Lecture 7

Introduction to JavaScript

Introduction

- **Types of Scripting**
  - Client Side Scripting
    - The code required to process user-input is downloaded and compiled by the browser.
    - An example of a client-side interaction is form validation.
  - Server Side Scripting
    - Information is sent to a server to be processed With server-side scripting, completing an activity involves sending information to another computer (server) across the internet.
    - The server then runs a program that process the information and returns the results, typically a webpage.
    - Search engines use server-side processing.
    - When a keyword is sent, a program on a server matches the word or phrase entered against an index of website content. (To complete the same search as a client-side process would require the browser to download the entire search engine program and index.)

DHTML

- **Dynamic HTML**
  - Client Side Scripting is an important part of the Dynamic HTML (DHTML) concept.
  - Enabling web pages to be scripted;
    - that is, to have different and changing content depending on user input, environmental conditions (such as the time of day), or other variables.
  - DHTML allows scripting languages to change variables in a web page's definition language.
  - which in turn affects the look and function of otherwise "static" HTML page content, after the page has been fully loaded and during the viewing process.

DHTML Uses & Features

- **Animate** text and images in their document, independently moving each element from any starting point to any ending point, following a predetermined path or one chosen by the user.
- **Embed** a ticker that automatically refreshes its content with the latest news, stock quotes, or other data.
- **Validate input:** Use a form to capture user input, and then process and respond to that data without having to send data back to the server.
- **Other Options:** Include rollover buttons or drop-down menus.
- **Dynamic Styles** using CSS, we can stylize the html dynamically.
- **Data Bindings :** you can bind individual elements in your document to data from another source, such as a database or comma-delimited text file.
  - When the document is loaded, the data is automatically retrieved from the source and formatted and displayed within the element.

Introduction to Java Script

- **JavaScript is:**
  - ➲JavaScript is a lightweight, interpreted programming language
  - ➲Designed for creating network-centric applications
  - ➲Complementary to and integrated with Java
  - ➲Complementary to and integrated with HTML
  - ➲Open and cross-platform
- **Syntax:**
  - ➲A JavaScript consists of JavaScript statements that are placed within the **<script>... </script>** HTML tags in a web page.

Synatx

- **Methods of using Java script:**
  - ➲**1.** You can place the <script> tag containing your JavaScript anywhere within you web page.
  - ➲**2.** but it is preferred way to keep it within the <head> tags.
  - ➲**3.** Or you can make an external file of script.
    - Just like we made external css file and linked it with html page.

The <script> tag alert the browser program to begin interpreting all the text between these tags as a script. So simple syntax of your JavaScript will be as follows.

```
<script  language="javascript" type="text/javascript">
  JavaScript code
------------
------------
</script>
```

Example 1

- **Writing script in <body>**

```
<html>
<body>
<script language="javascript" type="text/javascript">
<!--
   document.write(“Lecture 7 : Java Scripting")
//-->
</script>
</body>
</html>
```

Above code will display following result: Lecture 7 : Java Scripting

Example 2

- **Displaying Alert box.**

```
<script>
alert("My First JavaScript");
</script>
```

- **Writing to The Document Output**

```
<script>
document.write("<p>My First JavaScript</p>");
</script>
```

  - If you execute **document.write** after the document has finished loading, the entire HTML page will be overwritten.

Java Script Variables

- **JavaScript Variables Deceleration Rules**
  - As with algebra, JavaScript variables can be used to hold values (x=5) or expressions (z=x+y).
  - Variable can have short names (like x and y) or more descriptive names (age, sum, totalvolume).
    - Variable names must begin with a letter
    - Variable names can also begin with $ and _ (but we will not use it)
    - Variable names are case sensitive (y and Y are different variables)
  - You declare JavaScript variables with the "**var**" keyword:
  - Syntax:

    var name;

    var marks;

    ```
    var name="Doe", age=30, job="carpenter";
    ```

    - Multiple declarations:
    - Assigning values; marks="10";

Data Types (1)

