</th>
<th> Column 2 Header</th>
</tr>
<tr>
<td colspan=2> Row 1 Col 1</td>
</tr>
<tr>
<td rowspan=2>Row 2 Col 1</td>
<td> Row 2 Col2</td>
</tr>
<tr>

Column 1 Header	Column 2 Header
Row 1 Col 1	
Row 2 Col 1	Row 2 Col 2
	Row 3 Col 2

<td> Row 3 Col2</td>
</tr>
</table>

What will be the output?

<TABLE BORDER width="750">
<TR> <TD colspan="4" align="center">Page Banner</TD></TR>
<TR> <TD rowspan="2" width="25%">Nav Links</TD>
<TD colspan="2">Feature Article</TD>
<TD rowspan="2" width="25%">Linked Ads</TD></TR>
<TR><TD width="25%">News Column 1 </TD>
<TD width="25%"><News Column 2 </TD></TR> </TABLE>

HTML Images

The Image Tag and the Src Attribute

The <img> tag is empty, which means that it contains attributes only and it has no closing tag. To display an image on a page, you need to use the src attribute. Src stands for "source". The value of the src attribute is the URL of the image you want to display on your page. The syntax of defining an image:

This Code	Would Display
<img src="graphics/chef.gif">	

Not only does the source attribute specify what image to use, but where the image is located. The above image, graphics/chef.gif, means that the browser will look for the image name **chef.gif** in a **graphics** folder in the same folder as the html document itself.

The Alt Attribute

The alt attribute is used to define an alternate text for an image. The value of the alt attribute is author-defined text:
<img src="graphics/chef.gif" alt="Smiling Happy Chef ">

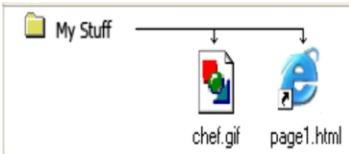
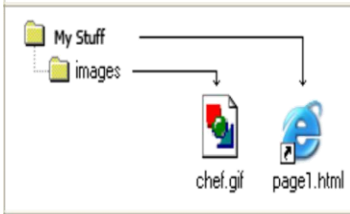
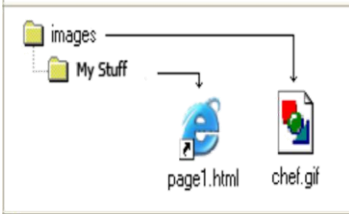
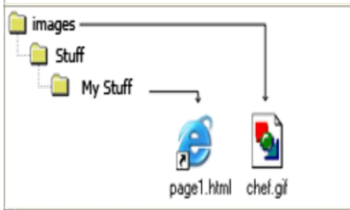
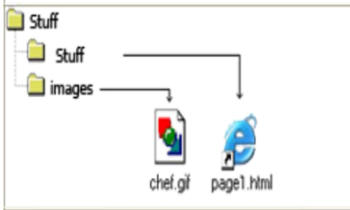
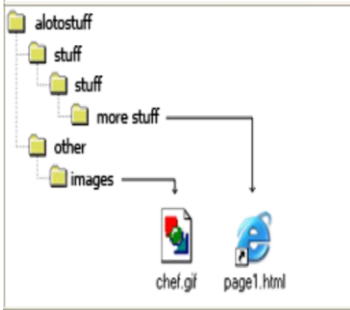
<IMG src="url" alt="text"> Embeds an image in the document at the location of the tag Adds an image with a text description.

<IMG src=”url” alt=”text” align=”direction”> Aligns an image to the left, right, center, bottom, or top

<IMG src=”url” alt=”text” border=”number”> Sets the size of the border around an image

<IMG src=”url” alt=”text” height=”pixels”> Sets the height of an image

<IMG src=”url” alt=”text” width=”pixels”> Sets the width of an image

	src="chef.gif" means that the image is in the same folder as the html document calling for it.
	src="images/chef.gif" means that the image is one folder down from the html document that called for it. This can go on down as many lavers as necessarv.
	src="../chef.gif" means that the image is in one folder up from the html document that called for it.
	src="../../chef.gif" means that the image is two folders up from the html document that called for it
	src="../images/chef.gif" means that the image is one folder up and then another folder down in the images directory.
	src="../../../other/images/chef.gif" means this goes multiple layers up

The browser puts the image where the image tag occurs in the document.

```
<html>
<head>
<title>My First Webpage</title>
</head>
<body>
<h1 align="center">My First Web page</h1>
<p>Welcome to my first webpage. I am writing this page using a text editor and plain old html.</p> <p>By learning html, I'll be able to create web pages like a pro....<br> which I am of course.</p>
<!-- Who would have guessed how easy this would be :) -->
```

** <img src="graphics/chef.gif" width="130" height="101" alt="Smiling Happy Chef" align="center"> **
** This is my Chef **
</body>
</html>

Save your page as **mypage5.html** and view it in your browser. To see how your page should look visit this web page: **http://profdevtrain.austincc.edu/html/mypage5.html**

