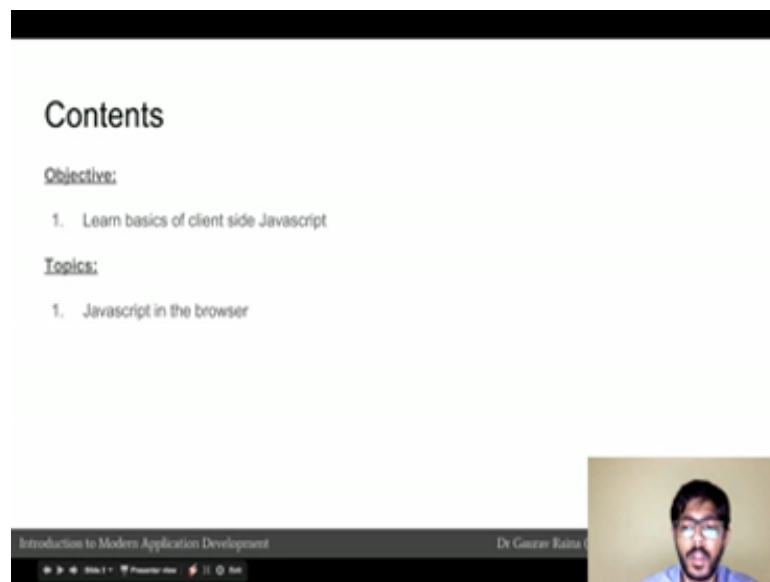


Introduction to Modern Application Development
Prof. Tanmai Gopal
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Module - P5
Lecture – 11
Introduction to Client side Javascript

Hi all. Welcome to Module - P5. In this module we will be getting an Introduction to Client side Javascript. Client side Javascript refers to javascript that executes in the browser.

(Refer Slide Time: 00:11)



Contents

Objective:

1. Learn basics of client side Javascript

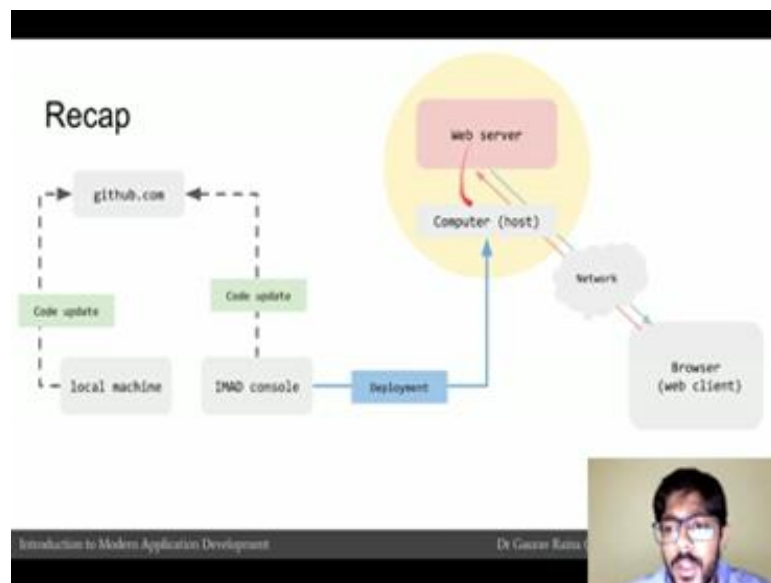
Topics:

1. Javascript in the browser

Introduction to Modern Application Development Dr. Ganesh Rajan

We will be doing a few simple exercises to understand what the power of javascript in the browser is.

(Refer Slide Time: 00:17)



Before we do that, let us take a quick recap of what we have covered so far. In the first week we understood that a browser can make request to a web server through a network and receive responses from the server. We also looked at how code can be deployed; we use either the IMAD console or the local machine for development.

We save our source code on github.com on a get repository on gethub.com. After we have written code we use the IMAD console to deploy our source code on to the computer which hosts web server software. The IMAD console takes care of deployment and takes care of restarting the web server with the latest changes.