- **String, Number, Boolean, Array, Object, Null, Undefined.**
  - A **string** can be any text inside quotes. You can use simple or double quotes: var name="Web engineering"
  - You can access each characters in a string with **[*position*]**
    - **var character=carname[7];**
  - JavaScript has only **one type of numbers**. Numbers can be written with, or without decimals.
    - var x1=34.00; //Written with decimals
      var x2=34; //Written without decimals
    - var y=123e5; // 12300000
      var z=123e-5; // 0.00123
  - **Booleans** can only have two values: true or false.
    - var x=true
      var y=false

Data Types (2)

- **Creating Arrays**
  - The following code creates an Array called cars:
    - var cars=new Array();
      cars[0]="Saab";
      cars[1]="Volvo";
      cars[2]="BMW";
      - OR
    - var cars=new Array("Saab","Volvo","BMW");
      - OR
    - var cars=["Saab","Volvo","BMW"];

Data Types (3)

- **Objects**
  - An object is delimited by curly braces.
  - Inside the braces the object's properties are defined as name and value pairs (name : value). The properties are separated by commas.
  - SYNTAX:
    - `var person={ firstname:"John", lastname:"Doe", id:5566 };`
- **Accessing Object Properties**
  - **name=person.lastname;**
    - **OR**
  - **name=person["lastname"];**

Conditions using if-else & switch

- **Syntax if-else and switch**

```
if (condition)
   {
   code to be executed if condition is true
   }
else
   {
   code to be executed if condition is not true
   }
```

```
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
}
```

Functions

- A function is a block of code that executes only when you tell it to execute.
  - It can be when an **event** occurs, like when a user **clicks** a button, or from a call within your script, or from a call within another function.
- Functions can be placed both in the <head> and in the <body> section of a document, just make sure that the function exists, when the call is made.
- SYNTAX:
  - **function *functionname*(arg1,arg2)**
    {
    some code here….
    }
- Returning values from functio

```
function myFunction()
{
var x=5;
return x;
}
```

Example 3

- **Using Functions**

```
<html>
<head>
<script>
function displayDate()
{
document.getElementById("demo").innerHTML=Date();
}
</script>
</head>
<body>

<h1>Introduction to JavaScript</h1>
<p id="demo">This is a paragraph.</p>

<button type="button" onclick="displayDate()">Display Date</button>

</body>
</html>
```

Important Notes

- JavaScript ignores **spaces, tabs, and newlines** that appear in JavaScript programs.
- Semicolons are **Optional**. Allows you to omit this semicolon if your statements are each placed on a separate line.
- JavaScript is a **case-sensitive** language. AGE, Age, age….
- **Comments**: /*…….*/ any thing inside them.
- Declaring Variable types using "**new**" key word.

```
var carname=new String;
var x=      new Number;
var y=      new Boolean;
var cars=   new Array;
var person= new Object;
```