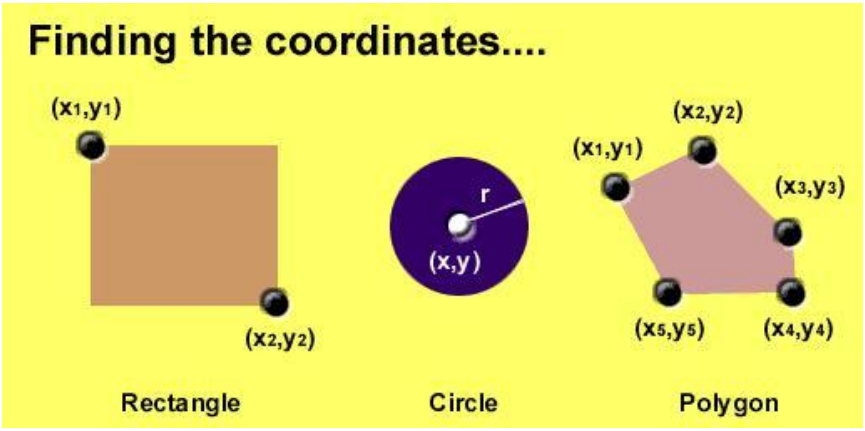
Image Maps

Image maps are images, usually in gif format that have been divided into regions; clicking in a region of the image cause the web server to be connected to a new URL. Image maps are graphical form of creating links between pages.

There are two type of image maps: Client side and server side

Both types of image maps involve a listing of co-ordinates that define the mapping regions and which URLs those coordinates are associated with. This is known as the map file.

Area Shapes Used



<IMG SRC="note.GIF" Width=200 Height=200 border="5"
useMAP="#map1">
<MAP NAME="map1">

<AREA SHAPE="RECT" COORDS="0,0,90,90" HREF="hi.html" ALT="see me…">
<AREA SHAPE="RECT" COORDS="100,100,160,160" HREF="divPara.html" ALT="see him…" >
<AREA SHAPE="CIRCLE" COORDS="150,50,20" HREF="house.html"
ALT="see it…" > </MAP>

- Types of Shapes
    - Rect : used for squares and ordered shapes.
    - Circle : used for circles.
    - Poly : used for unordered shapes.
- Number of coordenations for each shape:
    - Rect : 4 numbers for two corners
    - Circle : 3 numbers for the center & R
    - Poly : depends on the number of corners of the shape(2 numbers for each corner)

Frame tag and attributes

Frames allow you to divide the page into several rectangular areas and to display a separate document in each rectangle. Each of those rectangles is called a "frame". Frames are very popular because they are one of the few ways to keep part of the page stationary while other parts change.

<FRAMESET> defines the general layout of a web page that uses frames.

```
<FRAMESET cols=”30%,40%,30%”>
<frame src=”p1.html”>
<frame src=”p2.html”>
<frame src=”p3.html”>

</FRAMESET>
```

p1.html	p2.html	p3.html

is used in conjunction with <FRAME> and <NOFRAMES>.

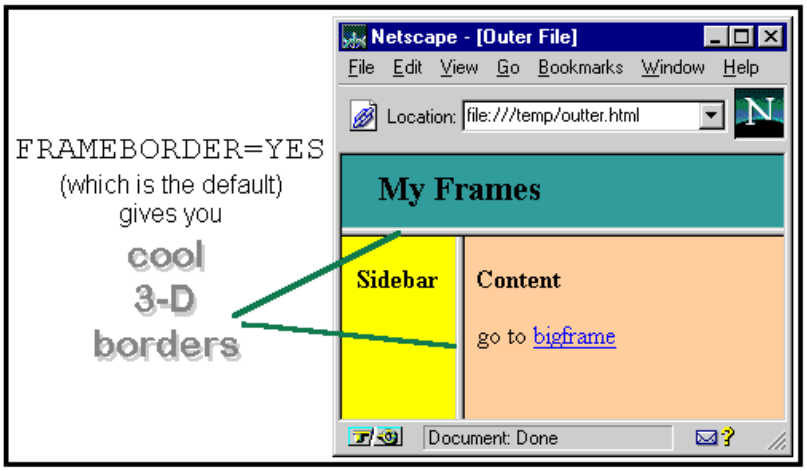
- COLS: how many cols in the frameset

- ROWS: how many rows in the frameset

FRAMEBORDER: if the frames should have borders

<FRAMESET>...</FRAMESET>	Specifies the layout of subsections in the main browser window
<FRAMESET rows=”value,value”>	Defines the rows within a frameset
<FRAMESET cols=”value,value”>	Defines the columns within a frameset
<NOFRAMES>...</NOFRAMES>	Provides alternate content for browsers that do not support frames
<FRAME src=”?”>	Defines the appearance and content of a single frame
<FRAME name=”name”>	Labels the frame for targeting by other frames

<NOFRAMES>: Frame – capable browsers ignore all HTML within this tag including the contents of the BODY element. This element does not have any attributes.



