**[PPT SLIDE No. 18]**

**Welcome back.**

**Let's now start to really learn the JavaScript language**

**starting with values and variables.**

**And let's start**

**So in this section, I will be using persons as examples**

**for example, a person's name, a person's age or a job.**

**All right?**

**But anyway, let's talk about values now.**

**So a value is basically a piece of data.**

**So it's the most fundamental unit**

**of information that we have in programming.**

**For example, this text, Jonas, is a value.**

**All right?**

**So this here is a value.**

**Values: It is basically a piece of data.**

**For example**

**‘Jonas’**

****

**And again, if we want to actually see this in the console**

**let's say console.log and then open up the parenthesis**

**and then close them here and then the semicolons.**

**So if I save and reload, then of course we get Jonas.**

**Okay?**

****

**And so Jonas here is the value**

**or we can, of course, have different values**

**for example, here 23 as the value,**

**right?**

**And so now we see 23 in the console.**

****

**And the same is true, here.**

**So in this line of code where the values are actually 40**

**and eight and 23 and 10.**

**So all these four are different values.**

**And then these mathematical operators here**

**joined them together to form just one value.**

**And that value is then 61.**

**Okay?**

****

**And so a value is basically the smallest unit**

**of information that we have in JavaScript.**

**Now, one extremely useful thing that we can do with values**

**is to store them into variables.**

**And so this way we can reuse them over and over again,**

**for example we can say, let,**

**and then the name of the variable**

**and then assign a value like Jonas to this variable.**

**Okay?**

**And so what we did here is called declaring a variable.**

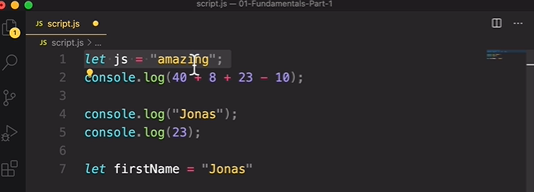
**Okay?**

****

**And so this will actually create a real variable**

**in your computer's memory**

**and we'll store this value inside of that variable.**

****

**And actually that is exactly what we did up here,**

**right in the beginning.**

**So here we declared a variable called JS**

**and assigned the value of amazing to that variable.**

**Right?**

**And so now it hopefully makes sense to you**

**what we did back then.**

**So I like to imagine a variable, like being a box.**

**So in the real world, a box can hold some object**

**for example, a book,**

**and we can then write a label on the box**

**to describe what's in it.**

**And then we can find the object later when we need it**

**by using that label.**

**And variables actually work in the exact same way.**

**So here, basically we have a box called firstName**

**and into that box, we put the value of Jonas.**

**And now if we want to use this value**

**all we have to do is to basically use this label**

**or in other words, this variable name.**

**So first name in this case.**

**And so let's use it actually.**

**And for that, once again, I will use console.log**

**and you will see me using console.log**

**throughout this entire course, actually.**

**So whenever we need to output something from our code**

**to the browser, we always use console.log like this.**

**And so here now, instead of passing a literal value**

**like we did here and here and here,**

**now we can pass in,**

**So we can write here, the variable name,**

**so first name.**

**Okay?**

**And if I save and reload, it let's see what happens.**

**Indeed, we get Jonas.**

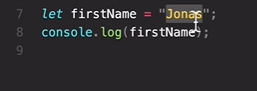
**So from line number eight.**

**And so that's right here.**

**And that means that indeed**

**or variable declaration up here worked.**

**And if we now change this**

****

**to Bob, let's say, then here we should get Bob.**

**Right?**

****

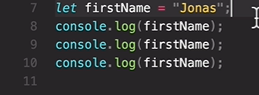
**Let's put it back.**

****

**And of course we can now use this variable name many times.**

**So let's just copy and paste this here a couple of times**

**and give it a save.**

****

**And so now we have it three times**

**and that means that whenever JavaScript sees**

**this variable name here it will basically replace it**

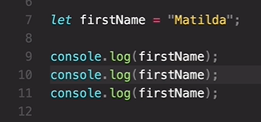
**with the original value that we assigned to the variable.**