(Refer Slide Time: 00:53)

The slide is titled "Recap" and features a diagram of a "Web server" represented as a rounded rectangle. Inside the rectangle, there is a grey box containing the following text: "nodejs code that matches a URL to a particular function.", "Function can respond with text OR", and "Function can respond with HTML". Below the diagram, there is a small video feed of a man with glasses and a beard, wearing a blue shirt. At the bottom of the slide, there is a footer with the text "Introduction to Modern Application Development" and "Dr. Ganesh Ramesh".

When we were looking in the web server that we wrote, we wrote a web server in nodejs and the code was matching a URL to a particular function.

(Refer Slide Time: 01:11)

The slide is titled "Recap" and displays four code snippets for different URL requests, each with an arrow pointing to a corresponding response type:

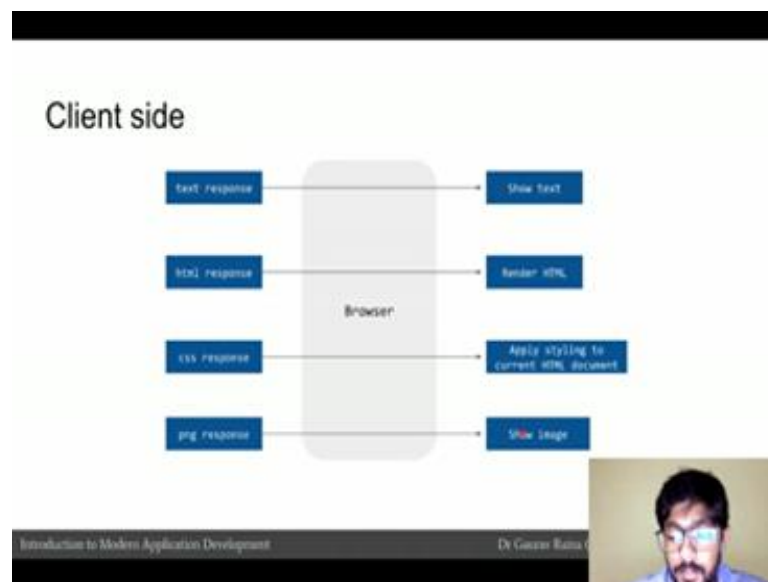
- `app.get('/article-one', function (req, res) {
 res.send('Article one requested and will be served here!');
});` → Text response
- `app.get('/', function (req, res) {
 res.sendFile(path.join(__dirname, 'ui', 'index.html'));
});` → HTML response
- `app.get('/ui/style.css', function (req, res) {
 res.sendFile(path.join(__dirname, 'ui', 'style.css'));
});` → CSS response
- `app.get('/ui/mad1.png', function (req, res) {
 res.sendFile(path.join(__dirname, 'ui', 'mad1.png'));
});` → PNG response

A red arrow points to the `res.sendFile` call in the last snippet. At the bottom of the slide, there is a footer with the text "Introduction to Modern Application Development" and "Dr. Ganesh Ramesh", along with a small video feed of the same man as in the previous slide.

The function can respond with text or the function could respond with HTML. Just to go with some examples this is what the code look like. The first piece of code here is saying that if there is get request made to the URL slash article one, use this function to determine the response, what does this function do? This function responds with the following text, article one requested and will be served here. This is a text response.

Similarly, if we look at the URL slash, in this case the response is not text response but the response is a file, and this file is an HTML file, which makes this an HTML response. Similarly for CSS for the URL slash ui slash style dot CSS we served a CSS response and so on for png.

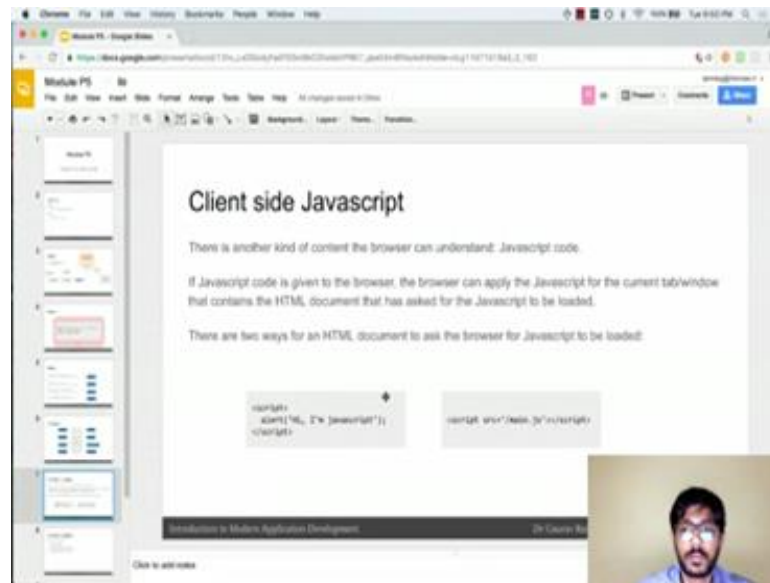
(Refer Slide Time: 01:50)