For example, you might use this code for a set of frames that has two rows and three columns:

<FRAMESET *ROWS="80%,20%" COLS="60%,20%,20%"*>
<FRAME SRC="a.html">
<FRAME SRC="b.html">
<FRAME SRC="c.html">
<FRAME SRC="d.html">
<FRAME SRC="rowcol1e.html">
<FRAME SRC="rowcol1f.html">
<NOFRAMES>*NOFRAMES stuff* </NOFRAMES>

a.html	b.html	c.html
d.html	e.html	f.html

H.W. write the code for this frameset?

<table>
<tr><td>Page1</td><td colspan="2">Page2</td></tr>
<tr><td rowspan="2">Page3</td><td colspan="2">Page4</td></tr>
<tr><td>Page5</td><td>Page6</td></tr>
</table>

The First course 2024-2025

Fourth lecture

Forms tag and attributes

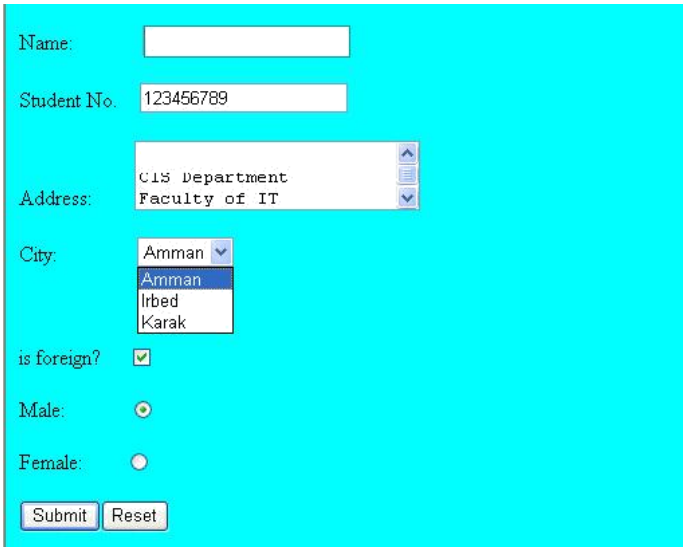
Forms are the most popular way to make web pages interactive. Like forms on paper, a form on a web page allows the user to enter requested information and submit it for processing.

- ACTION: URL of the CGI program (Common Gateway Interface) program that is going to accept the data from the form, process it, and send a response back to the browser.
- METHOD: how to transfer the data to the CGI, **GET** (default) or **POST** specifies which **HTTP** method will be used to send the form's contents to the web server
- NAME: name of this form by VBScript or JavaScripts.

To insert a form we use the <FORM></FORM> tags. The rest of the form elements must be inserted in between the form tags.

```
<HTML> <HEAD>
<TITLE> Sample Form</TITLE>
</HEAD>
<BODY BGCOLOR="FFFFFF">
<FORM ACTION = http://www.xnu.com/formtest.asp Method="get"
Name="Form1">
<P> First Name: <INPUT TYPE="TEXT" NAME="fname"  MAXLENGTH="50">
</P>
<P> <INPUT TYPE="SUBMIT" NAME="fsubmit1" VALUE="Send Info"> </P>
</FORM>
```

H.W/ write the code for this output?



INPUT

- <INPUT ...> creates the data entry fields on an HTML form. Attribute for <INPUT ...>
- TYPE: what type of field
- NAME: name of this form field
- VALUE: initial or only value of this field
- SIZE: how wide the text field should be

TYPE = TEXT | CHECKBOX | RADIO | PASSWORD | HIDDEN | SUBMIT | RESET | BUTTON | FILE | IMAGE

<INPUT> Element's Properties

TYPE= Type of INPUT entry field.

NAME = Variable name passed to CGI application

VALUE= The data associated with the variable name to be passed to the CGI application

CHECKED= Button/box checked

SIZE= Number of visible characters in text field

MAXLENGHT= Maximum number of characters accepted.

Text boxes*:* Used to provide input fields for text, phone numbers, dates, etc.

<INPUT TYPE= " TEXT " >
Browser will display
Textboxes use the following attributes:
TYPE: text.
SIZE: determines the size of the textbox in characters.
Default=20 characters.
MAXLENGHT *:* determines the maximum number of characters that the field will accept.
NAME: is the name of the variable to be sent to the CGI application.
VALUE: will display its contents as the default value.

<SELECT name=”NAME”> <OPTION></OPTION> </SELECT>	Defines each menu item Generates a pull-down menu
<INPUT type=”checkbox”>	Generates a check box
<INPUT type=”hidden”>	Conceals a field from view
<INPUT type=”image”>	Generates an image that acts like a Submit button
<INPUT type=”password”>	Generates a one-line password box
<INPUT type=”radio”>	Generates a radio button
<INPUT type=”text”>	Generates a one-line text box
<INPUT type=”submit”>	Generates a Submit button (send form)
<INPUT type=”reset”>	Generates a Reset button (clear form)