**So again, that's Jonas and this is extremely useful**

**because now if I want to change the first name**

**to something else, I only have to do it in one place.**

**So just here.**

****

**So let's change it here to Matilda**

**and if I save it now and reload,**

**of course it will change in all the three places.**

**So this is one of the big advantages of variables.**

**Without variables, I would now have to manually change**

**the value everywhere to Matilda.**

**But like this everywhere where I reference this variable**

**it will automatically changed to Matilda.**

**So that's one of the most important things**

**to keep in mind about variables.**

**Great.**

**And now that we know what a variable is**

**let's just very quickly talk about conventions**

**and rules for naming variables**

**because we shouldn't just give random names to variables.**

**So first, the way that I named this**

**variable here is called camelCase,**

**camelCase means that whenever I have multiple words**

**in a variable name, I write the first word with a lowercase**

**and then all the next words with upper case, like this.**

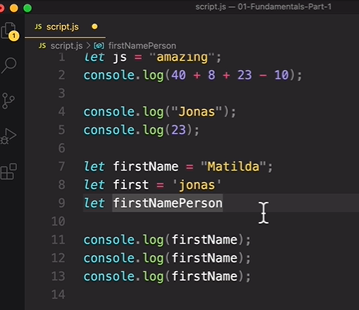
**So if it was just first, we would write it like this.**

**But if we had more than one word, like first name person**

**then you'll see that all the subsequent words**

**are written with this uppercase letter.**

**And this is kind of a standard in the JavaScript world.**

****

**But of course, there are other ways of naming variables.**

**For example, we could write**

**first\_name**

**with an underscore like this,**

**and this is very popular in other languages**

**like Ruby or like this.**

**Okay?**

****

**And you can use whatever you like most**

**just keep in mind that it's kind of a standard**

**in JavaScript to write variable names like this.**

**So usually whenever you see other people's code**

**the variables will usually be written**

**using a the camelCase notation.**

**So that's kind of a convention of**

**how to name variables in JavaScript**

**but there are also some actual hard rules in JavaScript**

****

**about how we can name variables.**

**For example we cannot write something like this**

**like three years and set it to the value of three.**

**So this is an illegal variable name**

**and VS code actually warns us right away here.**

**And again that's because this variable name**

**starts with a number.**

**And if we try it to load this**

**in JavaScript, we would get this error.**

**So that's an invalid or unexpected token here.**

**And it's important to actually start reading**

**these error messages right from the beginning of the course.**

**So we see that this is a so-called syntax error**

**which means that we did a mistake in writing or a code.**

**So that's a mistake in the code's syntax**

**and we can also see the line where the error occurred.**

**So line 13 so that we can then go ahead and fix it**

**and notice how this error was shown in the console.**

**Even though we didn't use console.log in this case.**

**Okay?**

**So we don't need to use console.log**

**for the console to show errors.**

**All the errors that we make**

**will always end up in the console.**

**And also here with this, well, error icon.**

**Now anyway, back to our naming conventions here.**

**So we already learned**

**that variable names cannot start with a number.**

**And in fact, variable names can only contain numbers**

**letters, underscores, or the dollar assign.**

**So for example, if we try to write**

**Jonas and Matilda**

**equals JM**

**we would, once again,**

**get a syntax error.**

**Right?**

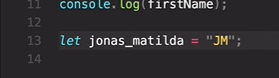
**And this time it's telling us unexpected token, &.**

