welcome back to web dev simplified my name is Kyle and in today's video I'm going to show you how to debug an application because for all of those thousands of videos out there showing you how to make something none of them actually show you how to properly debug a problem that you run into while building your own application and debugging is the most important skill any developer can have so let's get started now here is a sample application that we're going to use that's riddled with bugs right now but we're going to slowly debug all the different problems inside of it to show you what the application is supposed to do is it has a list of users it's going to loop through those users and for each user it just wants to print that users name and how old they will be in five years so let's run this and we can see immediately that we have a syntax error it says uncaused syntax error unexpected token curly brace and it tells us this occurs in script ojs on line number eight anything that comes after the colon here that's the actual line number the error occurred on and the thing before the colon is the exact file that that error occurred so we know that this error is right here line number eight in our script J s file which is this file right here and syntax errors are the most common type of error you're gonna run into when you're first starting to learn how to program and a syntax error just means that the code that you written cannot be parsed it's not written properly so we're missing here something and that's why the JavaScript engine right here cannot parse it and it says that there's a syntax error so we need to find it out and the bad thing about syntax errors is a lot of times the error message is really not correct right here it says that we're missing a left curly brace but in actuality we're not missing the left curly brace so if we go over here we see we have a left curly brace with a red squiggly under it and this is Visual Studio code our text editor telling us there's a syntax error if we highlight over it you see that it says comma is expected and that's because as you can see here on the line above with the name Sarah we forgot to put a comma at the end of our line this is really common and easy to forget so if we add a comment in here that error is gonna go away as you can see a visual studio code and if we reload our page you'll see we get another syntax error this one just says unexpected it identifier again it's on line eight but this error message again is just not at all useful but luckily with any modern text editor we're going to get that error highlighting in the text editor force and as you can see we highlight over this red squiggle age and it says comma expected again because we forgot to put the comma between our name and age properties of this object let's add that in and save and we're getting another error for our syntax error and this one is the exact same error unexpected identifier which is pretty much completely useless but it tells us that this occurs in script KS on line 16 since that's after the colon and one thing you can do is you can actually click on this and it'll open up that file where this error occurs inside of your developer tools here for example this is Chrome but I just find it easier to go into the file and Visual Studio code and look at the error there and if we do down on line 16 we see that there's no error on line 16 we don't have any red squigglies so clearly how is this correct but the problem is right below that on line 17 if we highlight over this we get a red squiggly here and it says declaration or statement expected and again this is really not that useful for an error message so anytime that I have a syntax error and I don't have a very easy or good error message to look at the first thing I look for is missing commas and this is missing semicolons or any other form of missing thing that goes between different things like maybe missing quotes for example and the next thing I look for is my curly braces because curly braces and parentheses those are really hard to get right sometimes we have super nested objects so the next thing I look for is curly braces as we can see here the very obviously this function does not have an opening curly brace for example this left curly brace here is completely missing now if we put that left curly brace on you see it matches our ending curly brace and we can click Save and now we're starting to get our information printed out and this actually looks like it's correct but there's a problem there's a bug in here that's not actually entirely inherent as you can see if we read these it looks like everything is going well until we get down to fill and this says user fill will be n a n years old in five years and n a n just stands for not a number and this is because as you can see fill doesn't have an age specified and so when we try to add nothing to five we're getting not a number so this is a little bit of a bug that kind of creeps into your code that you may not realize because there is no error to debug this but since we know that fill doesn't have an age and we're trying to use our age down here all we need to do is just put an if statement in here so we can just say if user oops user dot age is not equal to null then what we want to do is just our original console dot log so we can just copy this up and paste it here but if the users age is null we can just come down here and we can so that log something a little bit different we can say user whoops user is called and then we can just put whoops their name so we can say username now if we save that you see that it says user is called Phil so that's working perfectly fine because we're skipping over that age section so now that we've covered syntax errors as well as hidden bugs that come in that don't actually show up as errors let's take a look at another kind of error that's really common by looking at a different application here's the next application that we're gonna take a look at and it's really straightforward we have a button and an input element and every time we click the button we want to print whatever is typed in the input element and our index.html again has that input as well as that button so let's run this and see what happens go over to our JavaScript and immediately we see that we have an uncaught type error cannot read property add event listener of null and it says occurring in our JavaScript at line four which is this button add event listener right here and essentially this error is saying it cannot read add event listener of null and we're calling add event listener on our button so for some reason our button is null and this is an incredibly common error having something to be null and trying to call something on it is something that you'll have come up all the time when you're programming and it can be a little bit difficult to debug because you may not know why this element is null so one of the easiest things that you can do just to start off with to make sure you have the correct element that you're trying to debug is just throw in a console about long so we can just say console dot log button because this says that button is null at least we think it says that so if we save this in print you see that it is correctly saying that our button is actually null so now we need to figure out why our button is null so we need to do is find where we initialize our button right here Const button equals document dot get