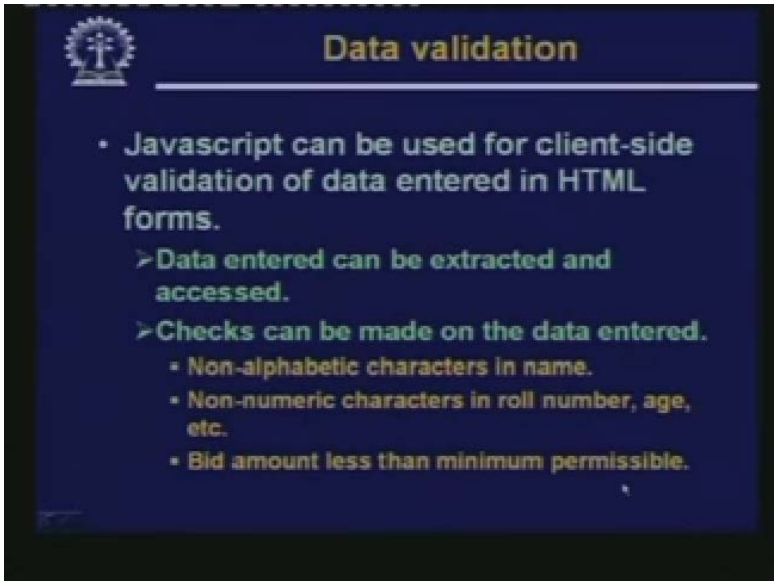
Internet Technology
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture No #26
Javascript Examples (Continued)

In this class we will be continuing with our discussion on Javascript. If you recall in our previous class we had talked about the basic object-oriented characteristics of the Javascript language. How it can be embedded inside an HTML document and through examples we tried to illustrate some of the basic properties or some of the capabilities of the Javascript language. How some of the very commonly used features of the language can be used and implemented as part some example code and we had looked at some of the examples in that regard. So today we shall be continuing in the same way. To start with we shall again be looking at some Javascript examples which will show how certain things can be or implemented using Javascript programming. So essentially we would continue with looking at Javascript examples.

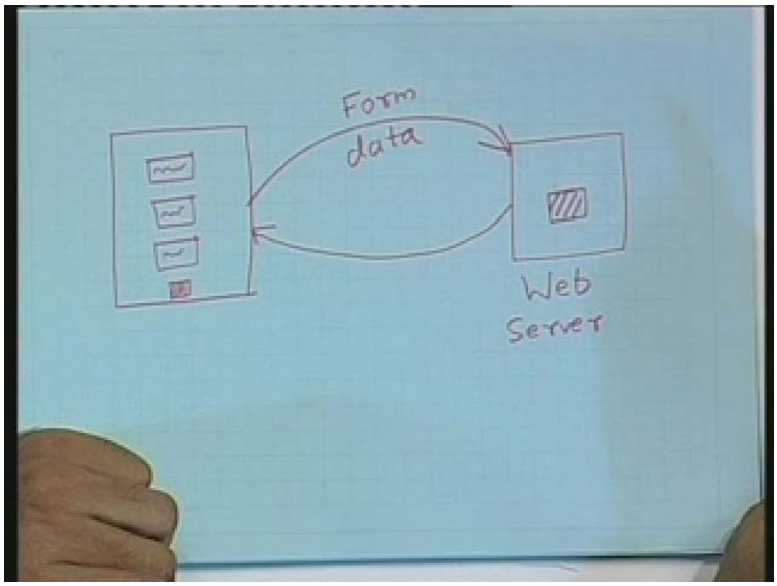
(Refer Slide Time: 01:35)



The first example we take today is a very important application of Javascript namely that of data validation. In fact Javascript can be used for client-side validation of data entered in HTML forms. Well you try to understand one thing. When we talk about data validation, we are basically talking about whether the users are entering data in a form correctly or not. There can be several ways in which you can check correctness in this regard. For instances you can say that in the telephone number fields you will not allow anything other than numeric digits. If it is a mobile telephone number, it will always consist of 10 numeric digits, if it is a roll number for example say consist of exactly 7 numerical digits and so on. A name can consist only of only of alphabetic characters and full stops possibly. So these kinds of simple rules you can formulate and at the time of data entry you can carry out online validation. Now without Javascript this

validation could be done using a server-side script after you enter a form you submit the form data to the server like this.

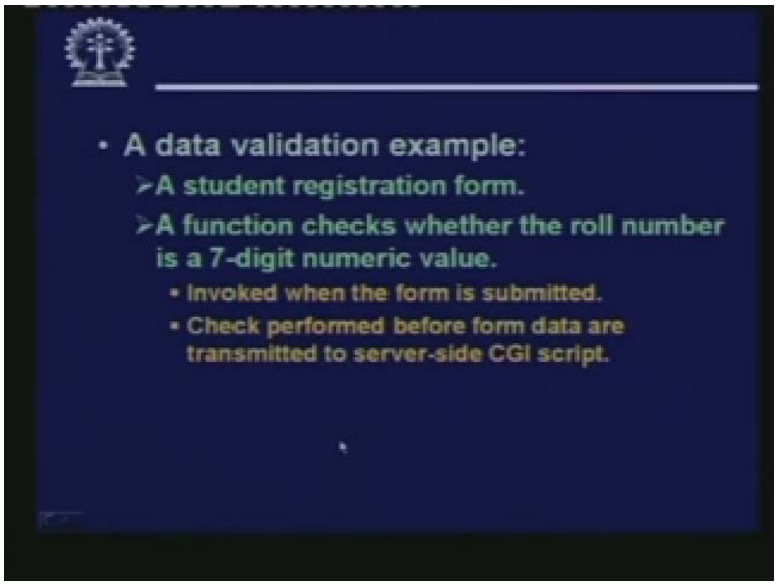
(Refer Slide Time: 03:12)



Suppose on the client-side, you have a form which is displayed. The user can fill in these forms and can press on the submit button. After the submit button is pressed, suppose this is your web server, the form data will go to the web server. The data that you have entered, the so called form data will go back the web server will call some CGI script program which is stored inside it. CGI script program will be providing data validation and will be sending back a confirmation to the user. This kind confirmation or data validation is of course sometimes essential like I am giving example. Suppose we are creating a new Yahoo mail account email account there at the time of registration you have to type in the name of the ID that you want, it is possible. Someone else is already holding that ID you are you are requesting for. So the Yahoo server has to check in its database to find out whether that ID already exists or is it available to you. So this check can be done only at the server-side because you need to check Yahoo database.

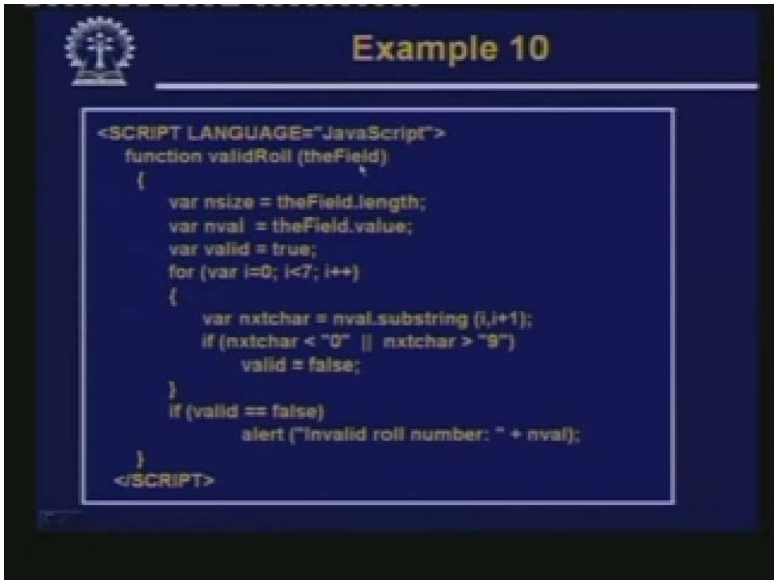
But there are many checks which can be done client-side event you need not disturb. The server with these kinds of simple checks right these are called so called client-side validation techniques. Using Javascript you can do client-side validation like the roll number and telephone number. Examples I have taken these checks can be very easily done at the browser end itself. So for data validation what you will have to do essentially is you will have extract the data entered in the form fields. You will have to ex you will have to access them and whatever data you have entered you will have to make some checks like. Some examples I have already given like some Non-alphabetic characters in names. Non-numeric characters in roll number, age, etcetera. And suppose in an online auction site you are supposed to submit a bid. But with every item there is a minimum amount of bid. So if the bid amount is less than permissible then also you will be getting an error message.

(Refer Slide Time: 05:56)



Here we show a data validation example which is essentially a student registration form. Will see that we have written a function for validation which checks whether the roll number is a 7 digit numeric value are not. This function will be submitted every time is trying to submit the form. So after this validation data will be sent back to the server-side script.

(Refer Slide Time: 06:29)

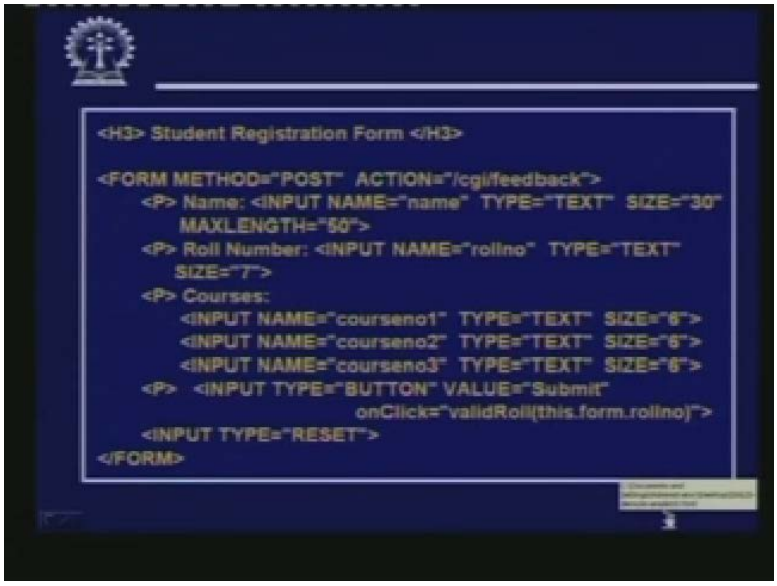


So the example is like this. First we show the function, name of the function is validRoll within parentheses, name of theField. Because in a form there may be several fields you will have to specify which field you are trying access or you are validate the name of that particular field will

have to be passed as argument or parameter to this method. Now in the body of the function there are several attributes of this field you are extracting; nsize of course you are not using in the function but I am showing you can access the value. This tells you the size of field, how many character this field size is. So if it is a roll number you are y are expecting this nsize to be 7. Then the actual value of the field, the field with the value attribute valid is a Boolean which we initialize to true. In a for-loop we will continuously be checking individual digits of the roll number. Or the individual characters there can 7 characters, 7 there will be characters expected. So we are looping seven times I equal to zero up to less than 7 in the loop.

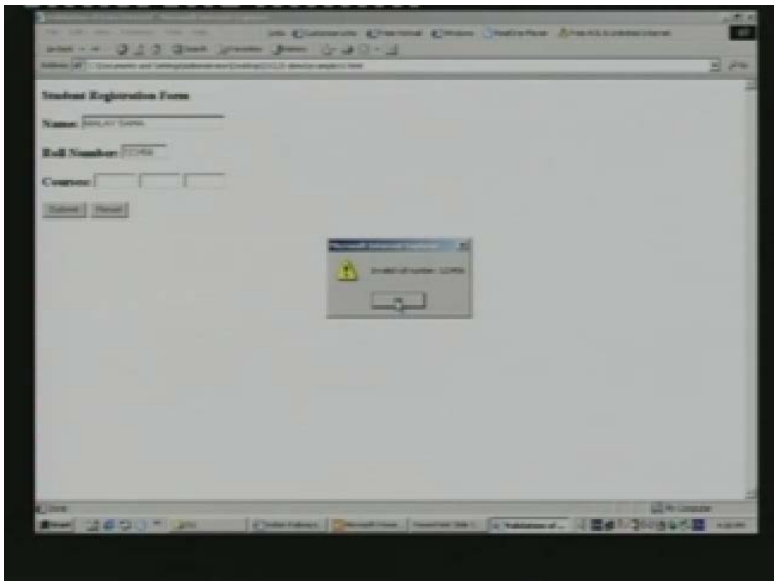
We are extracting a single character from this nval which we have taken out by calling the method substring. Substring I comma I plus 1 will extract a substring that will be starting at I less than I plus one which means a single character will be extracted at a time. So when I equal to zero the zeroth character will come out I equal to one first character will come out and so on. So here we are extracting one of the characters from the roll number at a time and we are checking whatever has come out if it is less than the digit zero ASCII code of 0 or greater than the ASCII code of 9 which means this is not a valid digit. If it is less than 0 or greater than 9, it will mean that the particular character is not a digit. So the roll number is not a valid one. So immediately what we do set the valid variable to false. But how we con the loop seven times at the end of it we check valid is true or false. If it is false, we raise an alert Invalid roll number. We also display the value of the roll number.