The browser in turn when it got the response, interpreted the response in different ways, for example, for the text response it just displayed the text if the response from the server was an HTML response; it converted that HTML response into a visual rendering and rendered the HTML.

Similarly for a CSS response the browser did not actually show the CSS, but it applied the CSS to the current HTML document, and for the png response it just showed the image. There is another kind of content that the browser can actually interpret and understand which we have not talked about so far - which is Javascript code.

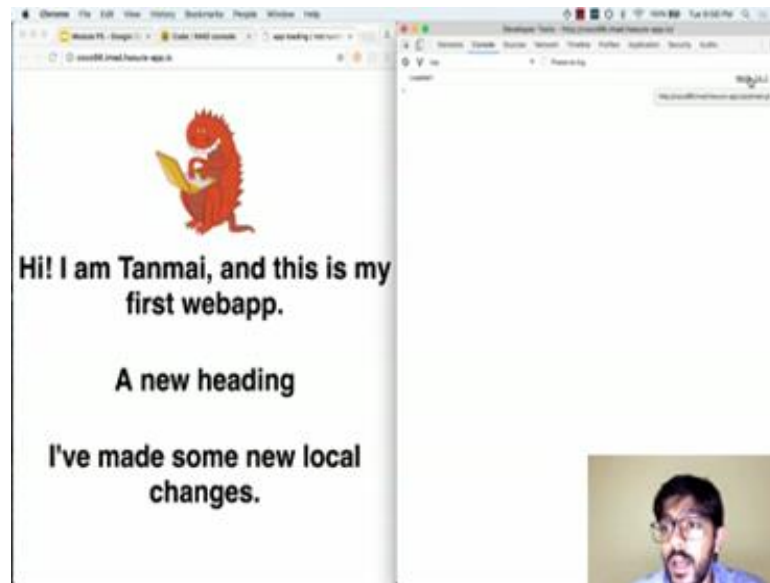
(Refer Slide Time: 02:20)



If javascript code is given to the browser the browser can apply the javascript inside the current tab or window, and within the context of the HTML document that has asked for the javascript to be loaded. How can an HTML document ask the browser for the javascript to be loaded, it can ask for the javascript in two possible ways, this is one HTML snippet that adds a certain amount of javascript. So, you can see if the HTML document has the tag script you can write some javascript inside and then you can close the script tag. This is quite similar to how CSS was added to the HTML document, but instead of script we would have used the word style. So, you would have said style and some CSS lines and then slash style.

So, instead of doing that we are using script, there is another way of including javascript in the HTML document, which is by referring to a particular javascript file. So, you would do script SRC main dot js and assuming that main dot js is a URL endpoint that responds with javascript, not that javascript code is injected in the context of the current window or tab.

(Refer Slide Time: 03:55)



The CSS analog for this was style and a link. So, we use style to inject inline CSS as this is called inline javascript when we include the code within the document it is called inline. So, you would have inline javascript or inline CSS and when you refer to an extraordinary CSS file, we were using the link tag, but for javascript we can just use a script tag.

Just to show you an example of what this looks like, I had to my code console and I had to the index dot HTML file. If you see this index dot HTML file is requesting for a particular js file, but I can also have code, but you can also have code that you write inline inside the document let us quickly come at this and take a look at what this looks like.

You can see that our javascript executed and you saw – Hi, I am javascript; if I click on the rest of the document loads. So, once again because this is important to understand this is what that browser is doing the browser sees that the response contains HTML, and then the browser starts interpreting the HTML and starts displaying the HTML. The browser then encounters a script tag and then it executes the javascript inside the script tag. So, here you have seen alert – Hi, I am javascript and that is why when the page loads you get an alert box that says, Hi, I am javascript.