Example on Password Box

```
<HTML><HEAD>
<TITLE>Form_Password_Type</TITLE></HEAD>
<BODY>
<h1> <font color=red>To Access, Please enter:</font></h1>
 <FORM name="fome2" Action="page2.html" method="get">
User Name: <INPUT TYPE="TEXT" Name="FName" SIZE="15"
  MAXLENGTH="25"><BR>
Password: <INPUT TYPE="PASSWORD"
NAME="PWord" value="" SIZE="15"
MAXLENGTH="25"><BR>
<input type=”submit” value=”login”>
</FORM></BODY>
</HTML>
```

Display page

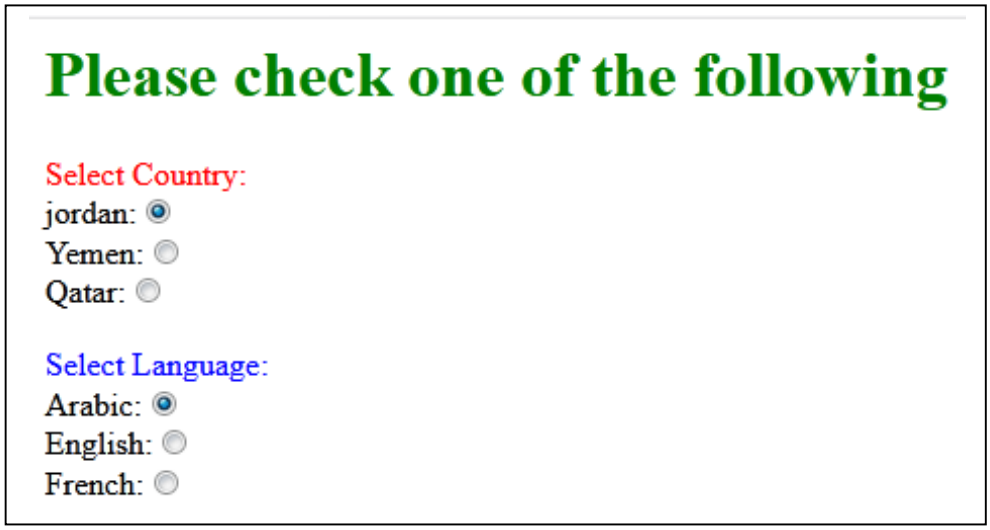
Example on checkBox

```
<HTML> <HEAD><TITLE>CheckBoxType</TITLE></HEAD>
<BODY>
<h1> <font color=green>Please check one of the following</font></h1>
<FORM name="fome3" Action="url" method="get">
<font color=red> Select Country: </font><BR>
        Iraq:<INPUT TYPE="CheckBox" Name="country" CHECKED><BR>
        Jordan:<INPUT TYPE="CheckBox" Name="country"> <BR>
        Qatar:<INPUT TYPE="CheckBox" Name="country"><BR><BR>
<font color=blue>Select Language:</font><BR>
        Arabic:<INPUT TYPE="CheckBox" Name="language" CHECKED><BR>
        English:<INPUT TYPE="CheckBox" Name="language"><BR>
        French:<INPUT TYPE="CheckBox" Name="language"> <BR>
</FORM>
</BODY> </HTML>
```

Display page

Example to Radio Button

```
<HTML>
 <HEAD><TITLE>CheckBoxType</TITLE> </HEAD>
<BODY>
<h1> <font color=green>Please check one of the following</font></h1>
<FORM name="fome3" Action="page3.html" method="get">
<font color=red> Select Country: </font><BR>
      jordan:<INPUT TYPE= "RADIO" Name="country" CHECKED><BR>
      Yemen<INPUT TYPE="RADIO " Name="country"><BR>
      Qatar:<INPUT TYPE="RADIO" Name="country"><BR><BR>
<font color=blue>Select Language:</font><BR>
      Arabic:<INPUT TYPE="RADIO" Name="language" CHECKED><BR>
      English:<INPUT TYPE=" RADIO " Name="language"><BR>
      French:<INPUT TYPE=" RADIO " Name="language"> <BR>
      </FORM> </BODY></HTML>
```



Push Button

Push Button: This element would be used with JavaScript to cause an action to take place.
<INPUT TYPE="BUTTON" >
Browser will display
Push Button has the following attributes:
TYPE: button.
NAME: is the name of the button to be used in scripting.
VALUE: determines the text label on the button.

Write code to this page

Submit Button

Submit: Every set of Form tags requires a Submit button. This is the element causes the browser to send the names and values of the other elements to the CGI Application specified by the ACTION attribute of the FORM element.

<INPUT TYPE="SUBMIT">

Submit has the following attributes:
TYPE: submit.
NAME: value used by the CGI script for processing.
VALUE: determines the text label on the button, usually Submit Query

Reset Button

Reset: It is a good idea to include one of these for each form where users are entering data. It allows the surfer to clear all the input in the form.

