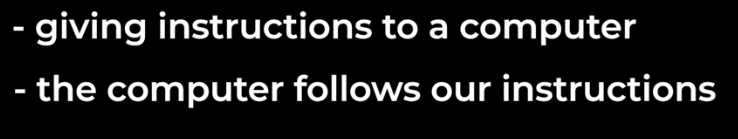
**Lesson-1:Basic of javaScript**

**1-what is javasript:**

****

****

**2-three technologies that we use to create website:**

****

**HTML creates the content of website like buttons,texts,images.**

**CSS changes the appearance of the website.**

**Javascript make our website interactive.**

**3-open google chrome developer tools and write this code in the console:  
**

**After enter:computer creates a pop-up with a text hello inside.**

****

**4-javascript terminology:**

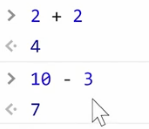
****

**Enetr means:**

****

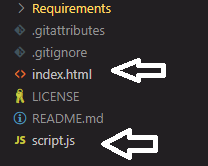
**5-MATH**

**Javascript can do many things like create pop-up and do Math:**

****

**6-how to type out this code:**

**Create index.html and script.js in the root of project and in the same folder:**

****

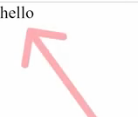
**Index.html:**

****

**In script.js file no need to enter code by now.and enter code in the chrome console:**

****

**After Enter:**

****

**What this code dose is that it removes everything on the page and replace it with text hello.**

**Note :the index.html file must have body tag.**

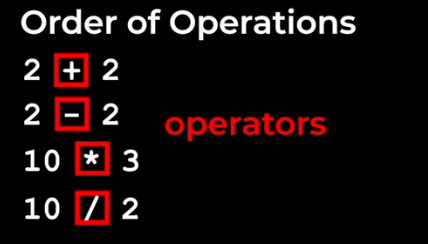
**In this example we used javascript to modify the web page.**

**Modifying the web page is on of the most important features of JavaScript.**

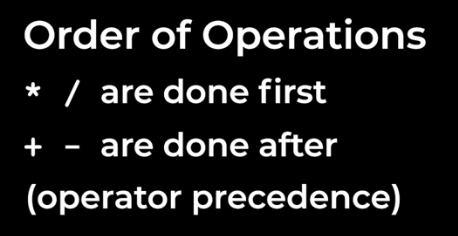
**The End-Lesson-1**

**Lesson-2:Numbers and MATH**

**1-operations:**

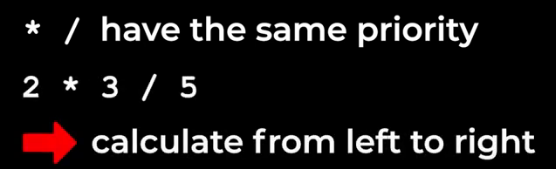
****

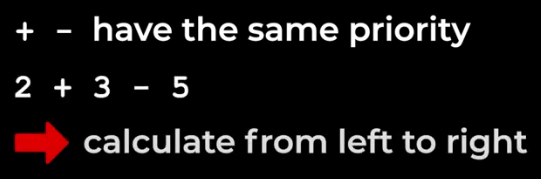
**2-order of operations:**

****

****

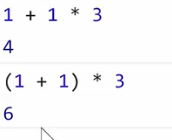
**If we have both \* , / in a calculation calculate from left to right:**

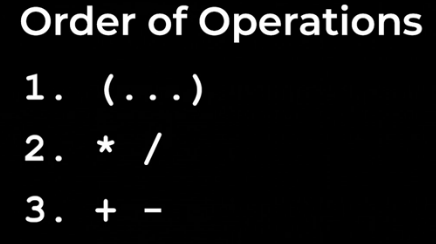
****

****

**3-using brackets:**

****

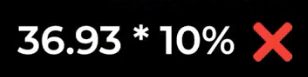
****

****

**4-claculate this 10 percent tax here:**

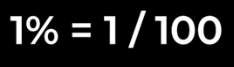
****

**How do we claculate 10% of something?Unfortunatley we can’t just multiply by 10 percent in javascript:**

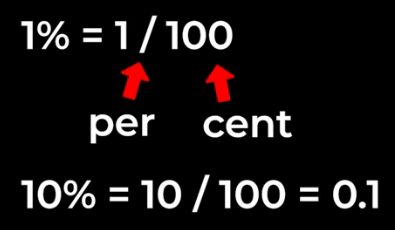
****

**Instead we have to calculate the percent manually:**

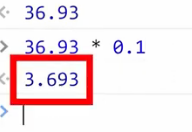
**On percent is acyually equal to one devided by one hundered:**

****

**It means:**

****

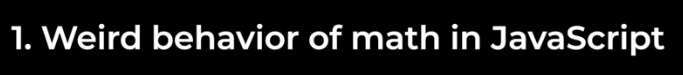
**So to calculate tax:**

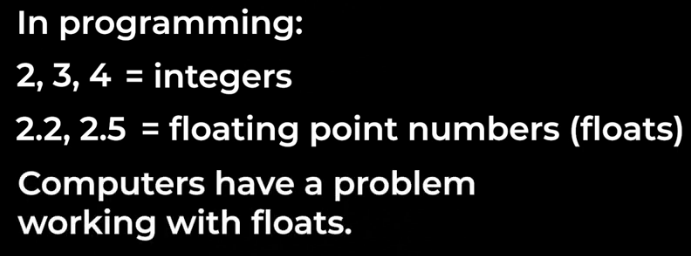
****

**That is close to this:**

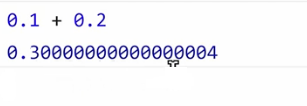
****

**5-more details about numbers and math:**

****

****

**If we type the calculation 0.1+0.2 and press enter it will give us this wired number here:**

****

**That’s close to 0.3 but a little bit inaccurate.this problem is not specific to javascript many programming languages also have this problem.**

**the reason this happens is because of how computers store numbers I'll give a simplified explanation:computers can only store zeros and ones while humans can count from 0 to 9. so**

**there's sort of a mismatch between our number systems for integers this is not a problem the computer can just use a bunch of zeros and ones to store it however for decimal numbers or floats sometimes the computer can't store the number properly **

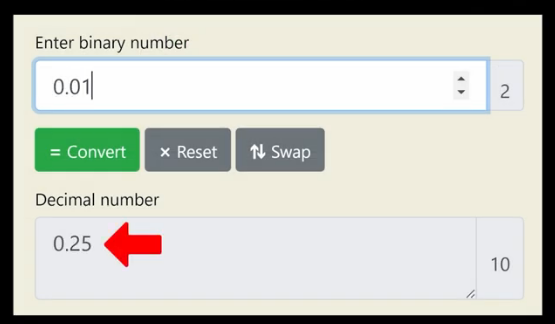
****

****

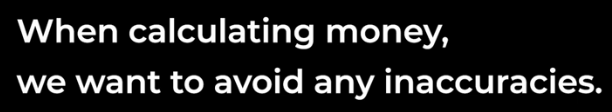
**for example here's how the computer tries to store 0.2 using only zeros and ones at the top is what the computer stores at the bottom is what this is equal to in human numbers we can see that the computer gets closer and closer to 0.2 but never actually reaches it and this is why we get these inaccuracies when calculating with floats because computers can't store some floats properly**

****

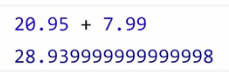
**note that this doesn't happen with all floats for example computers can store 0.25 without any problems so these inaccuracies only happen with some floats but not all of them.**

****

**however when calculating money we definitely want to avoid any inaccuracies so let's learn how to do that:**

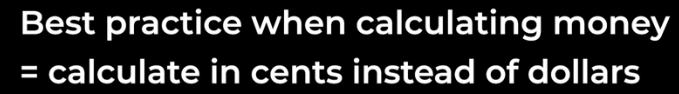
****

**Consider you are buying a basketball ball that cast is:20.95 and a T-ishirt casted:7.99**

****

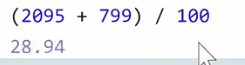
**We have inaccuracies because we calculate float numbers.**

****

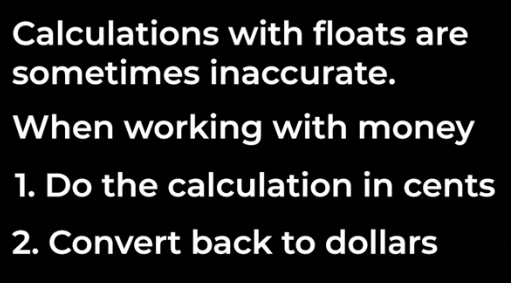
****

****

**To convert cent to dollar:**

****

**So:**

****

**6-how to round a number in javascript:**

****

**rounding just means taking a number and moving it to the nearest integer to round a number.**

****

****

**Let’s calculate tax again:**

****

**If we round the entire calculation:**

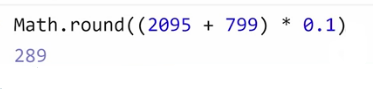
****

**Because the nearest integer is 3.instead we round this section first:**

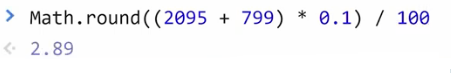
****

**Which is;**

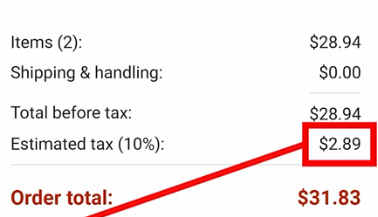
****

****

**Finally we convert this back to dollar:**

****

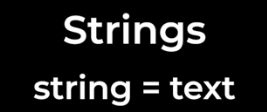
**That is good and exactlly same with out tax:**

