IIT Madras

ONLINE DEGREE

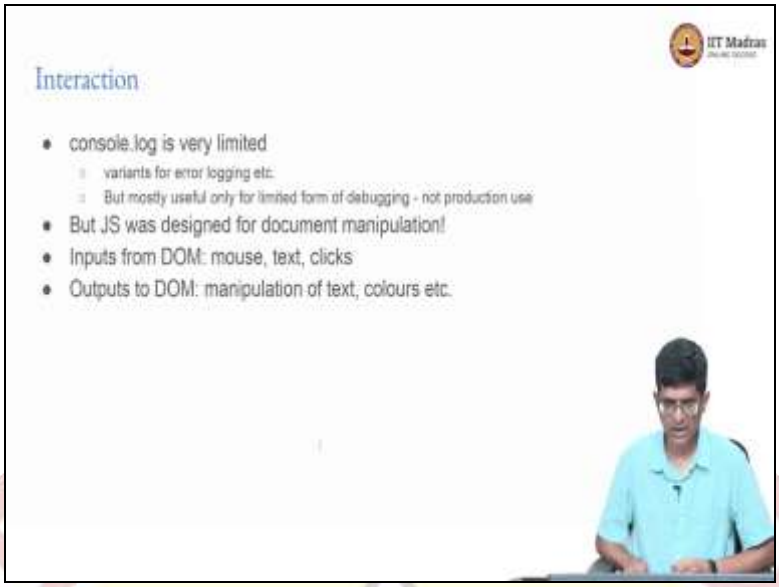
Modern Application Development II
Online Degree Programme
B. Sc in Programming and Data Science
Diploma Level
Prof. Nitin Chandrachoodan
Department of Electrical Engineering
Indian Institute of Technology- Madras

Javascript - DOM API

(Refer Slide Time: 00:16)

Hello everyone welcome to modern application development part two. Okay now before getting on to a little bit more advanced usages of Javascript we want to sort of see how Javascript can be used for user interaction right that is to say the main purpose over here ultimately is going to be for front end manipulation right. So, how can we actually use it to manipulate our DOM right the api that we are the api's that are used in order to manipulate the document that gets displayed to the user okay.

(Refer Slide Time: 00:42)



So, console dot log in other words is very limited it's primarily meant for logging basic errors and can be used for a limited form of debugging but is generally not useful in production okay. But the whole idea of Javascript was that it was designed from the get go for document manipulation. What does that mean? It means that it can get inputs from the document right mouse clicks ah values that are typed into input boxes and so on.

And it can also manipulate the DOM it can actually insert new text it can modify text that is already present over there and so on okay. We will be making extensive use of that moving forward. So, I am also just going to demonstrate a little bit of that just to see how we can you know tie some of these concepts together.

(Video Start: 1:32)

So, for that what I have done is I have now pretty much commented out everything in the script dot js let us look at this other file interact dot js and now I am going to slowly sort of start looking at manipulating things that are there in our html file. So, you might have noticed at the beginning that you know line 10 had this div id equal to d1 and line 12 had another div with id equal to d2.

And in between them I had this hello world being printed out now when I run all this file while I was running things this hello world was the only thing that got displayed because other 2 div's are there in the html but are basically invisible because there is no text in them okay. How can I change that? Let us go look at interact dot js and say right I have

this line document dot getelementbyid. So, this document is a sort of global variable in Javascript you don't have to declare it, it's already there available for you.

It's not a keyword it's actually a global variable right ah and in fact you can see that you know it's declared as var document of type document capital D document right. But this is a global variable you don't get to declare it you just use it. And this variable it's an object that has several different methods one of which is get element by id. So, I can basically get element by id d1 and that is going to pick up my first div.

And what I am going to do with that is to say d1 dot innerhtml equals welcome to d1. So, what do I expect is going to happen this html that I have out here when I run this new script right the page will load it will then run the Javascript and the Javascript should cause some new text to get inserted into d1 which means it should come about the hello world. Let's try that and sure enough we see that if you notice carefully there was a small break right.

It basically showed hello world and then welcome to d1 that gap is because the hello world was there in the html that got displayed first then the Javascript ran and resulted in welcome to d1 being inserted on top okay. What happens if I also put something into d2. Once again all going too fast but you can see that you know hello world came first then welcome to d1 and goodbye from d2 more or less you know in quick succession.

Because the Javascript once it starts running is very fast now this what I am going to show now this function out here is definitely something that is a bit advanced and will require a bit more uh explanation which we will have to get to after we have talked about asynchronous functions okay. For the time being what you can assume is that you know I am going to do a few things I am going to just print out something to the console.

And this await new promise is essentially going to wait for two seconds okay. After waiting for two seconds it's going to do something which is modifying the d1 dot inner html and also logging something to the console. After that it will wait for another two seconds and modify d2 dot innerhtml and log something to the console okay. Now you will notice that I can't just have those lines over there directly it won't run in fact.

I need to wrap it around this async function I need to create a new function called demo and then call that function that's the only way of running it okay. Now why is all that we will get to that later but for the time being let's see what happens as a result right when I run this I see hello world up there after two seconds welcome to d1 pops up and after another two seconds goodbye from d2 pops up okay.

So, I can bring in this notion of sort of timeouts and you know interactivity just based on time even into the picture by doing this. And of course the next most interesting thing that we could probably do over here is to say you know all very well to say that we have timeouts and you know changing text can I get input from the user okay and this is an example of that I declare a variable x and once again you know get d1 as get element by id of d1.

And I say d1 dot inner html is going to take this value click count dollar x notice the template with the back ticks over here right. But now I am going to add something else I am going to basically put in this thing which does an event listener okay this is a new another method which is bound to this d1 right. So, whatever came out of this document dot getelementbyid right it's some kind of an object.

And that object has a method called add event listener which takes in two things one is click and the second is a function I don't want to declare a function first and then use it I just directly use arrow notation. Okay so, this e arrow and open parenthesis and open brackets close brackets right is effectively a small anonymous function that I have created over here just for directly using within the ad event listener.

This is an example of where the arrow notation or the anonymous functions make sense I don't have to declare this function ahead of time because this is the only place I am going to use it. Okay having said that there is nothing wrong with declaring it and then you know putting that name in there and in some cases it might actually make it more readable. In this case arguably you know it's like easy enough to understand because this is like a sort of standard what's called an idiom of the code right a standard way of expressing certain things.

Now let's think about what this is going to do it's going to increment x each time I click on it hopefully set the value to be the count and I am doing something interesting which is that I am going to set the font size ok which is actually one of the style parameters corresponding to this div to something like x + 10. Why am I doing x + 10? Because if I give it 0 then it will probably be too small okay, now this looks interesting let us see what happens. When we run it right I got the hello world and then you know after a brief pause I got click count 0 okay interesting.

Let me go if I go click somewhere randomly out here nothing happens but if I click on d1 right I see that I see click count 1 it actually became a bit smaller because now it's using 11 pixel font size whereas previously it was using something bigger. What happens when I keep clicking? You can see that it starts getting bigger right because with each click the font size is correspondingly going at this point just like 22 point uh you know 25 point font uh 20 point 30 points right I can keep going this unless at some point you know the font size becomes too big and the system crashes right.

So, what's the you know idea behind this we have interactivity we had some way by which we could sort of get things to the point where you know I could take input from the user by means of mouse clicks associated with a function that gets called. And manipulate something on the DOM that then gets displayed to the user okay. Moving forward will be making extensive use of all of these things.

To see how we can you know put all of this together in order to build up more interactive and a better interactive applications and a better user experience.

(Video End: 09:14)