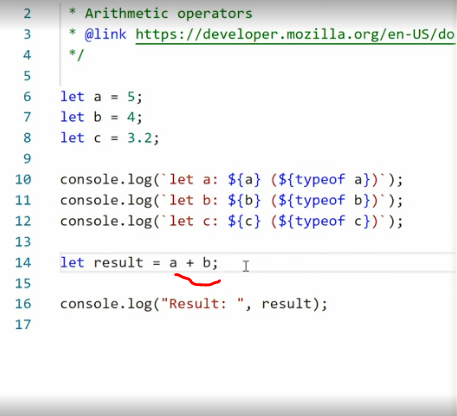
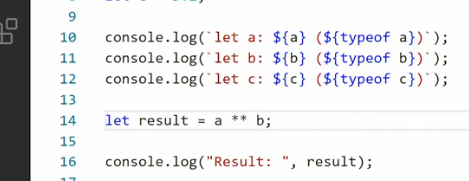
* - [Instructor] Computers are really good at computing as in doing math and in JavaScript, you'll often have to do math to convert or combine or subtract or multiply or divide values or do other things.
* For all of this we have standard arithmetic math operators.
* In the exercise files I've set up an example so we can play around with this.
* So you can see it here.
* We've set up some lets with different values.
* Let me just output those lets into the console so we can see what they are.
* And then we have this lets called results down here where we can do a math operation and the result is placed inside the result let and then we just output it.
* So right now we are doing an addition using the **plus** symbols.

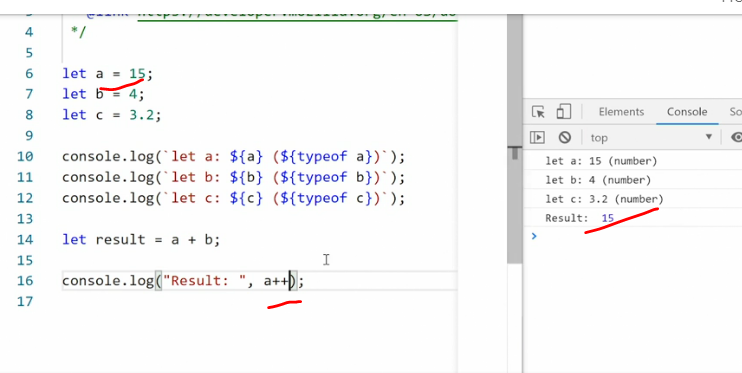


* So we are taking the value of A plus the value of B the result is nine because it's five plus four.
* We can also do a subtraction by putting in a **minus** symbol.
* This gives us A minus B the result is one.
* If we want to do **division, you put in a forward slash** this becomes A divided by B that's 1.
* 25, five divided by four.
* And finally, we can do **multiplication by adding an asterisk**.
* So A times B five times four equals 20.
* You can also do more advanced arithmetic here so we can **wrap this in parenthesis and then say A times B divided by.**
* And then we can add in a number, so we can say two.
* So you remember A times B five times four is 20.
* And if I divided that by two, we get 10.
* We can also pass in another value here so I can pass it in the C value, which is 3.
* 2, run that and we get 6.
* 25.
* So using these standard arithmetic operators, you can do pretty advanced math, right inside JavaScript.
* In addition to addition, subtraction, division and multiplication, we have some more advanced operators we can use.
* One is, this one, a **percentage** symbol this signifies a **modulo or modulus**, and it gives us the remainder left over when we divide the first number by the second number.
* So in this case, we're dividing five by four.
* This will give us one instance of four, and then we have one left over.
* So the modulus will give us one.
* If I then change the value here to 10, you'll see what I mean.
* So 10 divided by four gives us two times four which is eight.
* And then we have two leftover, save that and we get three.
* If I change it to 15, we get four by four by, four plus four plus four, which is 12.
* And that gives us three remainder, right? Save that and we get three.
* Modulus is useful when you're doing things like testing for prime value or other things.
* And it is something that's often used in math.
* We can also do **exponentiation** so if we have 15 and we want to say 15 to the power of four, we can use **two asterisks like this star star**.