****

**The End-Lesson-2**

**Lesson-3:strings**

**A string represents text:**

****

**Write ‘hello’ in the console:**

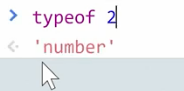
****

****

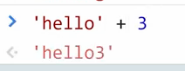
****

****

**2-Type of value:**

****

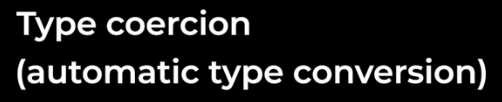
**3-Add a string and number together:**

****

**Javascript automatically convert this number into string.**

****

**This feature called:**

****

**Example:**

****

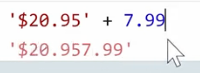
**we're going to add the cost of the basketball which is twenty dollars and 95 cents Plus the cost of the T-shirt which is 7.99 if we press enter unfortunately this doesn't look right instead of doing math and adding these two numbers together it just sort of put the numbers one after another into the string so why is this happening?**

**JavaScript adds from left to right so the first step is to add these two values and as we learned if we add a string and a number JavaScript will automatically convert this number into a string so the result of this calculation will be the string dollar 20.95 and quote:**

****

****

**the next step is to add 7.99 so it will add 7.99 and now we're adding again a string and a number so JavaScript will automatically convert this number into a string and combine them together that's why it puts 7.99 at the end of this string instead of doing math.**

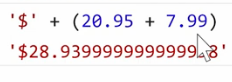
****

****

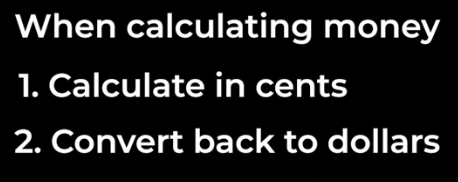
**what we actually want to do here is to add these two numbers first and then combine the total with the string so how do we do this the solution is that strings also follow the order of operations just like numbers and remember in the order of operations we can use brackets to control which part of a calculation gets done first.**

****

****

****

**unfortunately now we have another problem this math is inaccurate so remember from the previous lesson:**

****

****

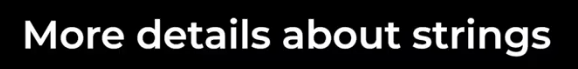
**Another example:**

**Create this:**

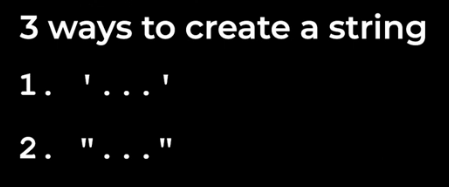
****

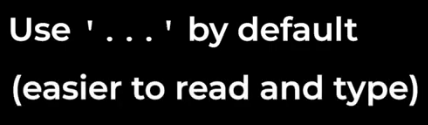
****

**4-more details about strings:**

****

**In javascript there is three ways to create a string:**

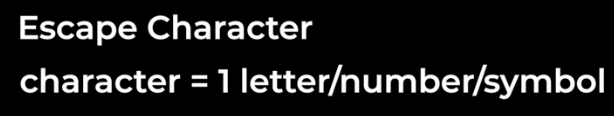
****

****

**When we have a single qoute in the string:**

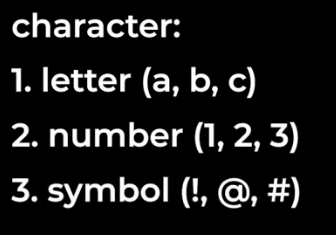
**1-we can use double qoutes**

**2-escape character:**

****

**For example:**

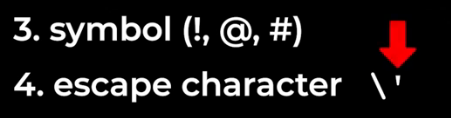
****

****

**in addition to these characters we have a special set of characters we can also use in a string called Escape characters.**

****

**an escape character looks like this a backslash and then another character like single quote :**

****

**this combination actually counts as one character in a string this character creates a single quote that is just text it doesn't have a special meaning.**

****

**this works because the Escape character backslash single quote creates a single quote that is just text it doesn't start or end the string another Escape character that's useful to know is backslash double quote this creates a double quote that is just text it doesn't start or end a string:**

****

**the last Escape character will learn in this lesson is backslash n this is called the new line character:**

****

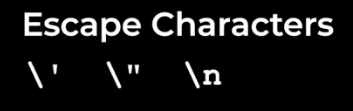
**this Escape character doesn't create the letter N anymore it creates a new line of text.**

****

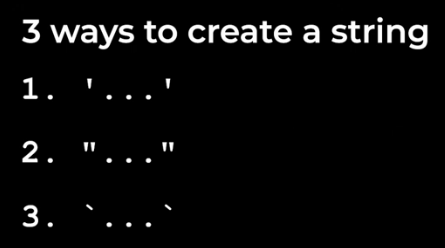
**Result:**

****

**So escape characters are:**

****

**Third way to create string in javascript:**

****

**This simbole called backtick:**

****

****

**this creates the same string as single quotes and double quotes strings created With .backticks actually have a name we call these template strings:**

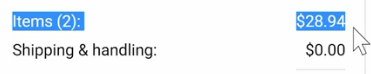
****

**and they have a name because they have some special features the first special feature of template strings is called interpolation.interpolation lets us insert a value**

**directly into a string:**

****

**for example let's say we want to create the first line of text in our project:**

****

**earlier in this lesson we created this text with concatenation which means we broke it up into parts and added themback together:**

****

**interpolation gives us a much easier way to do this:**

****

****

**Remember we use backtick.**

**So:**

****

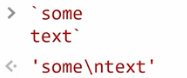
**So:**

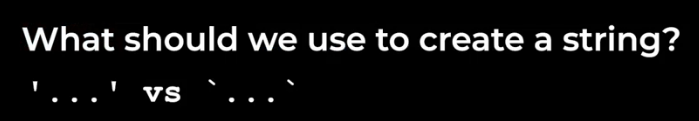
****

**next template strings have another special feature called multi-line strings:**

****

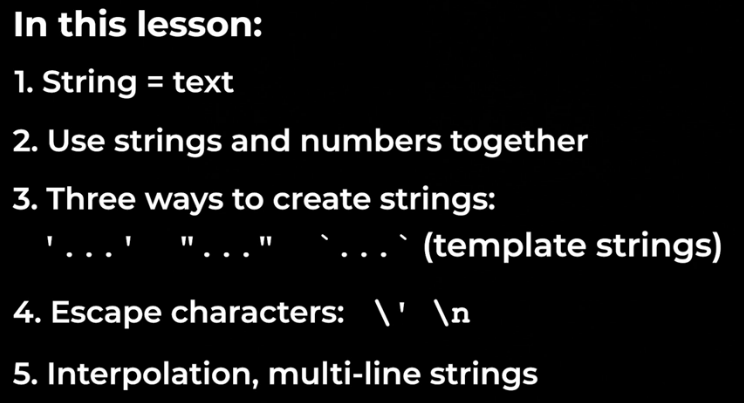
**and this feature is only available for template strings. if we press enter you can see that the new line in the string just creates the new line Escape character that we learned earlie.**

****

****

**Answer:  
**

**Sumary:**

****

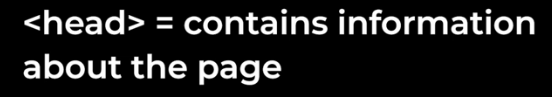
**in this lesson we learned about strings.a string represents text.we learn how to use strings and numbers to create the text in our final project. we learned three different ways of creating a string single quotes, double quotes and back ticks or template strings we learned about Escape characters and we learned two useful features of template strings interpolation and multi-line string.**

**The End-Lesson-3**

**Lesson-4:HTM,CSS Review,console.log**

**Unlike javascript we can’t use console to write and run html code instead we write HTML code inside a file using our code editor.**

**Create a file named:website.html inside our repository.**

****

**In general everything that is not visible on the page goes inside the head.**

**All time follow this structure in the HTML file:**

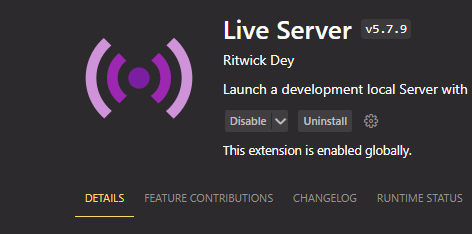
****

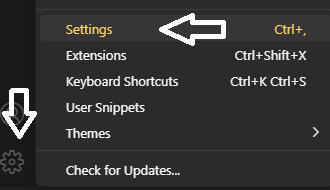
**What are the benefits following this struccture?**

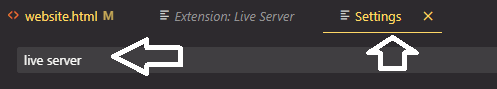
**One benefit is there are other elements that we can use inside the head like the title element.**

**Another benefit that we get from this structure is the ability to automatically refresh our web page when we change our code.every time we change our code we have to save and then manually refresh the web page we can actually automate this by following the structure and then installing a vs code extension:**