(Refer Slide Time: 09:12)



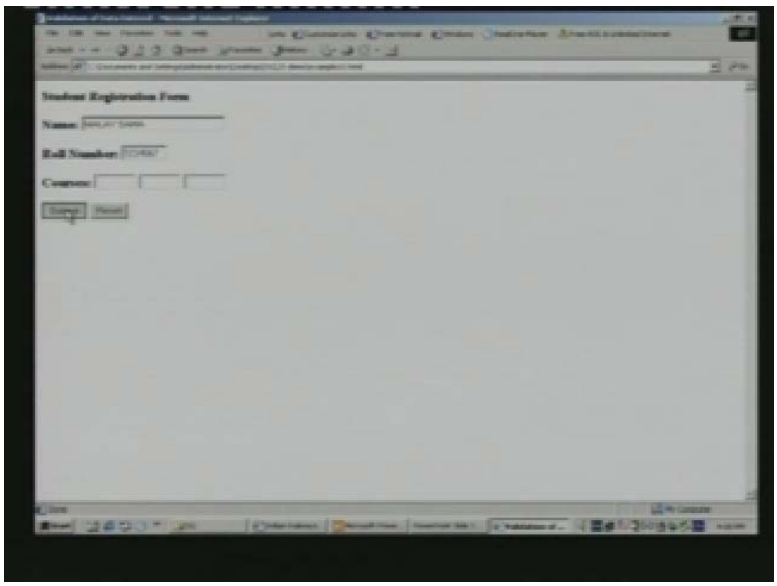
And this is the part of the student registration form as such. So you can see form method action cgi feedback. As usual you specify the cgi script, but in the button type you specify an event driven action at onClick. We call the validRoll fn with this .form, .roll number. Roll number field has the name r o l l n o. You call this fn with this parameter. So when this function will be called the roll number will be said sent here as the parameter. So now let us see this function working.

(Refer Slide Time: 10:00)



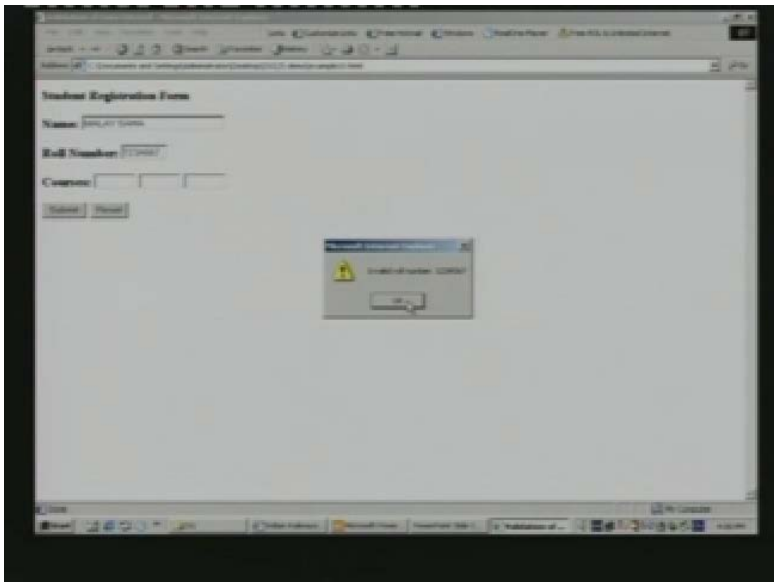
So when you run this you will get a student registration forms. So for student you can type the name Malai Shaha. Well just to show you I am typing in a 6 digit roll number 1, 2, 3, 4, 5, 6, this should be invalid Courses I have not checked anything. Now if I submit this, you see I immediately get a prompt Invalid roll number 1, 2, 3, 4, 5, 6. But if I add another digit to it seven.

(Refer Slide Time: 10:36)



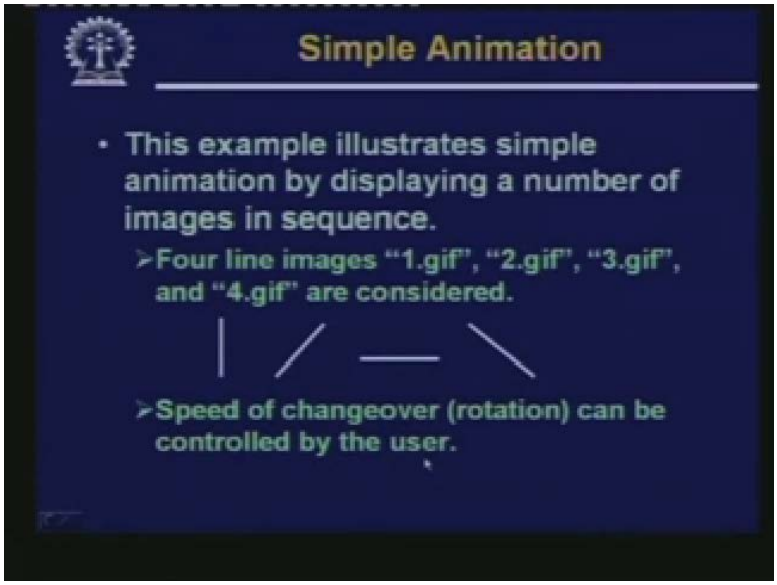
And if click submit again, nothing comes which means roll number is valid. So if I type 7 digits or if I add some alpha numeric characters, suppose I add an ‘a’ in between.

(Refer Slide Time: 10:51)



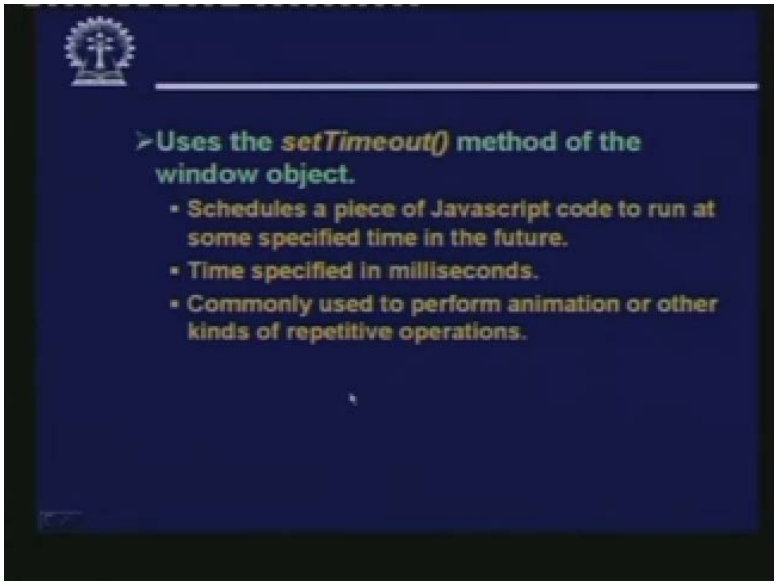
This will also give me an alert Invalid roll number 1, 2, 3, a, 5, 6, 7. So this example shows that how while suitably writing this kind of validation functions. Whenever a form is submitted I can check the individual fields in the form to the extent and try to ask user to correct the data as far as possible sitting at the current side itself, sitting at the client-side itself. Because some of the things as I had mentioned has to be done once the data has been sent at server-side. Because it may need accessing and validation from the database. Database content is not available on the client-side of course.

(Refer Slide Time: 11:39)



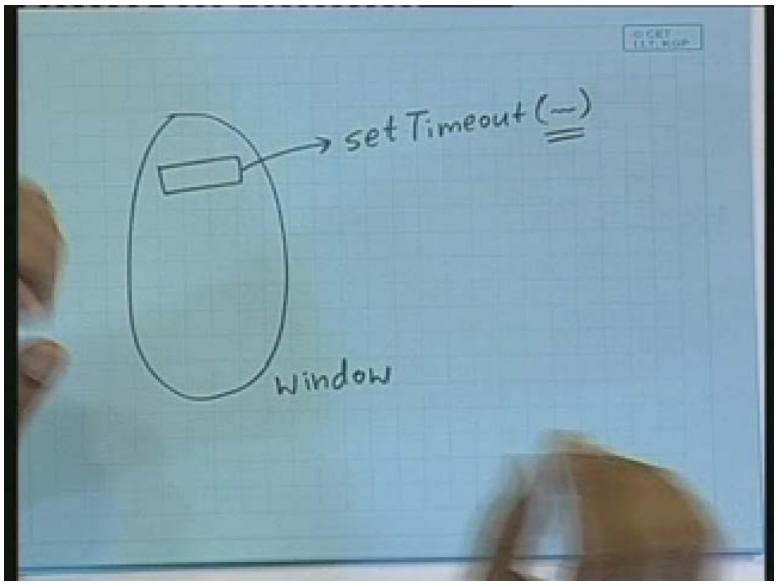
The next example we show is simple example of animation. The example is like this. It displays four images in sequence one after the other. The names of the four images are they are gif files, 1.gif, 2.gif, 3.gif and 4.gif. These are the names of the four files and the four images are like this. One is a vertical image, one is a straight line slightly slanted at 45 degrees. Another, the third one is a horizontal image and the fourth one is again a straight line slanted at minus 45 degrees. Now these 4 images are displayed one after the other. Now imagine what will you see on the screen? You will see as if the line is rotating there are 4 images. If you just display them pretty fast it will give you a feeling that there is an animation and the line is rotating. Now the example program that will show you, this will show, this animation plus it will also show you, how you can control the speed of the animation. This, the rate of rotation you can reduce or increase in speed by pressing two buttons. This also we shall illustrate. So as I have said here the speed of changeover can be controlled by the user.

(Refer Slide Time: 13:20)



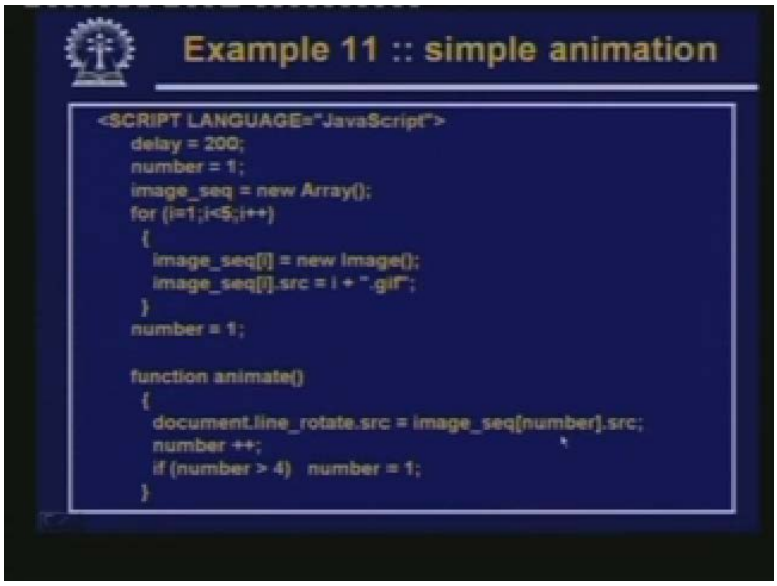
Now this example will show you that how we use the method called setTimeout. This is a method of the window object. Now let us try to illustrate what is the utility of this.

(Refer Slide Time: 13:38)



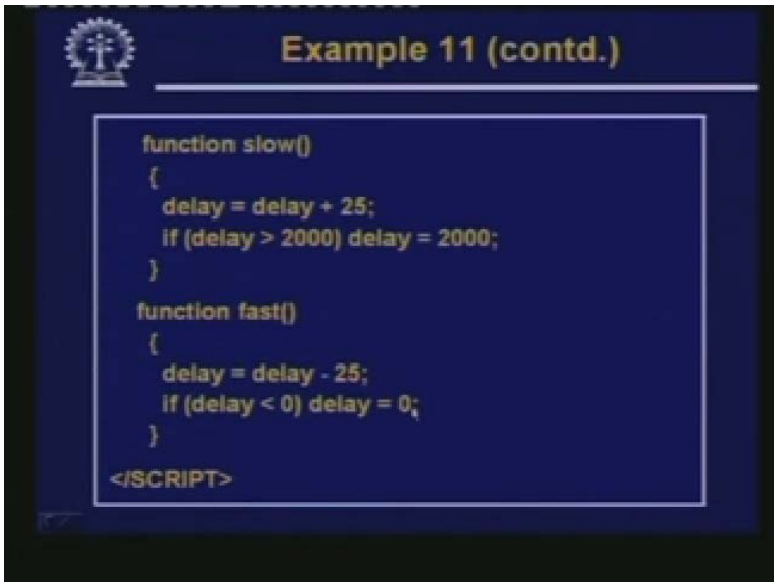
Suppose we have this window object. In the window object we have this method called setTimeout. Now we will see how we pass parameters to this. Basically what this setTimeout will do is this, setTimeout method actually specifies delay. That delay is specified as an integer number that represents the delay in milliseconds and this function is called with suitable parameters. As we shall see the effect is this a particular method can be invoked after a certain delay. And that delay you can change right. So earlier using the onClick event, you used to call an event immediately after clicking a mouse. But now using the setTimeout method what you can say well you click the method. But after one second like this, you can set a delay. So this example basically illustrates this kind of delay programming. This will be scheduling a piece of Javascript code to run at a specified time in the future. As I have said you specify an integer quantity that is basically the time in milliseconds. This setTimeout function can be used in a judicious way to either perform animation or other kinds of repetitive operations on some objects.

(Refer Slide Time: 15:29)



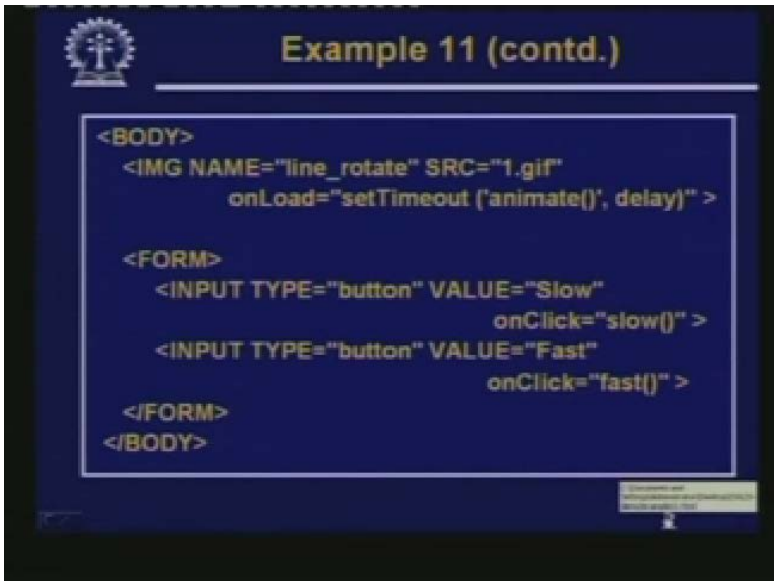
Let us see the example. This is the full example this is the first part of it.

