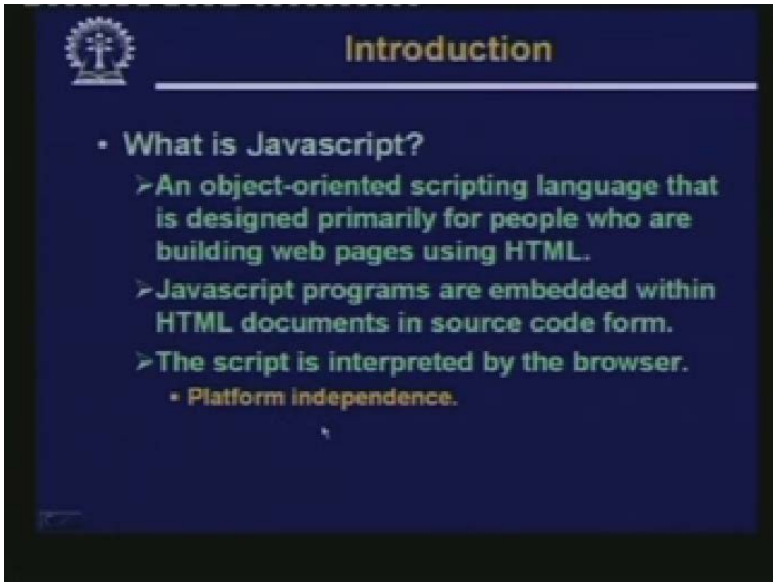
Internet Technology
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture No #25
Javascript – Part I

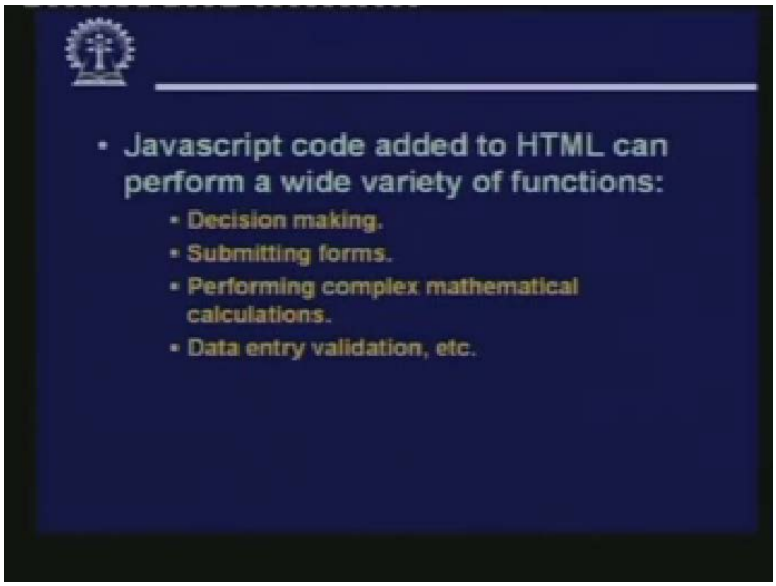
Today will be talking about a language called Javascript. Now in context of the internet you know we have discussed so many different languages, so many different technologies. JavaScript in yet another language which finds its place in the list of internet technologies and tools that are used we will see through examples why Javascript is so important. And in what ways we can use this to make our websites or web pages more attractive and interactive. Now first let us see what language JavaScript basically is.

(Refer Slide Time: 01:40)



Now Javascript is essentially a scripting language. Script language in the sense that you can write small code snippets in this language which will be executed as a part of a bigger html document and Javascript is object-oriented. Javascript is an object-oriented scripting language and as I have this language is designed primarily for the purpose of building web pages. So as you know HTML is the primary technology through which we develop. Our web pages Javascript can be considered as some kind of supplement. We can use Javascript along HTML to produce or create better web pages. Now Javascript programs are essentially embedded within HTML documents and the point to note is that they exist within HTML document in source code form. And when a document is downloaded on to a browser the document is downloaded along with the embedded Javascript source code which gets interpreted by a Javascript interpreter which is present within the browser itself. Now this helps in making Javascript code platform independent. Now essentially what we mean is that Javascript exists as a part of bigger HTML document and when a document gets downloaded on the browser there is a Javascript interpreter which is resident within the browser itself that interpreter helps in the execution of Javascript segments of the code.

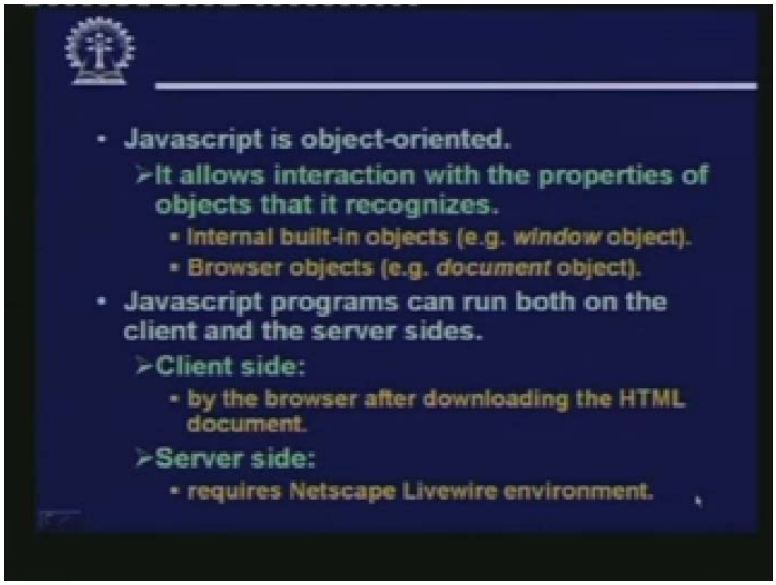
(Refer Slide Time: 03:49)



Now with respect to applications when we add Javascript to HTML naturally we are trying to perform some kind of functionality. Over and above what is allowed or can be done by pure HTML. Now we shall see through examples that there are several ways or several applications where we can use Javascript to our advantage. Like for example you can create or you can have some simple decision making. Based on an interactive user response depending on what the user is types in or clicks on the mouse the Javascript program can be instructed to take an appropriate action. Secondly you can make form submission more flexible and more powerful using Javascript in built mechanism for submitting forms and responding to various kinds of events you can have these.

Thirdly what you cannot do in HTML is some kind of mathematical or logical calculations some string manipulation. Javascript provides functionality to do that, you can make simple calculations and depending on the result of the calculation you can produce contents to be displayed in a suitable form on the HTML page. You may be aware of many websites which performs automatic online data validation while you are entering some data in a form. For example when you are entering a telephone number, the system will immediately raise an alarm or an error if you try to enter some alpha numeric value in the field of the telephone number. This kind of simple data validation can also be done by Javascript.

(Refer Slide Time: 06:01)



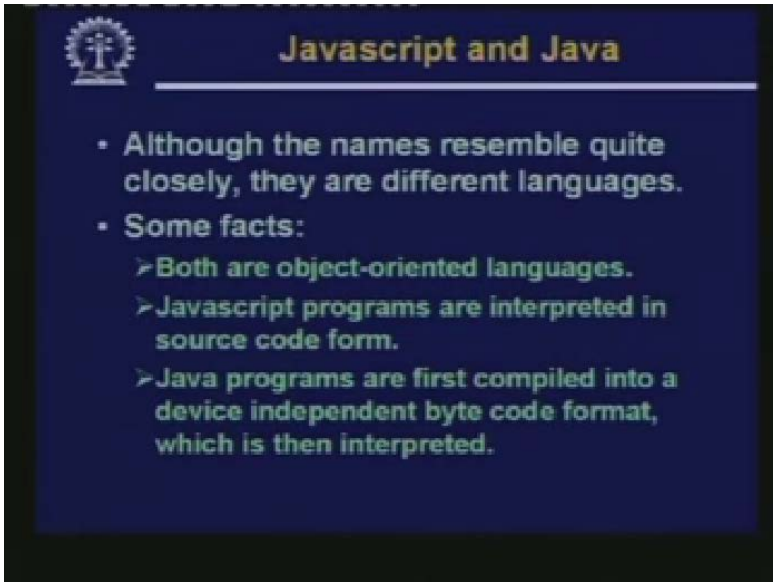
Now as we have said the language Javascript is basically object-oriented, this is an object-oriented language. Now since it is an object-oriented language, program is considered to consist of a set of objects and their interactions. So basically Javascript allows one to write a program which can access the various properties of the different objects that are recognized by the language and accordingly produce or provide some functionality. There are a number of different objects in JavaScript. Some of which are as follows there are some internal built-in objects like window. Window is the name of an object. So whenever you display a document on a window or a frame there can be multiple frames within a given window. So each of the frames or the window itself there considered as the instances of the window object.

So, when you have a window object there are number of methods, number of properties that you can associate with window objects to changed way it looks to change the way. It will function when you are trying to use it in a certain context and so on. There can be a browser objects also. See when some items are displayed on the browser there can various things you want to display. For example document is an object which is considered to be a browser object and when you are displaying a document there are some document related features like color, font side, foreground, background colours, font size styles, etcetera. So if you associate these properties along with a document object the document object will be rendered or displayed in an appropriate way.

Now Javascript programs technically can run either on the browser. That means on the client side or on the side of the web server. But the point to note is that most Javascript code. Today in the internet scenario are written to be run on the client side. All the browsers today are capable of executing embedded Javascript code. So a browser can always download an HTML document with embedded Javascript code and execute them on the browser itself. Now in addition to this you can also have server side Javascript program being executed. However this is not automatic you need some special functionality. On the server side to execute Javascript code, for example Netscape Livewire environment is one such program or

tool which you can install on the server side to have Javascript based server side scripts written and executed.

(Refer Slide Time: 09:29)

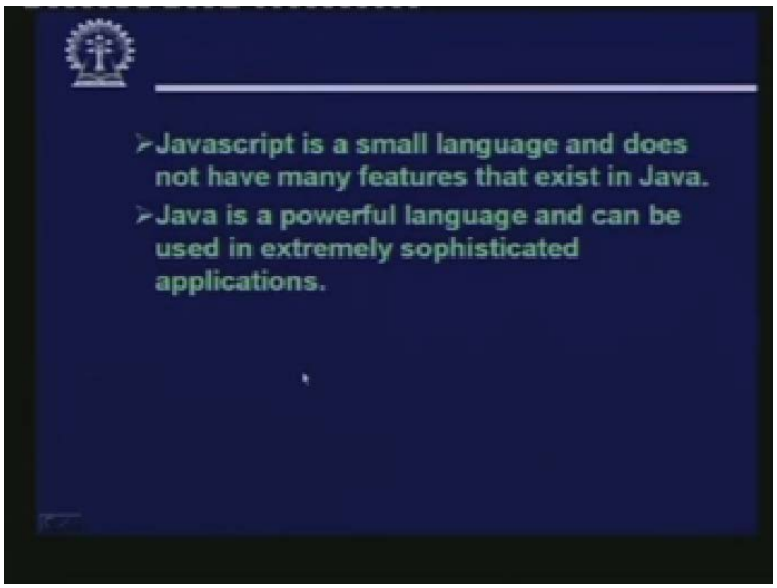


Now form the name it may imply that Javascript or Java are same or very similar. Because they contain the common word java in them. But is should be remembered that although the names are similar they are distinctly different languages. There are some similarities though there are similarities both are object-oriented languages. But the main difference between them the first main difference is the way. They are executed Javascript programs are interpreted in source code form. So what we mean here is that Javascript programs reside in the source code form on the web browser. On the web server as well as the web browser as they download it. So after downloading on the web browser these Javascript source codes get executed. So the Javascript interpreted is part of the browser which can directly interpret Javascript source code. In contrast if you look at java programs, java programs first need to be compiled in to so call byte code. There is an interpret format of java program called byte code which needs to be converted in to before you can attempt to run a Java program. And whenever you have a Java programs linked or embedded as part of an HTML document it will be these byte codes which will be linked.

And with respect to the browser you can download an HTML document along with the Java programs called an applet which is link from the HTML code and the browser must have an interpreter add-on install for Java for interpreting the Java byte code. Now you try to see in this way both Javascript and Java are made platform independent. Javascript is made platform independent by having the source code form existing as a part of the document and by adding the functionality of interpreting Javascript with the browser itself. So it maybe Netscape, in maybe Mozilla, it maybe Internet Explorer. Whatever browser you use whatever platform you use Windows, Macintosh, UNIX, Linux. If the browser has the functionality, it can always download Javascript code and execute them. Similarly Java platform independence is achieved by translating the Java program in Java byte code and then the Java byte codes are

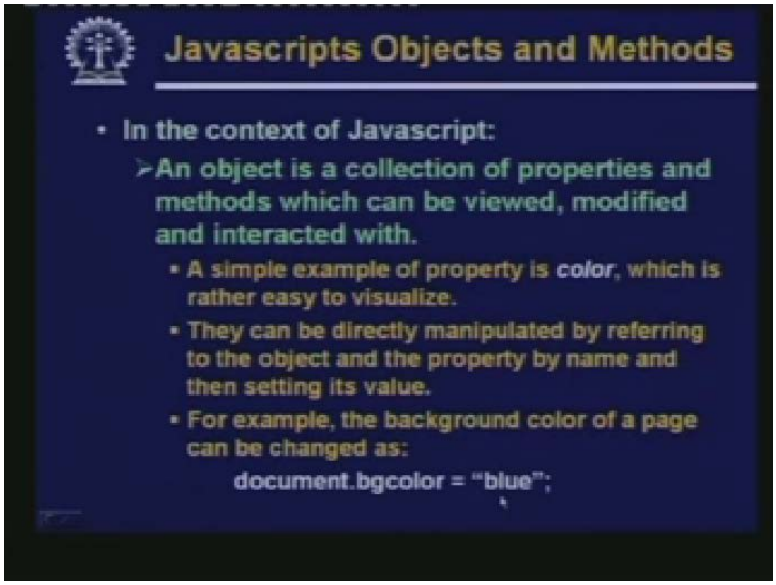
downloaded on the browsers and interpreted out. There will be executing the Java byte codes. So this is how Javascript and Java then execute.