**1-install live server in vscode:**

****

**2-go to setting:  
**

**The search live server:  
**

**Then:**

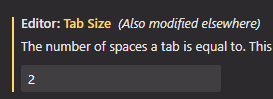
****

**3- instead of opening this file directly in the browser like before we're going to use live server to open this file to do that we're going to right click this file in our code editor and then click open with live server if we open a file with live server this webpage will automatically refresh when we change our code.**

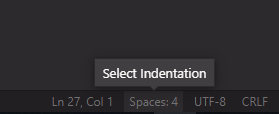
****

**by default vs code uses 4 spaces per indent however in HTML CSS and JavaScript we usually use 2 spaces per indent so we're going to update this setting:**

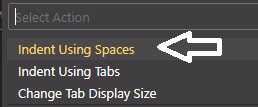
**go to settings and search tab.then change the tab size from 4 to 2:**

****

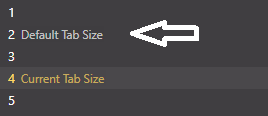
**To change tab indention for website.html go to down page and click on spaces:**

****

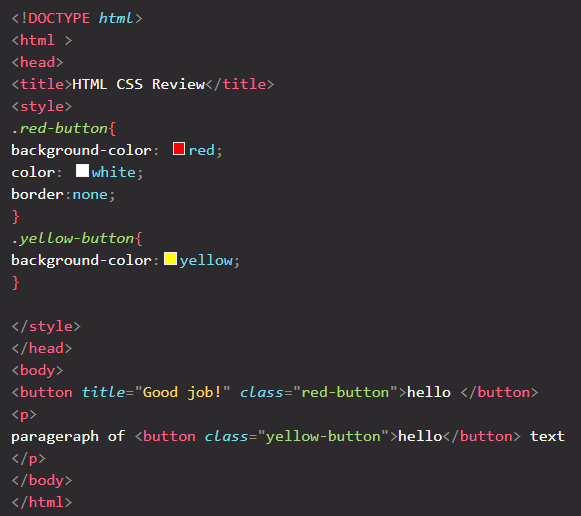
**The select this:**

****

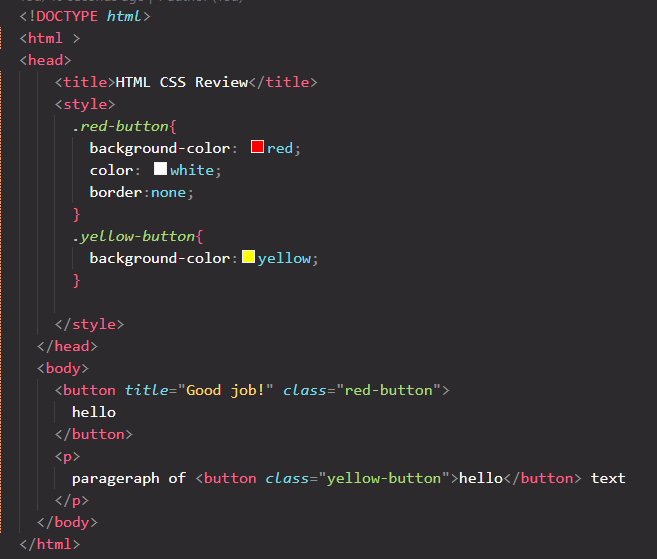
**Then select this:**

****

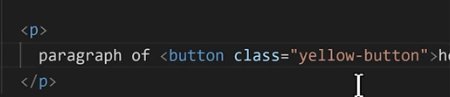
**Then select all entire website.html and press shift+tab until to remove all the indents:**

****

**Then select head tag and press tab then select everything inside head tag and press tab:**

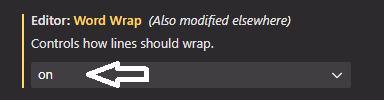
****

****

**if you look at this line of code you can see that it's really long and we need to scroll horizontally to see the rest of the line:  
**

**instead we're going to make this line wrap around if it gets too long let's go back into our settings:**

**we're going to search for wrap and we're going to look for editor word wrap:**

****

**and we're going to turn this on.**

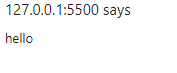
**So vscode setting are:**

****

**finally we're going to add JavaScript to this website**

**so we learned that we can use the console to write and run JavaScript well there's actually another way to run JavaScript which is inside an HTML file. let's learn how to do that so we learned that HTML has a special element that runs CSS code called the style element while HTML has another special element that runs JavaScript code and this element is called the script element so inside our body at the bottom of the body we're going to add script tag:  
**

**result:**

****

**for the rest of this course we're going to run our JavaScript code using an HTML file. the console that we used in the previous lessons is more of a way to try out JavaScript code we wouldn't use the console for a real website in addition to the script element we have another way to run JavaScript code inside HTML and that's by using an attribute.on click attribute:**

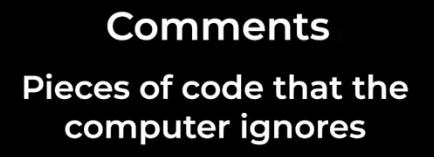
**this attribute will run JavaScript code whenever we click this button that's what on click means between the double quotes we can write some JavaScript:**

****

**The code in the script elemt run first when the page is loaded.the code in the onclick attribute runs after when we click the button on the page.**

**Some javascript features:**

**1-comments:**

****

****

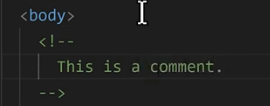
**Multiline comment:**

****

****

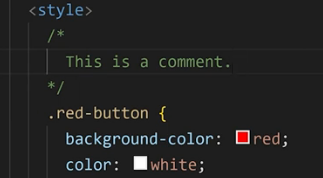
**Comment in html :**

**Has different cyntax**

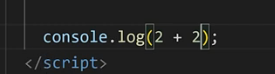
****

**Comment in CSS :**

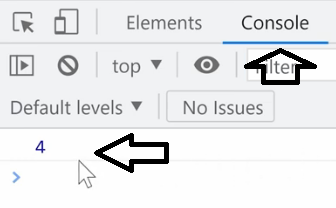
**Same with multiline comment in javascript:**

****

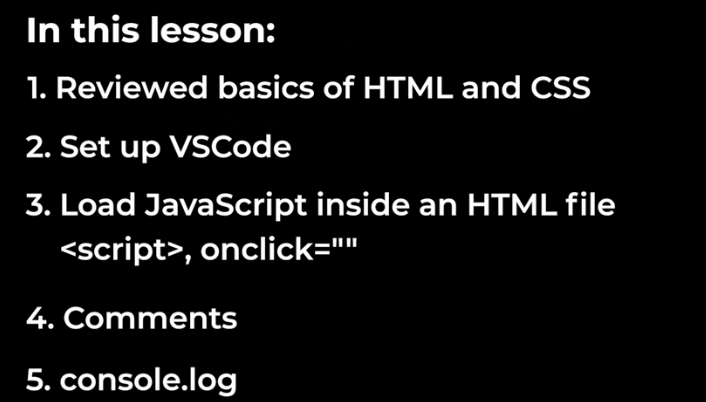
**console.log:**

****

**Result:**

****

**What we learned in this lesson:**

****

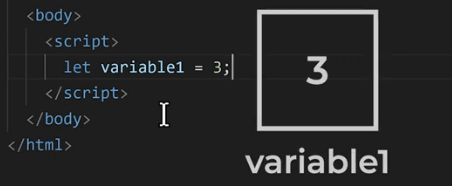
**The End-Lesson-4**

**Lesson-5:variables**

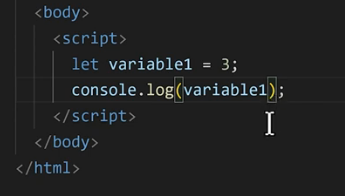
**Create a new file called 05-variables.html:**