(Refer Slide Time: 15:34)



This is the second part of it and this is the body and form.

(Refer Slide Time: 15:40)

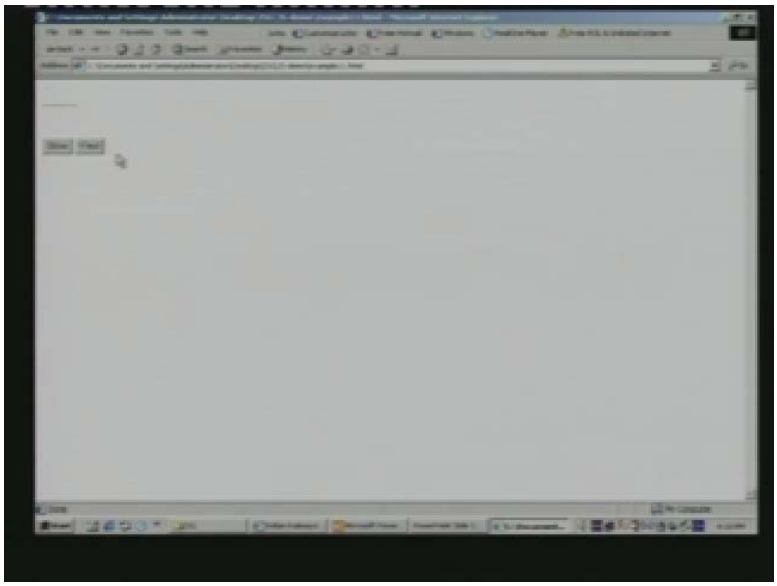


So let us start with the BODY HTML document part first. Here you see in the IMG. Here IMG is the tag which displays an image. This we already know. Now this particular object I am giving a name called line underscore rotate. Line rotate the source of the image is 1.gif and what we are saying here the event is onLoad. For IMG tag this onLoad event will be activated every time you load an image. For example to start with that 1.gif is what I have specified. So whenever that 1.gif will be loaded on the screen automatically, that event will get triggered a also and what the event will tell event is a call to the setTimeout method. There are two parameters. The first parameter is a user defined method called animate which actually been changing the source attribute to the next image we will see how and a specified amount of delay. So idea is that whenever you are loading an image you are also getting ready to load the next image after a certain specified period of time. This will help you create the animation and for varying the speeds of the animations you have put two buttons.

One we are calling Slow other is calling Fast. So the slow one if you click you will be calling a method called Slow fast will be calling a method called Fast. Now first let us look at the Slow and Fast methods these are very simple. The slow method it implements the value of a variable called delay by 25. But if it exceeds 2000, it remains saturated. There it does not exceed any further fast does the reverse it decreases delay by 25. But it does not allow delay to fall negative delay 0 is the minimum. Now let us look at the main function. This animate and some initialization, the SCRIPT LANGUAGE Javascript it starts with an initial delay of 2000 number is a variable which we use for this image. Because as I have said, the four images are named as 1.gif, 2.gif, 3.gif and 4.gif number is that 1, 2, 3, 4. So initially since it is 1.gif it is set to 1 and I am creating an array and I am storing it in image seq and in a loop I have 4 images. I am assigning a new image object to each element of this array. Image seq is an array of size we have not specified of the type we have not specified.

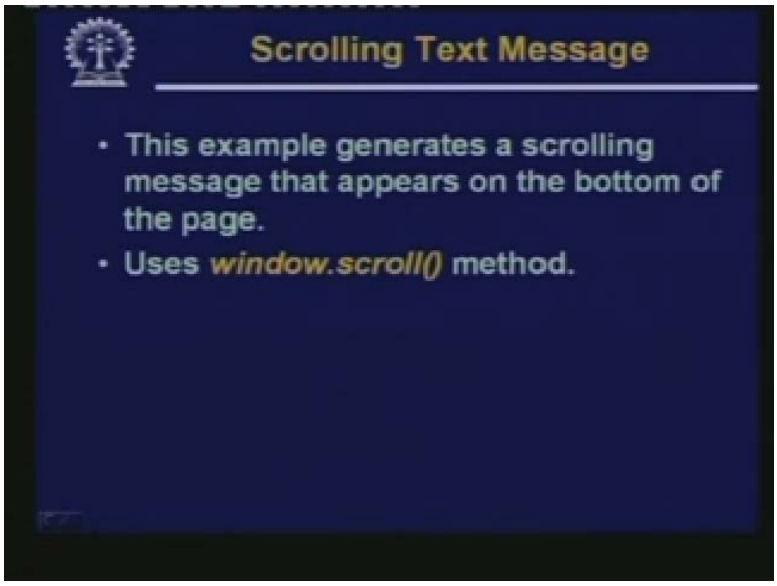
But here we have saying that the type is of type image. Image is a built-in type. Now in the image array there is an attribute called src source. There we are putting this, I concatenate dot gif. So in this loop the first one will get 1.gif, second one will get 2.gif and so on. Number to start with is one. Now in the animate function you see what I am doing here. Line rotates if you recall this is the name of the image. Just see in this image line rotate was the name of the image attribute. So there is a source attribute of line rotate src which means which image is currently being displayed. Now I can directly change this source attribute of an image to load another image. So you see that even the animate function I go and changing src attribute of that image then it is the same image. Location is the same but the image is going on changing. So this kind of animation I can implement very easily and every time is doing this. I change the number to one more number plus because I am accessing a particular element from this array and if the number is greater than 4, I again set it to one. So let us see this program working.

(Refer Slide Time: 20:22)



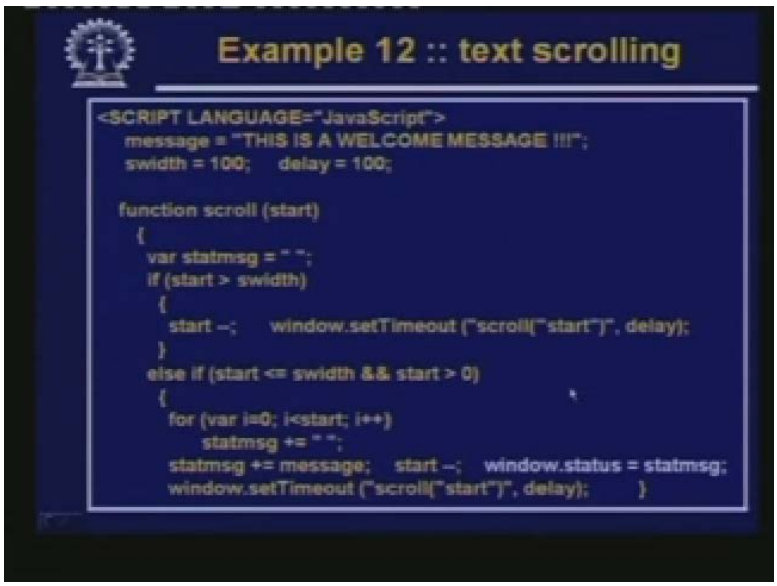
So use see there is a small arrow which is rotating, if I press Fast the speed will go on increasing. It is becoming faster and faster. But if I press Slow it will become slower and slower. So in this way you can see, you can create very simple animations.

(Refer Slide Time: 20:52)



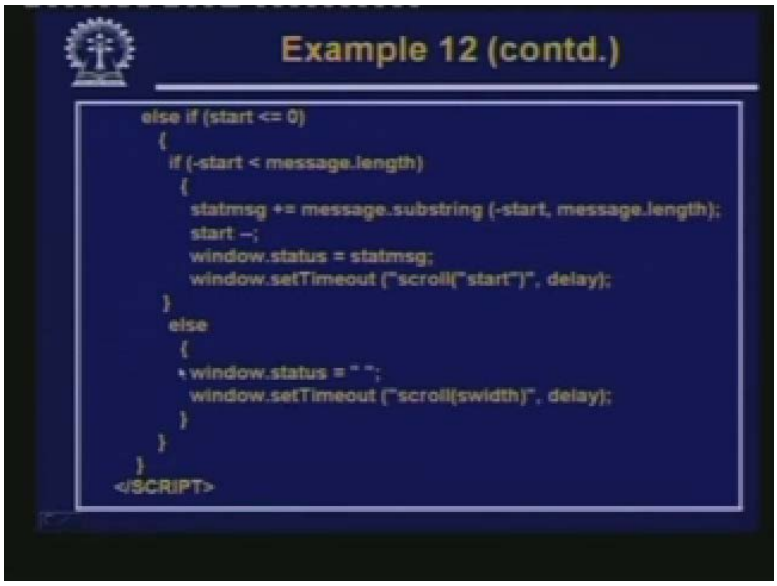
The next example will look at is that of creating some simple scrolling text messages. Now sometimes we need to have some messages scrolled either on the main BODY of document or on the bottom of the screen the example that we show the scrolling will take place at the bottom of the screen. Now here we see would be using a method called scroll which is a method you can use for scroll which is the mechanism for scrolling.

(Refer Slide Time: 21:31)



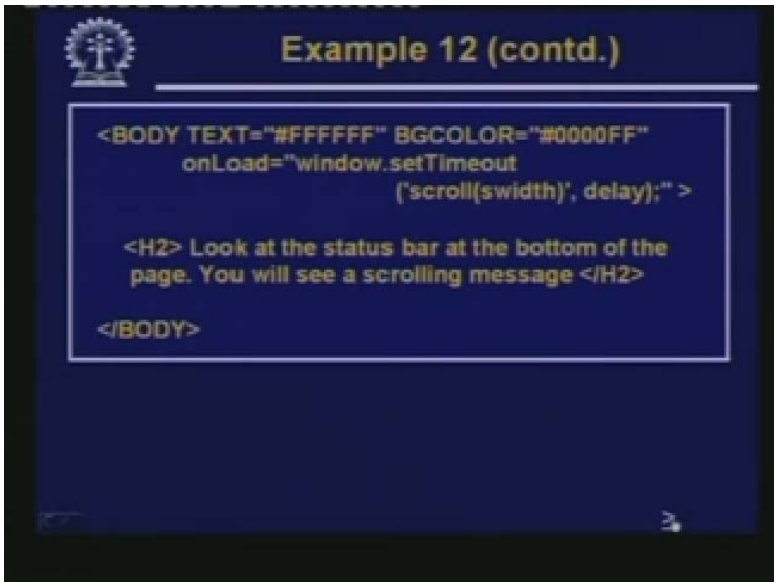
So let us see the program. This is the first part of it.

(Refer Slide Time: 21:37)



This is the second part.

(Refer Slide Time: 21:40)



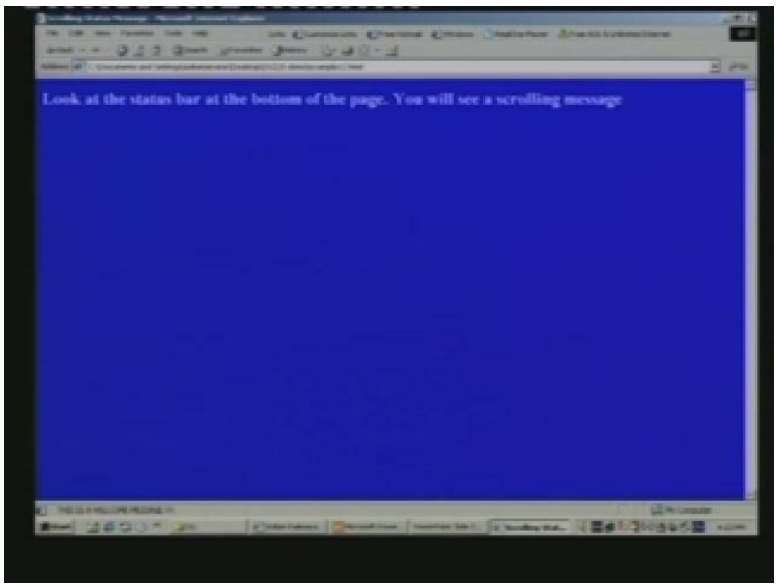
And this is main HTML part. Let us start again with the HTML part. Here the TEXT BODY we have set as white background color is blue, onLoad window.setTimeout you are saying whenever the page is loaded, this is the event. We are immediately setting a Timeout on this window object and the Timeout we are specifying scroll within parentheses swidth. We will see the significance of this and delay is the amount of delay so with this delay you call this. Then there is a message "Look at this status bar", etcetera. Let us see what is there in the BODY of the code. Message is

a string which we have initialized “THIS IS A WELCOME MESSAGE”, swidth is the width of the string maximum by default 100 we have taken. And delay also we have taken 100.

If you change this value the speed, scrolling will differ in the function scroll well here I am not going into details of calculations but actually what we are doing. This is status message initially set to null. If the starting point is greater than this swidth 100, then you go on decrementing to start and you again set window start tama Timeout scroll from start. Because you see when you are displaying this scrolling message, there is a beginning portion, there is an ending portion. If during scrolling it crosses start, it will again have to start from that point. That is why what I am saying if start exceeds the width that is allowed you start decrementing start by one and you scroll starting from that position. This is the beginning point from where you start the scroll.