(Refer Slide Time: 12:56)



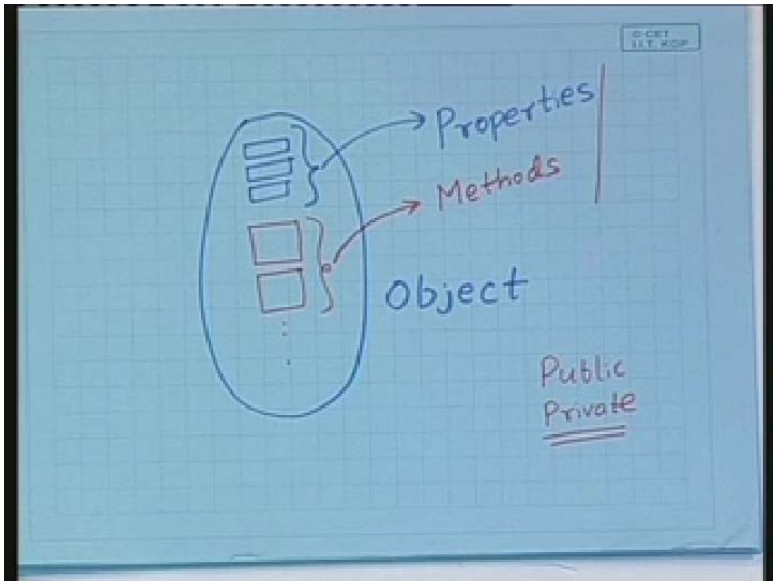
There are a few other differences. Javascript is primarily meant to enhance the features of the HTML language. So it is basically a small language and does not have many features. Most of the features that exist in Javascript have something to do with, how the documents are displayed how user interactivity can be implemented and so on. In contrast Java is a standalone language it is quite powerful and it can be used in extremely sophisticated applications. So in that sense Java is a complete language it can be considered as a substitute to existing standard languages like C or C++ with respect to Javascript. Let us now see how the concept of objects and methods can be used and processed and accessed as part of code snippets.

(Refer Slide Time: 13:55)



First let us try to understand what is an object in the context of a Javascript program. Now an object essentially is a collection of properties and methods. Now when you say properties and methods this is something to do with an object. Let us try to understand.

(Refer Slide Time: 14:33)



Suppose this is an object. So within the object we can have several variables which can be accessed publically by other variables. These variables we can call as the properties of the object. In addition we can have several functions within the object itself. These functions in object-oriented terminology are called methods. Now both these properties and the methods they can be either public or private. Now if the properties or the methods are public they can be accessed by other methods other objects. But if they are private they can be accessed by the methods only belonging to that particular object. Now the concept of objects and

properties in Javascript is exactly this there are objects there are some properties there are some methods.

You can write the methods to implement certain functionality which can access or modify some of the properties of the object or it can even call public methods of other objects. So an object is essentially a collection of properties and methods which can be viewed, modified and interacted with viewed modified and interacted. These are important terminologies in the context of Javascript because Javascript is primarily meant for creating documents which will be viewed on the browser. So viewing is an important component of processing when we talk about JavaScript. Now as in simple example a simple example of a property of an object is colour. Suppose you have a document. Document is an example of an object. So colour will be a property of the document you can change the colour of a document.

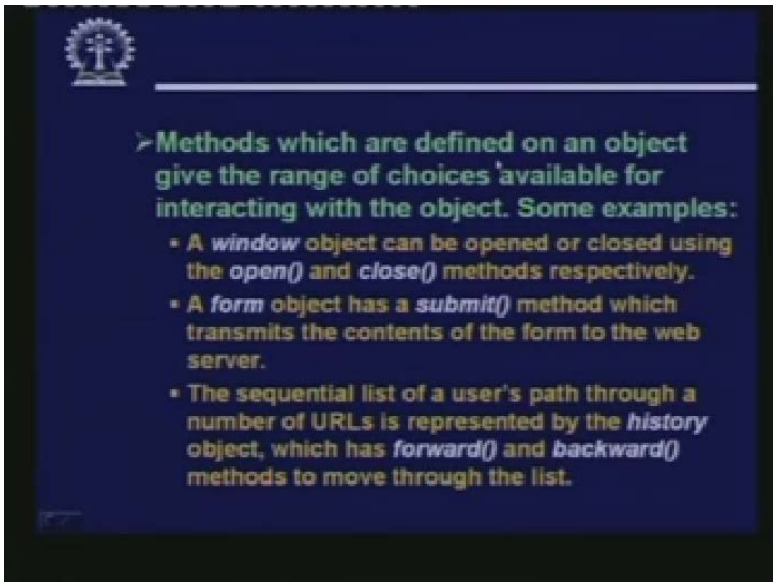
Now when we talk about the colour of a document it can be background color of foreground color. We can change your access any such color of the document with respect to display they will be immediately affected. So these kinds of properties like color, we can directly manipulate them by referring to object and the property name, as an example document dot bgcolor. This means document is the name of the object, bgcolor is the name of the property of that object and this separating dot means bgcolor is a property of the document object. And when we assign a value blue to this, this will mean that the property bgcolor of the object has changed to blue. So immediately the background color of the display document will change to blue. In this way we can the background color to any desired color red, blue, green, white, whatever you want, similarly foreground color also you can change fgcolor. Similarly so this is how we can change colors.

(Refer Slide Time: 18:39)

Now another example. This example shows me the use of methods. Now as I have said in object-oriented paradigm methods is nothing but functions which carry out certain manipulations on objects. And as a result of the manipulation some properties of the objects might get modified or you can also call or invoke methods of other objects. Now in the

example that we will show here we will see how to invoke one method of a particular object, so that some property or some visual aspects of that object gets modified. Now in this Slide we have an object called document. As I have said the document object refers to any displayed information which gets displayed on either a window or on the frames write is a method. So whenever we invoke write on the object called document like this document dot write as a parameter any string within quotes. What will happen is that whatever string of characters that we output here, that will get outputted on the particular window. See you may be having an HTML document in between somewhere you have this document write command. So whatever is being outputted by this document that output string will be merged by the surrounding HTML document, so as the output you will be getting ultimately an HTML file where some part of it has been outputted by this Javascript method or functions.

(Refer Slide Time: 20:43)



So these methods which we define an objects. Actually they give to users some choices available for interacting with the objects like the methods or the functions. For an object there can be 10 methods available. Now these 10 methods will give the choice or the option to interact with the object in several different ways. You can also change the properties of the objects in several different ways by calling or invoking correct method for that object. Like some examples I have shown here, there is a window object. A window object refers to an object where we have a window within that window certain items are getting displayed. So some methods are there for window. These methods some of these methods are open and close.

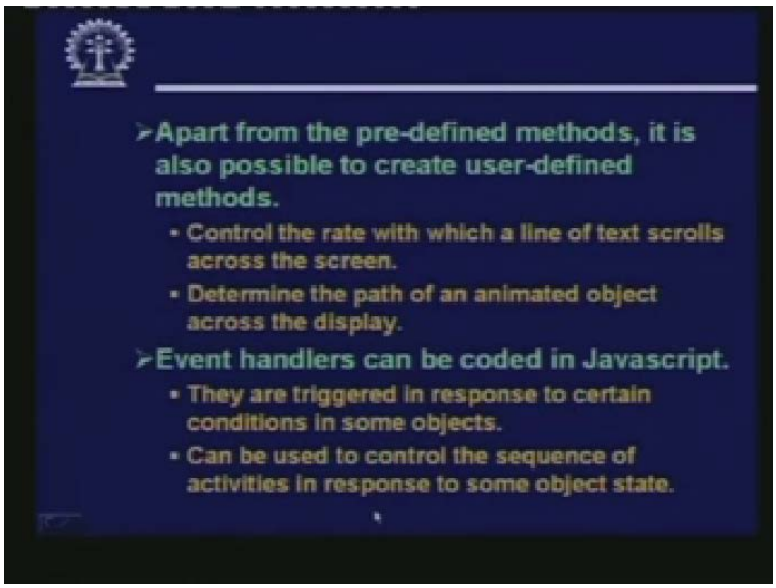
As the names imply when we invoke the open method a window will be created it will be opened and we will see later that whenever we open such a window during opening. We can also provide certain properties of the window like what will be the size of the window in length, in height, whether there will be a title bar, whether there will be scroll bar on the side of the window, whether the window should be resizable or should be of fixed size. You may have seen some windows in some web sites where you cannot resize the window. So that wind has been marked or created as non-resizable. So all these properties you can specify

when you are creating or opening a new window. Similarly when you are done with the window you can close it the window will get closed.

Similarly you can have a form object which you can submit. There is a submit method associated which each form and we you invoke submit. There submit method of the form object would get invoked and the form data would be submitted in response to what you have specified in terms of the Javascript. We will see that we can specify number of different ways in which form can be submitted. We can say that if we submit a form you send the form data back to web server or else if we submit you call a function which is written in Javascript. So you can specify a number of different options. In this way these we can do to provide a number of different functionalities as we shall see later. The third example talks about the history object and under the history object we have methods called forward and backward which can traverse in the forward and backward direction through the history list.

Now the concept of history, for those of you who have used the browser who have quite familiar with accessing internet. You know what this history is when you continue browsing the internet you will be visiting the web pages one after the other. Now at any certain point in time you can press the back button of the browser and you can go back to the previous page. Similarly you can press the forward button and go to the next page actually the list of the pages that have been visited in the access of time. There are stored in some history object list. Now the Javascript methods forward and backward. They serve the purpose of moving one step forward and one step backward along that list in order to move forward and backward in time.

(Refer Slide Time: 24:58)



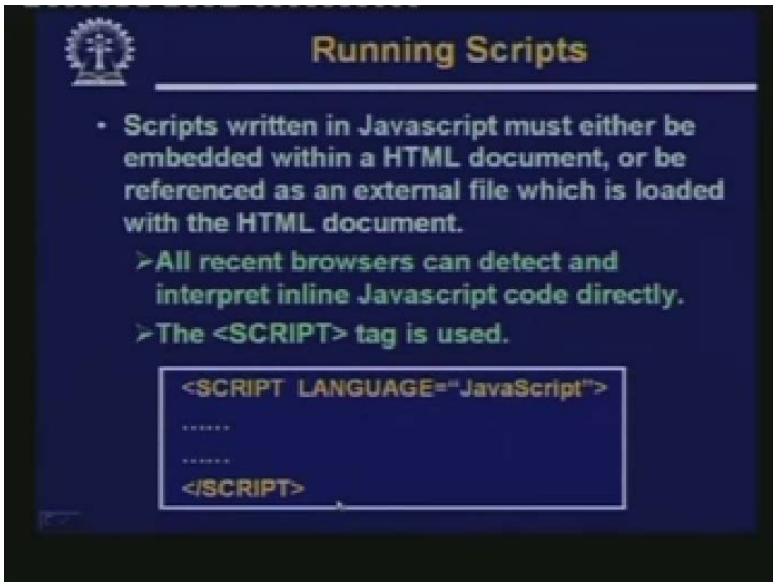
Javascript also allows you to create user-defined methods. The examples that we have taken are all pre-defined methods. But when you write Javascript code snippets in HTML you can write your own functions or methods. And some examples, suppose you want a scrolling text to be displayed on the screen, some text which will be scrolling on the screen or on the bottom bar of the window. Now there you can have a mechanism using which you can

control the rate of scrolling of the text. It can be slow, it can be medium, it can be fast. Now you can have function you can write you own function to control the speed with which the text would be scrolled. Similarly you can have some kind of animation on the screen and your method can control the path or the way the animation would be carried out. The direction or path of the animation may also be the speed of the animation all those things you can control by writing your own function.

Say I am giving a simple example say you are viewing an online album. There are number of pictures you are trying to view. Now the pictures will be appearing one after the other. There can be a function which can control how fast or how slow you can continue viewing the photos right. Now in addition to such methods which are explicitly called you can have event handlers coded in Javascript. Now let us try to explain here what is meant by event handlers. Most of the modern languages support event handling. You see if you look at the conventional methods and functions as it exists in almost all languages. There are some pieces of code which can be called from some other program or from some other method. There are explicitly called and when they are called control gets transferred to the new program and after they finish control comes back right.

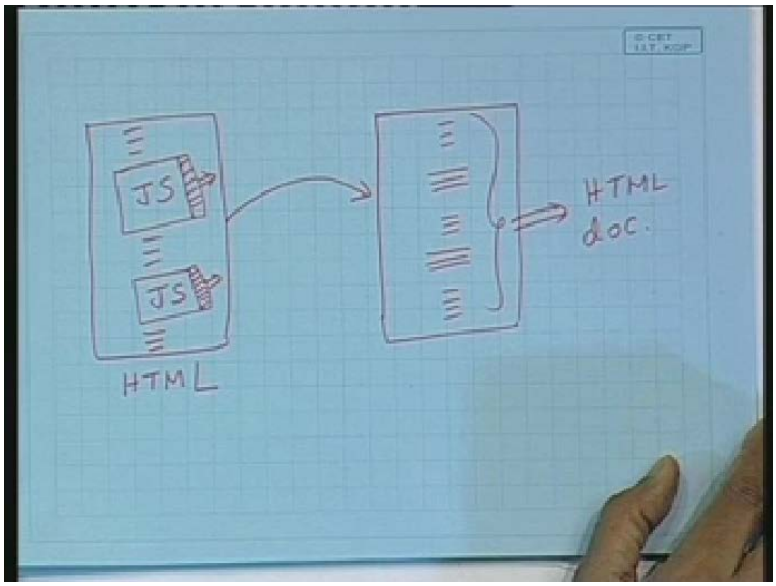
Now in contrast event handler will be something like this. I can tell that if someone clicks the mouse then do this, otherwise you continue whatever you are doing. So this clicking this mouse is defined as an event and I can write a function and I can associate this function along with this event. If this mouse is pressed that is the event and executing this function that is the action. So event and action I can specify as many such associations as I want and whenever event of a particular type occurs the corresponding action can get executed automatically. Javascript allows you to do that. So basically this event handler routines or function or methods. They get triggered in response to certain conditions or events. So as I have said these event handlers can be used to control certain activities in in response to some object state or some user controlled events that have taken place.

(Refer Slide Time: 28:41)



Now let us see how we can actually run a Javascript program as part of an HTML document. The first thing is that there are two ways in which you can embed Javascript code in HTML. The first alternative is Javascript code snippets are explicitly embedded in HTML. Or you can refer Javascript as a link to an external file which will be loaded along with the HTML document. The second option we shall see later, but for the time being we shall see some examples where Javascript code is present as part of the HTML document. And Javascript code will get executed and whatever is outputted by that; that will be merged with the surrounding HTML document. So the final output will be an HTML document without any Javascript the idea is like this.