**What is a variable?**

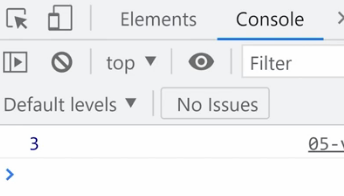
****

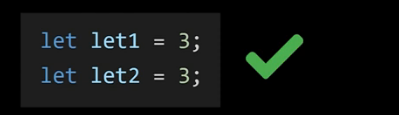
****

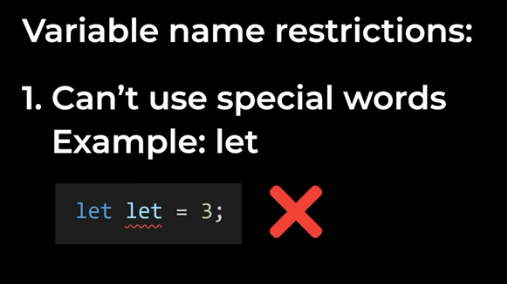
**Example:**

****

**Result:**

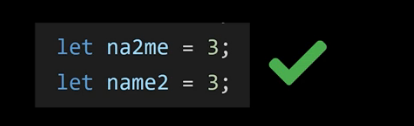
****

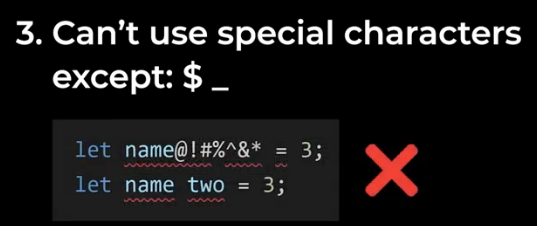
****

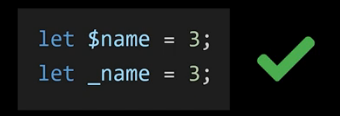
****

****

**But we can use numbers in the middle or the end:**

****

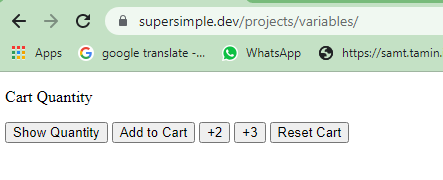
****

****

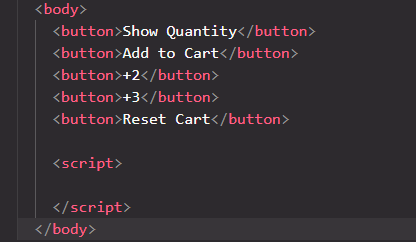
**Project:**

**Project url :**

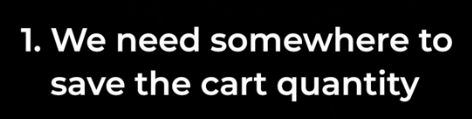
****

****

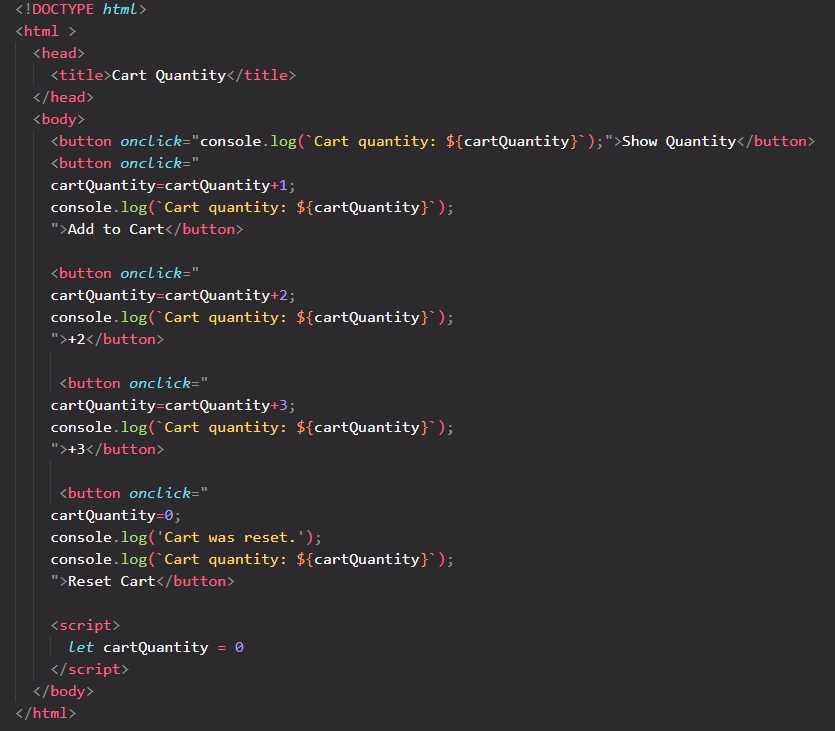
**1-create buttons in the body:**

****

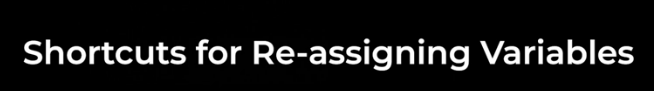
**2-**

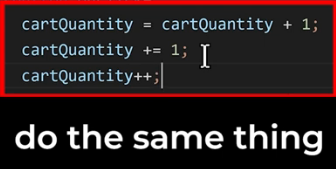
****

**Solution:**

****

**Shourtcaus for re-assigning variables:**

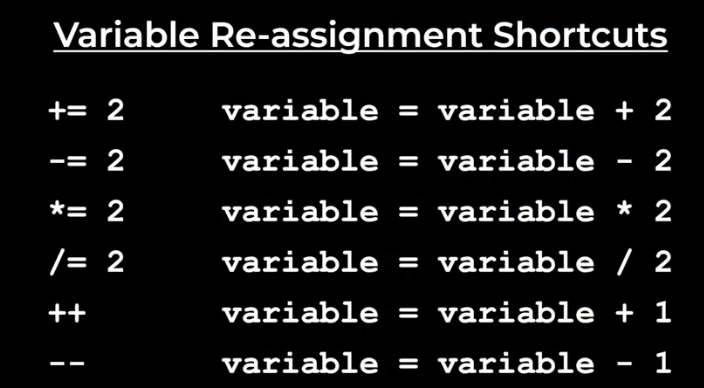
****

****

**And:do the same**

****

**Other shourtcuts:**

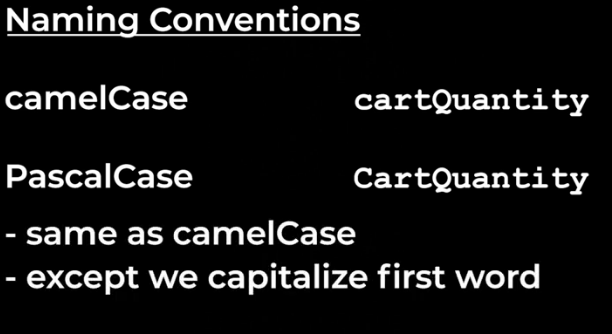
****

**Best practice for naming variables:**

****

****

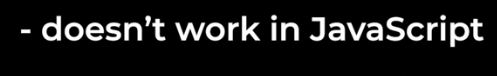
**The other way:**

****

**The other way:**

****

**Kebab-case dose’nt work in javascript because the dash is already a minus symbol.**

****

**But we can use in HTML and CSS:**

****

**Finaly we have snake\_case:**

****

**Snake\_case is used in other languages but it not used in javascript.**

**Ways to create variables in javascript:**

****

1. **Use let**
2. **Use const**
3. **var**

**const variable2=3**

**but we can’t change it’s value later.but const makes our code safer and easier to understand.**

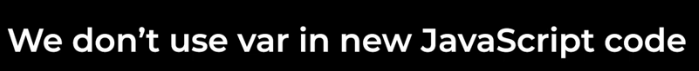
**What is best practice?using const or let?**

****

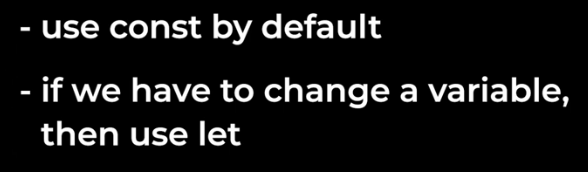
**Third way to create variables is using var:**

**var variable3=3**

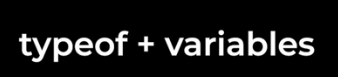
**it is just like let and we can change variable value later.but var have some issues.and because of this issues we don’t use VAR in new javascript.**

****

**Finaly how to create variables:**

****

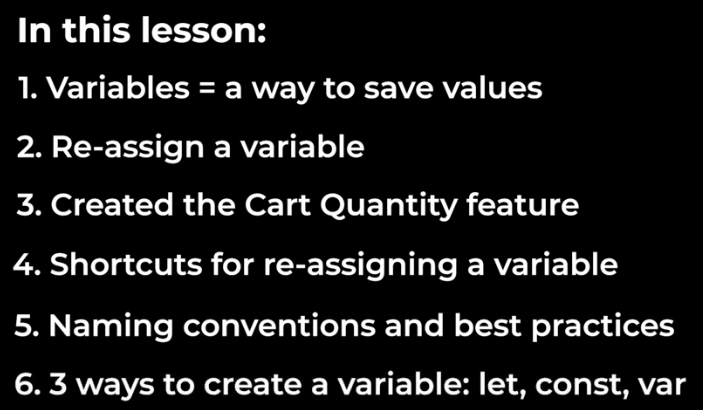
**typeof:**

****

**Example:**

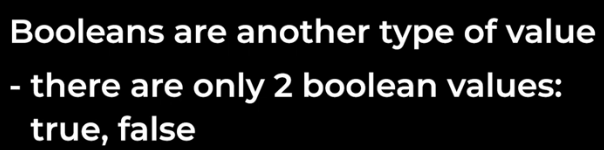
****

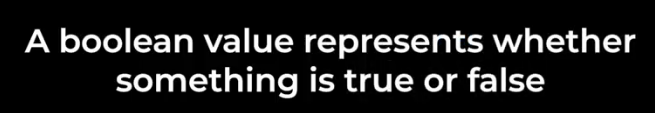
****

****

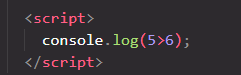
**The End-Lesson-5**

**Lesson-6:Booleans and if statements:**

****

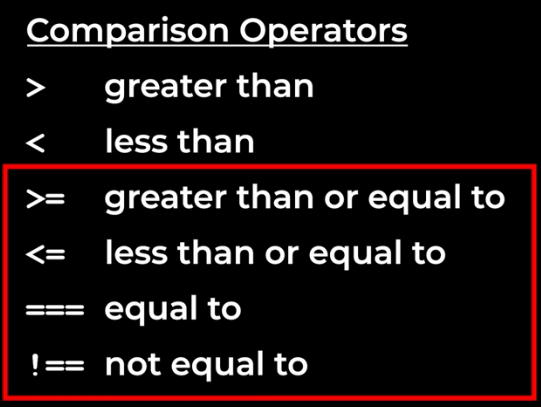
****

**Example**

****

****

**comparision operators:**

****

**Javascript has 2 ways to check if 2 values are equal.triple equals and double equals.**

**The differnce is that double equals tries to convert both values into the same type.**

****

**Example:**

****

**Result:**

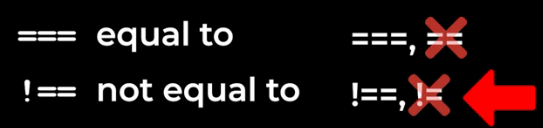
****

**It will tell us that these twwo values are equal to each other.even though one of them is a number and the other is a string.that’s because the double equals convert both values into the same type.so convert both of these into the number 5 and then compare them.**