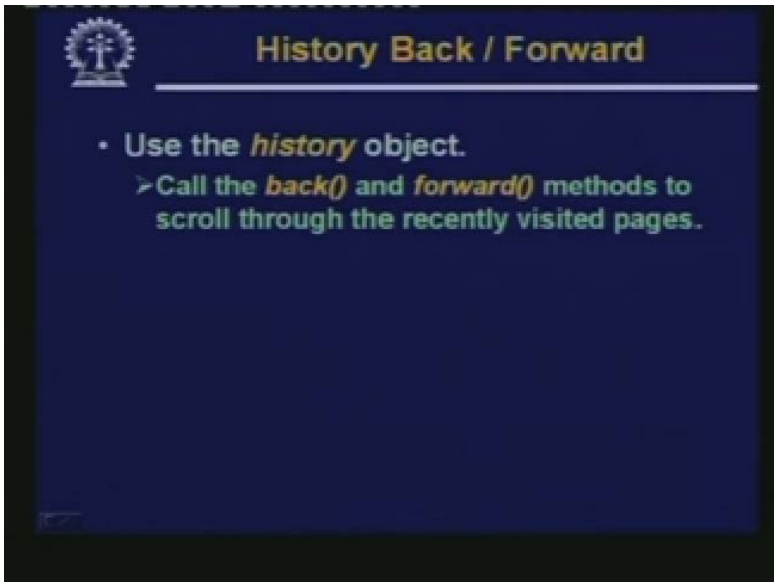
Else if it is less than swidth and start greater than zero, these are all calculations you can find out for the effective scrolling on the screen. I am not going into detail on this. But the place where the message actually gets displayed on the status bar is here. Status is an attribute of the window object window.status is whatever gets displayed on the status bar on the bottom. So the string I calculate based on a certain point in time. Then I assign it to the status bar. What happens is that every time I make some modification to the message I go on assigning and modification is such that it will get shifted by one place. So I will effectively see that the message is scrolling on the status bar. So every time I do it I also set the Timeout to again call scroll after a certain delay. So again this continues if it crosses the other end we do the same thing. So I am not going into detail of it. Let us see it working.

(Refer Slide Time: 24:56)



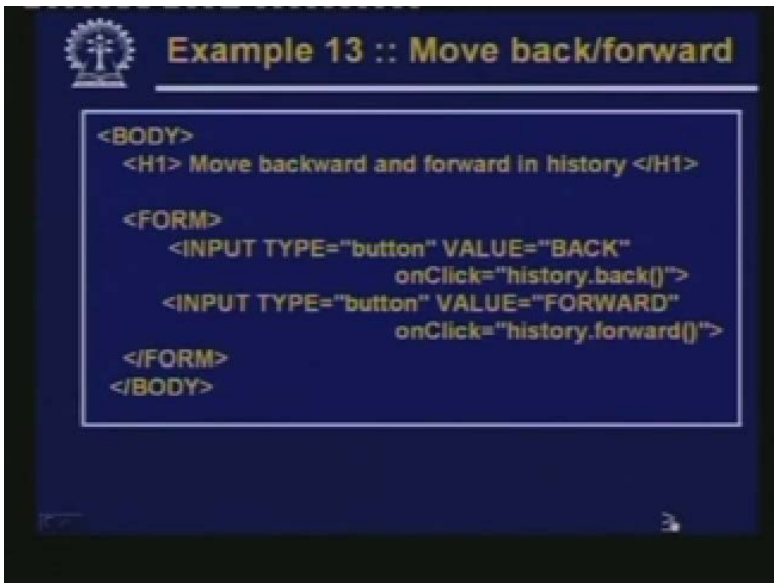
You see at the bottom out here, there is a message which is going on scrolling. This is the welcome message. This is the message, see it goes out, goes out, goes out and again it will still start coming out. Here the whole width is 100; it starts from this 100 100th position. So this is how it is implemented.

(Refer Slide Time: 25:23)



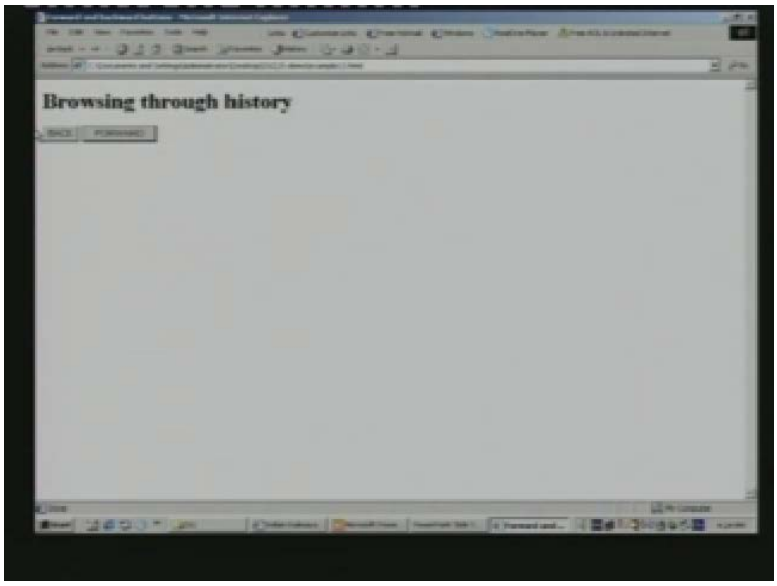
Now let us take an example which will talk or which will browse through the history of pages viewed by the browser. So here we are talking of the history object. Now in the history objects the methods we need to look at are back and forward. Now as I mentioned earlier this back and forward methods can be used to trace through the history of the URL's of visited pages either in the backward direction or the forward direction. A very simple example to illustrate this.

(Refer Slide Time: 26:10)



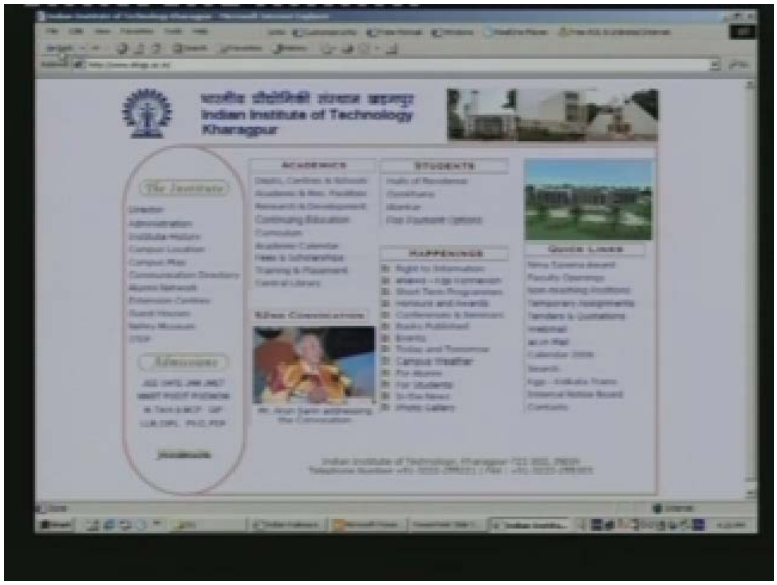
There is a form, there are two buttons. The first button is labeled as BACK; the second button is labeled as FORWARD. Now what I am saying is that if I click on the first button, onClick I am invoking a built in method called BACK which is a part of the history, object history.back. But if click on the other button I am calling history.forward. So let us see.

(Refer Slide Time: 26:47)



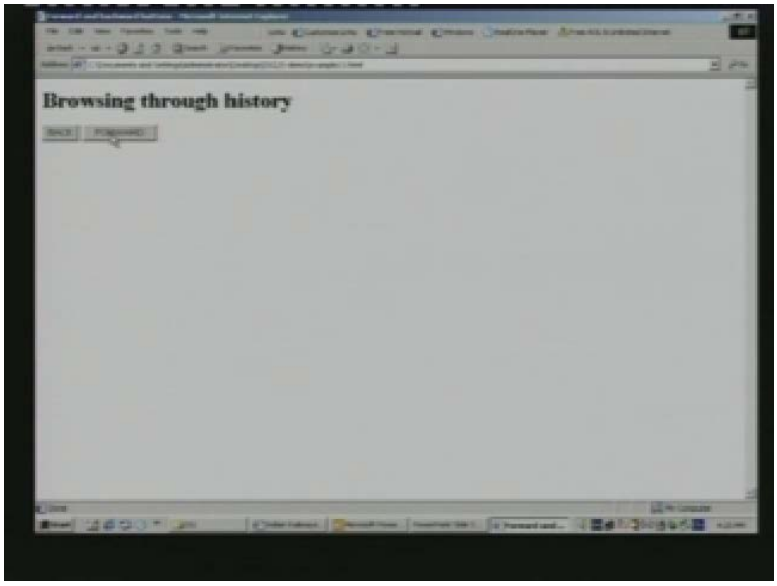
This is the screen, as it looks, but since it is open, there is no history. If I press forward and backward nothing will happen. Backward will make me come to the previous place from where I had gone. Now let us suppose from here I type a URL I type.

(Refer Slide Time: 27:13)



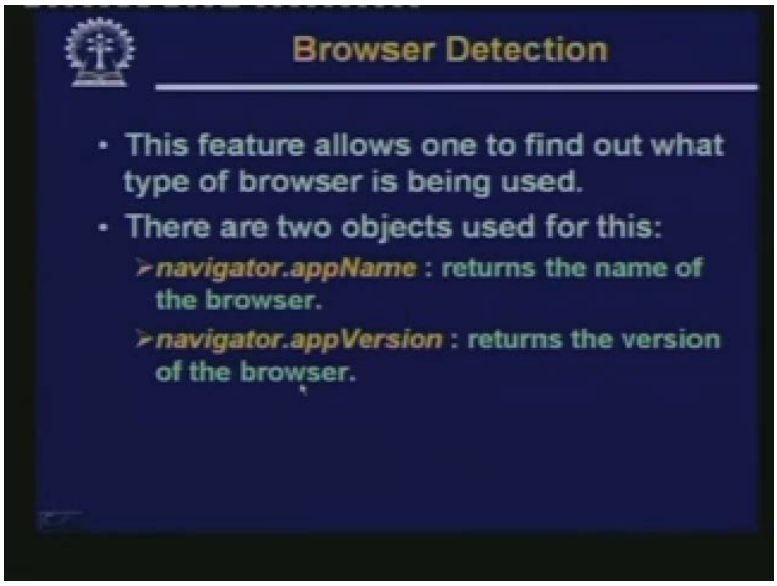
The URL of the IIT Kharagpur website. So I go to the IIT Kharagpur web site. Now I press the back button on the browser to go back to my original page.

(Refer Slide Time: 27:20)



Now here if I press Forward, I should go back to the next page in the history. I press Forward, so you see I go back to my page here. Now if I press back I again come back here. So like this in many pages when you design a page you also have a mechanism to browse through the sequence of pages you are going through. Suppose you have a web site, you create a web site where you have created some online learning materials, where you are going through the chapters or different sections of a book. So as you go forward, sometimes you may have to back and see what as the previous page. So there can the forward and backward navigation buttons as part of the page itself which if you press will make go back and go forward.

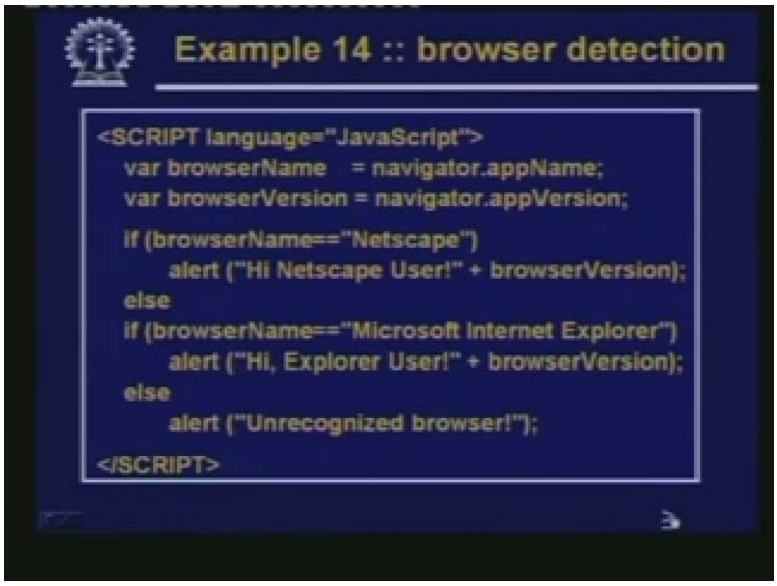
(Refer Slide Time: 28:13)



Sometimes you may want to what kind of a browser the user is using. Because you now that there are so many facilities or features or tags that are available in HTML. There are so many other utilities that are supported by browser that are not supported by some other browsers. So a document which you have created, it may load and get displayed very nicely on Internet Explorer. But it may not display properly on a navigator browser for example. So in order to take care of this problem it would be nice if the document itself can be, you can say powered with some Javascript code which can check or find out the kind of browser and according would expand some code. Or write in general some HTML code which can be properly handled by that particular type of the browser.

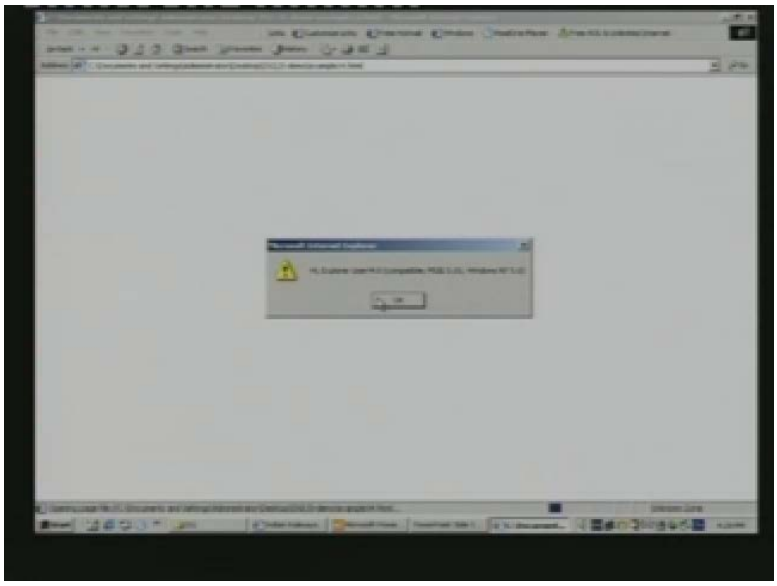
Now let us see how we can do? This how we can detect the type of browser we are using. Now whenever we are using object that we are using we would be using is called navigator. Navigator is the kind of object. This traditionally, this name navigator comes from Netscape because these objects are initially developed by them. But actually navigator means a browser kind of object. An appName and appVerison are two attributes of it. appName actually returns the name of the browser and appVerison returns the version of the browser. Whether you see this name or version is not just a very simple string. But a very complex string which consists in addition to the browser name and version a number of other information also. So let us look at a very simple example.