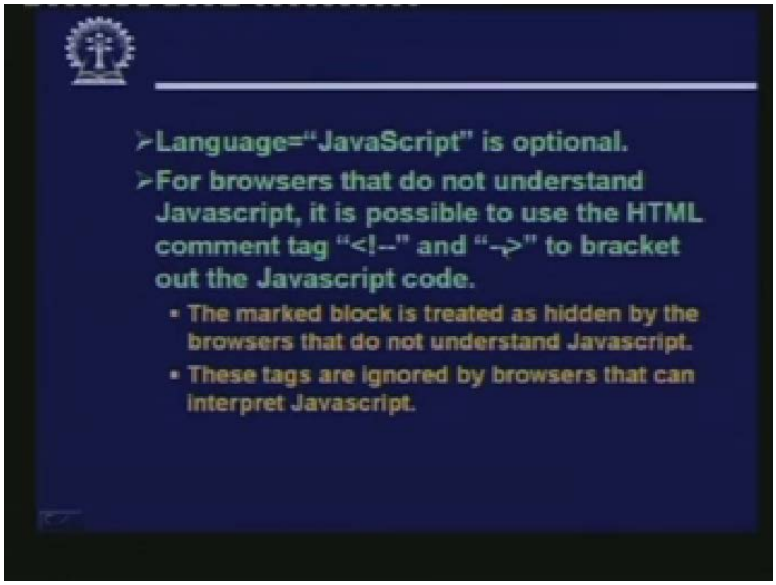
(Refer Slide Time: 29:48)



You have basically HTML document where there are certain code segments which are written in Javascript. There are Javascript code segments in addition to this there are HTML codes around it. Now when this HTML document is downloaded on the browser, what the browser will do is this. First it will execute these Javascript code segments and each of these Javascript code segments will be outputting certain textual information. After processing, so whatever is being outputted by this segment, they will get merged along with the surrounding HTML. So what we get finally is a single HTML document without any Javascript code.

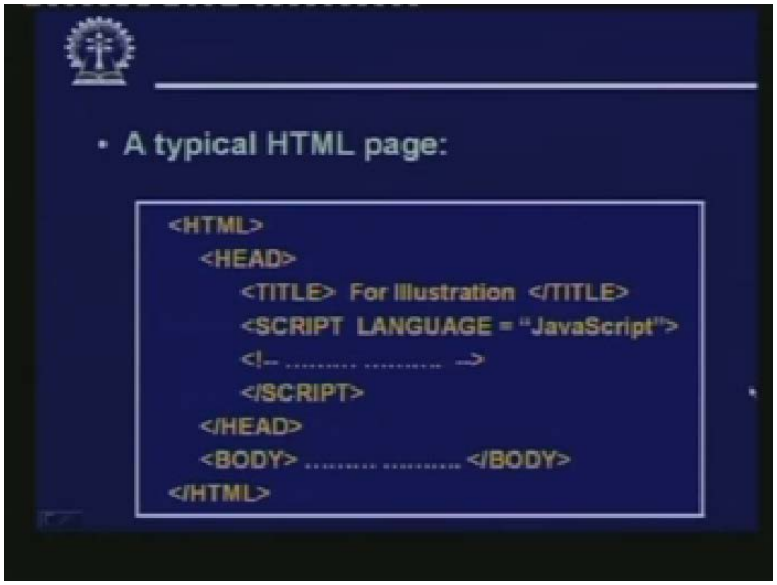
This is the basic idea of behind embedding a Javascript program inside HTML and the way there are interpreted and processed. So the recent browsers as I have said all the recent browser you see, Internet Explorer, Netscape, Mozilla, they can all detect and interpret inline Javascript code. Now in order to identify that, the code that follows is Javascript code we can the SCRIPT tag. The tag begins with SCRIPT with an optional parameter called LANGUAGE. If you do not use this LANGUAGE equal to JavaScript, then by default the browser will take it to Javascript code followed by any number of lines of Javascript code followed by the end SCRIPT tag.

(Refer Slide Time: 31:53)



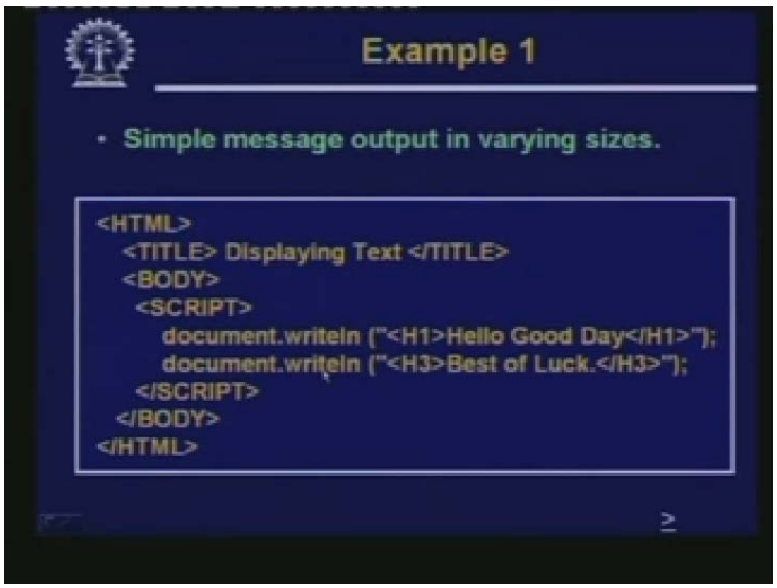
As I have said the Language JavaScript, this is optional and there is one interesting thing here that all browsers may not have the capability of interpreting Javascript. Say for instance you may be running a very old version of Internet Explorer that may not have the capability of interpreting or understanding Javascript codes. So what do you do for them? For them you can use this less than exclamation dash, dash and dash, dash greater than sign to bracket out the Javascript code. Now the way they are processed is this browser which does not understand Javascript, they will consider these tags as commons and the entire Javascript code in between will be bracketed as commons. They will not be processed at all. But for browsers which do understand Javascript for them it will be slightly different. For the browsers which do understand Javascript these special tags will be ignored and whatever there is between the tags they will be taken and processed as Javascript segments or programs.

(Refer Slide Time: 33:19)



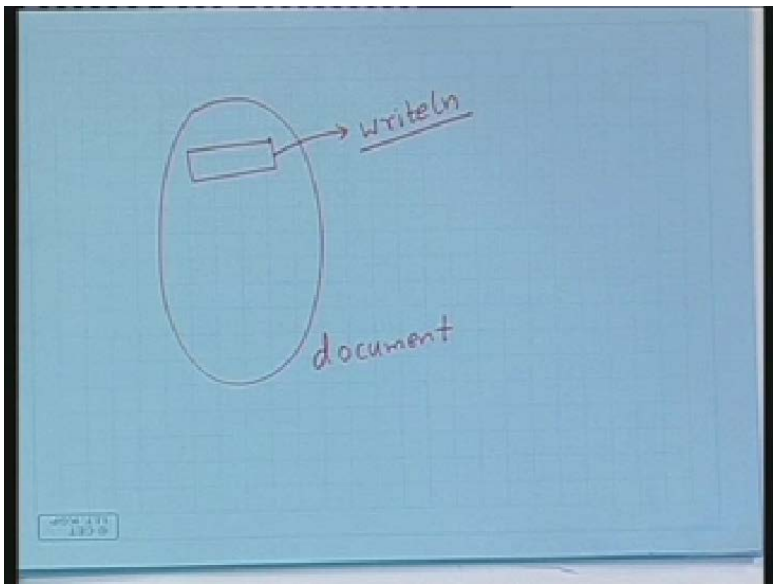
Now a typical HTML page will look like this. It begins with HTML all the standard tags HEAD, begin HEAD, end HEAD, begin BODY, end BODY, something in between. This is TITLE, end TITLE, see this Javascript code can be inserted anywhere the HTML document either within the HEAD portion or within the BODY portion. Here I have shown to insert within the HEAD portion. So SCRIPT LANGUAGE equal to JavaScript and here we have end SCRIPT and this is the bracketing where the Javascript code is there. So some browsers will be taking this as a comment and some other browsers will be ignoring this being and end and will be taking this as the Javascript code to be interpreted right. So now let us take some Javascript examples. Instead of formally explaining the syntax and semantics of the language we would rather show the capabilities of the Javascript language through some illustrative examples. Let us start the process.

(Refer Slide Time: 34:36)



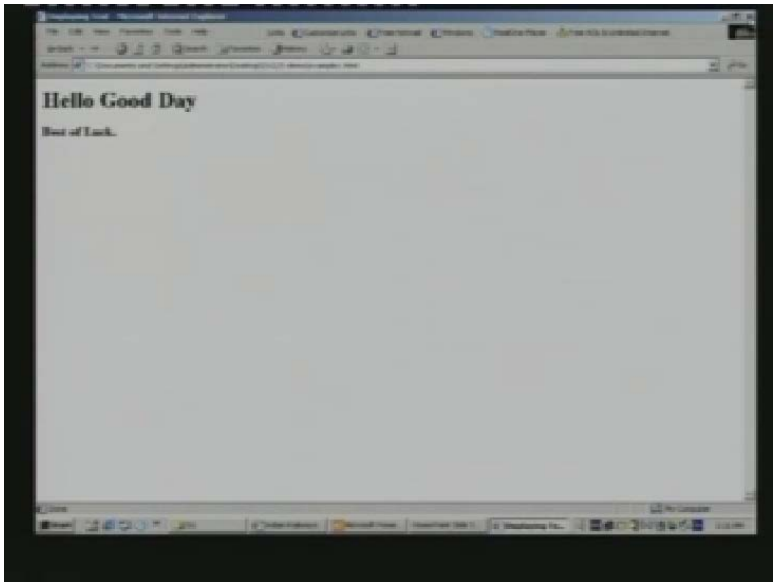
This first example is possibly the simplest. This example shows how we can output a simple message in varying sizes. You look at this code of interest is the portion within the SCRIPT. Now within the SCRIPT we have used the document dot writeln, these methods.

(Refer Slide Time: 35:10)



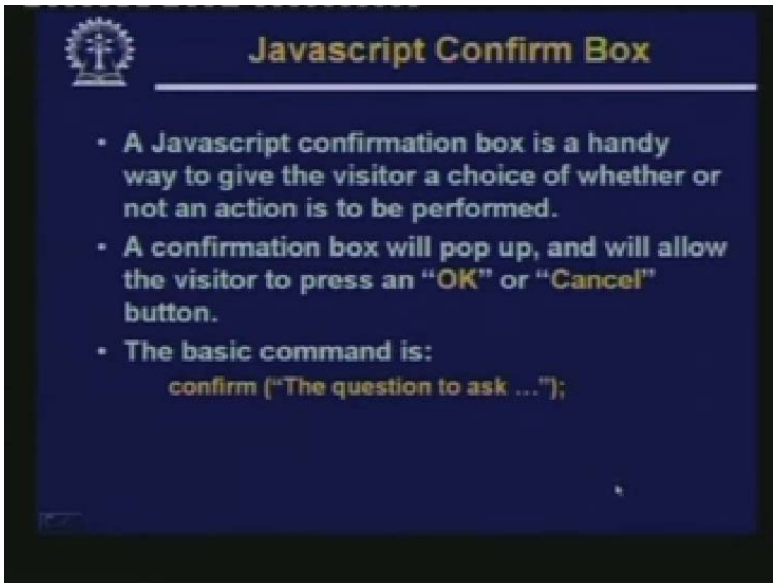
So here let me tell that we are basically trying to manipulate the document object and within the document object there is a particular method called writeln. Now the writeln what it basically does? It will take the document object and will output the specified string with a line break at the end with at the correct point inside the document. So whenever you are calling or invoking the writeln method in association with the document, the string that is specified will get outputted or inserted at the correct position inside the document. So here in this example there is nothing else in this HTML file just two document writeln. So within quotes, you see within quotes you can add even the HTML tags like H1, H3. So this “Hello Good Day” will be displayed in a bigger font. “Best of Luck” will be displayed in a smaller font. So let us see this code running. So if you run this your display will show like this.

(Refer Slide Time: 36:26)



Hello Good Day will be in bigger font. Best of Luck as you can is in smaller font.

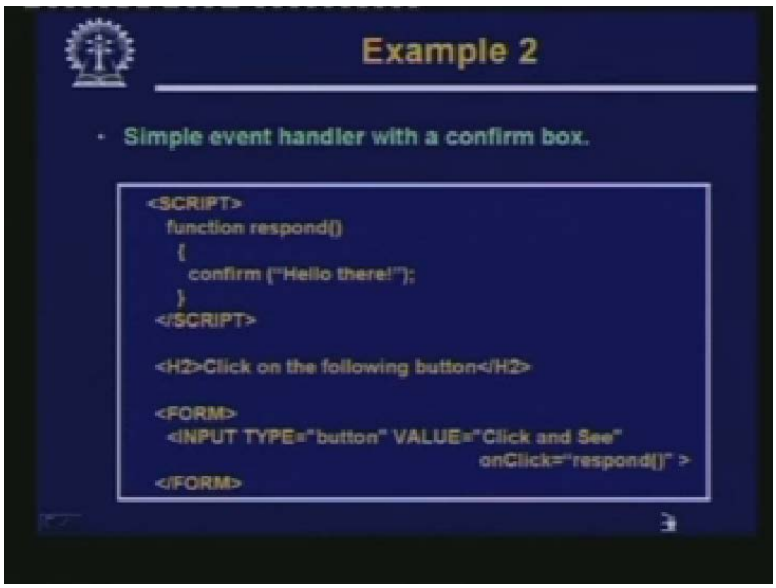
(Refer Slide Time: 36:39)



So let us now move to a second example which will illustrate something called a confirm box. Now a Javascript confirm box or confirmation box is basically a way to allow the user or the visitor to give a choice as to whether certain specified action is to be taken or not. Like you may be familiar with window based applications where you are often asked through a popup window. Certain message is displayed; then you are asked whether you want to continue this or cancel this. This is an example of a confirmation box. The system or the application is asking the user whether the specified operation or action is to be taken or is to be cancelled. Confirmation box can be used for this purpose. So when you use the Javascript confirmation box, a box will pop up and the box will be having two buttons called OK and

Cancel. And the visitor is now allowed to press either an OK or Cancel. The syntax of the confirm box is the name of the method is confirm within quote. You can give a string and that string will be used as the label of the confirmation box. It will display that string and then it will ask you whether OK or Cancel right.