<INPUT TYPE="RESET">

Example

<FORM Action="URL" method="get">

First Name: <INPUT TYPE="TEXT" Size=25 name="firstName">
Family Name: <INPUT TYPE="TEXT" Size=25 name="LastName">

<FONT Color = red>
<STRONG>Press Here to submit the data:</STRONG>

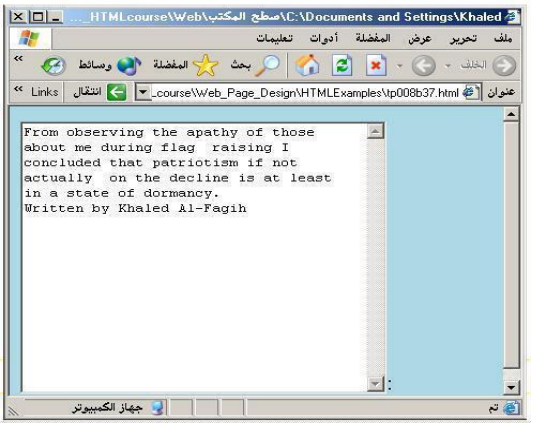
<INPUT TYPE="submit" VALUE="SubmitData">
<INPUT TYPE="RESET" VALUE="Reset"> </FORM>

Other Elements used in Forms

1-<TEXTAREA > </TEXTAREA>: is an element that allows for free form text entry.

Textarea has the following attributes:

NAME: is the name of the variable to be sent to the CGI application.
ROWS: the number of rows to the textbox.
COLS: the number of columns to the textbox



2- <SELECT></SELECT> elements, where the attributes are set differently.

The Select elements attributes are:
NAME: is the name of the variable to be sent to the CGI application.
SIZE: this sets the number of **visible** choices.
MULTIPLE: the presence of this attribute signifies that the user can make multiple selections. By default only one selection is allowed.

```
<BODY bgcolor=lightblue>
<form>
Select the cities you have visited:
<SELECT name=—list‖ size=5>
<option> London</option>
<option> Tokyo</option>
<option> Paris</option>
<option> New York</option>
<option> LA</option>
<option> KL</option>
</SELECT></form> </BODY>
```

H.W/What the output of this code

```
<BODY>
<h2><font color=blue>What type of Computer do you have?</font><h2>
<FORM>
<SELECT NAME="ComputerType" size=4>
<OPTION value="IBM" SELECTED> IBM</OPTION>
<OPTION value="INTEL"> INTEL</OPTION>
<OPTION value=" Apple"> Apple</OPTION>
<OPTION value="Compaq"> Compaq</OPTION>
</SELECT>
</FORM></BODY>
```

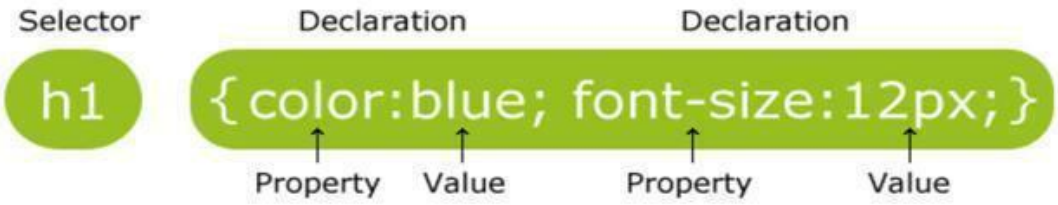
The First course 2024-2025

Fifth lecture

Cascading style sheets

- CSS stands for Cascading Style Sheets
- Styles define how to display HTML elements
- Two Ways to Insert CSS: Internal Style Sheets and External Style Sheets

CSS Syntax:



A CSS rule has two main parts: a **selector**, and one or more **declarations**:

- The selector is normally the HTML element you want to style.
- Each declaration consists of a property and a value. The property is the style attribute you want to change. Each property has a value.

Internal Stylesheet

First we will explore the internal method. This way you are simply placing the CSS code within the <head> </head> tags of each HTML file you want to style with the CSS. The format for this is shown in the example below:

```
<head>
<title>page style</title>
<style type="text/css">
H1 {color:red;}
P {color:blue;}
</style>
</head>
<body>
<h1>All header 1 elements will be red</h1>
<p> elements will be</p>
```

With this method each HTML file contains the CSS code needed to style the page. Meaning that any changes you want to make to one page will have to be made to all. This method can be good if you need to style only one page, or if you want different pages to have varying styles.

```
p {color:red;text-align:center;}
```

Example1 of Internal CSS:

```
<html>
<head>
<style type="text/css">
h1 {color:red;}
h2 {color:blue;}
p {color:green;}
</style>
</head>
<body>
<h1>All header 1 elements will be red</h1>
<h2>All header 2 elements will be blue</h2>
<p>All text in paragraphs will be green</p>
</body>
</html>
```

Example2 of Internal CSS:

```
<html>
<head>
<style>
p {text-align: center;
   color: red;
}
</style>
</head>
<body>
<p>Every paragraph will be affected by the style.</p>
<p> Me too!</p>
<p>And me!</p>
 </body></html>
```

Style the element with id=" para1": The *#id* selector styles the element with the specified id.

```
<html>
<head>
<style>
#para1 {
    text-align: center;
    color: red;}
#para2{
    text-align: right;
color: blue;

}
</style>
</head>
<body>
<p id="para1">Hello World!</p>
<p id=”para2”>This paragraph is not affected by the style.</p>
</body>
</html>
```

Select and style all elements with class="intro". The (.class) selector selects elements with a specific class attribute. To select elements with a specific class, write a period (.) character, followed by the name of the class.