(Refer Slide Time: 30:24)



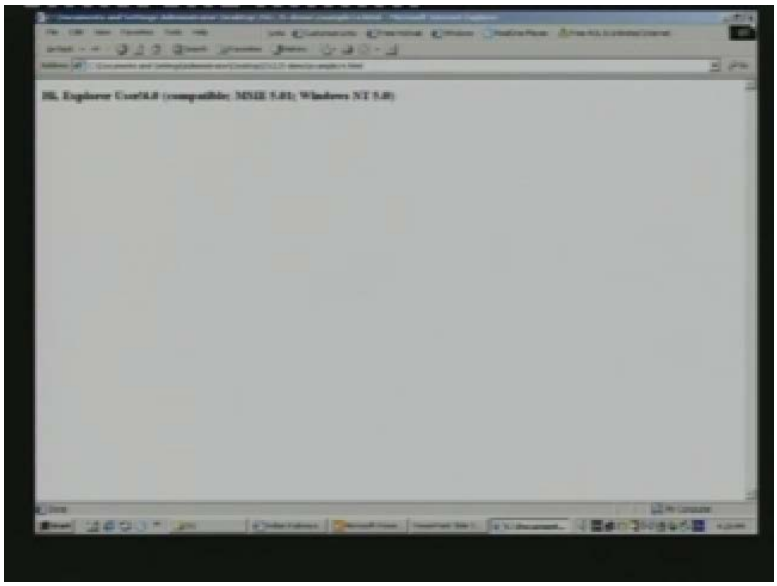
To illustrate, so I am showing only the Javascript portion. So first we are extracting the appName portion of the navigator. We are assigning it to browserName. Secondly we are storing the appVerison attribute value into browserVersion. Then you are checking the name whether it is Netscape or Microsoft Internet Explorer. These are the default names of the browserName. So if it is Netscape then we are giving an alert “Hi Netscape User” and concatenate with the browserVersion. Or otherwise we give an alert “Hi Explorer User” and browserVersion. Otherwise we say it is an “Unrecognized browser”. Now here we show an example here we are using Explorer, so we should get an alert like this “Hi Explorer User” and concatenate with browserVersion.

(Refer Slide Time: 31:31)



You see, as you run you get an alert. Here the font size is small I will show you “Hi Explorer User” “4.0” compatible MISE Windows. Well I have just inserted a document writeln after that, so that you will know what is exactly shown here after pressing you can see it.

(Refer Slide Time: 31:52)

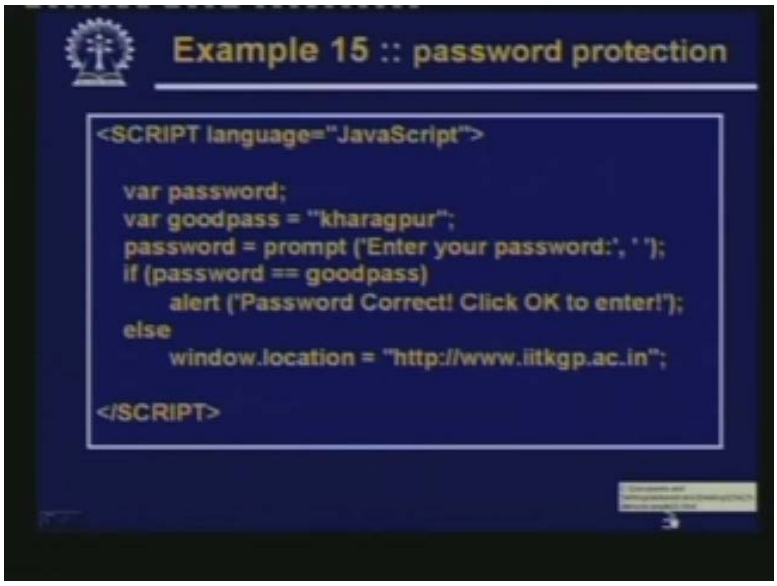


Like this Hi Explorer User, the version is “4.0” compatible, MSIE “5.0”. One these are the versions Windows NT “5.0” which version operating system is running. So in addition to the browserVersion you get a lot of other information.

(Refer Slide Time: 32:30)

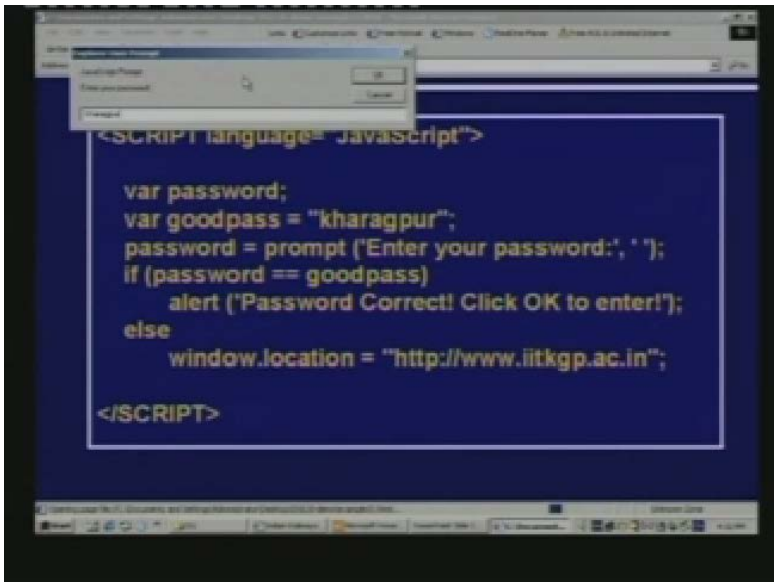
The next example shows that how we can make or means how we can carry out simple password protection in your document. Because this processor password protection is sometimes quite important, this is a very simple SCRIPT that is been shown an actual password protection. SCRIPT will be more complex because you would not like to store password in plain text as here. So let we first explain this one password is variable you are defining good password is say “kharagpur”. This is my good password, then you are calling a function called prompt. Prompt is another kind of a popup box. A feature which Java supports in prompt, this Enter your password is the string that you have passed and what will here is that when this prompt box will come, there will be a empty box to be coming below where you are prompted to typing the password. So prompt box is something similar to alert, but something more also here. The user is supposed to type in something. So here for instance I am asking the user to type in the password.

(Refer Slide Time: 33:44)



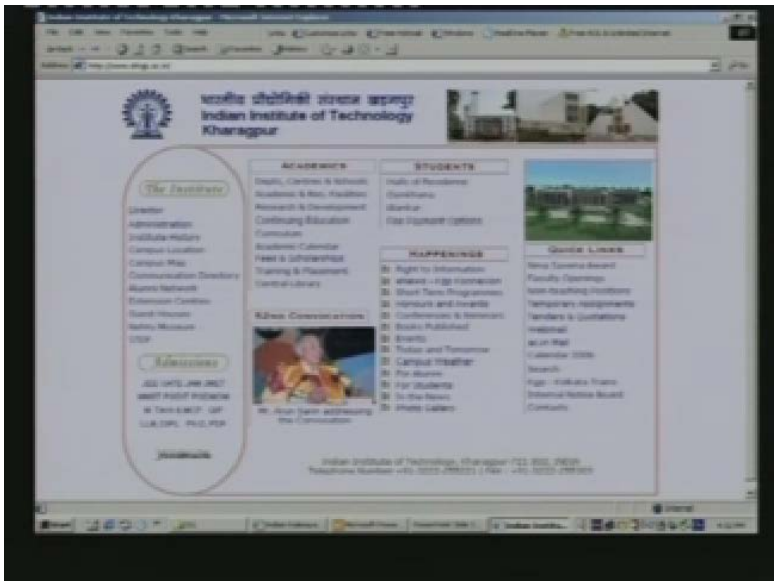
So now the password is compared to this good password. If it is equal then alert password correct, click to enter. If it is not correct then you are setting window location to iitkgp. So if the password is incorrect you are taking the user to the iitkgp website. Let us see it running.

(Refer Slide Time: 34:09)



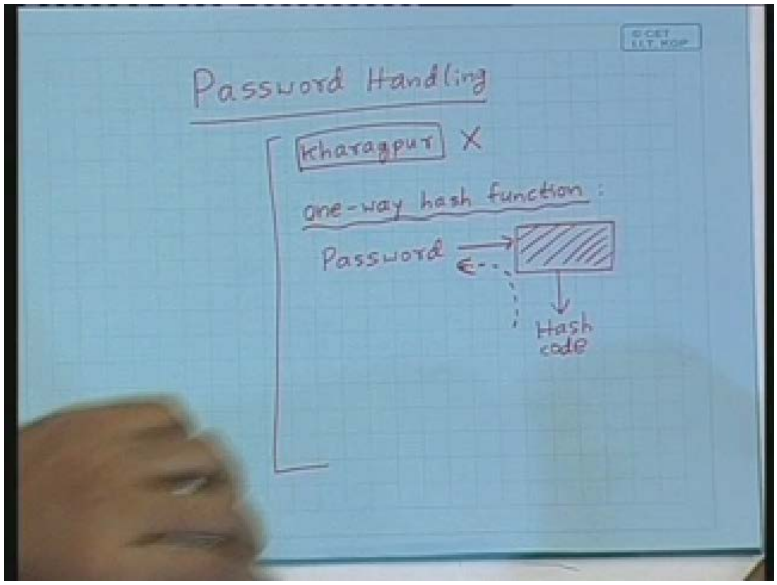
So here when you run the code you get a [34:19 word not clear] like this, so here you are prompted to type in a password; kharagpur is the password I am typing, the kharagpur I press OK.

(Refer Slide Time: 34:29)



So I go to the page. So this way you can create simple pages link password. But in this context let me also tell you one thing.

(Refer Slide Time: 36:35)

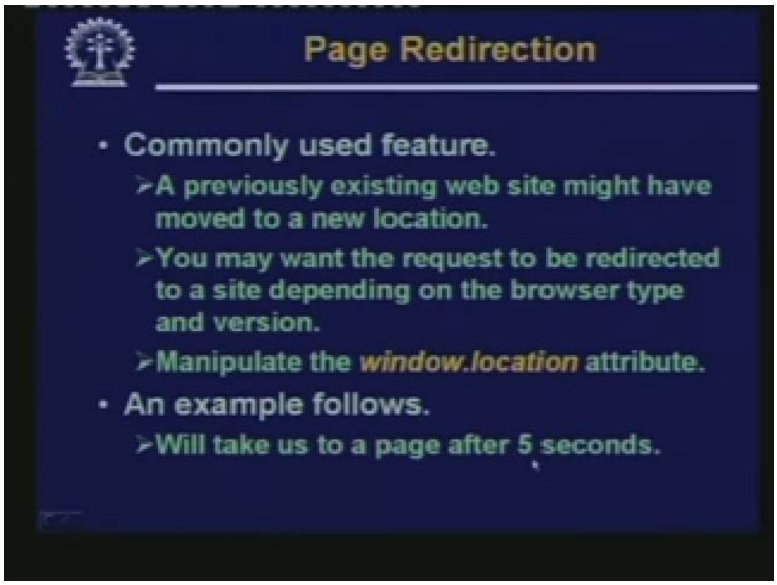


That for more complex password handling. You need to do something more, that something more is like this. See here for instance I have stored, the password called kharagpur as a plain text as part of the document. Now if you do this a hacker or intruder can easily get hold of the password. So is not what you are supposed to do. Rather what you will be doing is that you will be using something called a one-way hash function. What this one-way hash function will do is

that the Password that you type, this Password will be sent to a function which will be generating some hash code. Now this function can be very easily written in Javascript and this function can be executed at the client-side itself. So what you do here is that you are not storing the password or you are not showing it to anybody.

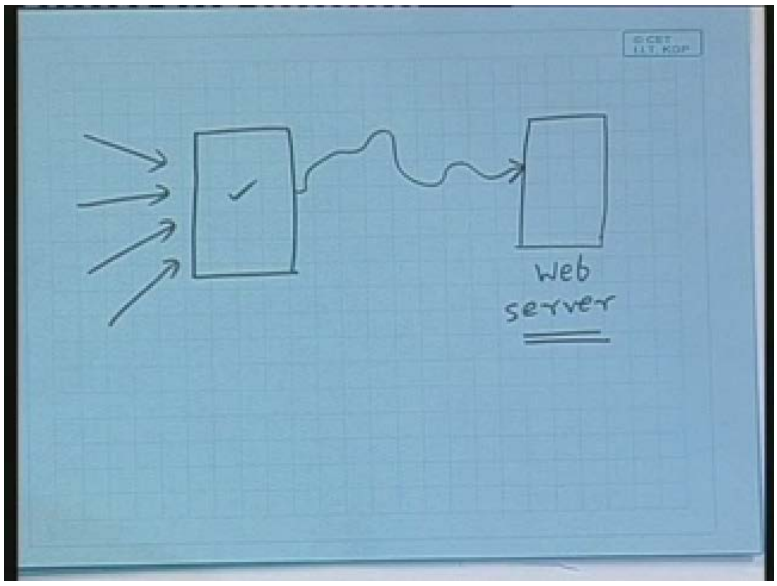
You are actually transferring the password in to a hash code which is a cryptic code which is very difficult to, you can say memorize or remember. Even if you remember it, this is called one-way hash function because reverse process is not possible. Given the hash code you cannot get back the password. So now the idea is this instead of storing the passwords you can store the hash codes as part of the browser. When the user wants to wants to authenticate himself or herself user will be typing in, the password and that hash function will be computed and the computed hash code will be compared. In this scheme in no way can an intruder steal the password. So this is what is done normally for good password protection scheme. The example that we have taken is a simple one.