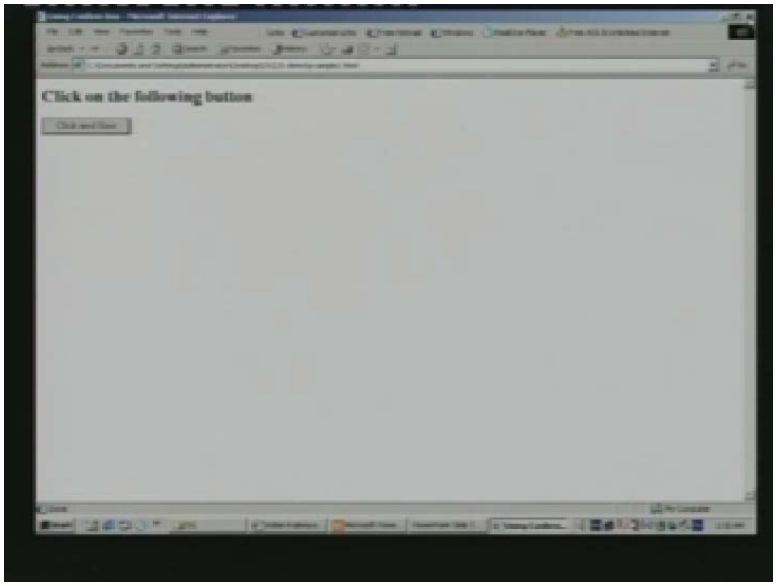
(Refer Slide Time: 38:20)



So, a very simple example is illustrated here. This is a very simple event handler. This example also shows for the first time how an event handler in Javascript can be written. See here for the sake of simplicity I have omitted the HTML, HEAD, BODY, all those tags in the HTML file. I am assuming all those standard tags are already present. I am only showing the relevant portion of the Javascript code snippets. So in the first part begin SCRIPT end SCRIPT. Here I have written a method or a function whose name is respond in the BODY of this function. As you can see there is just a call to confirm nothing else it simply confirms. It simply just opens a confirm box with the message Hello there and now comes the remaining part of the HTML. This is heading Click on the following button and as if we are creating a FORM, but the point to remember is that here we are creating a FORM not to allow the user to enter something which will be sent back to the server.

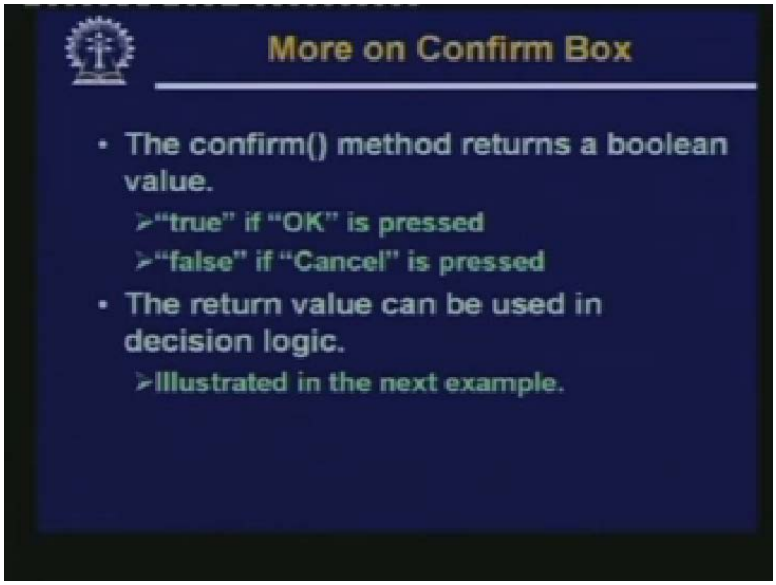
Rather we are allowing you take some local actions forms with Javascript can be used to take some local interactive actions also like this example shows. Let us look at the description of this form. This form says it is an INPUT box of type button. There is a button and the label of button will be Click and See this is given is value and there is a property called onClick. This is new. This is a Javascript specific property equal to respond within quotes. This means if the user clicks the button with the mouse which is the onClick event, then automatically the function or method respond will be called. This is an example of an event handler. So here I have written a method called respond. But I am not explicitly calling it I am only specifying whenever you press it, with the mouse, the button the method will be called. So let us see this script working.

(Refer Slide Time: 40:57)



So as you can see when you run this program you get this click on the following button and then a button whose label is click and see. Now if I click on this button, you see I get a pop up window. In this pop up I get this Hello there as the caption and there are two buttons OK and Cancel. Now I can press OK or Cancel whatever but there is no special action taken. Because in the program or the code I have not mentioned what action is to be taken if I press OK? What action will be taken if I press Cancel? So using confirm button we can also take some decision based on which button the user has pressed.

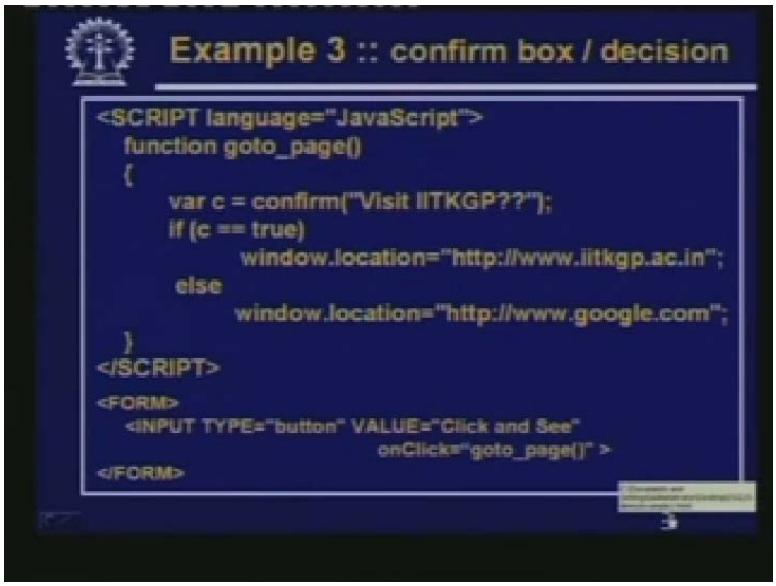
(Refer Slide Time: 41:46)



So for such decision making you have to understand one thing that this confirms method which you are calling, this actually will be returning a Boolean value. It will be written true if the button OK was pressed. It will be returning the value false if the button Cancel was

pressed. Now this return value either true or false. This you can use in an if-statement. For example to take some decision. So the applications like this can be easily created out of this. You ask the user for some option which option he wants yes or no. The user will click an appropriate button. Based on the appropriate button you can read the return value. You can use some if statements Java allows if statements using the, if statements you can take appropriate course of action.

(Refer Slide Time: 42:46)



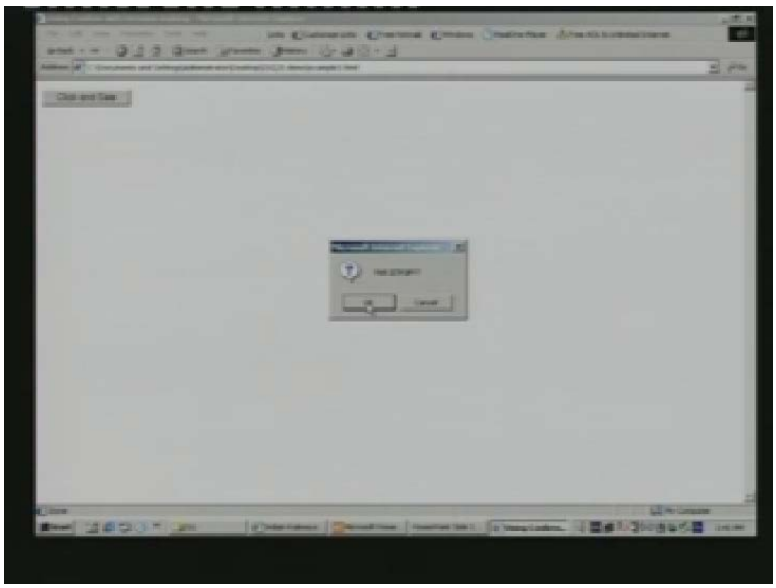
Now an example is shown here which illustrates how such decision making can be done. You see here again we have written a function or a method whose name is goto page. So as the name implies in this function we are trying to move to some specified page before explaining the BODY of the function. Let us see the FORM part first the FORM is exactly similar to what we had seen in the previous example. There is a single INPUT TYPE of type button, the label of the button will be Click and see that is the value and onClick event. I am specifying that you would be invoking goto page function. So whenever you click on the mouse, on the button you will be invoking this particular function. Now is this particular function you see what is the BODY of function?

We have called confirm with a string Visit IITKGP. The return value we are storing in a variable c this is the syntax var c equal to this. So whatever is the return value it gets stored in c. So this example shows you need not specify the type of the variable whatever returns back will automatically carry the type of the return value in to the variable. Then you have if statement, if else syntax is very similar to C. You check if the value of c is true or not. If it is true you are executing certain statements. If it is not you are executing certain other statements. Let us see what this statement means. Window dot location equal to something. Now as I have said window is the name of an object location is the property of that object in the window we are displaying something.

Location property specifies which URL is presently being displayed on the window. If we change or modify the location property of the window object automatically the URL and the

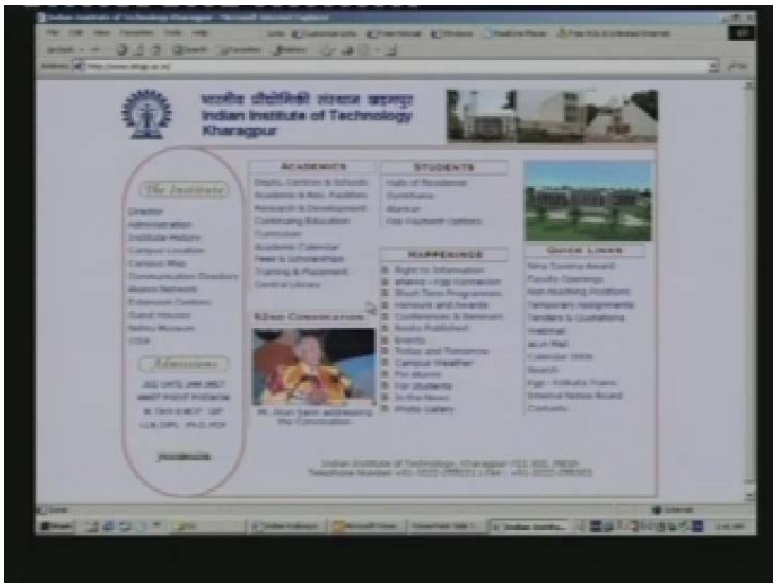
displayed value on the page will be changed. So just by assigning some string to the location property we can change the contents of the window by getting another page to loaded and displayed. So in this example what we have said is that if the user has confirmed by pressing OK, c is true. Then window location we are setting to the IIT Kharagpur web site which means the user has confirmed that I want to Visit IIT Kharagpur. But if the user has pressed cancel then by default we are taking the user to www.google.com. Now let us see that the example works correctly or not.

(Refer Slide Time: 46:08)



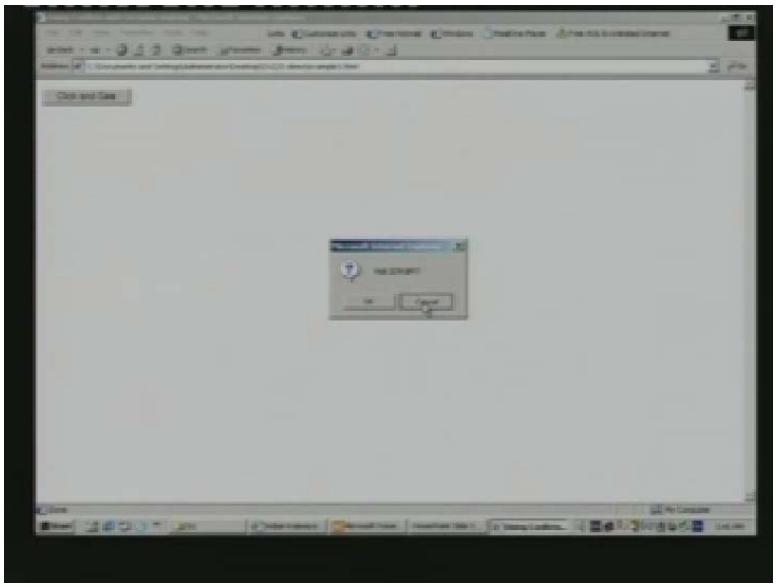
So when this example runs you will see that there is a button whose label is Click and See. Now I click on this button. I get a pop-up. This is a confirm box. It displays Visit IITKHG OK or Cancel. First I am showing if I press OK what happens if I press OK?

(Refer Slide Time: 46:31)

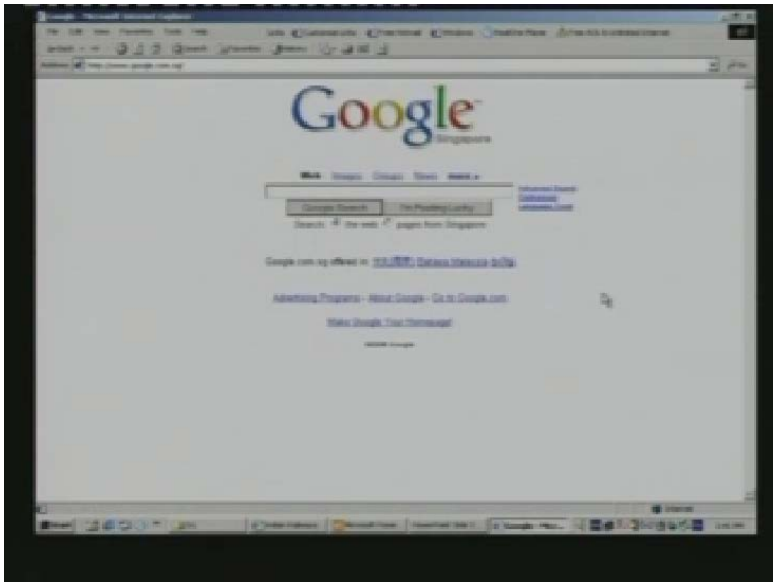


You see I have come to IIT Kharagpur web site. Let us again go back, again run it. This time let us place press Cancel. I press Cancel.