Here would be trying to include a javascript file. So, just script SRC equal to u i slash main dot js and let us look at what happened to that request. So, I am going to refresh

this to see what our network requests were and I say. So, here you can see that they tried to make a request on main dot js, but it did not receive the main it did not receive the javascript from the server and the server responded to the 404. This is happening because there is no code here which is actually responding to the main dot js.

Let us quickly correct that by adding the main dot js code. So, let us save this, wait for a server to restart. So, there are server has restarted and now let us look at the output. You can see that the main dot js is now a 200 which means that the main dot js has been served and what was this main dot js code. So, what we can do is actually go to the tab called sources on inspect element, and you can see that main dot js is a resource that has been requested. So, if you double click on this you can see the javascript file that was loaded and this is the same javascript as we had here which is main dot javascript.

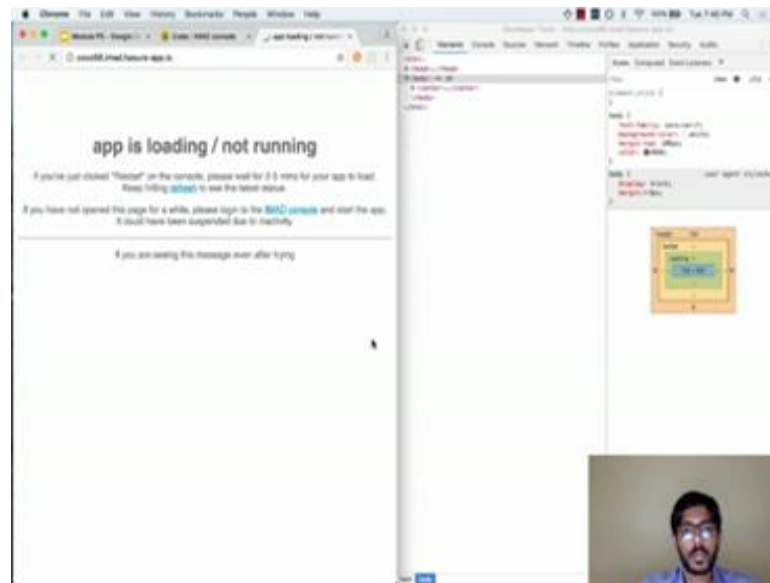
Now, let us go back to our app. Now, if you look at what this main dot js is do anything console dot log loaded and what this does is printout the string loaded in the console and so; that means if I go to the console I see the string loaded has been printed here. So, we are seeing both the methods of javascript being used, first this executes because this is their first and then you see u I slash main dot js.

(Refer Slide Time: 07:18)



Which is being loaded and once it is loaded you can see that the javascript is printing this line here, inspect element also tells you what line of javascript has caused this to be printed here. In fact, if you click on this you will go back here.

(Refer Slide Time: 07:36)



Now let us try to write some javascript and see what kind of things we can do with the javascript, we are going to try to do four interesting exercises one will make an alert box which we have actually already done. We will output something on the console which is also something that we have already done; we will change the content of an HTML element which is something you are going to do. I have to go to make an element move if we click on the element.

Now, the first interesting thing to understand is that I do not I can actually write javascript inside the developer tool which is kind of the window that you get when you click on I mean inspect element and, let us go to inspect element let us go to console. So, I can actually just write javascript here; for example, I can say alert hi and. So, you see that an alert comes up I can also just change the size I can also say console.log, and so the string a s d f comes up here let us do something a little fancier, let us select a particular HTML element and once we select an HTML element let us change the content of that HTML element.

Let us look at elements to see what our HTML elements are? So, we have an image we have the content here let us say I want to change this content. So, the first thing that I need to do is I need to select this element to be able to modify it. Now another way like we had a class tag, another way to add a tag to an element or kind of label in element is using an id. The difference between an id and a class is that an id must be a unique name

while a class can actually repeat. So, many elements can have the class, but center. But if I give it in id called main text, only one HTML element should have the id called main text. Now, that I have given it an id called main text.