(Refer Slide Time: 37:16)



Now let us look at the feature which we find quite often in many of the web sites. This is called page redirection. See it happens like this. Suppose I have advertised the URL of a particular page where I am supposed to provide some information to a larger community of users.

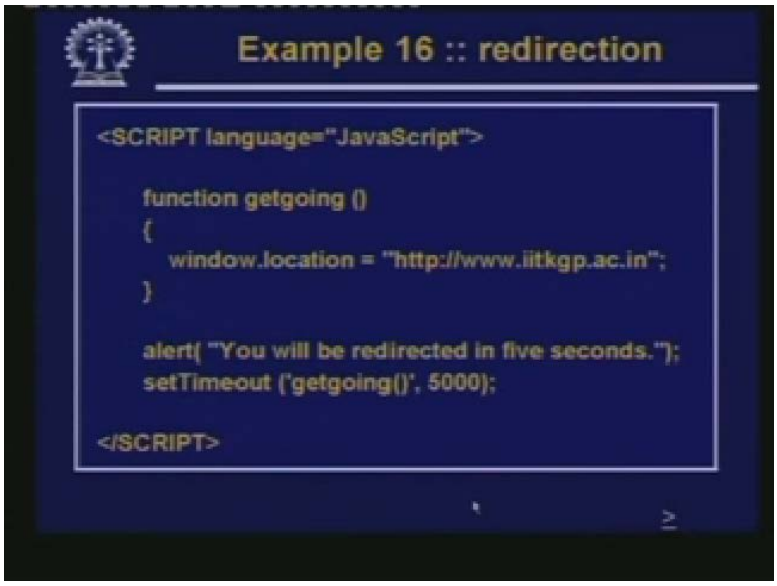
(Refer Slide Time: 37:50)



Now what the situation will be that, this is the website which I advertise so that many users will be trying and accessing this website. But what might happen is that the actual information that I want to advertise, I want to share with all these users that may be residing in some other webserver. This kind of an event happens quite often in many situations. For instance we also had such a situation some time back where you are holding a conference and our conference website with all the advertisements data everything details that we had hosted in one of our internal web servers. But to the outside world the URL that was advertised was some other web site; not our internal web site. So what could be done here, what was done was like this in that outside website which is advertised, a very small Javascript controlled HTML code was written whose sole purpose was to redirect the requests back to our internal website from where the actual information can be obtained. So from this diagram actually what I am saying is that whatever requests are coming, they will be redirected back to this web server.

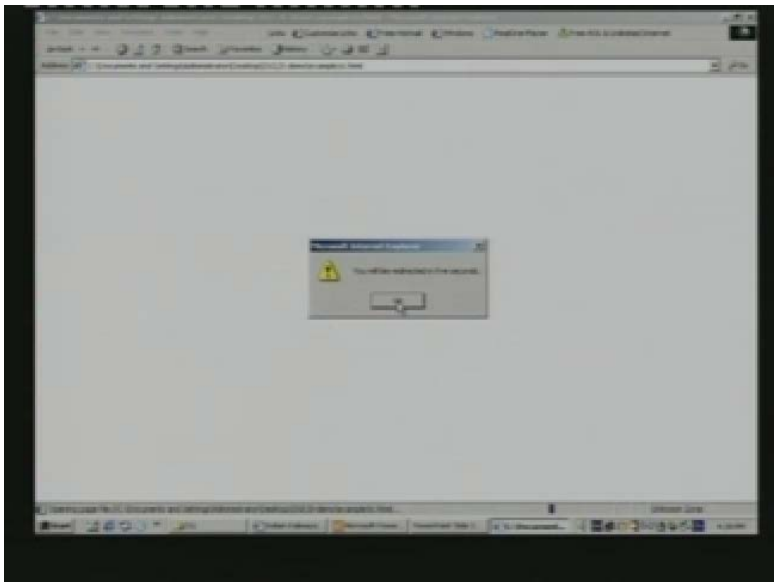
And this redirection can be done very easily in Javascript and we shall be showing through an example. How this can be done? So as I had said, this is a quite commonly used feature and there one reason I had mentioned. The other reason maybe there was web site which was existing. But due to some reason the website has moved to a new location, maybe due to administrative reasons the administrator has moved. So you need this redirection facility. So users would be sending request to the previous URL you want. A means to redirect the request to the new location. Now redirection can be very easily done by manipulating the window.location attribute. Because as I had said earlier that window.location, this attribute mentions that which URL is presently displayed on my screen? So if make some changes to this URL whatever is displayed on my screen will also change. So for redirection I have to do this. But in addition I will also show how we can do this after a specified delay. This redirection will take place after specified delay example that I will show the delay will be 5 seconds.

(Refer Slide Time: 40:59)



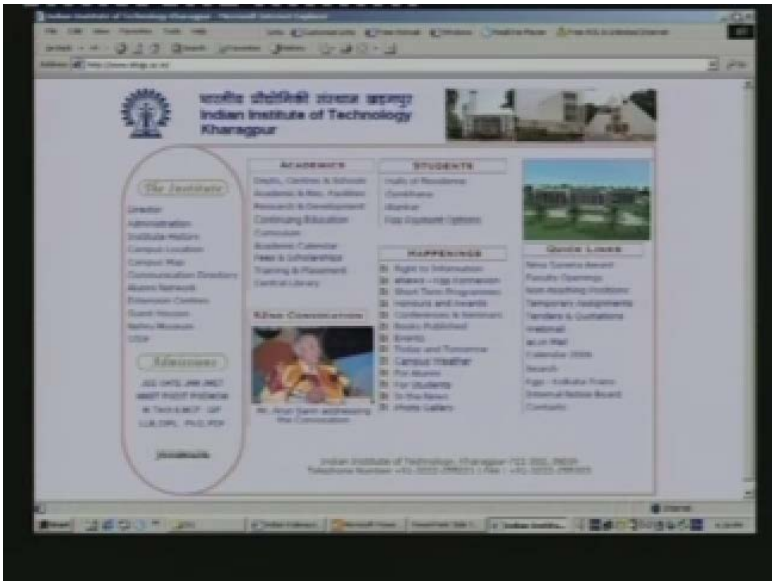
The example is like this. So we have written a very small function a single line function called getgoing where we have set window.location to our iitkgp website. Where we want to redirect and before doing that we are sending an alert to the user “You will be redirected in 5 seconds” and after the user presses OK, there is a function setTimeout which is invoked again. So here we are calling this getgoing method after a delay of 5000 milliseconds. So effectively the alert box will be displayed and once you click on you will be redirected to IIT Kharagpur website after a delay of 5 seconds. So let us see if this works.

(Refer Slide Time: 41:56)



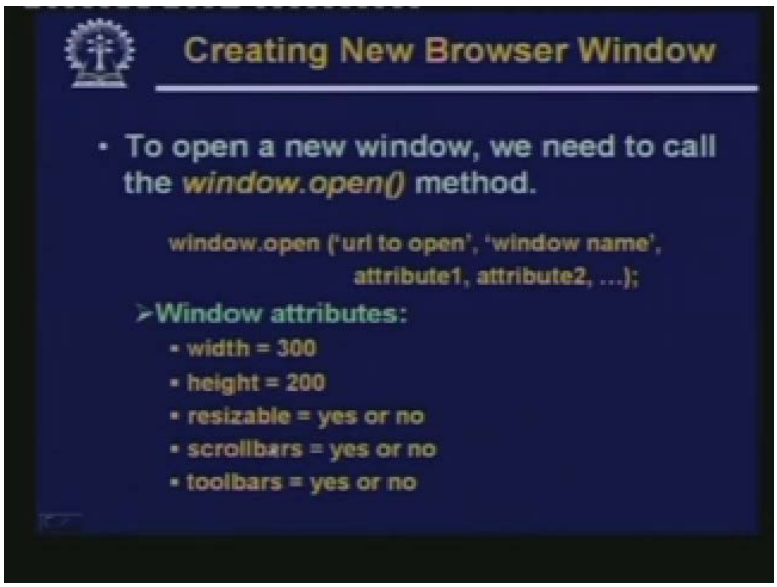
You see that the alert box gets displayed “You will be redirected in 5 seconds”. Let me click on I click on and let me wait. It should go to IIT Kharagpur website in 5 seconds, you see it has gone after 5 seconds.

(Refer Slide Time: 42:11)



So this is how you can have redirection facility based on some delays. This is also a pretty useful facility that you can have.

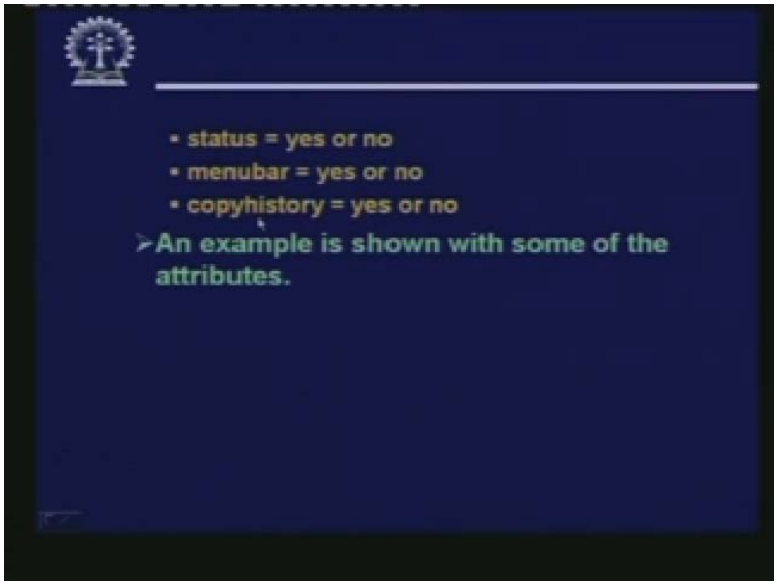
(Refer Slide Time: 42:30)



Now let us take an example where you will see how we can create a new browser window under Javascript control. Now you know that you have seen that in many websites there are so many popup windows that come up well whether you like it or not. There can Javascript code which you have downloaded along with the HTML document you have got and that Javascript code can have lines where some new windows are getting created. So creating new window in Javascript is fairly easy. You will to call or invoke the open method of the window object. The syntax is like this, window open has to be called with a number of different parameters. The first one and second one is mandatory. The first one says you have specified which URL you want to load on the window.

Because just creating a window is not enough you will also have to say in that window, what is the HTML page you want to load and display so the URL also you need to mention. Second one in order to refer to the window. For example if want to close the window later you will have give a name to the window followed by a list of optional attributes. Now there are many window attributes which you can give. I am just mentioning them, width and height specify the width and height of window in pixels. For example you can give window width equal to 300, height equal to 200. For ins example, resizable is another attribute which can be either yes or no. This tells you whether it is possible to change the size of the window later or not by drag dragging the corners. Scrollbars again yes or no whether you need scrollbars with your window. If you say yes scrollbars will appear automatically otherwise scrollbar will not appear.

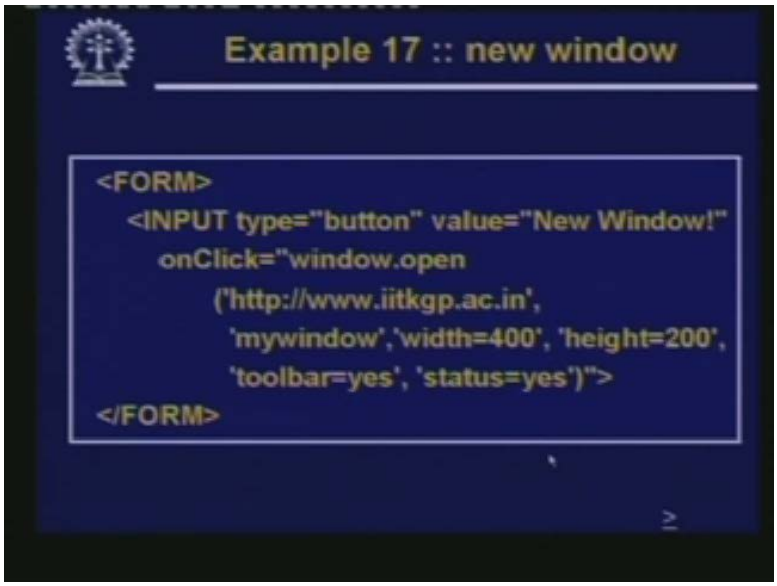
(Refer Slide Time: 44:53)



Toolbars again yes or no, status bar again yes or no, menu bar yes no. So means all these optional bars which you typically see on a window on a browser window when you create a new window, you can tell whether you really want this. This bars or these features or not you can say either yes or no and the last attribute says copyhistory yes or no. Copyhistory means say from a master window you are creating a new child window. If you say copyhistory equal to yes, then the URL history that was existing in the current window that will get copied into the child

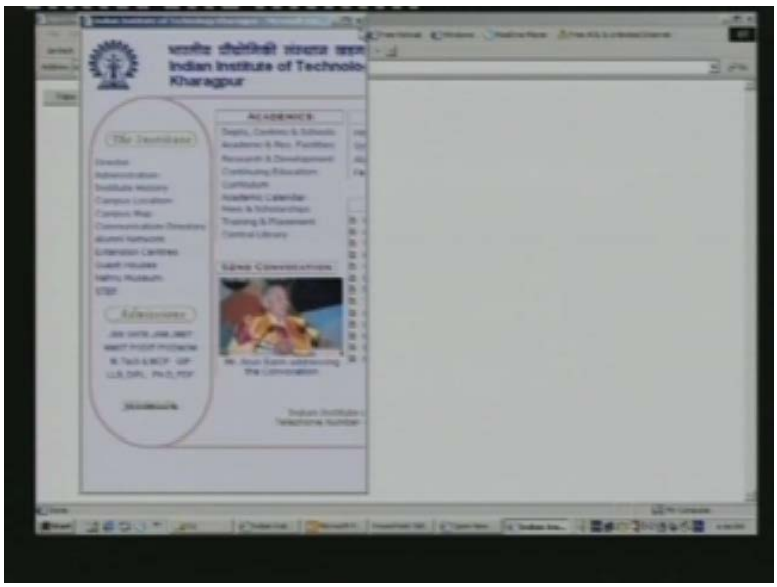
window. If you say no then the child will start with a null history to start with. But subsequently as you go on the history list will start growing. So I show an example here.