(Refer Slide Time: 46:40)



(Refer Slide Time: 46:49)

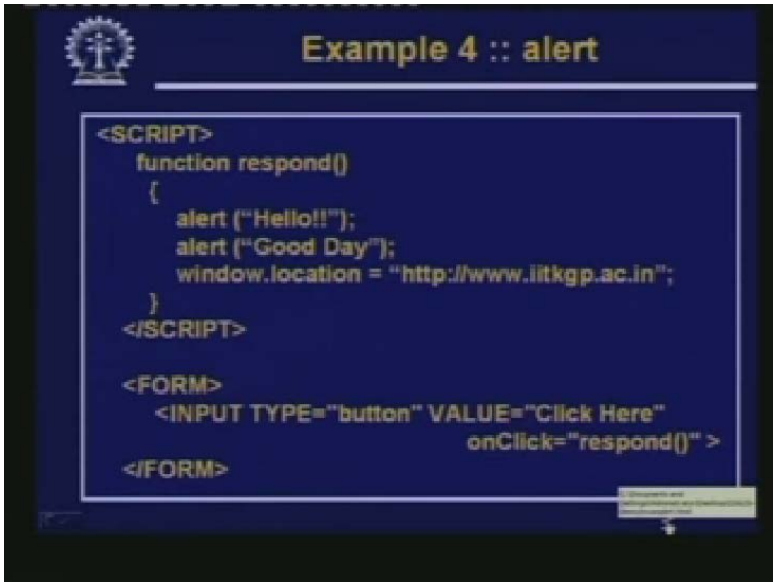


So we are been redirected to the Google web site and you can see we have gone to Google web site. So just by using the confirm box along with the checking of the return value of the confirm box we can basically. Take some decision based on what the user have entered.

(Refer Slide Time: 47:10)

Similar to confirm box there is something called Alerts which are primarily meant to convey some information to the user. Not necessarily getting feedback from the user. In the confirm box you are essentially trying to get a feedback from the user in the sense that you are asking the user to press either OK or Cancel. But here just a simple message box is displayed and after the user have read the message will simply press the only button available this is OK. There will be only one button available. User does not have any choice. So after reading the user can press the OK button. So the basic syntax of the Alert method is alert and any string in quotes.

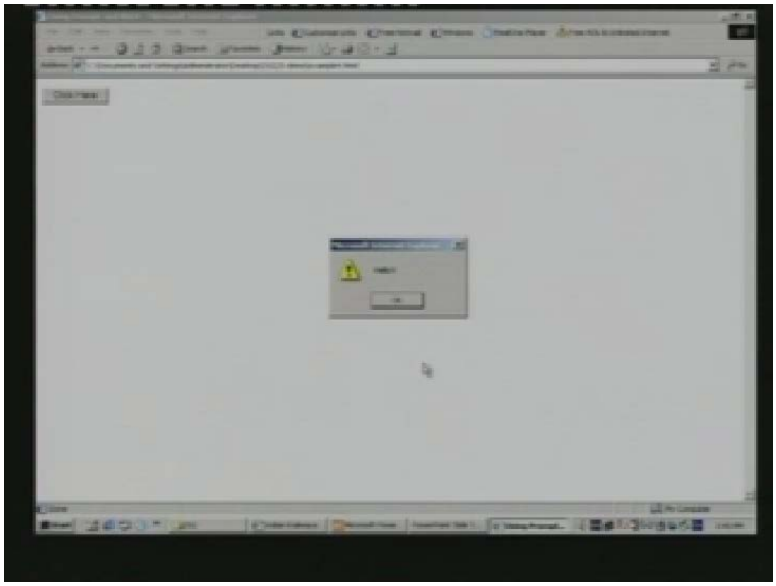
(Refer Slide Time: 48:02)



Let us take an example to show how Alert can be used. This is again a function called respond which you have written where there are two Alerts one after the other. First there

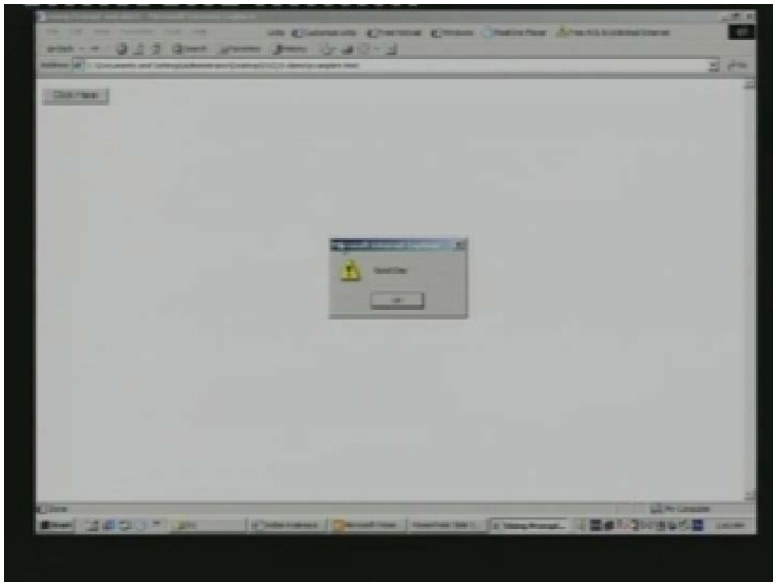
will be an Alert box displayed with the message Hello. Second there will be another Alert box displayed with the message called Good Day. Then after these two Alerts are executed then we would be going to a location iitkgp.ac.in and again in the FORM we are calling this in response to a mouse click. Now here these two Alert box will be displayed one after the other then only iitkgp will be executed. Let us see.

(Refer Slide Time: 48:51)



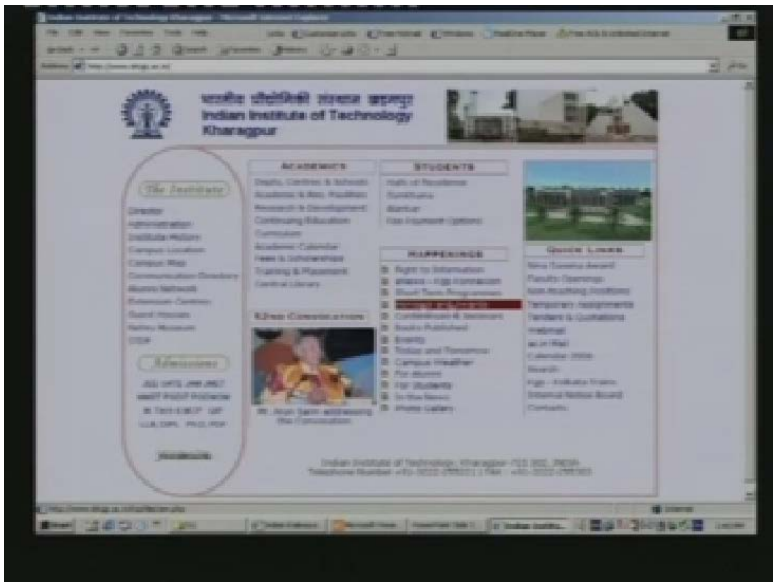
So first the button is displayed with a label Click here. Let us click on this. So the first Alert box is with a message Hello. You press ok.

(Refer Slide Time: 49:05)



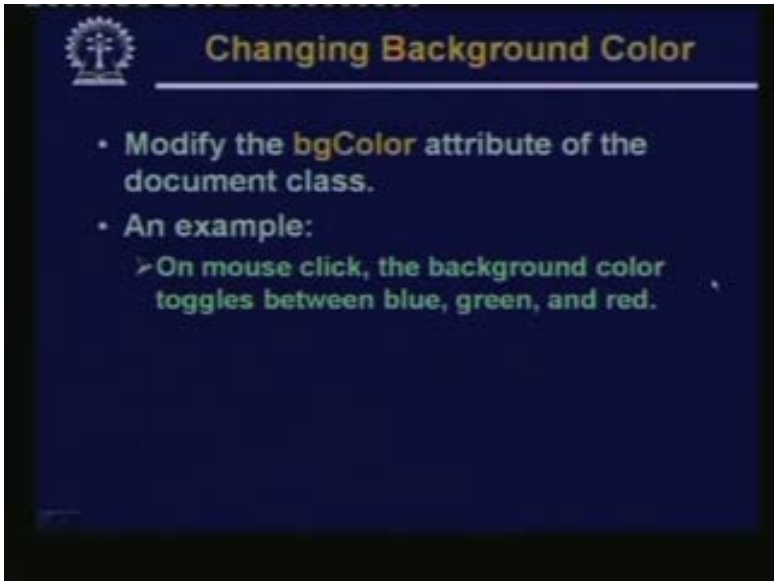
Now another Alert box the second one displays the message Good Day you press OK. So now you go to IIT Kharagpur right. So this is what was expected.

(Refer Slide Time: 49:09)



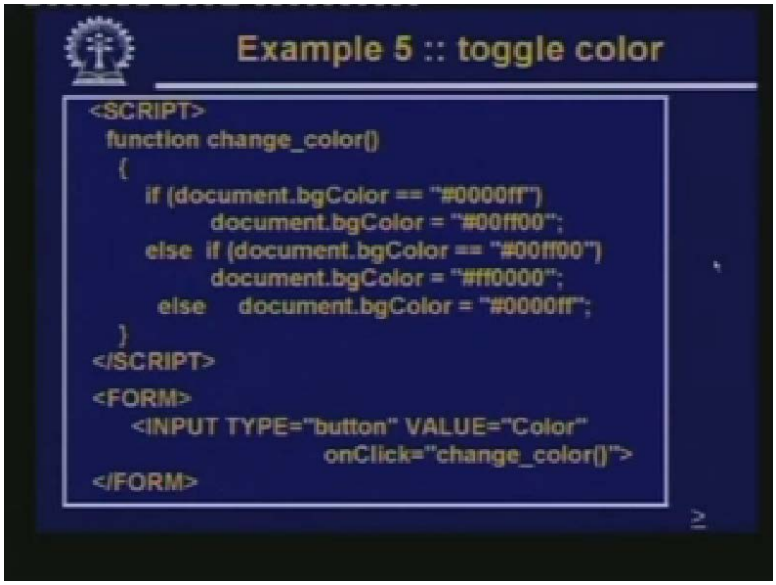
So next let us take another example which concerns color.

(Refer Slide Time: 49:17)



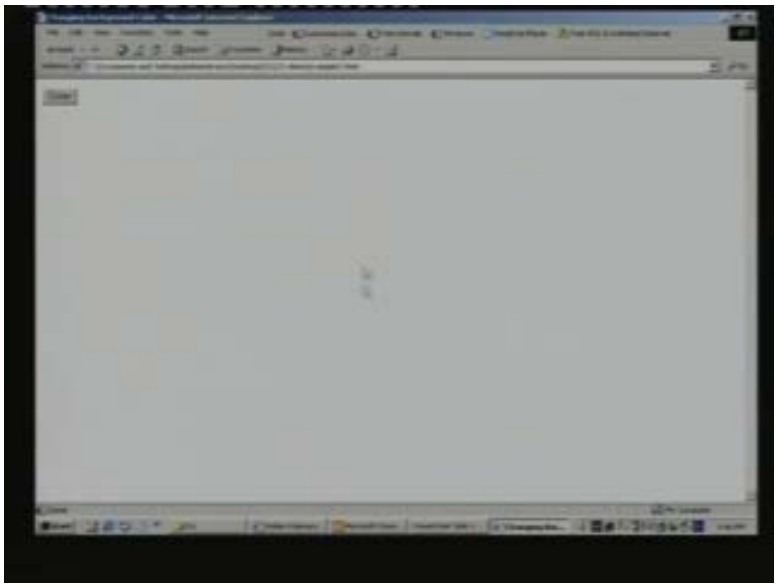
Now we had said that the bgColor property or attribute is used to specify the background color of a certain document. So here we shall see how we can manipulate that color to change the background color of a displayed document. So essentially we will have to modify the value of the bgColor attribute of the document class. Now the example that we have taken in this example, we have written a Javascript code which will toggle the background color between blue, green and red. Repeatedly as we click a particular button with the mouse. This is what we try to do.

(Refer Slide Time: 50:09)



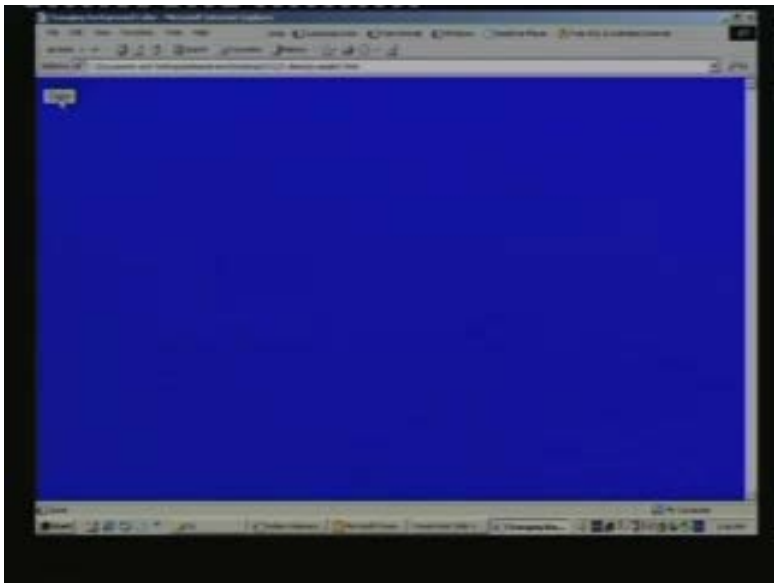
The code snippet is like this the form portion. Let us see this is again same. TYPE is button the label is Color and when you click with the mouse, we call the function or method change_color. In the change you see there are three if-statements, if-nested, if-statements. If document.bgColor is an attribute of document object equal to hash 0000ff, this is color in the RGB format. Red is zero, green is zero, blue is ff. This means blue instead of in this format I could have also specified the color by mentioning string b l u e blue. These are two alternate ways of specifying. So what I say here is that if the present background color if blue then you assign 00ff00 RGB, this means green make it green. If is green make it red else. Well else means it was red make it blue. So blue, green and red in this order color will go on changing.