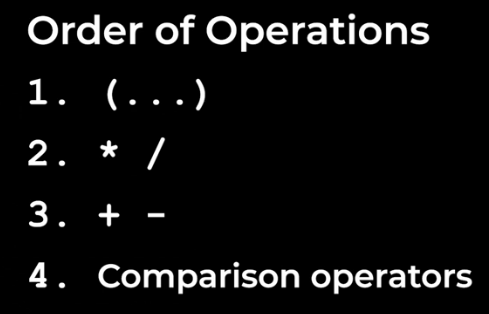
**However this not a good idea.in javascript we always use triple equals:**

****

****

****

**Order of operations:**

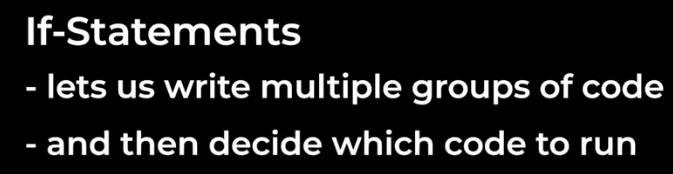
****

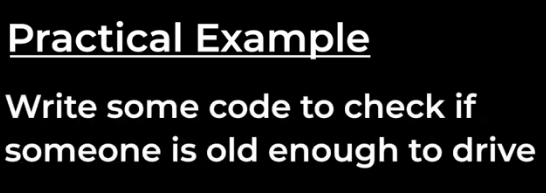
**Example:**

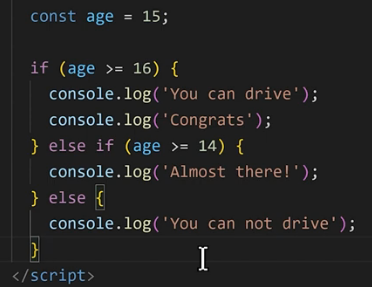
****

****

**If-statements:**

****

**Example:  
**

****

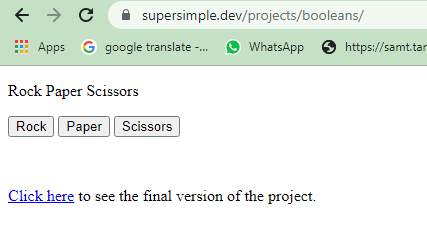
**Result:**

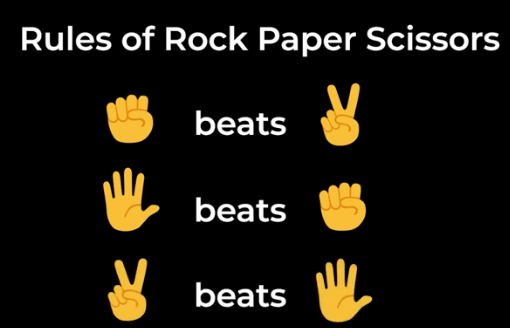
****

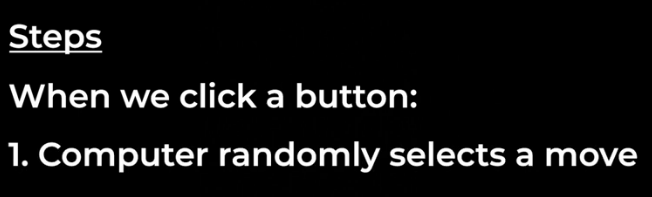
**Project:**

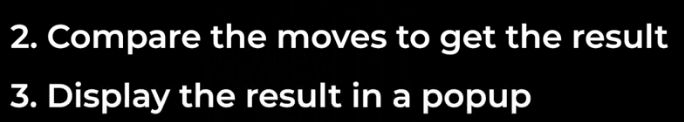
**Project url :**

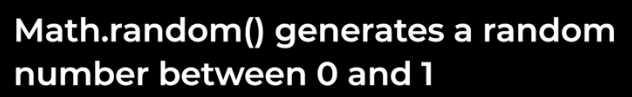
****

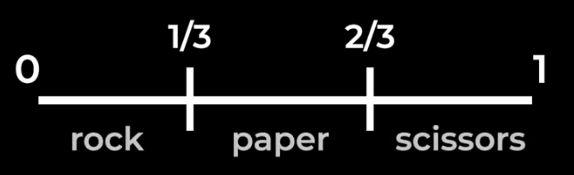
****

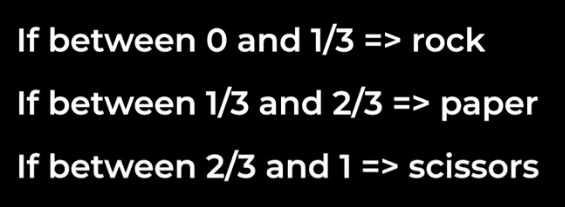
****

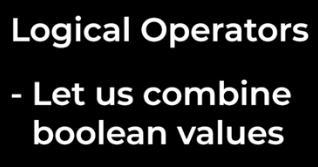
****

****

****

****

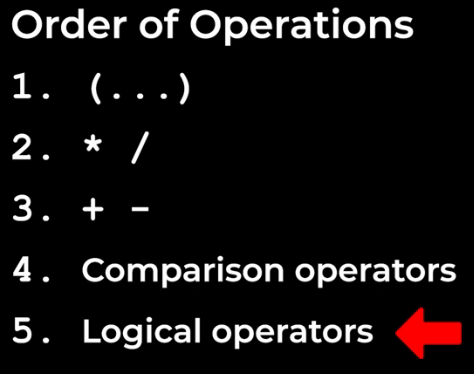
****

****

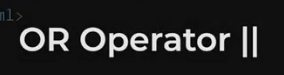
**1-**

****

**Order:**

****

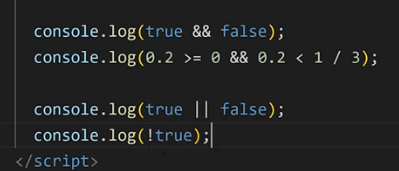
**2-**

****

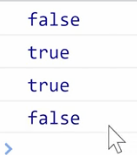
**3-**

****

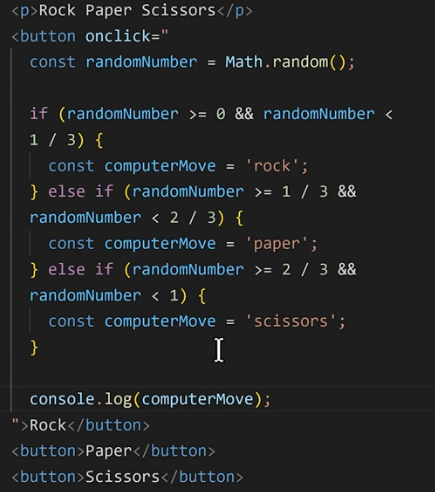
**Example:**

****

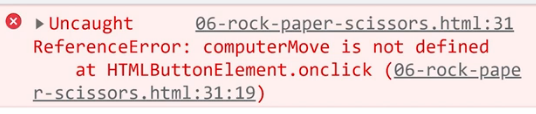
**Result:**

****

**Project solution:**

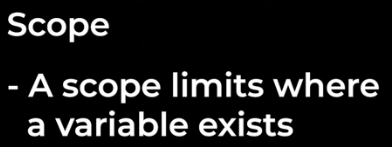
****

**We we run we have an erro:**

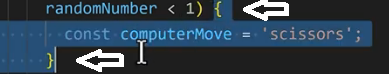
****

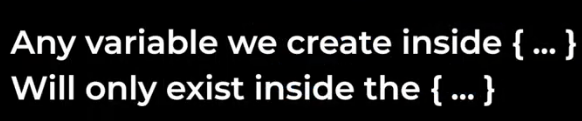
**It tell us compuetMove is not defined.but we difiend it.**