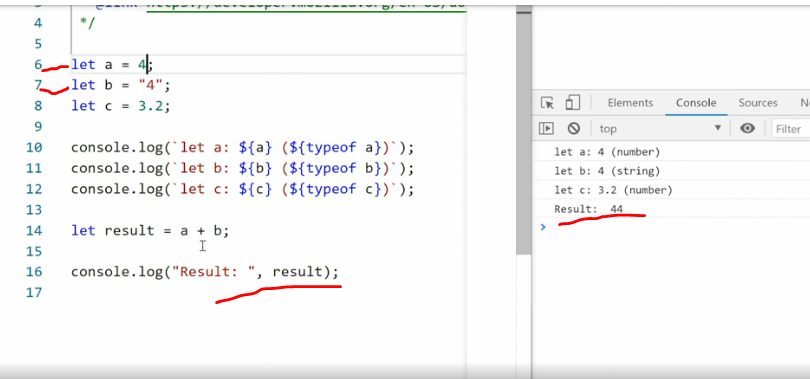


* So A asterisk, asterisk B is A so 15 to the of power B four.
* Save that, and we get a huge number, 50,625.
* Now there's one thing that you often do in JavaScript which falls a little bit outside of normal math.
* And that is, y**ou often have a number and then you want to increment it or decremented by one.**
* So you want to add one to the number or you want to take one away from that number.
* Because this is so common, we have a specialized formula for it in JavaScript and it works a little bit differently.
* So take a look at this, if I go down into the results here, and I say, plus plus A,
* Graphical user interface, text

  Description automatically generated
* what do you think the result will be over here? Now A is 15, but when I save this, the output is 16 because I've taken A, and that I've incremented by one.
* If I add two minus symbols in front of it instead, I get 14 because I took A and then I took one away.
* But it gets more interesting than that because sometimes you don't want to add the number first.
* You want to do something with the number and then add an increment afterwards.
* For that we can say **A plus, plus.**



* Now, if I run this in the browser you'll see we get 15 as a result but we also added that plus plus.
* To see what happens, I can then console log out just A, when I save this you'll see A is now 16.
* So what happened is I took A then output the value 15, then I incremented it.
* So then the next time we use A it'll be 16 instead.
* So this is an easy way of incrementing or decrementing a number.
* And by placing the plus symbol either before or after the value, we can control how that value comes out and when the instrumentation happens.
* One more thing worth noting here, and I need to reset my example.
* Remember how I said JavaScript often interprets a single number inside a string as the number itself.
* Well, things get weird if you try to mix true numbers with string numbers and then use the plus symbol for addition.
* So if I go up here you can see right now it says 15 and 14, right? So A and B it's 15 plus 14, save it and we get 19.
* What happens if I put quotation marks around the four? And you'll remember if I did a comparison here from between A and B and A is a number and B as a string with just a number four, JavaScript, we'll go, sure they're the same.
* But, if I do this using the plus symbol, look what happens.



* I get the number 44.
* That's because the plus symbol in this context is used as a string combiner.
* So we are putting the number from A next to the number from B.

Graphical user interface, text, application

Description automatically generated

* If I put 15 as the value in A here, it will say the number we get out is one, five, four, which is confusing.
* What is even more confusing is **this only happens for the plus symbol.**
* So if I used a minus symbol instead, then JavaScript will still interpret B, which is a string as a number.
* Graphical user interface, text, application

  Description automatically generated
* So if I save this, we suddenly get 11 that's, 15 minus four.
* And so this issue around making two numbers placing next to one another only happens with the plus symbol.
* *That means when you're doing math, using JavaScript and you are doing addition, you have to make sure you're actually operating with real numbers and not strings.*
* Otherwise you get weird output that makes no sense From a math perspective.