element by ID button so we can see okay do we have an element with that ID we go to our index and we see it actually has the ID of BTN so if we come back into our script all we need to do is just change this to BTN now when we save you see that it correctly prints out our button because it's not an old and we no longer get that error so now let's try to type something in here and click print now you'll notice we get another error this is the exact same error essentially says uncaught type error cannot read property value of null and it's saying this occurs on line 7 which is right here so we have again a problem with something being null so in order to check this all we need to do is first print out our and because we think our input is noble because we're trying to call that value on it which is what we were saying over here it cannot read so let's print our input we save that and we see again our input is null so we need to do is figure out why our input is null we go to here we see that we're getting an element with ID text elem but if we go into our index we have this called text element so instead if we just copy that over whoops paste that in here and now when we save we type something in here and click print you see that's properly printing out and in general you're not going to be able to have this problem with debugging be so easy because most of the time you'll have for example your input and then you're gonna do something down here we're going to set input equal to something else and maybe again later inputs going to get set to something else and it's constantly going to be getting changed over and over so what you want to do is you want to first put your log here at the very beginning right when you initialize your element and you want to make sure it's correct then you just move it down to the next section where you set it and make sure it's correct after that and again you do the same thing after the next time you set it and you find where in that chain of events this input gets set to null or undefined or something that you do not want and once you figure out where that point is you know that if for example here it is null but here it's not null you know that somehow this line right here line six is what caused your input to become null another thing that can really help if you're stuck on an error for example you've tried all these different tricks and nothing is really letting you figure out exactly what the error is is you take the error message exactly as it occurs over here on the side you copy it and paste it into Google and search for that and that'll drastically help you find the error because most likely someone else's ran into that error and they actually have an answer on how to solve that exact problem it may not be exactly the same as your issue because their code is obviously going to be different but the steps to solve that problem will be hopefully the same as what you'll need to solve your problem so now that we've talked about a couple of the really common debugging issues you'll run into let's talk about the different tools that you have to debug these problems and the first one we've already talked about is the console log essentially just being able to print out a value at the exact point in that code whatever it is this is probably the most common way you'll debug problems and in my opinion one of the easiest and quickest ways to debug things because putting in a console dot log is incredibly easy it just takes a second to type that out and then rerun your application to see what that is value is but sometimes just logging out a value isn't enough because you actually want to mess with things when you're in that exact point in the code and this give me done by using breakpoints so let's set this code back to its working version save it and what we want to do is we want to go over to this sources tab inside of our debugging tools and here we have our in our script J s if we just click on it here and as you can see this is our exact file and what we can do is we can click on one of these numbers over here for example let's click on number five and this is going to set a breakpoint so now whenever our code gets to this section it's going to stop executing and pause so let's show an example that we're going to come in here type something out and click print and immediately as you can see our code is completely paused and we're at this line and we have access to what everything is for example hovering over something we have all of the different contents of that element for example here's our input here's our button when we highlight over it instead of here here's our input value when we highlight over it and it just pauses our code right here we even have down here our scope which has all of our different variables for our local context global context script context it just has a ton of information and best of all is if we have our console open we can actually use our console so we can say input and it's going to print out our input we can say input dot value and for example this is our value and we can actually change our code we can say input dot value and we just want to set that equal to null for example whoops I spelled null incorrectly there we go let's put it as null so now and put that value is no so if we say input value we can see that this is empty and now when we let our code run by just going back over to our sources section and clicking the play button is just going to resume and now we go back to our console you'll see that we've printed out nothing because we changed our input value to nothing and then the rest of our code ran that console dot log over here ran to print the value but since we've changed the value to nothing it now printed nothing so that really shows you the power of when you put in these breakpoints you can actually change your code modify your code run things you can really deep dive into exactly what your code looks like at that exact moment another thing you can do let's hit that break point again is you can click some of these buttons step-over will allow you just to go to the very next line of your code while I'll step in will actually go into the function that you're calling so the quick step in for example if we are calling a function here it would go into that function that we just called but since there is no function just go to the very next line as you can see Ron line six we can just click step again and now we're stepping into some other code that's not actually written by us so we can just click the play button to continue but these step buttons are really useful for allowing us to granularly step by our code line by line by line to really deep dive into what is happening inside of our code debugging your code this way though is a bit more advanced and not as useful if you're trying to debug a small problem but when you have a difficult problem that you're having a hard time figuring out the solution to and the console dot logs aren't doing it for you I highly recommend throwing in some breakpoints because it really gives you a ton of information about what your code is doing at the exact moment it hits that breakpoint and that's all you need to know to get started debugging your own code if you enjoyed this video make sure to check out my other videos which are linked over here and also subscribe to the channel for more videos of me simplifying the web for you thank you very much for watching and have a good day