(Refer Slide Time: 45:56)



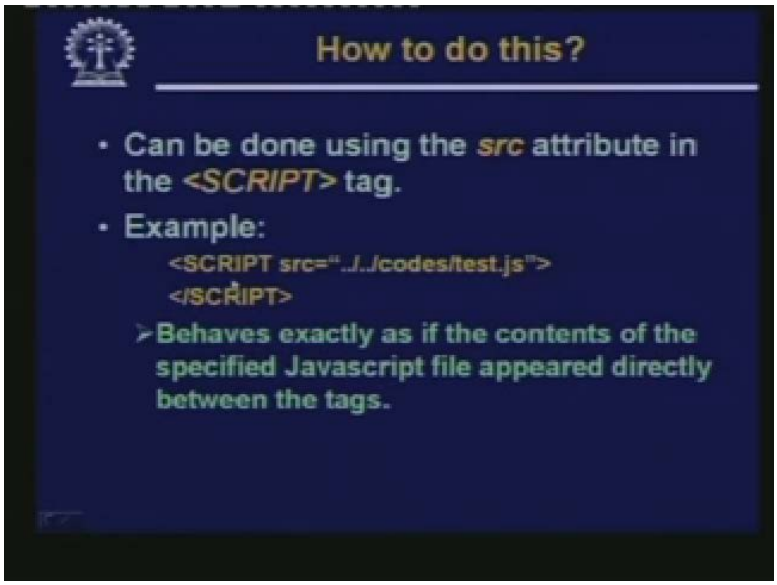
Here is a simple example of a form where there is a button a single button. The label of the button is New Window and if you click it onClick, here we are calling this window open method with some parameters. We are again invoking the IIT Kharagpur website. Name is mywindow width is 400, height is 200, toolbar yes, status yes, all orders by default will be no. So let us see this running.

(Refer Slide Time: 46:41)



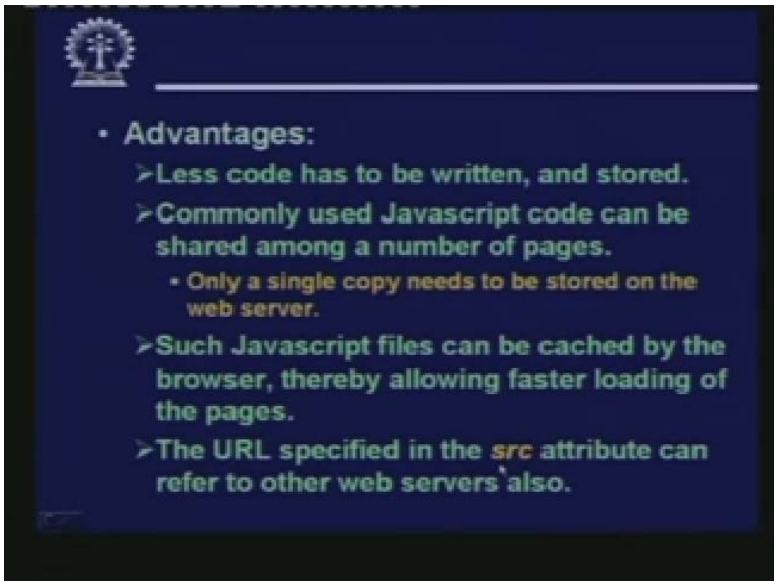
Let us click on this button and see what happens. So you see a new window comes up. So here since there are no scroll bars I cannot scroll and see the remaining part. This is also a non-resizable window. I cannot drag; I cannot get hold of the corners and drag it. So in this way you can create windows of any desired characteristics. So far in the examples we had seen we had shown that the Javascript code is embedded or included as part of the HTML document itself. But we will now show how we can add external Javascript files within in an HTML document which will also be loaded along with the HTML file when you loading it. So here the point is how to include external Javascript files from within an HTML doc.

(Refer Slide Time: 47:42)



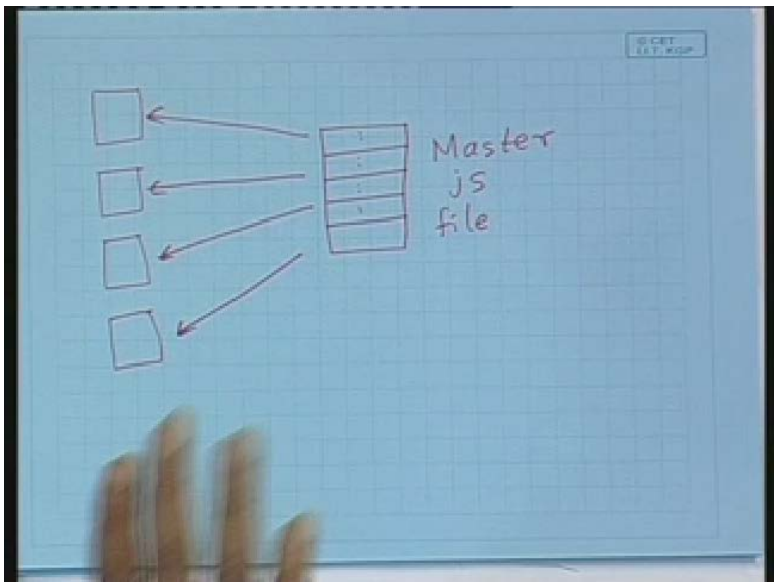
So first for this we need the source or the src attribute. In this SCRIPT tag, the usage is like this SCRIPT. Well language equal to Javascript is optional I told you. So you can have language equal to Javascript. But in addition you can also have SCRIPT equal to source dot dot dot slash dot dot. This is we are specifying the path and we are specifying the name of the Javascript file. So this is equivalent to the situation where the contents of the Javascript file is included between the begin SCRIPT and end SCRIPT. Both are equivalent. So instead of including all the lines between begin and end SCRIPT's you can put the contents in a file. You can store them as a file and include them using the src attribute to link to that particular file. So this, as I had said, this will behave exactly in the same way as if the contents of the specified Javascript file appeared directly in between the tags.

(Refer Slide Time: 49:02)



There are certain advantages here. Less code has to be written and stored. Well. This comes from the fact that many different pages may be having similar requirements. So you may have to write similar Javascript functions for a number of different HTML pages. So instead of writing them along with each and every such document or page if you can store or create a master Javascript file and include it from many different files.

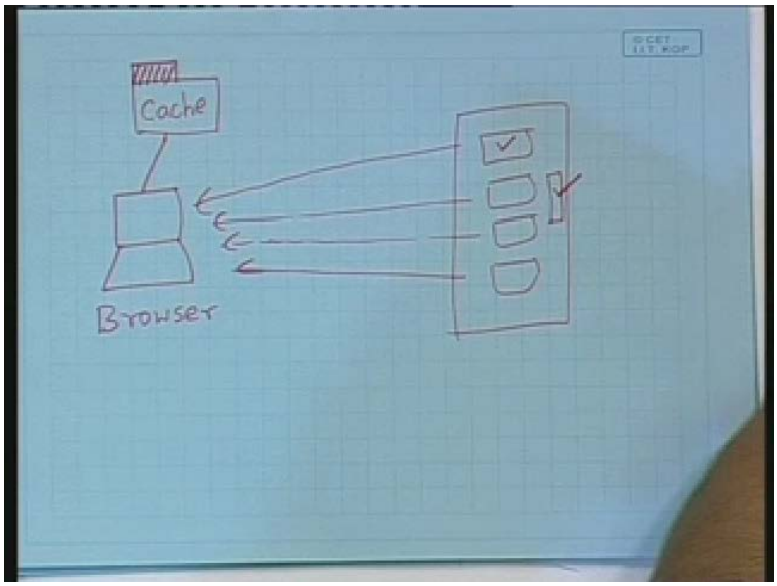
(Refer Slide Time: 49:46)



It would be fairly simple like what I am saying is that suppose you have a Master Javascript file js file, which you have created which contains the commonly used Javascript methods and

functions you can have several web pages and all these web pages can include this master Javascript file with it. So this will obviously reduce the number of times you need to type the same code in the different files. That is why we say that less code has to written also less code has to be stored. So as I had said the Commonly used code can be shared. You need to store only a single copy of this shared code on the web server. There are some other advantages. It is faster.

(Refer Slide Time: 50:55)

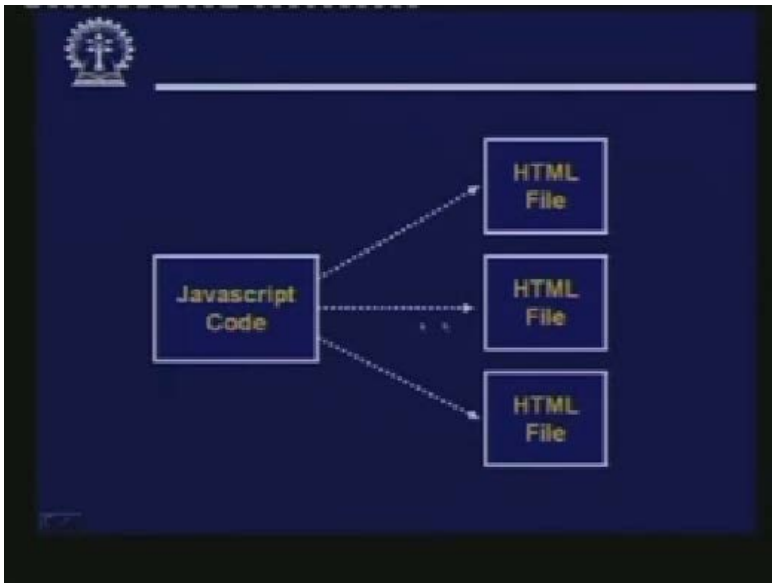


Like what I am saying is that suppose here I have a client, here I have a browser. Through the browser I am browsing. Now as you know that as part of every browser, you can configure some local cache. Cache means these are the recently visited pages or the documents from the browser in question. So if a particular document is already there in the cache, if you still want to access it, the browser typically will directly pick it up from the cache and not send an explicit request and get it again. This will make the access faster. Now here what I am saying is that suppose the browser is accessing several web pages from a server. The browser will be accessing these different web pages from the server and it is possible that along with all these servers there is a common Javascript file which they are sharing. So what will happen is that when the first time the first page is downloaded, this Javascript file will also get downloaded and the Javascript file will be stored on the cache.

Now for all subsequent page accesses all the other subsequent pages also will be having a link to the Javascript file. But that Javascript file is already in the cache. So access will be much faster you need not have to go back to the web server every time to load the Javascript file. This is what we mean by saying that can be cached there by allowing faster loading of the pages. And secondly the src attributes specifies a Javascript file. It is not necessarily true that the Javascript file need to reside on the same web server as the pages from where they are referred to. Javascript files can be loaded on a web server xyz.com my web pages which link to that Javascript file maybe storing. Some other web server abc.com, so in that this is more general

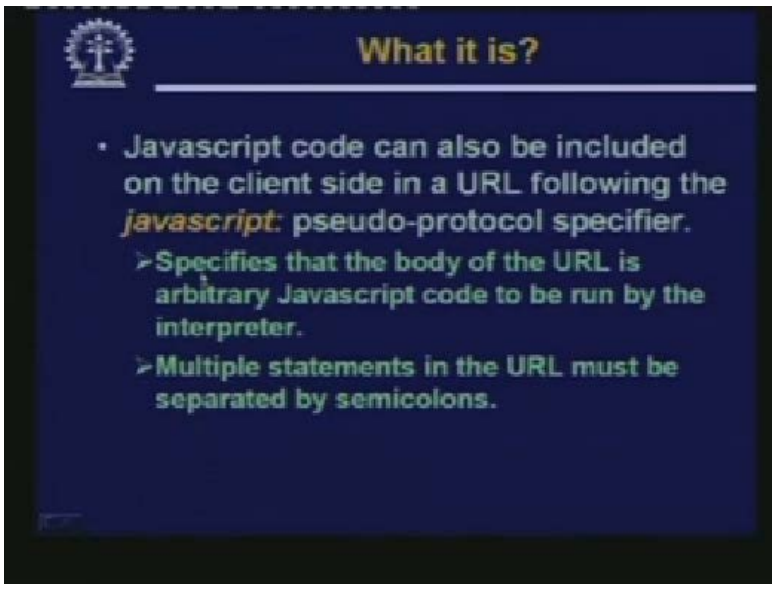
over the internet. You can have a library of Javascript code in one place and from all other pages you can have linkages to them right. So in that sense this is quite flexible.

(Refer Slide Time: 53:40)



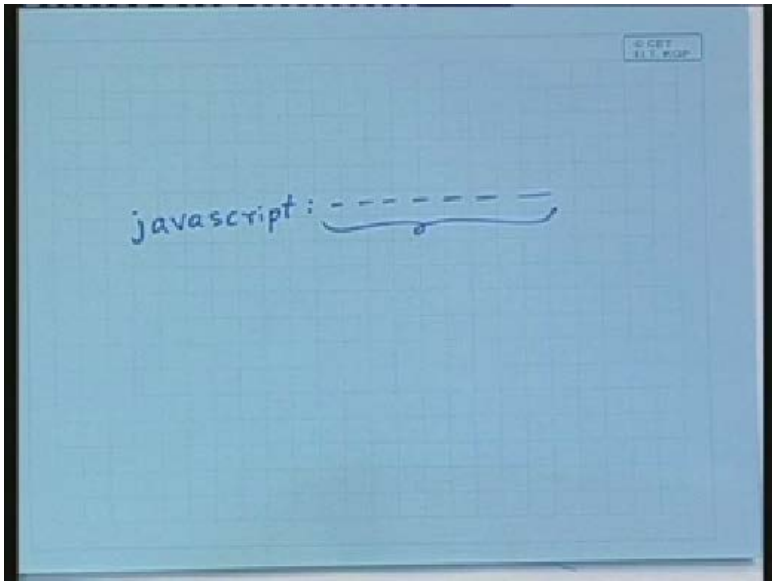
So this is what I have already mentioned, there is something called Javascript URLs. Just like you can specify a URL to identify a particular object on the internet.