Reserve Words for JavaScript

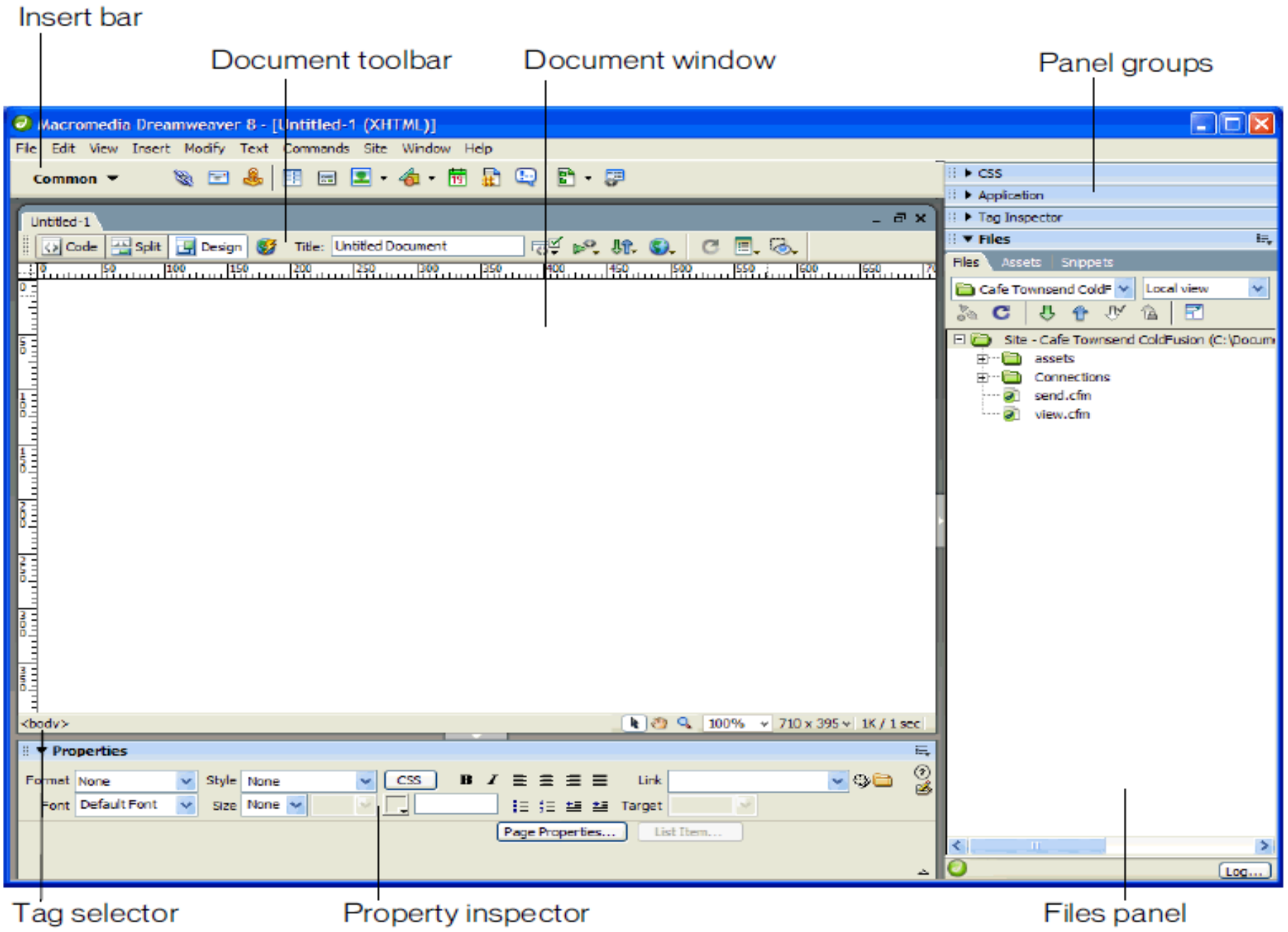
abstract	else	instanceof	switch
boolean	enum	int	synchronized
break	export	interface	this
byte	extends	long	throw
case	false	native	throws
catch	final	new	transient
char	finally	null	true
class	float	package	try
const	for	private	typeof
continue	function	protected	var
debugger	goto	public	void
default	if	return	volatile
delete	implements	short	while
do	import	static	with
double	in	super	