Example

```
<html>
<head>
<style>
.center {
    text-align: center;
     color: red;
 }
 </style>
 </head>
 <body>
 <h1 class="center">Red and center-aligned heading</h1>
  <p class="center">Red and center-aligned paragraph.</p>
 </body>
 </html>
```

Group Selectors with examples

```
<style>
h1, h2, p {
    text-align: center;
    color: red;}
</style>
<body>
<h1>Hello World!</h1>
<h2>Smaller heading!</h2>
<p>This is a paragraph.</p>
</body>
```

External Style Sheet:

With an external style sheet, you can change the look of an entire website by changing just one file. Each page must include a reference to the external style sheet file inside the <link> element. The <link>element goes inside the <head> section. The format for this is shown as below:

```
<head>
<link rel="stylesheet" type="text/css" href="stylefile.css"> </head>
```

An external CSS can be written in any text editor. The file should not contain any html tags. The style sheet file must be saved with a .css extension.

Example

```
body {
    background-color: darkslategrey;
    color: azure;
    font-size: 1.1em;
        }
h1 {
    color: coral;
     }
#intro {
    font-size: 1.3em;
        }
#first {
    font-size: 1.3em;
    color: coral;
        }
.colorful {
    color: orange;
          }
```

styles.css

```
<html>
<head>
<title>My Example</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<h1>External Styles</h1>
<p id="intro">Allow you to define
styles for the whole website.</p>
<p class="colorful">This has a style
applied via a class.</p>
<p id="first">Allow you to define
styles for the whole website.</p>

</body>
</html>
```

page1.html

Create the Style Sheet

Type CSS code into a plain text file, and save with a .css extension (for example, styles.css).

Here's an example of some CSS code.

1- Link to the Style Sheet from the HTML Documents

Add the following code between the <head></head> tags of all HTML documents that you want to reference the external style sheet. This code uses the HTML <link> element to link to the external style sheet.

<link rel="stylesheet" href="styles.css">

2- Style .css

```
body {
    background-color: darkslategrey;
    color: azure;
    font-size: 1.1em;
}
h1 {
    color: coral;
}
#intro {
    font-size: 1.3em;
}
.colorful {
    color: orange;
        }
```

3- Link

```
<html>
  <head>
    <title>My Example</title>
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <h1>External Styles</h1>
    <p id="intro">Allow you to define styles for the whole
    website.</p> <p class="colorful">This has a style applied
    via a class.</p>
  </body>
</html>
```

The First course 2024-2025
Sixth Lecture

Introduction to JavaScript

JavaScript was released by Netscape and Sun Microsystems in 1995. However, JavaScript is not the same thing as Java.

JavaScript is the most popular scripting language on the internet, and works in all major browsers, such as Internet Explorer, Firefox, Chrome, Opera, and Safari.

What is JavaScript?

- JavaScript was designed to add interactivity to HTML
- pages JavaScript is a scripting language
- A scripting language is a lightweight programming language
- JavaScript is usually embedded directly into HTML pages
- JavaScript is an interpreted language (means that scripts execute without preliminary compilation)

Put a JavaScript into an HTML page

To insert a JavaScript into an HTML page, we use the <script> tag. Inside the <script> tag we use the type attribute to define the scripting language.

```
<script type="text/JavaScript">
...some JavaScript
</script>
```

The **document.write** command is a standard JavaScript command for writing output to a page.

By entering the document.write command between the <script> and </script> tags, the browser will recognize it as a JavaScript command and execute the code line. In this case the browser will write Hello World! to the page

```
<script type="text/javascript">
document.write("Hello World!");
</script>
```

The example below shows how to add HTML tags to the JavaScript:

```
document.write("<h1>Hello World!</h1>");
```

JavaScript is a sequence of statements to be executed by the browser.

JavaScript is Case Sensitive

Unlike HTML, JavaScript is case sensitive - therefore watch your capitalization closely when you write JavaScript statements, create or call variables, objects and functions.

JavaScript Statements

A JavaScript statement is a command to a browser. The purpose of the command is to tell the browser what to do, JavaScript is a sequence of JavaScript statements. Each statement is executed by the browser in the sequence they are written.

This example will write a heading and two paragraphs to a web page:

```
<script type="text/javascript">
document.write("<h1>This is a heading</h1>");
document.write("<p>This is a paragraph.</p>");
document.write("<p>This is another paragraph.</p>");
</script>
```

JavaScript Blocks

JavaScript statements can be grouped together in blocks. Blocks start with a left curly bracket {, and ends with a right bracket }, the purpose of a block is to make the sequence of statements execute together.

This example will write a heading and two paragraphs to a web page

```
<script type="text/javascript">
{
document.write("<h1>This is a heading</h1>");
document.write("<p>This is a paragraph.</p>");
document.write("<p>This is another paragraph.</p>");
}
</script>
```

JavaScript Variables

Variables in JavaScript are much like those in any other language; you use them to hold values in such a way that the values can be explicitly accessed in different places in the code.