**If statement have special feature.they create something called a scope:**

****

**Basically whenever we have these curly brackets like this:**

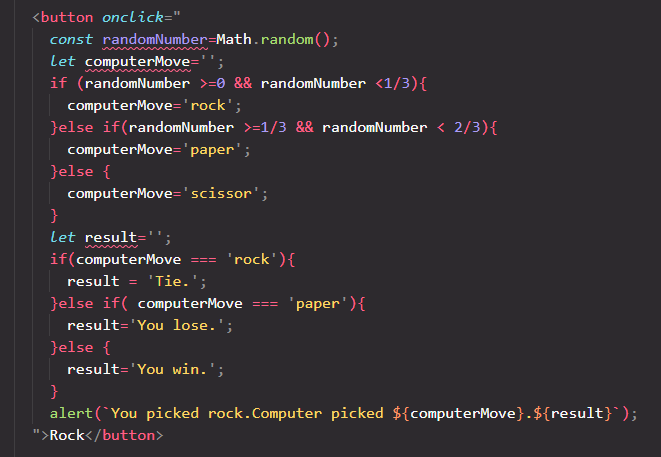
****

****

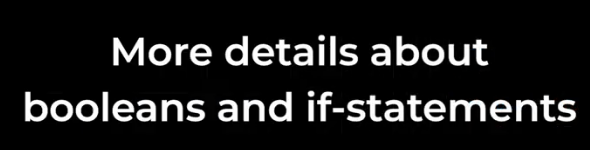
**And we can’t use these variables outside the curly brackets.so this is called a scope.scopes are a feature of many programming languages and they help us avoid naming conflicts.**

****

**So by creating a new scope all the variables names inside the scope only exists between the curly brackets and it won’t effect anything outside in the code.we don’t use VAR to create variables because var dose’nt follow the rules of scope.**

****

**Other buttons is like this with some changes.**

****

****

****

****

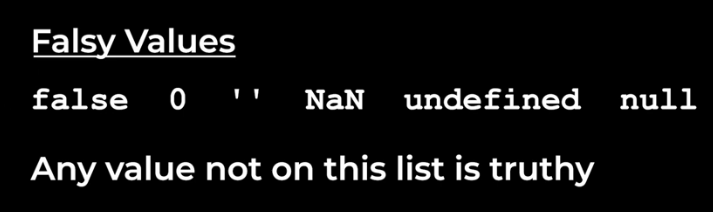
**The number 5 behaves just like true.so we call this value truthy value.**

****

****

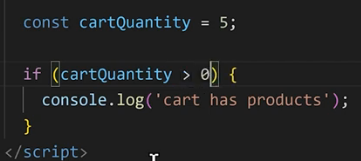
****

**So how we know a value is truthy or falsy?**

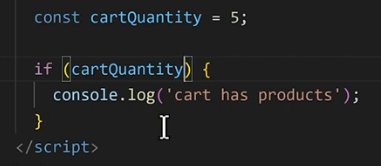
****

**Why do we use truthy and falsy values?**

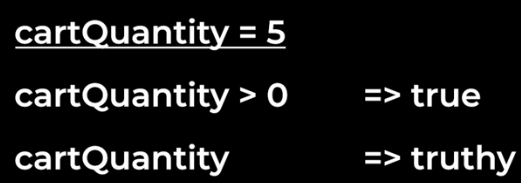
**One way we use them is like a shourtcut in our code.for example:**

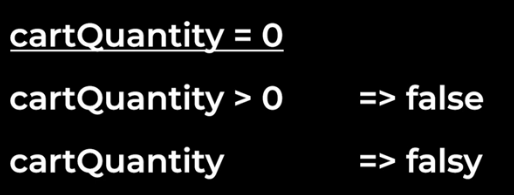
****

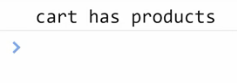
**But we can do like this:**

****

**It will behave the same way.**

****

****

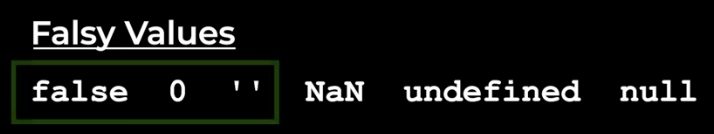
****

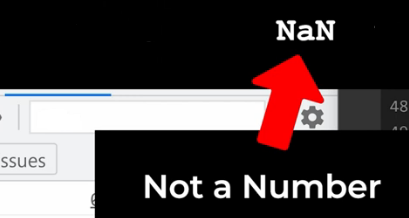
**As you can see instead of doing comparision we can use truthy and falsy values as shourtcuts.Truthy and Falsy values also work with logical operators.for example:**

****

****

**We have learned false , 0, ‘ ‘ that is empty string before:**

****

****

**We het NAN if we do some invalid MATH.for example:**

****

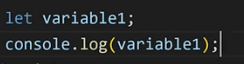
****

**Next falsy value:**

****

****

**For example:**

****

****

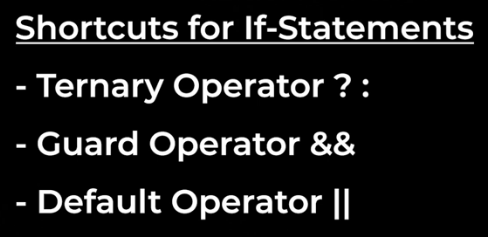
**One thing to note with undefiend is that we can only use this syntax with let.we can’t use it with const.if we really want to use this with const we have to type out equals undefined directly:**

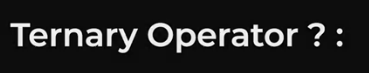
****

**The last falsy value is null:**

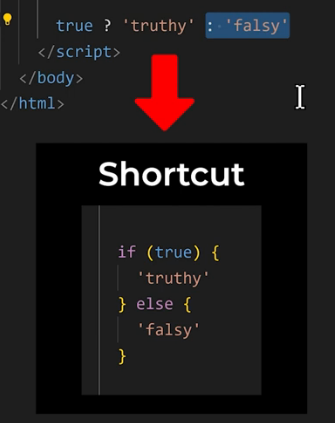
****

****

**Some shourtcuts for if-statement:  
**

****

****

****

****

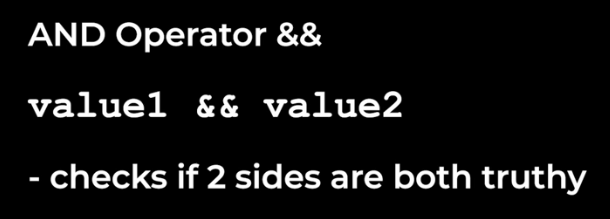
****

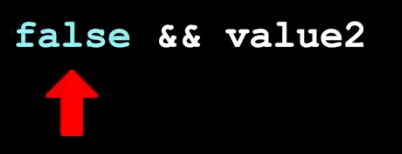
**Another example:**

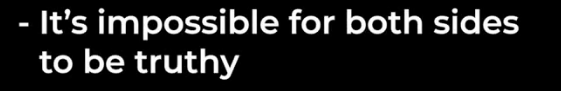
****

****

**The second shortcut:  
**

****

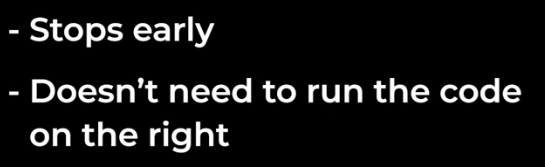
****

****

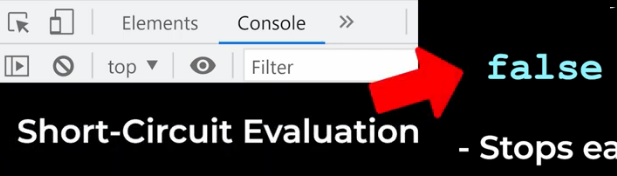
**So we don’t need the check right side:**

****

**So the end operator actually stops early:**

****

**This is called:**

****

**For example:**

****

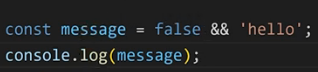
****

**As you can see we can use the value on the left to block or guard the code on the right.**

**So we use the and operator like this.we call it the guard operator:**

****

**Just like the ternary operator we can save the guard operator in a variable.**

****

**If the first value is Falsy the guard operator will stop early and the result will be the first value:**

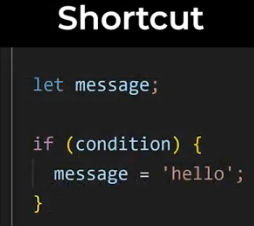
****

**Let’s change this value to truthy value like the number 5:**

**Now it will not stop early and it will give us the second value as the result:**

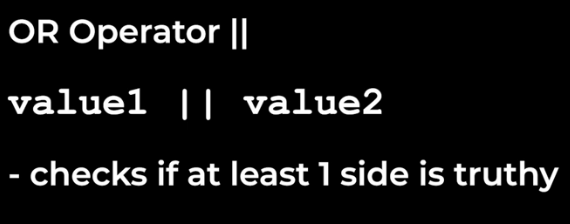
****

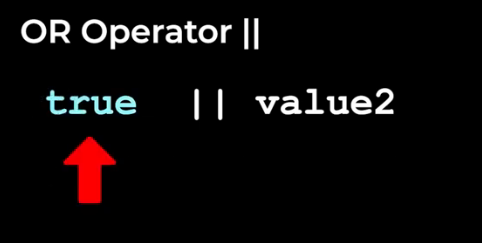
****

****

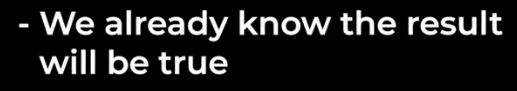
**The last shourtcut is called default operator:**

****

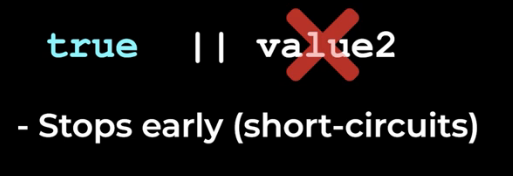
****

****

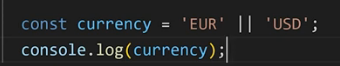
**In this situation we already know the result will be true:**

