Since we are learning JavaScript we had to do some research on different types of topics and this Is all the information that I've found.

The typeof operator will determine which type of data something is. So typeof 32 will return "number", because 32 is a number, the same as typeof 12 and so on. It will also consider infinity and Nan as a number, even though Nan is Not-a-number.

Number isn't the only type of course, you also have strings, which are sequences of one or more characters and you can recognize them by the " ". This means that typeof " " or typeof " 0" will always return "string" no matter what's inside of them. You can also convert anything into a string with string(), so typeof string(34) will return "string".

Boolean determines if something is true or false, so typeof true will return "boolean" and typeof false will return "boolean". Boolean() can convert values, so typeof Boolean(1) will return "boolean".

An object is a standalone entity with properties and you can create an object with object(), so typeof Object(1) will return "object"

If you want to check if your data is a function you can check it by using either typeof function () {} or typeof class a {}

All these function can save you a lot of time when your checking what type of data something is, instead of going through your code line by line you can use these simple operators.

Variables on JavaScript are used as containers to store data and you can do that in three different ways, you can use var, let or const. Var is the oldest of the three and is the best to use if you want your website to be able to run on older browsers. Const is the best to use if you want to use a variable, that you won't change. If you want to make a sum, you can use const to set the values of the sum and the outcome off the some will be a new variable which you can best declare by using let. Let is great to use for variables which can change.

Logical operators are used with Boolean values, so they can help determine if certain values are true or false. You can do that in three different ways using logical

Type coercion is when JavaScript converts data from one data type to another, for example number into strings. If you make a sum consisting of a string + a number it will convert the number into a string, so "3"+1 will be 31 instead of 4, because it converts the 1 into a string, so the calculation it make is "3"+"1" JavaScript convert it into string + string.

JavaScript has tons of different expressions and operators. An arithmetic operator will take numerical values and make a calculation that ends up being one singular number. The arithmetic operators we use are +, -, * and /, but arithmetic operators can also be string operators like + is. Just like + adds number on to each other it can also add strings to each other, for example "java"+ "script" will be JavaScript, but you could also make sentences with it. When trying to make a sentence you have to include spaces and spaces in strings will stay put, so "I "+"am "+"Quinn" will return I am Quinn, while "I"+"am"+"Quinn" returns IamQuinn.

In JavaScript you have multiple types of value comparisons. The equality operator == will compare two variables and check if they are equal if so it will return true if they are not equal it will return false. So 4 == 4 will return true and 4 ==2 will return false, same for strings.

The strict equality operator === works a bit differently it will also compare two variables just like the equality operator, but in this instance it will consider different types of variables as false. So 4===4 will still return true, but “4” === 4 will return false, because even though it’s the same number it will not convert the types and so it is different and therefore false.

The inequality operator != will compare if two values are not equal. So know 4!= 4 will return false just like strings and 4!=2 will return true, since they are not equal.

The strict inequality operator !== which also is really similar to the inequality operator, but in this instance just like the strict equality operator it will it not convert the types and therefore the variables are different and true. So 4!==4 will return false and 4!==4 will return true, but now “4”!==4 will return true instead of false, since they are the same number, but not the same type.