Let say var element is equal to document dot get element by id and so I will say main text. So, that click on element you will see that the HTML element is sort of represented here. Now if I want to change the HTML I can just do element dot inner HTML at and you can just said to a value a new value and you can see that the HTML on the document has changed. So, this is what javascript can do and I am writing this javascript in the console this javascript would have also come from our main dot js file.

And I mean the dot js file we can just write, I am writing a comment here saying change the text of the main text div and so I can write var element is equal to document dot get element by id main text and then I can say element dot inner HTML is equal to new value. If I go to index dot HTML I have to remember to change the main means I can say id is equal to main text, let us save this, cool. So, we have alert box and then you can see that this is already been changed because it happens. So, fast that we could not see it and it was already changed to the New Value.

Let us now try to make something move. Let us try to make this image move if I click on the image, and let us see how we can do that. So, the first thing to do is again get the ability to select this image. So, let us go to our index dot HTML and let us give this image a name. Let us call it id is madi and now it has go to a main dot js file and write some codes so and so as put a comment you are saying move the image. So, the first thing I need to do is again select the image. So, I say document dot get element by id img and I can say img dot on click. So, on click execute a function.

So, here I am going to write the function that should be executed if I click on if I click on the img element and I want to move it a bit little to the right. So, the way I can do that is I can change it is CSS from javascript and increases the left margin and make that 100 pixels make that a 100 pixels.

Let us save what we have. So, restart our server, all right. So, you can see that I have gotten an error which says cannot set property on click of null. Let us try to understand what this error is. So, the browser could not execute this javascript so; that means, that it could not find img and that is because I made a mistake when I set get element by id it

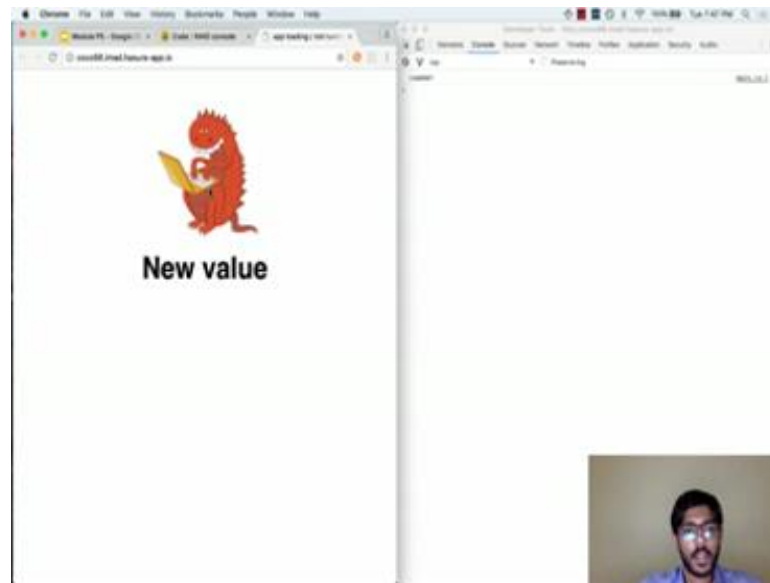
there is no id called img it should have been it should have been madi. Let us change this to a madi. So, let us go back to our code save this, let us go back to our browser restart, and see you can see that we have not got java script error which means that our javascript is probably executed.

Now, let us try clicking on it. So, I clicked on it and it jumped a little bit to the right because the left margin had increased. In fact, a few click on inspect element you can see that a margin left property suddenly got added right and this did not exist before, but this got added because of a javascript. Now, it is trying to do something else let us try to make it move to the right a little gradually, and let us see what the code for that looks like the way we can approach animation in javascript is by changing a property gradually, and the way we can affect a gradual change is by using something called an interval type.

So, let us create an interval. So, this code that I have written here reads as every 100 milliseconds apply the move left function I am sorry this should be move right. So, every 100 million seconds apply the move right function right and where is this move right function well we do not have this function. So, let us just quickly create this function.

So, function move right, says that create a variable if you have a variable called margin left make sure that the variable called margin left increases by ten and then set the CSS of the image and set the margin left of the image. This will convert to a string called 10 p x right I am concatenating the number 10 with p x, but this margin left variable does not exist. Let us initialize that margin left variable. So, margin left is equal to zero. So, every 100 milliseconds the move right function would be called move right will first increment the value of margin left which was initially zero and which will now go up by 10.