(Refer Slide Time: 51:30)



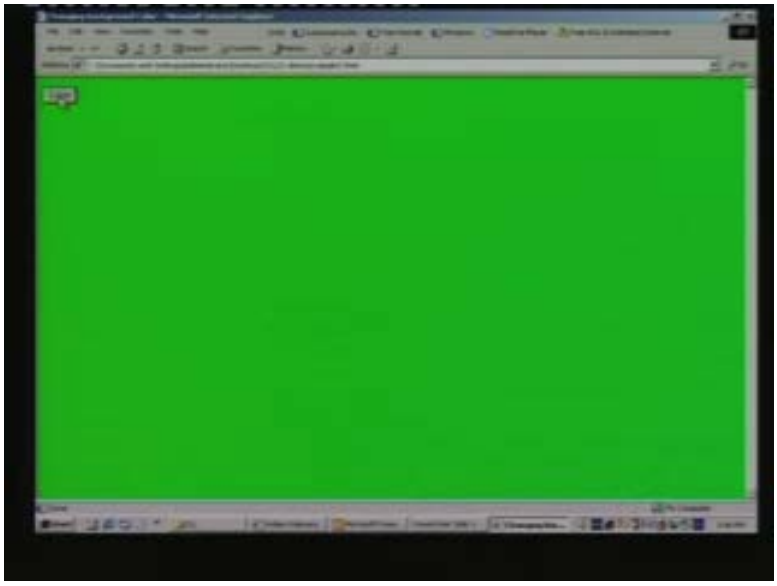
Let us see this is the window with this label color. So I will be clicking on this color.

(Refer Slide Time: 51:37)



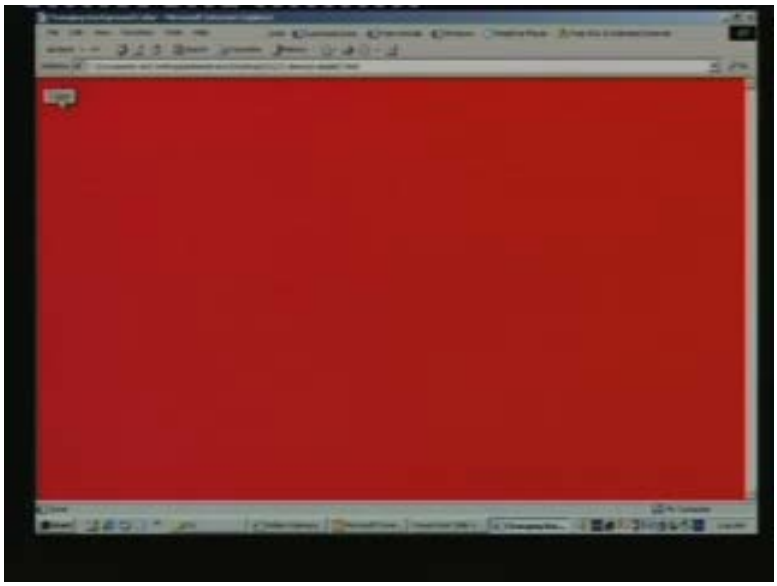
I click it once the window color becomes blue.

(Refer Slide Time: 51:39)



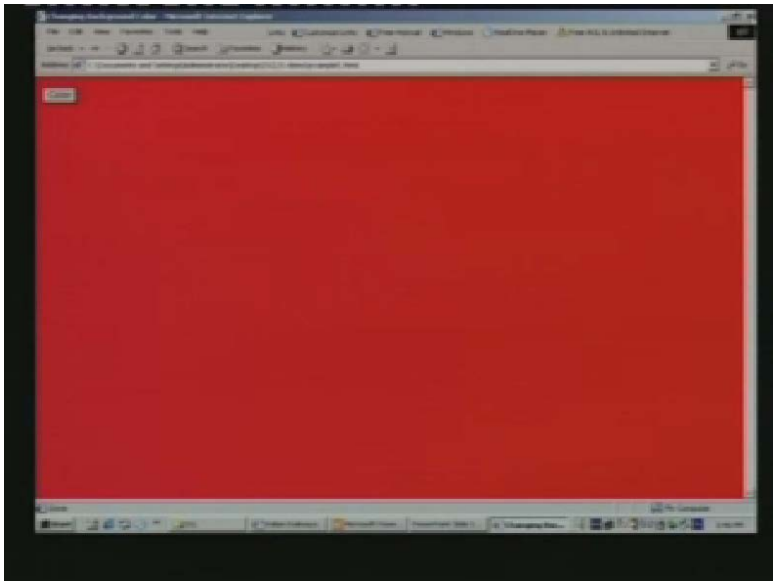
I click it again, becomes green.

(Refer Slide Time: 51:41)



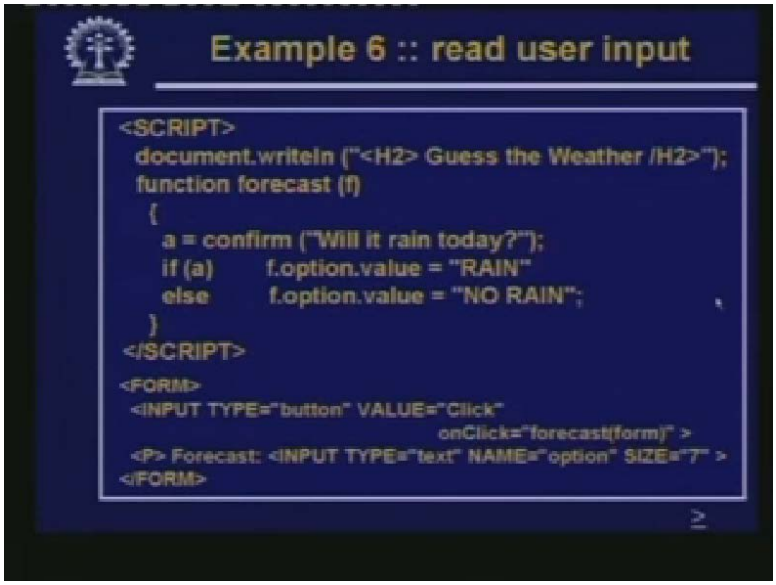
I click it again, it becomes red. I can continue clicking and the color will go on changing.

(Refer Slide Time: 51:49)



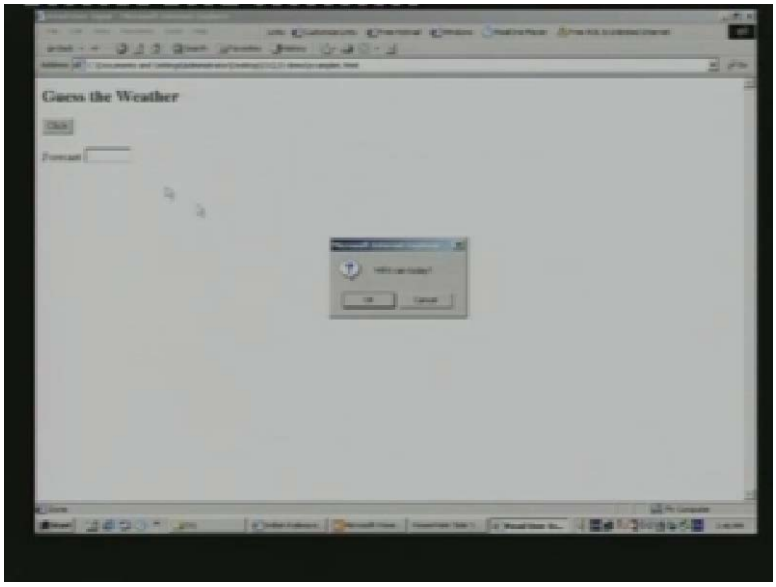
So this is how you can change the background color for document. Similarly we can also change the foreground color. The attribute of the foreground is fgColor, bgColor is background color, fgColor if foreground color here.

(Refer Slide Time: 52:11)



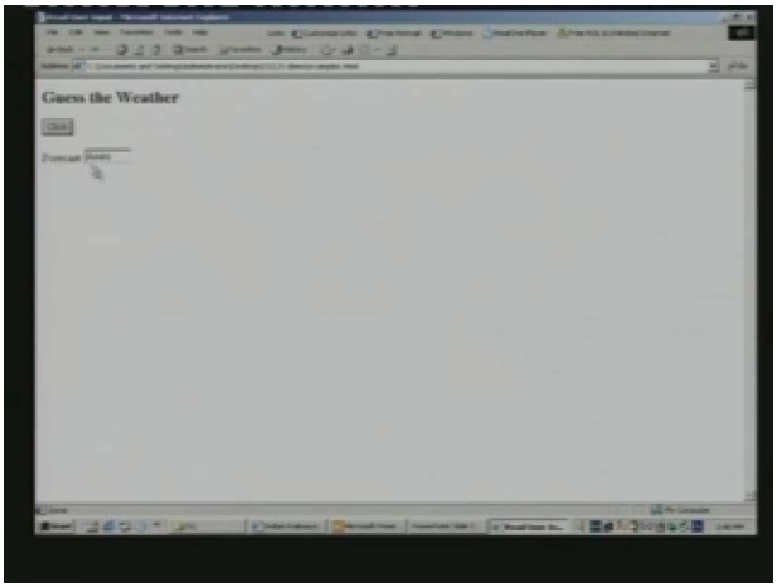
We are showing an example using confirm. Based on some user input you can fill up some data in a form. Like the example is like this you first look at the form. In this form there are two input boxes. First is a is a conventional button where the label you have put as click and if you click you would be calling the forecast function with a parameter form. Means the present form as the parameter. So whatever you do is on the present form and in the second one, there is a text forecast which is displayed and it is a text type input. The option is the name of the field and size is seven and the script says that if you call confirm, you will be pressing OK or Cancel depending on a. See f is the parameter. So this form will come in. So the option attribute, name option the value part of it you change to RAIN or to NO RAIN. So the content of that form will change. Let us see showing an example.

(Refer Slide Time: 53:39)



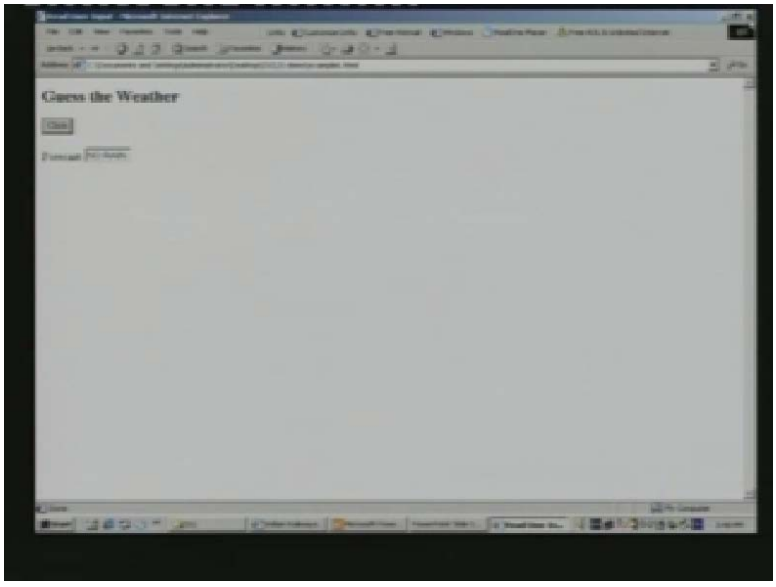
Guess the weather click if I get a box will it rain today OK Cancel if I press OK. I get the value RAIN, here you can see.

(Refer Slide Time: 53:50)



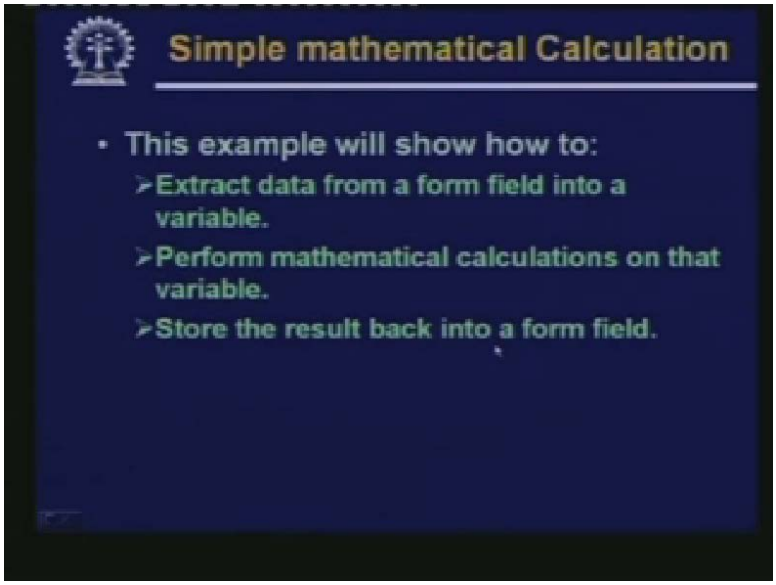
And if I press Cancel I get the value NO RAIN.

(Refer Slide Time: 53:53)



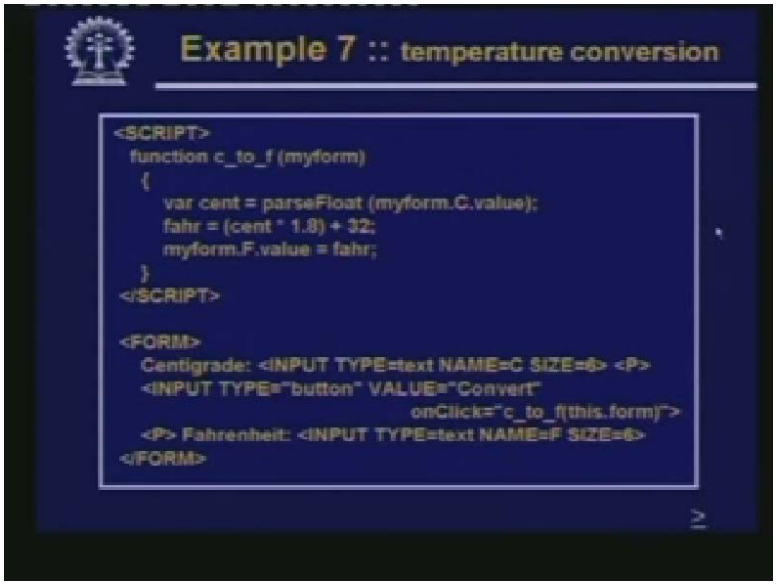
So this is how you can fill up the data in a form.