(Refer Slide Time: 54:56)



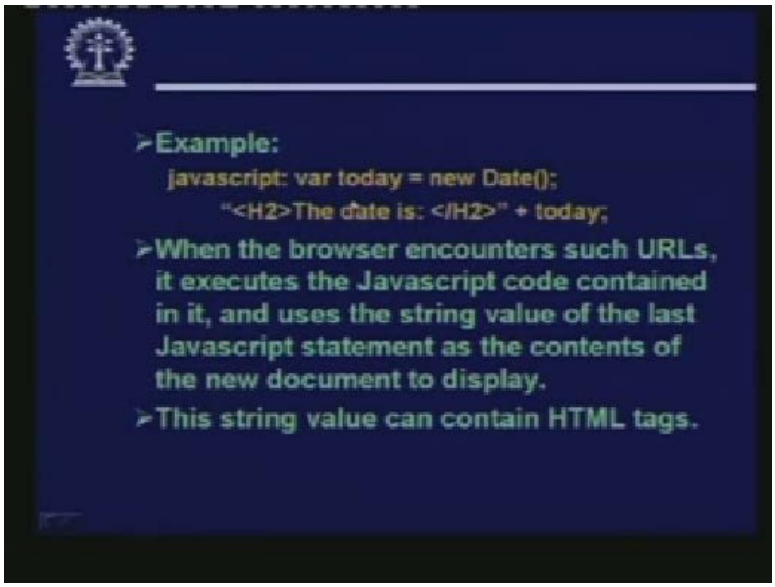
You can use Javascript URL. This is a URL following the keyword javascript colon. This actually specifies that the body of the URL is an arbitrary Javascript code.

(Refer Slide Time: 54:12)



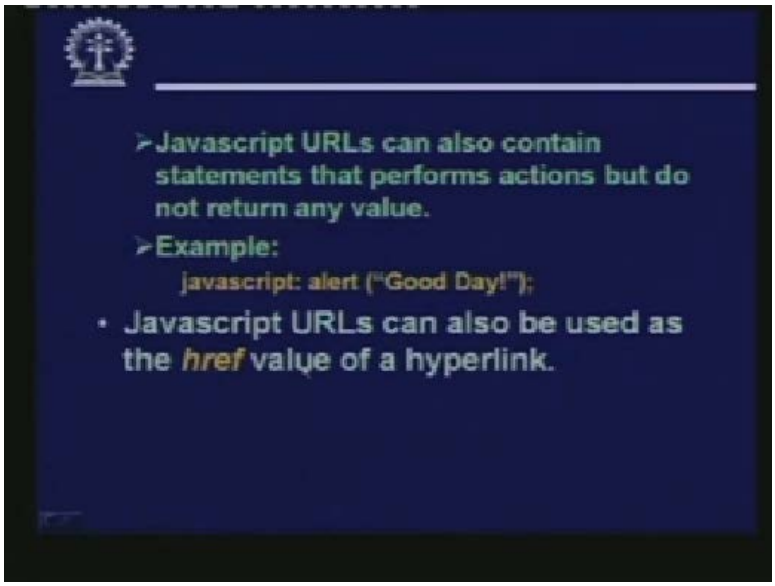
What is mean is this. Suppose you have a URL javascript colon something. This something is considered to be Javascript code and if it is included as part of the URL that is directly executed in line, there is no need to have that begin SCRIPT, end SCRIPT coding before and after. And within this URL you can have multiple Javascript statements also. And if there are multiple statements they need to be separated by semicolons.

(Refer Slide Time: 54:49)



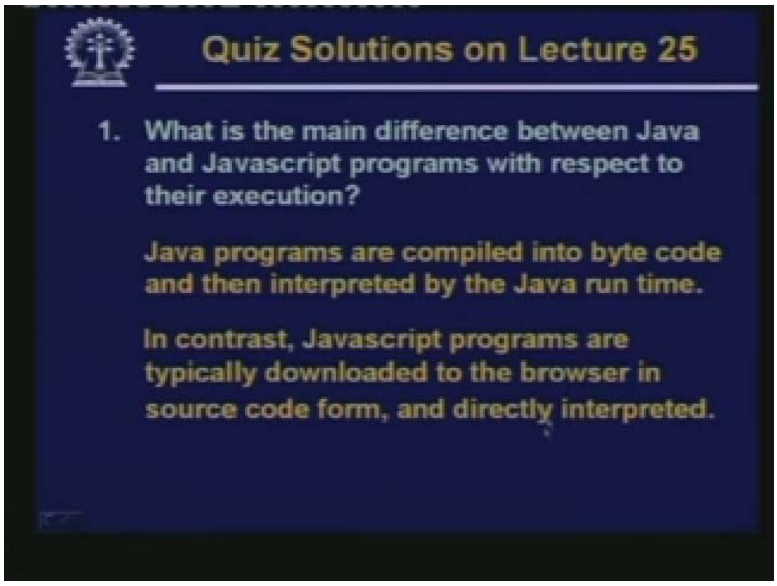
Some example here javascript colon var today equal to new Date. This is a Javascript statement where a new object of type Date is created it is assigned to today colon. There are semicolons; there are two statements the next statement. This is just and just a part of an HTML document H2, “The date is today”. So in a Javascript URL you can have several such Javascript snippets separated by semicolons. But what happens if such a Javascript appears in an HTML document? What will happen is that all the Javascript code will be executed and whatever is outputted by the last Javascript statement, that will be included as part of the HTML document. The others will be ignored only the value outputted by the last Javascript statement. That will be included in the surrounding HTML text.

(Refer Slide Time: 56:05)



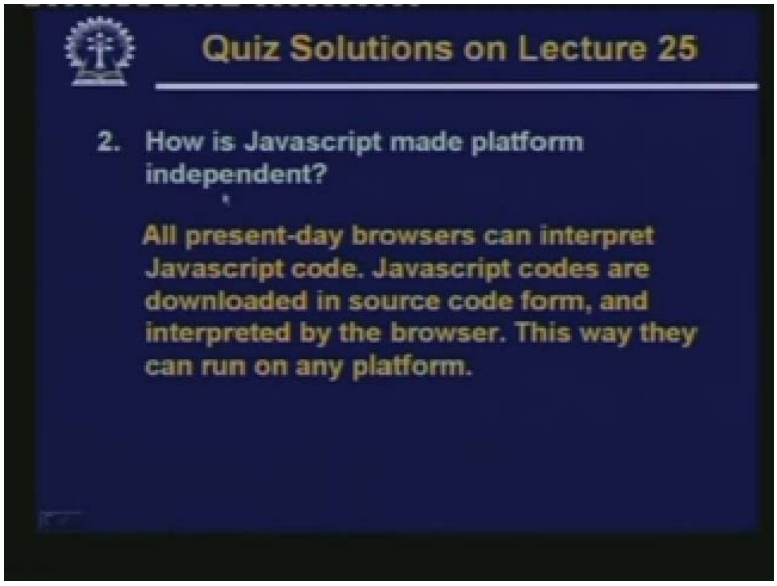
So some examples javascript alert Good Day. So this will produce an alert and you can also have Javascript URLs as part of the href value of a hyperlink. So with this we end the discussion on today’s lecture. So now let us very quickly look at the solutions to the questions which we had raised during our last lecture; lecture number 25.

(Refer Slide Time: 56:42)



Do the first questions was what is the main difference between Java and Javascript programs with respect to their execution?
Java programs as we have mentioned are compiled into byte code which are then interpreted by the Java run time which are typically part of the browsers. Well Javascript programs are downloaded to the browser in source code form and are directly interpreted; this is the main difference.

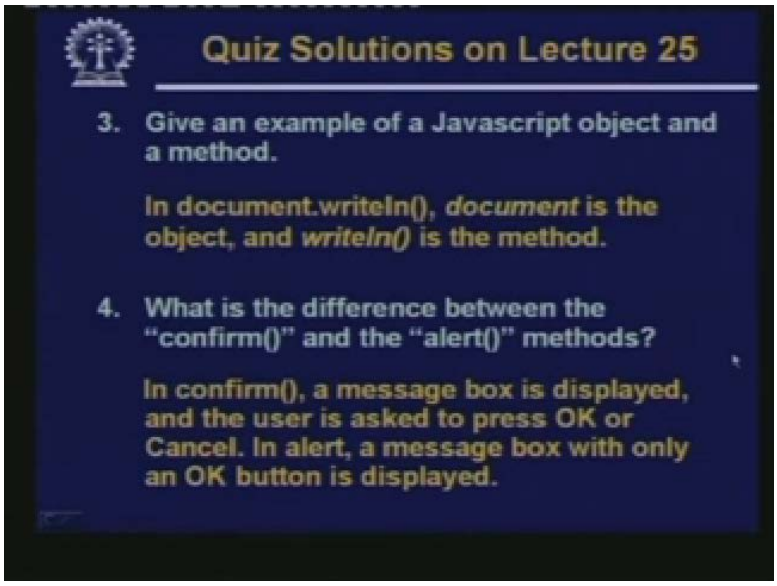
(Refer Slide Time: 57:14)



How is Javascript made platform independent?

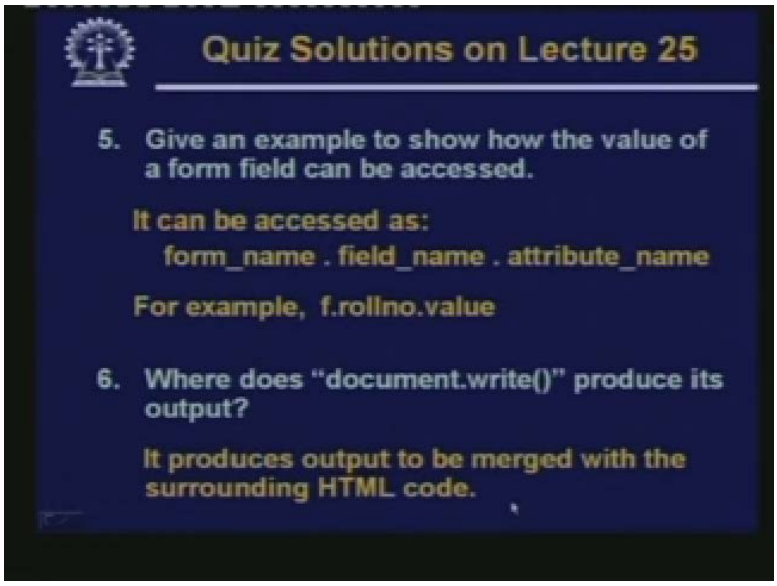
Now all present day browsers have the capability of interpreting the Javascript code. Javascript codes are downloaded on to browser in source code form which is interpreted. So in this way Javascript code can run on any platform, on any browser. This way they can be made platform independent.

(Refer Slide Time: 57:41)



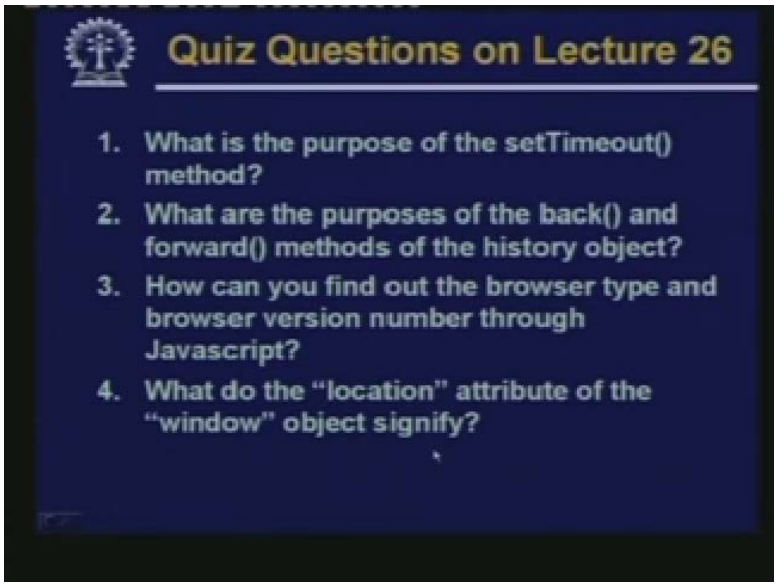
Give an example of a Javascript object and a method.
For instance document dot writeln, if you write like this document is the object writeln is the method these are separated by dot.
What is the difference between the confirm and the alert methods?
In confirm a message box is displayed with a choice being asked from the user to press either or Cancel it also returns a boolean value depending on the key pressed. But in alert only a message box is displayed where a single button is there.

(Refer Slide Time: 58:21)



Give an example to show how the value of a form field can be accessed.
It can be accessed as the name of the form dot name of the field, dot name of the attribute like this, f is the name of the form roll number is the name of the field value is an attribute of that field.
Where does document writeln produces its output?
This I have mentioned repeatedly, this will produce the output to be merged with the surrounding HTML document. So now some questions from today lecture.

(Refer Slide Time: 59:03)



What is the purpose of the setTimeout method?
What are the purposes of the back and forward methods in the history object?
How can you find out the browser type and browser version number through Javascript?
What do the location attribute of the window object signify?
So in our next lecture we shall be continuing with Javascript. So with this we come to the end of this lecture. Thank you.