****

**Because left side is already truthy.we don’t even need to check the right side.so the or operator also stops early or short circuits.if the first value is truthy.**

****

**For example:**

****

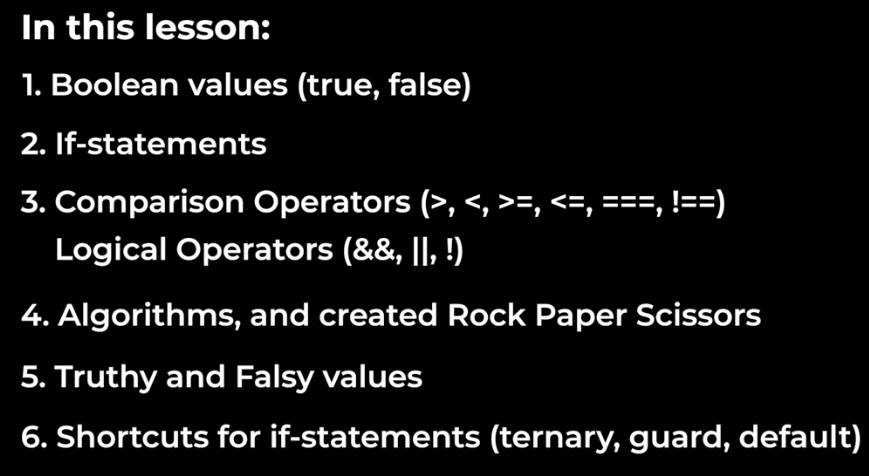
****

****

****

**As you can see we can use OR operator to set a default value.so we use the OR operator like this we call it the default operator.the default operator is sort of a shourtcut for an if statement like this:**

****

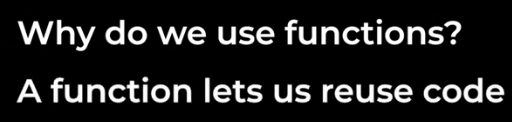
****

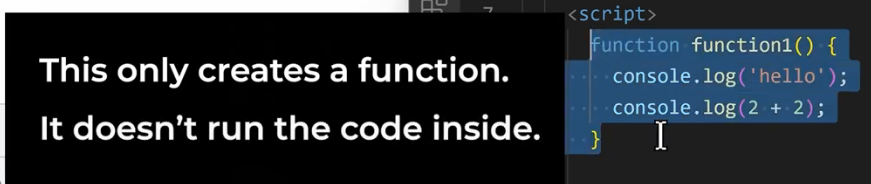
**The End-Lesson-6**

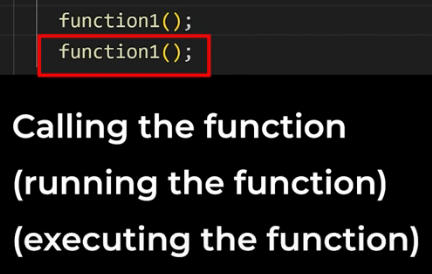
**Lesson-7:Functions:**

****

****

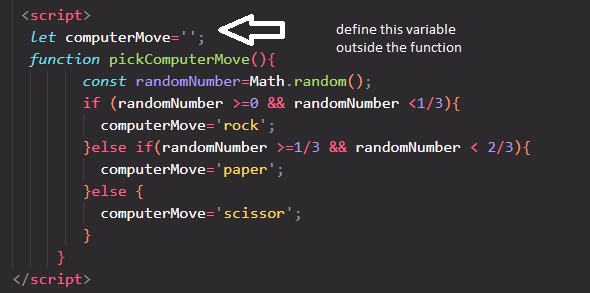
****

****

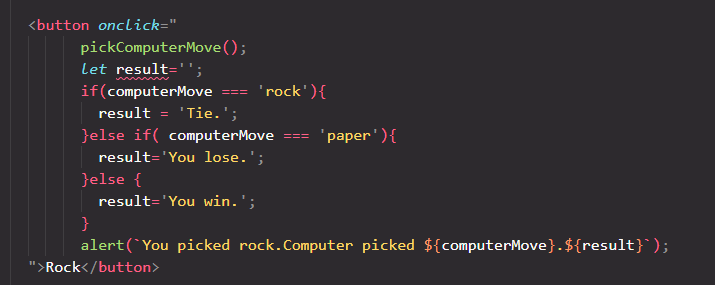
****

**Improve Rock-Paper-Scissor project:**

**Using function:**

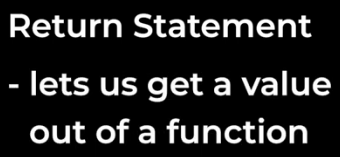
****

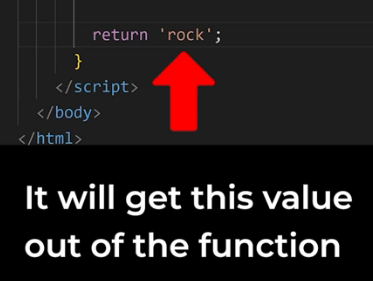
**Because function has scope too.computerMove is global variable now.**

****

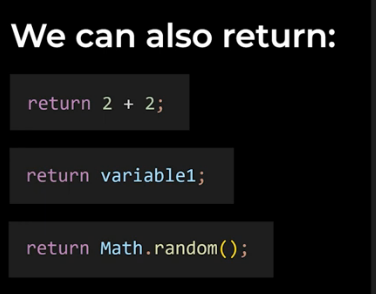
**Other buttons is like this one.**

**Return statement:**

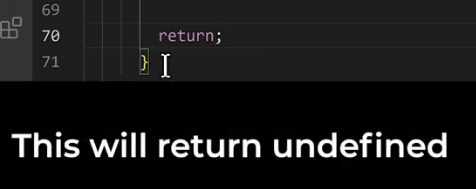
****

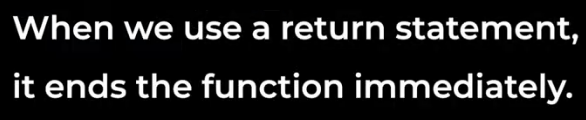
****

**Instead of returning a value we can also return:**

****

**Return undefiend:**

****

****

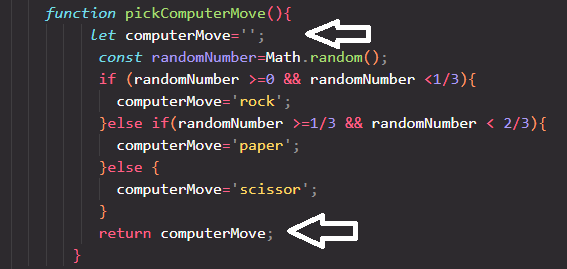
**Example:**

****

**The code after return does’nt run because once we return the code returns back to where we call the function.that’s why it’s called a return statement.**

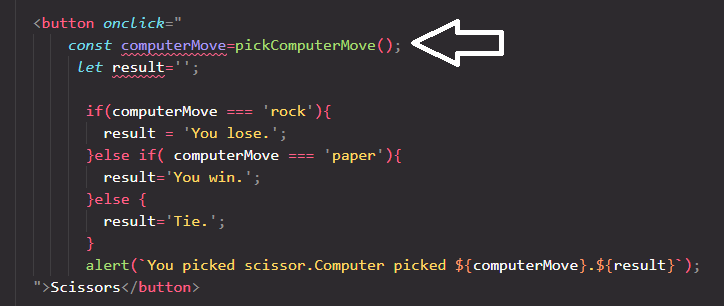
**Imporove rock paper scissor game:**

**We move computerMove variable inside the function.so how do we access this varable outside of the function?we use return:**

****

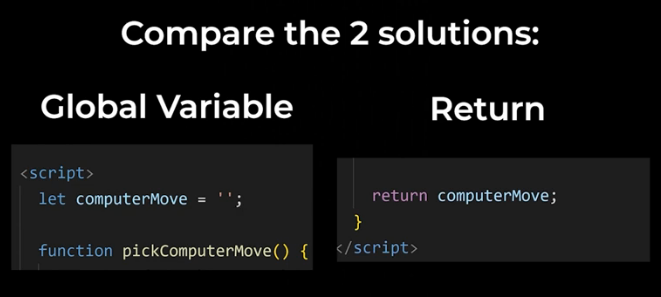
**We have used compuerMove variable out of the function because it was global variable and it was not in the scope. but now it is undefined out of the function because it is now in the scope.**

**Solution is:we can save the return value in the variable.**

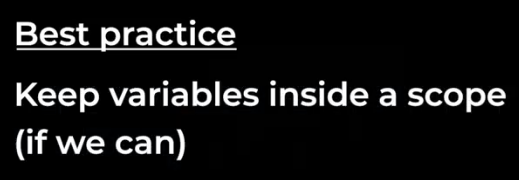
****

**This computerMove variable is different from compueterMove inside function.**

**Compare 2 solution:**

****

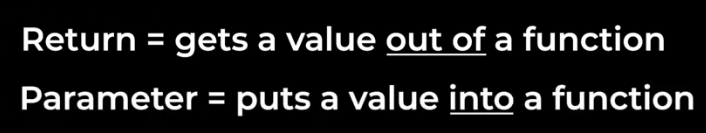
**Returning a variable is actually prefered over using a global variable.and that’s because a scope can help us prevent naming conflicts.so all the variables inside the scope will not conflict with anything outside the scope.**