(Refer Slide Time: 53:56)



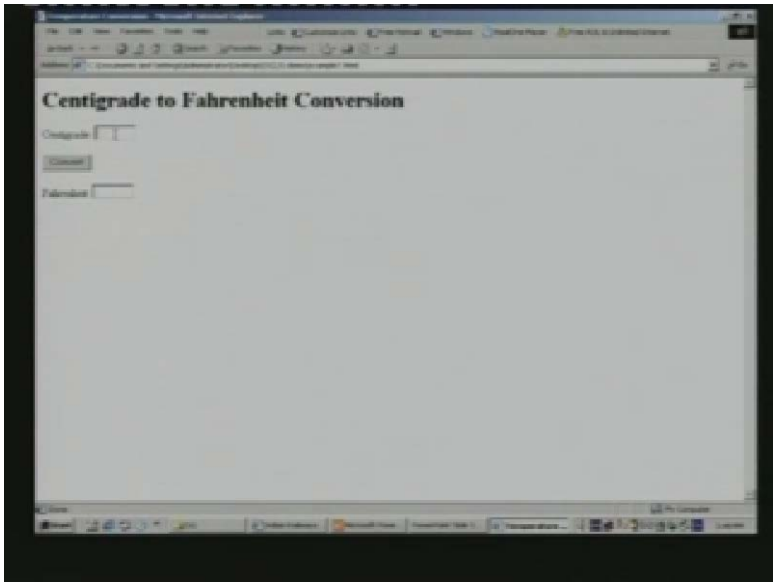
The next example shows you how you can carry out simple mathematical calculation. So here you will be extracting some data from a form, perform some calculations and store the result back.

(Refer Slide Time: 54:13)



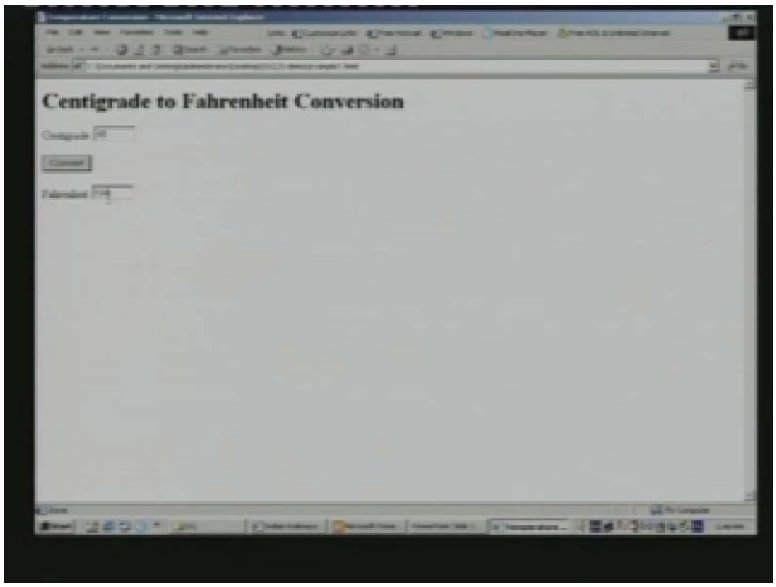
The calculation is fairly simple we are doing a Centigrade to Fahrenheit conversion. There is a form with two fields; one with Centigrade field whose name is C, there is another Fahrenheit field whose name is F. And there is a button field whose label is Convert. If you press it then this function will be called. This function essentially reads the values of C dot value of this myform converts to floating point assigns to cent. Convert it n to Fahrenheit and assign Fahrenheit to this F value, the value part of the F field right. So let us see if it working.

(Refer Slide Time: 54:52)



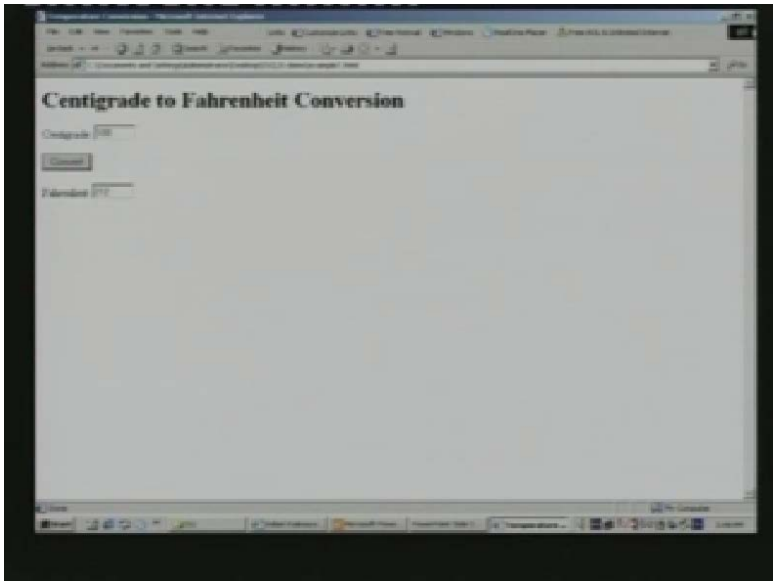
So if you specify Centigrade [54:54 word not clear] if you specify Centigrade as forty degree press Convert.

(Refer Slide Time: 55:10)



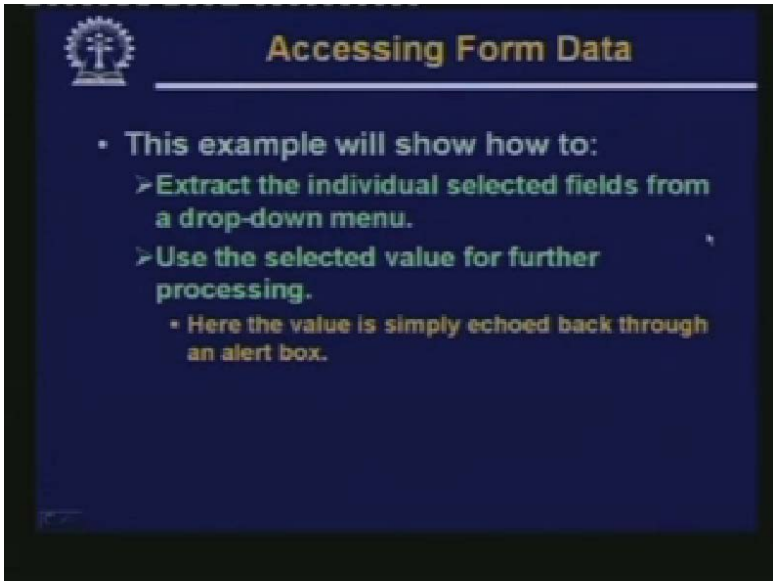
You get the equivalent Fahrenheit 104. Now specify some other value.

(Refer Slide Time: 55:12)



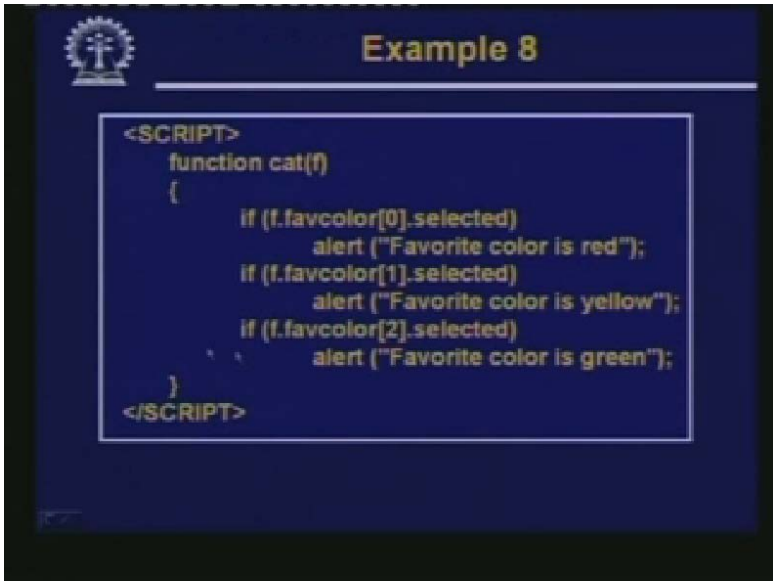
100 degree Centigrade, you convert it becomes 212. So this shows how you can read the value from a form. How you can store back the result in to another form right and also it shows how you can carry out simple calculations.

(Refer Slide Time: 55:20)



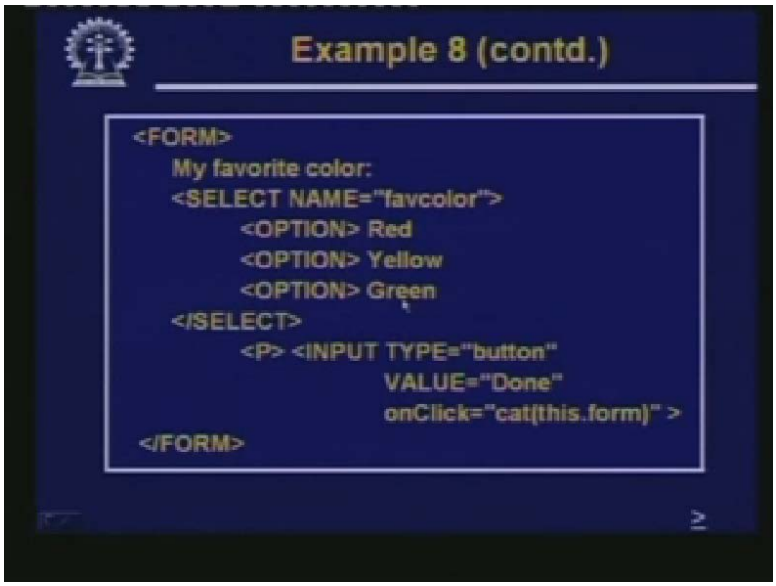
The second example will show you how you can access some form data. Now here you can extract some individual selected items from a drop down menu like this example shows.

(Refer Slide Time: 55:37)



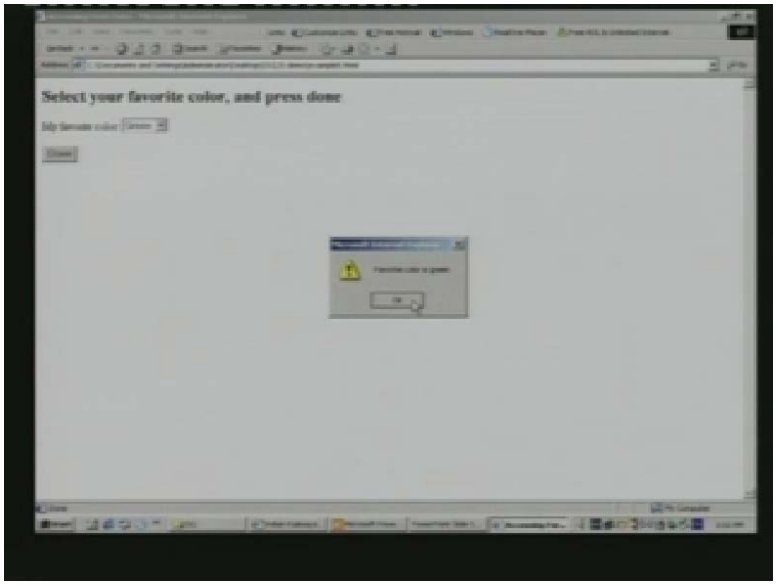
Like you can have a particular function, which can the different elements of a favcolor select box of a form. If it is selected then you give the appropriate Alert.

(Refer Slide Time: 55:56)



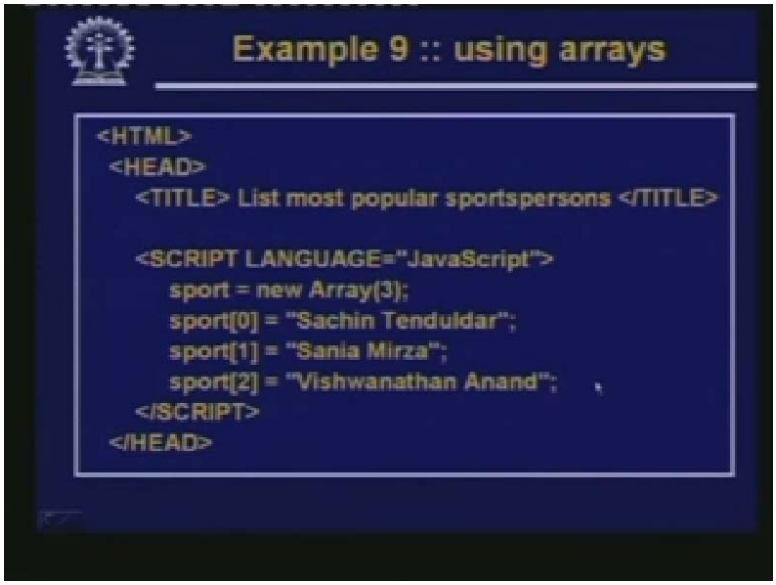
This is the form, this is the select box whose name is favcolor options are Red, Yellow and Green and you call that function called cat showed before cat if you press on mouse and here this one if you execute it. So you can change your favourite color Red, Yellow or Green. Suppose it is Green I press Done, there is an Alert box favourite color is Green this comes.

(Refer Slide Time: 56:22)



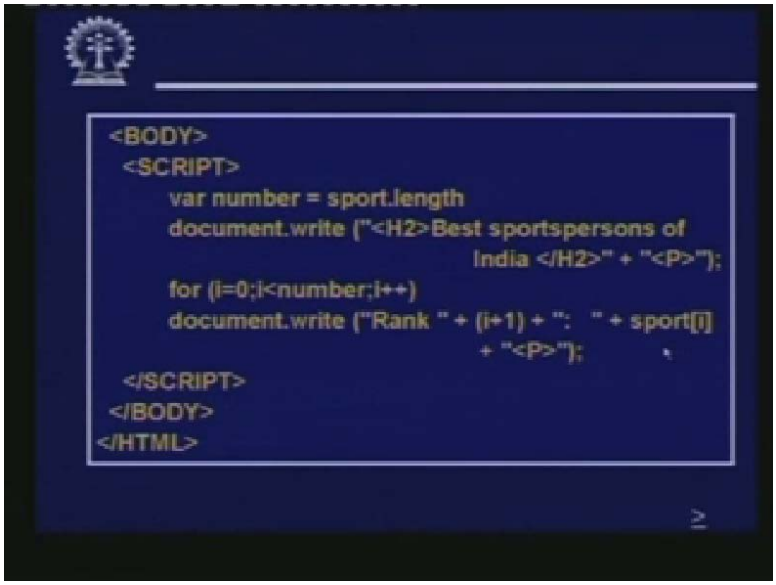
So this shows how we can access the individual items of a select box.