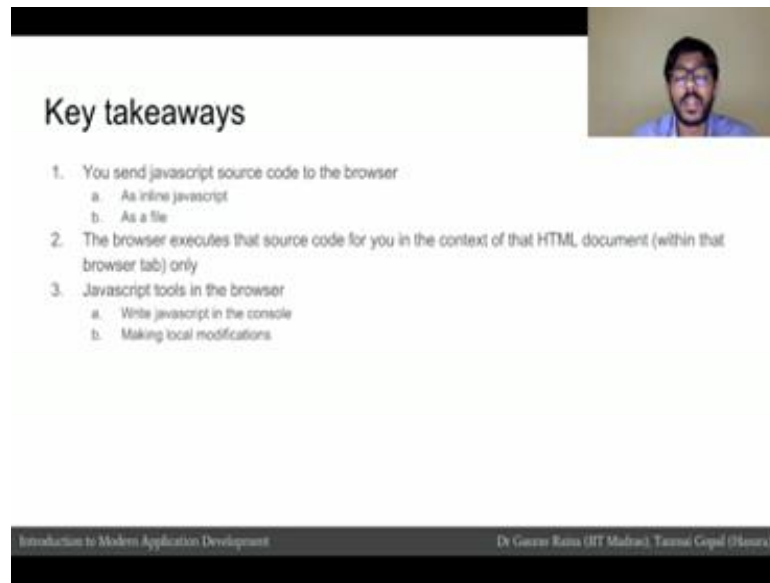
(Refer Slide Time: 14:39)



Once it goes up by 10, the CSS will be set to that value. So, let us save this and restart our server our javascript is also loaded fine the browsers have not found any errors and now let us click on this element and you can see that the images gradually moving towards the right, but you can see that it is a little jumpy. So, we can actually make this a little smoother.

Let us instead of every 100 milliseconds. Let us do this every 50 milliseconds. So, that the movement is more gradual and let us change this value to a smaller number value let us say 5 pixels. So, we just move it by or maybe we can even move it by a smaller value let us say one pixel, right. So, that will be a slow movement, but it is seem to be a more continuous movement.

(Refer Slide Time: 15:30)



Key takeaways

1. You send javascript source code to the browser
 - a. As inline javascript
 - b. As a file
2. The browser executes that source code for you in the context of that HTML document (within that browser tab) only
3. Javascript tools in the browser
 - a. Write javascript in the console
 - b. Making local modifications

Introduction to Modern Application Development | Dr. Ganesh Ramesh (HT Madras), Tamasz Csapodi (Hamburg)

Now, let us click on `make` and you can see that the movement is a lot more gradual and seems to be a lot more continuous, the main things that we looked at in this module was that we can send javascript source code to the browser. So, which the browser will execute for us the javascript can be inline javascript meaning it can be embedded inside the HTML document or it can be a separate file, the browser executes that source code only in the context of the HTML document that loaded in the javascript.

So, javascript for many HTML document cannot effect the javascript of another HTML document which might be in different tab in your browser or a different window in your browser we also looked at the few nice javascript tools in the browser that allowed us to see what javascript is loaded, and allowed us to execute some javascript or write some javascript in the console and we also make some local modifications to the HTML using the javascript from the console.

(Refer Slide Time: 16:11)



The slide features a white background with a black header bar at the top. The main content is titled 'Further reading' in a bold, black font. Below the title, there is a numbered list of three items. The third item includes a blue hyperlink. At the bottom of the slide, there is a black footer bar containing white text.

Further reading

1. We just scratched the surface of Javascript
2. Read up and think of things to do and implement them in Javascript
3. <https://www.codecademy.com/skills/make-an-interactive-website> is a good course to learn JS basics

Introduction to Modern Application Development | Dr. Gaurav Rana (IIT Madras), Tanmay Gupta (IIT Bombay)

In this module we just scratched the surface of javascript, the point in this module was to introduce you to the kind of things that javascript can do and how javascript in the browser works. The code academy course to make an interactive website is a good example to learn and the kind of things you can use javascript for.