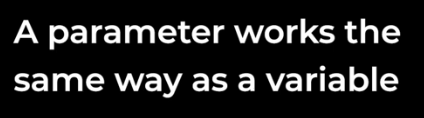
****

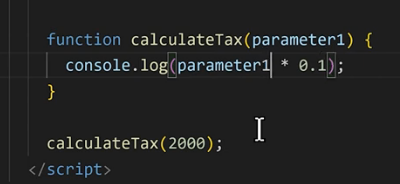
**Parameter:another feature of functions**

****

**Parameters are sort of the opposite of a retrun statement:**

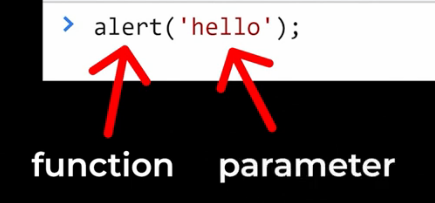
****

****

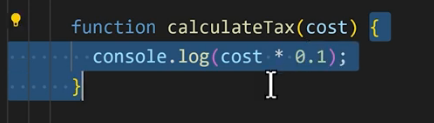
****

****

**We used functions with parameter before:**

****

**Note:a parameter only exists inside the function scope.so we can only use this paramemeter between the curly brackets we can’t use it ouside the function:**

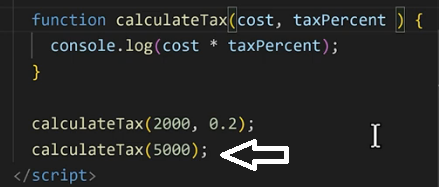
****

**A function can have more then 1 parameter.**

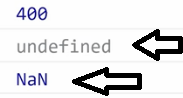
**Default value:**

****

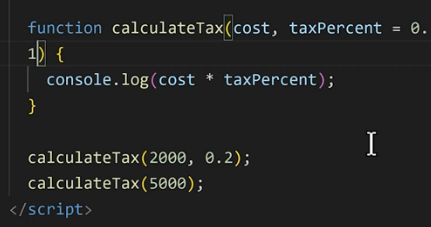
**Example:**

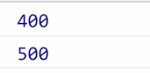
****

**We didn’t use second parameter so:**

****

**To prevent that we can use the default value:**

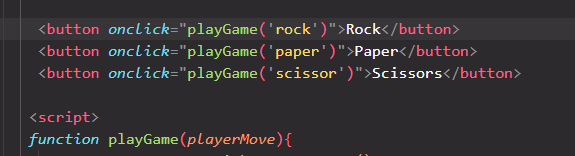
****

****

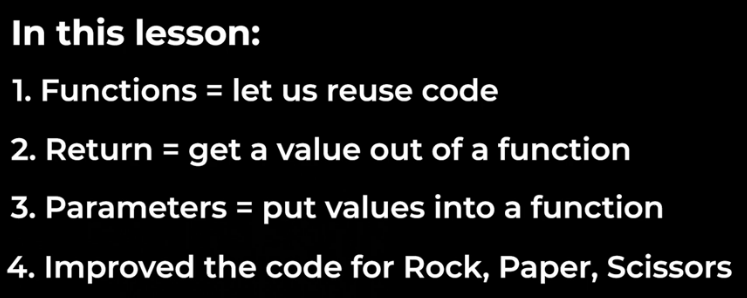
**Imporove rock-paper-scissor game:we add another function with parameter**

****

**Then use this function:**

****

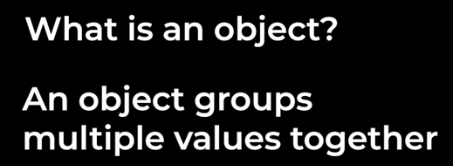
**And its work fine.**

****

**The End-Lesson-7**

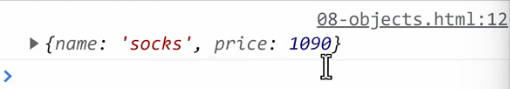
**Lesson-8:objects:**

**What is an object:**

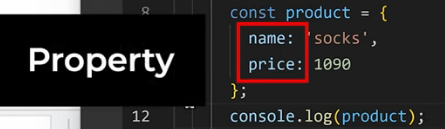
****

**Example:**

****

****

**Property:**

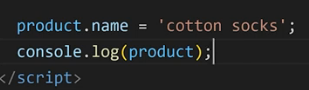
****

**We can access each property:**

****

****

**Use property to change the value inside the object:**

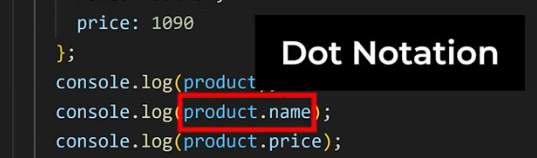
****

****

**Property-value pair:**

****

**This syntax is called dot notation:**

****

**If we access the property that doesn’t exist the value wil be undefiend:**

****

**We can add new property with dot notation:**

****

****

**We can also remove a value from an object:**

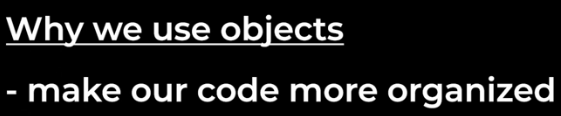
****

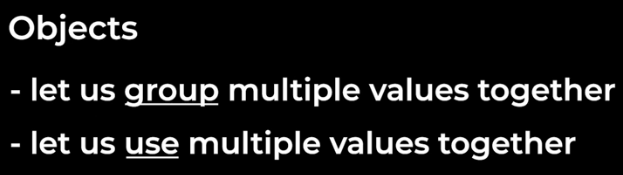
****

****

****

****

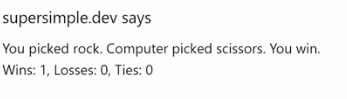
****

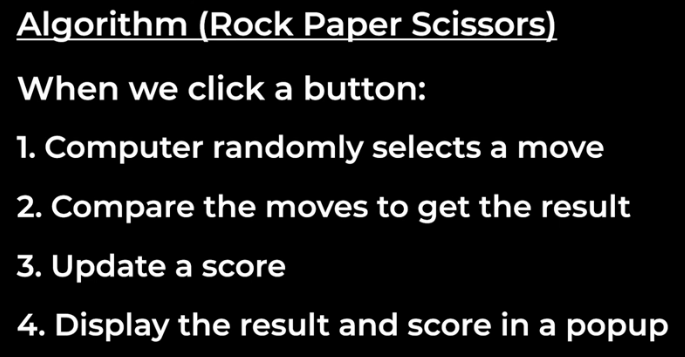
****

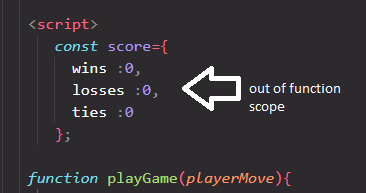
**Practical example of object.we are going to add score to rock-paper-scissor game:**

****

**After play game:**

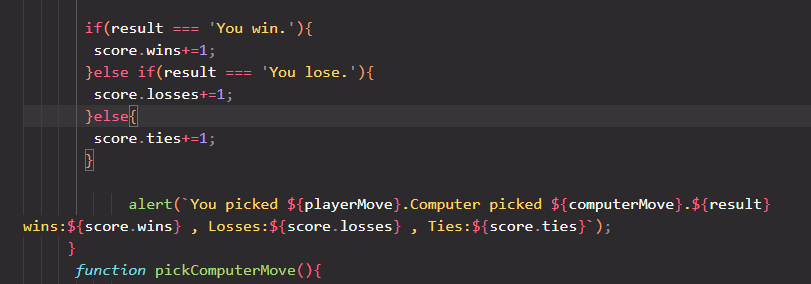
****

****

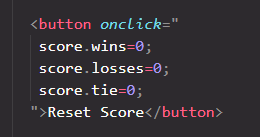
****

**we’re creating this variable outside of a function and outside of a scope and that’s because every time we make a move we need to update the score from last time.so in order to save the score from last time we need to keep it outside the function.if the score was inside the function,the function would create a new score every time which is not what we want.**

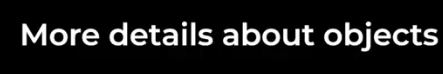
**Then:**

****

**For reset button:**

****

**Everything works fine.**

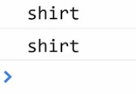
****

**Is another way to access the value inside.we used dot notation before:**

****

**In javascript there is another way to accessing value inside objects which is using square brackets:inside the squre brackets we have string:**

****

****

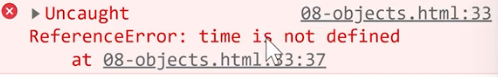
**When do we use bracket notation instead of dot notation?**

****

**Example:**

****

**If we save this property gve us an error:**

****

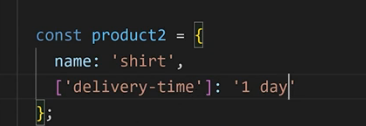
**And that’s because javascript thinks this dash is a minus sign and it’s trying to do MATH.so in oreder to use a property like this we have to use bracket notation:**

****

****

**It no longer give us an error and because delivery-time doesn’t exists in the object we get the value undefined.**

**We can also add property with bracket notation:**

****