Introduction

Design Using Dream weaver

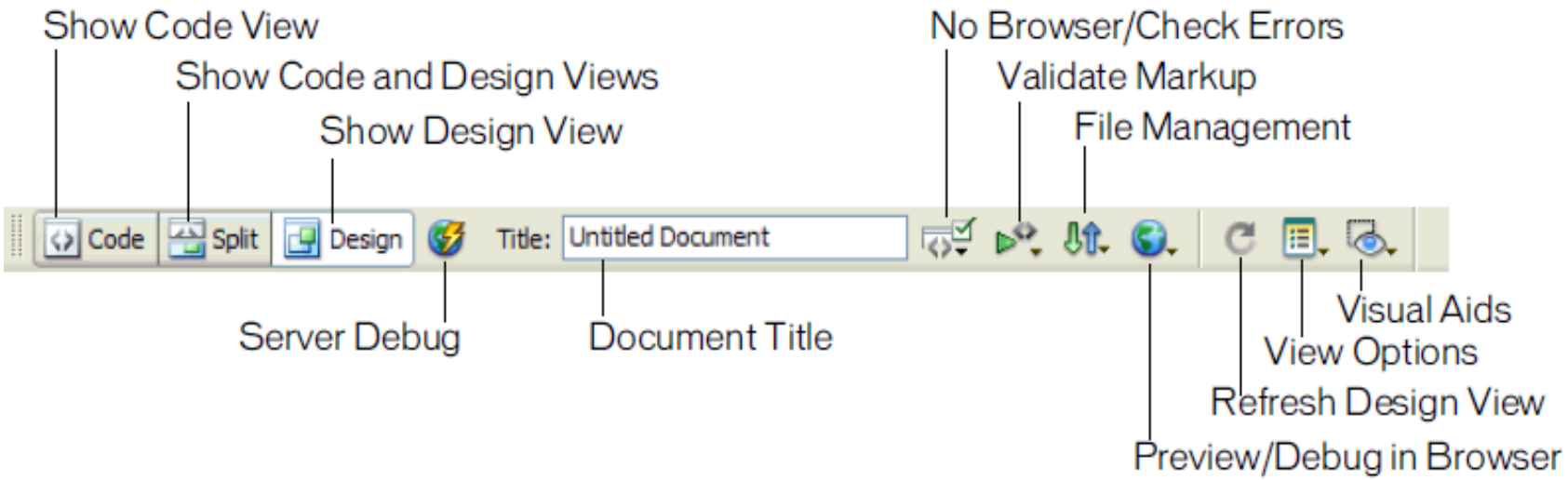
macromedia
DREAMWEAVER 8
Getting Started with Dreamweaver

Introducing Layout

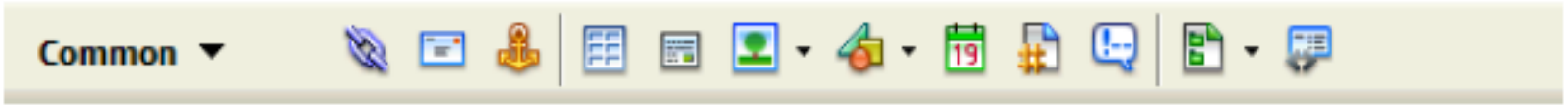


Tools & Features

- **Code & Design views**

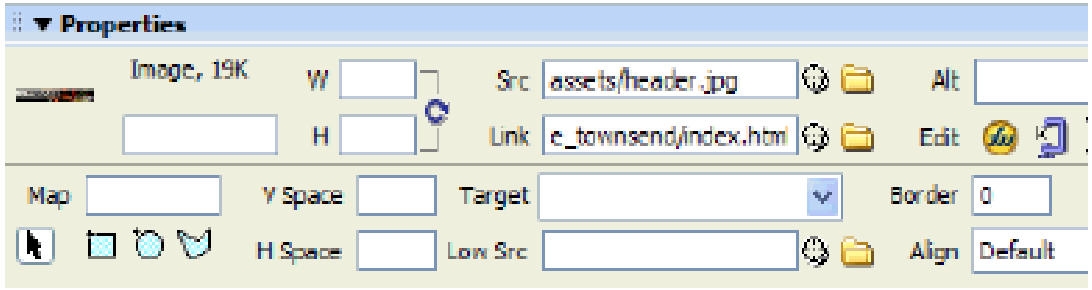


- **Insert Tool bar**



Tools & Features

- **Property Inspector**



  - The contents of the Property inspector vary d
    selected.

- **CSS & HTML Pannel**

Assignment # 2

- **Think of some topic to design a web form.**
- **Using Dreamweaver ,design the form.**
- **Your form should contain all types of form elements.(textbox, password, lists, radio buttons, buttons, text area, picture, file browser button.)**
- **When you click on submit button, system should validate your name. if name is correct, then it should display message that page is saved.**
  - ➲ If name is not correct, then message should be displayed.
- **Design form layout using your common sense.**
- **Also attach external style sheet with it to stylize it.**