Rules for JavaScript variable names:

Variable names are case sensitive (y and Y are two different variables) Variable names must begin with a letter.

Table JavaScript keyword

break	else	new	var
case	finally	return	void
catch	for	switch	while
continue	function	this	with
default	if	throw	

Creating JavaScript Variables

Creating variables in JavaScript is most often referred to as "declaring" variables. You can declare JavaScript variables with the **var** keyword:

```
var x;
var carname;
```

However, you can also assign values to the variables when you declare them:

```
var x=5;
var name="Alex";
```

After the execution of the statements above, the variable **x** will hold the value **5**, and **name** will hold the value **Alex**.

When the string is used in a dialog window, the escape sequence, **\n**, is interpreted literally, and a newline is published:

```
var string_value = "This is the first line\nThis is the second line";
```

This results in:

This is the first line

This is the second line

JavaScript Arithmetic Operators

Arithmetic operators are used to perform arithmetic between variables and/or values.

Given that **y=5**, the table below explains the arithmetic operators:

Operator	Description	Example	Result
+	Addition	x=y+2	x=7
-	Subtraction	x=y-2	x=3
*	Multiplication	x=y*2	x=10
/	Division	x=y/2	x=2.5
%	Modulus (division remainder)	x=y%2	x=1
++	Increment	x=++y	x=6
--	Decrement	x=--y	x=4

JavaScript Assignment Operators

Assignment operators are used to assign values to JavaScript variables.

Given that **x=10** and **y=5**, the table below explains the assignment operators:

Operator	Example	Same As	Result
=	x=y		x=5
+=	x+=y	x=x+y	x=15
-=	x-=y	x=x-y	x=5
*=	x*=y	x=x*y	x=50
/=	x/=y	x=x/y	x=2
%=	x%=y	x=x%y	x=0

The + Operator Used on Strings

The + operator can also be used to add string variables or text values together.

To add two or more string variables together, use the + operator.

```
txt1="What a very";
txt2="nice day";
```

txt3=txt1+txt2;

After the execution of the statements above, the variable txt3 contains "What a verynice day".

```
txt1="What a very";
txt2="nice day";
txt3=txt1+" "+txt2;
```

After the execution of the statements above, the variable txt3 contains:

"What a very nice day"

Adding Strings and Numbers

The rule is: If you add a number and a string, the result will be a string!

Example

```
x=5+5;
document.write(x);

x="5"+"5";
document.write(x);

x=5+"5";
document.write(x);

x="5"+5;
document.write(x);
```

JavaScript Comparison and Logical Operators

Comparison and Logical operators are used to test for true or false.

Given that **x=5**, the table below explains the comparison operators:

Operator	Description	Example
==	is equal to	x==8 is false
===	is exactly equal to (value and type)	x===5 is true x==="5" is false

Logical Operators

Logical operators are used to determine the logic between variables or values.

Given that **x=6 and y=3**, the table below explains the logical operators:

Operator	Description	Example
&&	And	(x < 10 && y > 1) is true
||	Or	(x==5 || y==5) is false
!	Not	!(x==y) is true

Conditional (Ternary) Operator

JavaScript also contains a conditional operator that assigns a value to a variable based on some condition.

Syntax

```
variablename=(condition)?value1:value2
```

Example

```
var age =16;
voteable = (age < 18) ? "Too young":"Old enough";
```

The output for this example is:
Too young

The First course 2024-2025

Seventh Lecture

Conditional Statements

Very often when you write code, you want to perform different actions for different decisions. You can use conditional statements in your code to do this.

If Statement

Use the if statement to execute some code only if a specified condition is true.

Syntax

```
if (condition)
  {
  code to be executed if condition is true  }
```

Note that if is written in lowercase letters. Using uppercase letters (IF) will generate a JavaScript error!

```
<script type="text/javascript">
//Write a "Good morning" greeting if
//the time is less than 10
var d=new Date();
var time=d.getHours();
if (time<10)
  {
  document.write("<b>Good morning</b>");
} </script>
```

Notice that there is no ..else.. in this syntax. You tell the browser to execute some code **only if the specified condition is true**.

if...else Statement

Use the if....else statement to execute some code if a condition is true and Another code if the condition is not true.

Syntax

```
if (condition)

  {
  code to be executed if condition is true
  }

else
  {

  code to be executed if condition is not true
  }
```

if...else if ...else Statement

Use the if ....else if...else statement to select one of several blocks of code to be executed.

Syntax

```
if (condition1)
  {
  code to be executed if condition1 is true
  }
else if (condition2)
  {
  code to be executed if condition2 is true
  }
else
  {
  code to be executed if condition1 and condition2 are not true
  }
```

The JavaScript Switch Statement

Syntax

```
switch(n)
{
case 1:
  execute code block 1
  break;
case 2:
  execute code block 2
  break;
default:
  code to be executed if n is different from case 1 and 2
}
```

This is how it works: First we have a single expression *n* (most often a variable), that is evaluated once. The value of the expression is then compared with the values for each case in the structure. If there is a match, the block of code associated with that case is executed. Use **break** to prevent the code from running into the next case automatically.