(Refer Slide Time: 56:32)



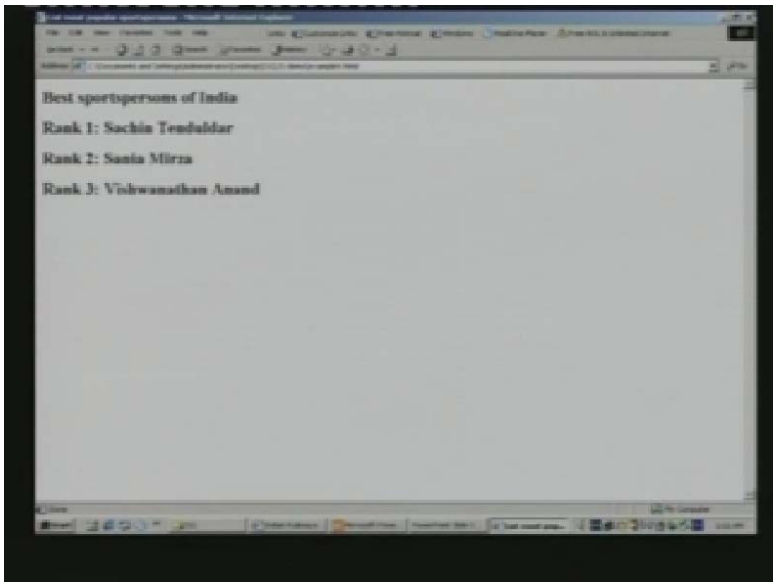
The last example I show is that of using arrays. Suppose I have a simple code snippet. Here I have defined an Array of size 3 and I have assigned values to the three elements of the array. These are strings by default Sachin Tendulkar, Sania Mirza, Vishwanathan Anand.

(Refer Slide Time: 57:03)



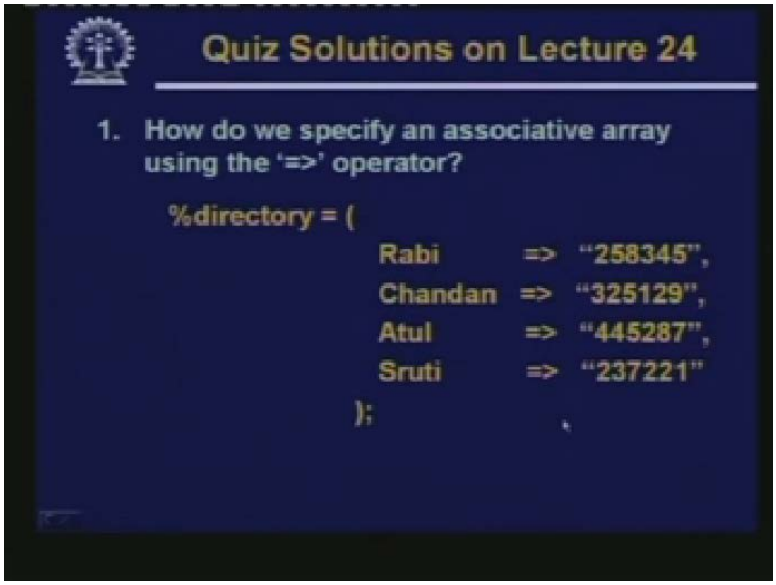
And in the BODY part I have return another code snippets in Javascript. Which tells best sportspersons in India and the number elements in the array and the for loop we are displaying all of them Rank, the number and then the contents of the array. So if you run this code you will get an output like this.

(Refer Slide Time: 57:26)



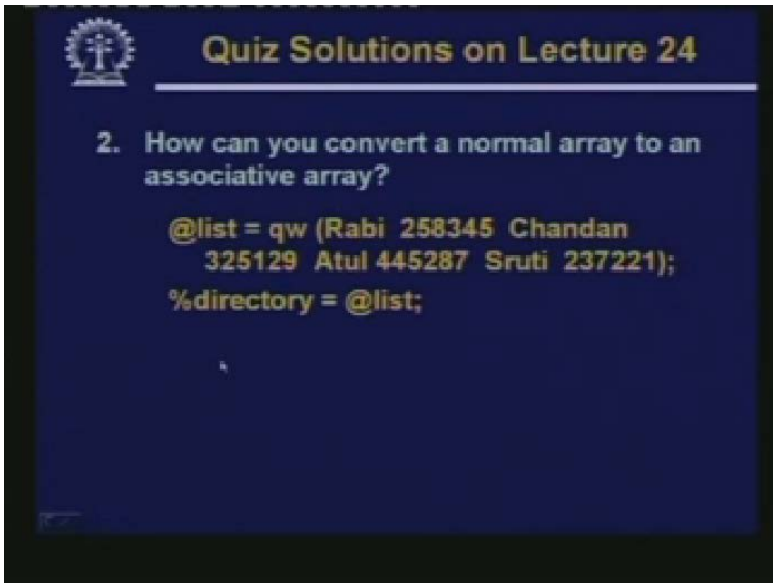
Best sportspersons in India. Rank one Sachin Tendulkar, Rank two Sania Mirza and so on. So with this we come to the end of today's lecture where we have basically looked at some the basic features of Javascript and some examples. Let us now quickly go back and look at the solutions to the questions we raised in our last lecture.

(Refer Slide Time: 57:48)



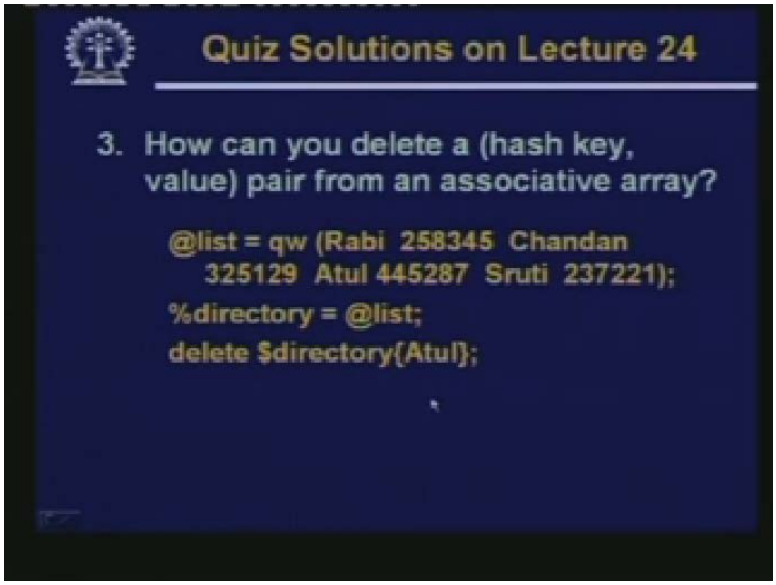
The first question was how we specify an associative array using the equal greater than operator?
It is simple you simply follow this syntax within bracket, the first element of the associative array followed by this symbol. Then the value part name, value name, value name, and value in this order you can specify.

(Refer Slide Time: 58:11)



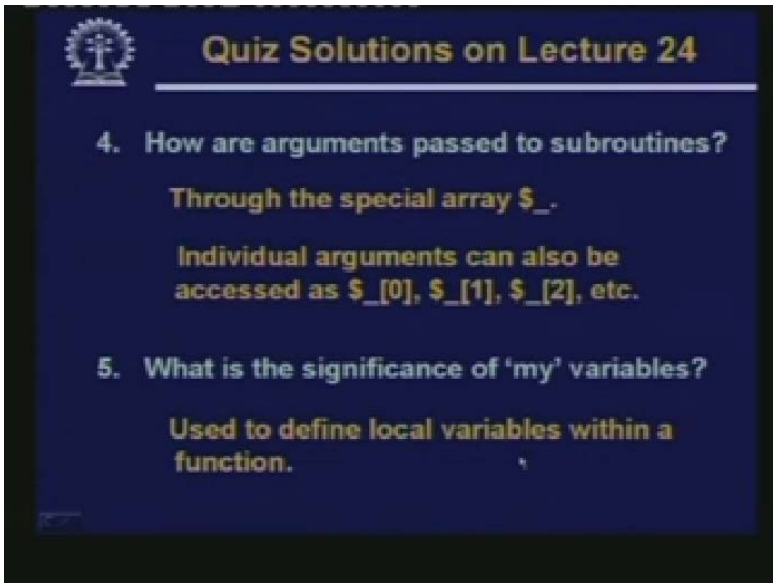
How can you convert a normal array to an associative array?
Simple, you create a normal array where the elements are in that order name, value name. Value in that order and simply assign that normal array in to an associative array. So the association will be done automatically.

(Refer Slide Time: 58:29)



How can you delete a particular hash value pair from an associative array?
Suppose we have an associative array created like this. We can simply write delete. The name of the associative array within curly bracket. The name of the entry we want to delete. Suppose Atul is one we want to delete, so Atul followed 445287, both these entries will get deleted.

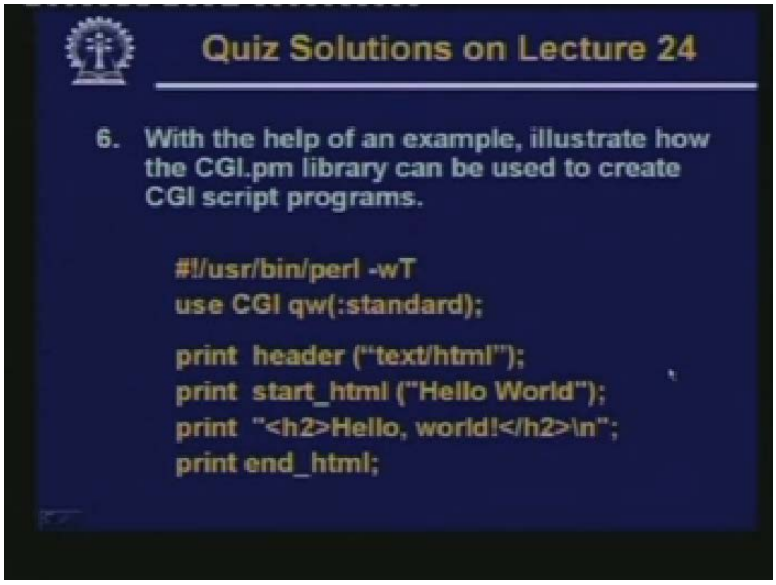
(Refer Slide Time: 58:58)



How are arguments passed to subroutines?
They are passed through a special array called dollar underscore. But however individual arguments of the array of the arguments can also be accessed by accessing the array elements 0, 1, 2, etcetera.
What is the significance of my variables?

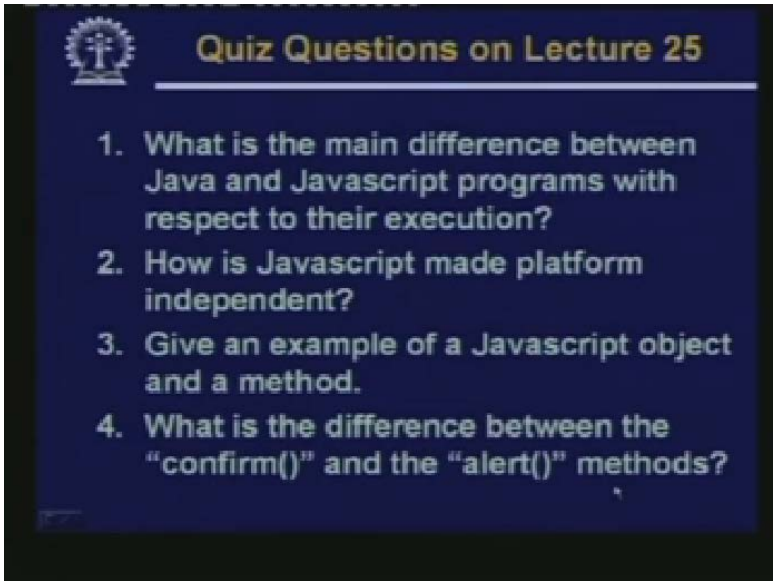
There are used to define local variables within a function.

(Refer Slide Time: 59:22)



Lastly With the help of an example illustrate how the CGI.pm library can be used to create CGI programs?
This we had shown as examples in the previous class. But we can basically call or we can print using some certain pre-defined functions header, start html, end html which will automatically put in the standard html code snippets in the output. So it becomes really easy to write CGI script programs like this. So some questions from today's class.

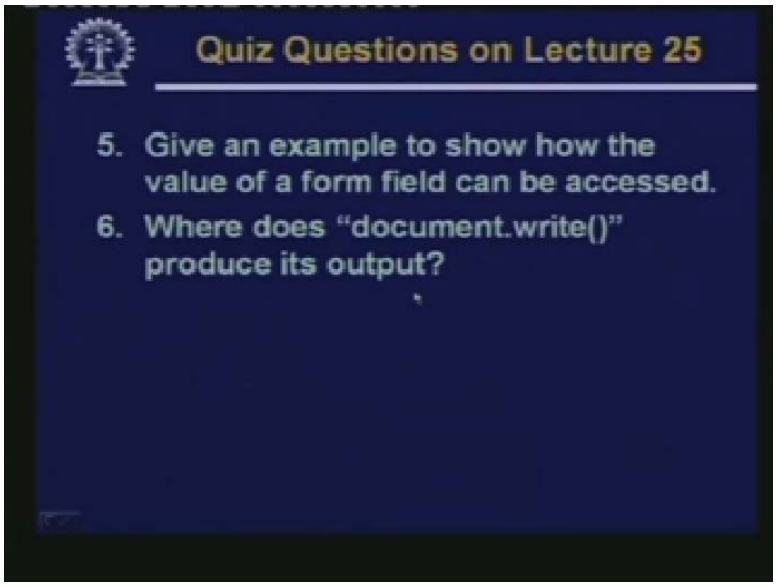
(Refer Slide Time: 59:58)



What is the main difference between Java and Javascript programs with respect to their execution?

How is Javascript made platform independent?
Give an example of a Javascript object and a method.
What is the difference between the confirm and alert methods is Javascript?

(Refer Slide Time: 01:00:27)



Give an example to show how the value of a form field can be accessed.
Where does document dot write produce its output?
So with that we come to the end so today's lecture.