**And, and again that's because this symbol here**

**is illegal in variable names.**

**They can only contain numbers,**

**letters, underscores, or the dollar sign.**

****

**So this here, for example, would be allowed.**

**So if we reload now, we get no more errors here.**

**Another error might occur when we try to name a variable**

**using a reserved JavaScript keyboard.**

**For example, if we did this, so new,**

**for example set it to 27,**

**then this would also not be allowed.**

**So you see unexpected token new.**

**And that's because new is a reserved keyword in JavaScript**

**as we will see a little bit later.**

****

**And the same goes for something like function.**

**So again unexpected token here,**

****

if we wanted to fix that we could,

**for example, start this variable name with an underscore**

**or with the dollar sign.**

**So that's the only two symbols that are allowed**

**besides letters and numbers.**

****

**Another variable name that's kind of reserved**

**but still actually allowed to use is the word name.**

**So we could do this.**

**And it would actually work.**

**But in some cases, this creates some problems.**

**Because again, this is kind of a reserved keyword**

**but it's still legal to use.**

**But since it is a keyword,**

**never call your variables just name.**

**Okay?**

**Let name = “Jonas”**

**That's why I always call them first name, like this.**

**Let's actually get rid**

**of this one because we already have first name up there.**

**Now another convention is**

**that we should not start a variable name**

**with an uppercase letter, so we should not**

**do**

**this.**

**So that person, Jonas.**

**Now again, it's a convention, so that's not illegal.**

**It's just that we use this kind of variable names**

**with an uppercase letter for a specific use case**

**in JavaScript, which is object-oriented programming**

**as we will see later in the course.**

**Let person = ‘jonas’**

**as we will see later in the course.**

**So for now never do this, but instead you should write it**

**like this, with a lower case letter starting.**

**On the same note variables that are all in uppercase**

**are reserved for constants that we know will never change.**

**For example, the number PI**

**which is like 3.1415, and so on and so forth.**

**So we know that this number is never gonna change.**

**And so that's a constant.**

**And for that, we have a convention of writing it**

**in all upper case.**

**and VS code actually marks this variable name**

**with a different color**

**because it knows about this convention that we use.**

**Of course you could also write it like this,**

**but then this would be kind of weird actually.**

**So if it's a real constant, write it in uppercase**

**like this, so that's a pretty normal convention there**

**in programming as a whole.**

**Let PI = 3.1415; --- use this**

**Let pi = 3.1415; ---- not this**

**Finally, to finish this lecture**

**let's talk about one final convention,**

**which is to make sure**

**that our variable names are descriptive**

**and that's very important to write cleaner code.**

**So when you name your variables**

**it should be really easy to understand**

**exactly what value the variable is holding**

**just by reading the name of the variable.**

**And that's kind of what we did up here**

**by calling this one first name.**

**But let me show you another example.**

**Let myFirstJob = “Programmer”**

**Let myCurrentJob = “Teacher”**

**For example, let's say my first job,**

**programmer**

**and then my current job,**

**teacher.**

**So this is much, much better than writing this.**

**Let job1 = “Programmer”**

**Let job2 = “Teacher”**

**So job one programmer**

**and job two a teacher.**

**So which set of variables**

**do you think is more descriptive?**

**Is it these ones or these ones?**

**And it hope that you agree with me**

**that it's much easier to understand**

**what programmer and teacher are in this case here**

**by simply looking at the variable name.**

**So we know that my first job was a programmer**

**and that my current job is teacher.**

**While down here, well, we would simply know**

**that these are two different jobs.**

**We don't know anything about them.**

**And so this approach up here is a lot better.**

**So keep that in mind,**

**whenever you write your own variable names**

**and actually keep all of this in mind,**

**that we just talked about,**

**for naming your variables.**

**Okay?**

**So just to quickly recap what a variable actually is,**

**it is basically a box into which we can store a value.**

**So we give that box a name**

**which in this case here is first name**

**or in this case here, it's my first job, for example.**

**And then into that box, with that label,**

**we can store a value**

**which in this case is this programmer string here.**

**Then later into code,**

**we can reference that variable over and over again,**

**console.log(myFirstJob);**

**for example, like this,**

**my first job.**

**And if I save this, then of course here we see programmer.**

**Once again you probably saw**

**my code changing here automatically**

**from the single quotes, two double quotes**

**but that is just my automatic formatting doing its work.**

**And**

**now if we want it to change this here, let's say coder.**

**Let myFirstJob = “Coder”**

**Then it would change across the entire program.**

**And now we get coder here.**

**And that's what variables are all about.**

**They're one of the most important things of programming.**

**And so make sure to really understand them**

**before you move on.**

**Variables**

* **In JavaScript, variables are used to store and manage data. They are created using the var, let, or const keyword.**

**var Keyword**

* **The var keyword is used to declare a variable. It has a function-scoped or globally-scoped behavior.**

**var x = 10;**

**Example: In this example, we will declare variables using var.**

**var a = "Hello Geeks"**

**var b = 10;**

**var c = 12;**

**var d = b + c;**

**console.log(a);**

**console.log(b);**

**console.log(c);**

**console.log(d);**

**Output**

**Hello Geeks**

**10**

**12**

**22**

**let Keyword**

* **The let keyword is a block-scoped variables. It’s commonly used for variables that may change their value.**

**let y = "Hello";**

**Example: In this example, we will declare variables using let.**

**let a = "Hello learners"**

**let b = "joining";**

**let c = " 12";**

**let d = b + c;**

**console.log(a);**

**console.log(b);**

**console.log(c);**

**console.log(d);**

**Output**

**Hello learners**

**joining**

**12**

**joining 12**

**const Keyword**

* **The const keyword declares variables that cannot be reassigned. It’s block-scoped as well.**

**const PI = 3.14;**

**Example: In this example, we will declare the variable using the const keyword.**

**const a = "Hello learners"**

**console.log(a);**

**const b = 400;**

**console.log(b);**

**const c = "12";**

**console.log(c);**

**// Can not change a value for a constant**

**// c = "new"**

**// console.log(c) will show error**

**Output**

**Hello learners**

**400**

**12**