Example

```
<script type="text/javascript">
 //Monday=1, Tuesday=2, etc.
var Day=1;
 switch (Day)
{
case 1:
  document.write("Monday ");
   break;
 case 5:
   document.write("Finally Friday");
   break;
 case 6:
   document.write("Super Saturday");
   break;
 case 0:
   document.write("Sleepy Sunday");
   break;
 default:
   document.write("I'm looking forward to this weekend!");
 }
```

</script>

The break Statement

The break statement will break the loop and continue executing the code that follows after the loop (if any).

The continue Statement

The continue statement will break the current loop and continue with the next value

JavaScript Popup Boxes

JavaScript has three kind of popup boxes: Alert box, Confirm box, and Prompt box.

Alert Box

An alert box is often used if you want to make sure information comes through to the user.

When an alert box pops up, the user will have to click "OK" to proceed.

Syntax

```
alert("sometext");
```

```
<html>
<body>
<h1>Demo: alert()</h1>
<script>
        alert("This is alert box!"); // display string message

        alert(100); // display number

        alert(true); // display Boolean
</script>
</body>
</html>
```

Confirm Box

A confirm box is often used if you want the user to verify or accept something.

When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed.

If the user clicks "OK", the box returns true. If the user clicks "Cancel", the box returns false.

Syntax

```
confirm("sometext");
```

```
<html>
<body>
        <h1>Demo: confirm()</h1>

  <script>
                 var userPreference;

                if (confirm("Do you want to save changes?") == true)
                        { userPreference = "Data saved
                successfully!"; }
                        else {
                        userPreference = "Save Canceled!";
                }

                document.write( userPreference);

    /script>
</body>
</html>
```

Prompt Box

A prompt box is often used if you want the user to input a value before entering a page.

When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.

If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.

Syntax

prompt("text","defaulttext");

Parameter	Type	Description
text	String	Required. The text to display in the dialog box
defaultText	String	Optional. The default input text

```
<html>
<body>
        <h1>Demo: prompt()</h1>

        <script>
                var tenure = prompt("Please enter preferred tenure in years", "15");

                document.write("You have entered " + tenure + " years");
   </script>
</body>
</html>
```

What output of this code

```
<script>

var name = prompt("Please correct your e-mail address:", "un@edu.iq");
document.write("Your e-mail address is ", name);

</script>
```

parseInt() or **parseFloat()**

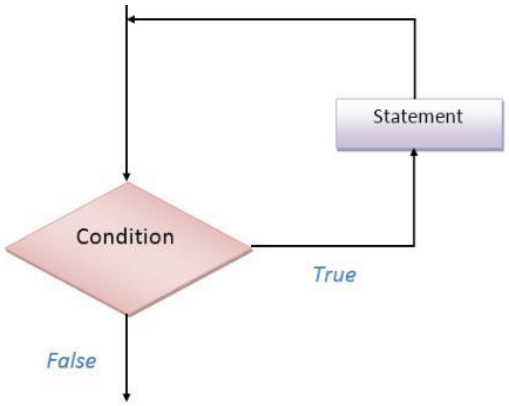
are functions in JavaScript which can help you convert the values into integers or floats respectively.

```
var x = prompt("Enter a Value", "0");
 var y = prompt("Enter a Value", "0");
 var num1 = parseInt(x);
```

var num2 = parseInt(y);

How to use Loop?

Loops are useful when you have to execute the same lines of code repeatedly, for a specific number of times or as long as a specific condition is true. Suppose you want to type a 'Hello' message 100 times in your webpage. Of course, you will have to copy and paste the same line 100 times. Instead, if you use loops, you can complete this task in just 3 or 4 lines.



Different Types of Loops

There are mainly four types of loops in JavaScript.

- for loop
- for/in a loop (explained later)
- while loop
- do…while loop

➢ For loop

Syntax:

```
for(statement1; statement2; statment3)
{
lines of code to be executed
}
```

- The statement1 is executed first even before executing the looping code. So, this statement is normally used to assign values to variables that will be used inside the loop.
- The statement2 is the condition to execute the loop.
- The statement3 is executed every time after the looping code is executed.

```
<script type="text/javascript">
        document.write("<b>Using for loops </b><br />");
        for (var i=0;i<9;i++)
        {
        document.write("*" + "<br />");
        }
    </script>
```

➢ While loop

Syntax:

```
while(condition)
{
lines of code to be executed
}
```

The "while loop" is executed as long as the specified condition is true. Inside the while loop, you should include the statement that will end the loop at some point of time.

```
<script type="text/javascript">
    document.write("<b>Using while loops </b><br />");      var
    i = 0, j = 1, k;
    document.write(" series less than 40<br />");

    while(i<40)
    {
    document.write(i + "<br />");
    k = i+j;
    i = j;
    j = k;
    }
</script>
```

➢ do…while loop

Syntax:

```
do
{  block of code to be executed
} while (condition)
```

The do…while loop is very similar to while loop. The only difference is that in do…while loop, the block of code gets executed once even before checking the condition