













[MUSIC] In this lecture we're going to talk

about JavaScript built-in types. Now, first of all, what is a type? A type is a particular data structure. Now each language defines some built in

types, and JavaScript is no different. Built in types, can be used to build other

data structures, that you can then use for your particular business logic. Now JavaScript has seven built-in types,

six primitive, and one object type. So what is an object in JavaScript? Well in JavaScript

an object is nothing more than a collection of name/value pairs. You can have one name value pair, or

it can have zero name value pairs. But it is a collection

of name value pairs. Let's take a look at one a quick

example might look like, visually. So, here I have a person object and it has

a bunch of different name value pairs. First name, Yaakov,

last name Chaikin, social and so on. So first name would be the name and

Yaakov would be the value and so too a value doesn't necessarily have to be a

single value, it could be a nested object. So a name here, for example is social,

the name of the property of the object. And the value of that property,

is an entire other object, that has again, name value pairs. So that's object type. What about primitives? Well primitive type is a type that

represents a single, immutable value. So what do that mean? A single value means it's not an object,

remember an object is collections of name, value pairs, even simplest object there

is there is a name of value pair. Well the very simplest one

has zero name value pairs. But the very simplest one that has

some data in it is a name value pair. It's not a single value. What does immutable mean? Immutable means once it's set, once

that value is set It cannot be changed. It means that once you set it, the value

of the variable becomes read-only. You could certainly create another

value based on an a existing value but the memory space that was allocated for

the first value is not changed. Instead, you create a new memory space for

the new value. So let me run down one by one of all the

primitive types that JavaScript defines. First of all, boolean. Well boolean can only have two values,

either true or false. And true and false are reserved

keywords in JavaScript language. Now if you've ever done any programming at

all and this course assumes that you have. You should be very

familiar with this type. Enough said, really. Another type is called undefined. The undefined data type signifies

that no value has ever been set on this particular variable of this type. This is the value that every variable

gets when the JavaScript engine sets up the variable in memory when it defined but

has never been assigned any value yet. And it can only have one value. And it's a reserved keyword undefined. Now you can set a variable to

the keyword undefined yourself manually, but you should never ever do that. And the reason is, is because it's

meaning is that it's never been defined. So defining it and then setting it to undefined Is

really counter to its core meaning. If you ignore this advice and

explicitly in your code, set a variable to the value of undefined, that will

create tremendous confusion in your code. And confusion and

code is never a good thing. Another value that the JavaScript

language defines is null. Null signifies the lack of value. Now, the lack of value

as opposed to undefined which signifies the lack of definition. There's only one value that

could ever set for that type and that value is the reserved keyword, null. And it's also perfectly normal and perfectly okay to explicitly in your

code set a variable to the null value. Another type is number. Well number is actually the only

numeric type in JavaScript and it's always represented under the hood as

a double-precision 64-bit floating point. In other words,

JavaScript does not have integer type. Integers in JavaScript

are just a subset of doubles instead of a separate data type. Another data type is strings and

string in JavaScript is very similar to the string type in other languages in

that it is a sequence of characters used to represent text and you can

define strings either using single or double quotes, either one is 100% legal. The last primitive type that I'll mention,

is symbol. Now we're not really going to cover

this data type in this class. And the reason is, is that it's new to

the ES6, otherwise known as ecmoscript 6. Now ecmoscript 6,

was released in 2015, around mid 2015, and it just isn't widely supported. Were really used yet. It will probably become quite popular

in the next number of years, but it's not there yet, so

were going to skip it for this course. Okay, so time to go to the code editor and

see some examples.

[SOUND]

Okay. So here I am in Sublime code editor. And I'm located in the script.js, which is located in the js

folder of the lecture 42 folder. And again, I opened up Sublime the same

way I've been opening it up for the last couple of lectures. Let's go ahead and close the file browser. And let's take a look as

to what we have over here. So here we defined the variable X and

then print it out straight to the console. Let's go ahead and

do that by saving the file and as we can see,

the value of the variable X is undefined. Well the reason it's undefined is

because it's never been set to anything. So if the next line I say X=5 and

save it again, the value would be five. Okay, so let's go erase that for a second. And let's go and

uncomment this line of code right here. So this line of code

directly tests whether or not X is actually equal to undefined. And if it is, it will go ahead and

print out to the console X is undefined. So let's save this and

see what our console says. And sure enough, X is undefined. This kind of demonstrates to you

that this is a reserved keyword, that you can actually test against

to see if it's undefined or not. You have to be a little bit careful,

because JavaScript has this concept of type coercion which we

haven't covered yet. But in general, this is something

you could definitely test. Now let's see what happens

if we uncomment this. And what this does is

actually sets the value to 5, so now X is explicitly defined. And if we test it against

the undefined keyword, it should jump out of the if statement. And we haven't covered if statements,

but if statements are pretty generic across many languages so

I'm sure you understand this code here. So if x is equal to undefined, then we'll say x is undefined, otherwise

else we'll just say x has been defined. So what is it going to do? It first should tell us x is undefined and

then it should jump and tell us x has been defined. Let's go ahead and save that. So we get our undefined,

x is now undefined and x has been defined. Okay, so, all this is well and good. Let's go ahead and

comment this out for a minute. And, let's do this, let's comment

out the actual definition of x. If we comment out the actual definition

of x and let's try to print out x. And, if we go ahead and

save that we're going to get an error. And the error message is

a little bit confusing. It says, "Uncaught ReferenceError:

x is not defined". Well, what, is this a joke? We just said that x is undefined, so

is undefined not the same as not defined? And that is correct. Undefined in JavaScript

has a specific meaning and the meaning of it is, is that it has

been declared, or defined, so to speak. But it has not been set to any value. So in this case the variable has been

declared and a memory space has been allocated for it, but no value has

been placed in that memory space. That's the state that assigns this

variable to be a type of undefined. Now if I save that, now it will tell me

that the variable is actually undefined. So to summarize, JavaScript defines seven built-in types,

one object and six primitives. Now object type is a collection

of name value pairs. It is nothing more than

that in JavaScript. A primitive type can only contain

a single, immutable value. And remember thar immutable means that

once you set it, it cannot be changed. You could create new values

based on this values, but you cannot change

the original value anymore. Now, there's a particular interesting type

called undefined among the primitives, and undefined means variable

memory has been allocated, but not value has ever been

explicitly set yet. And it's important to distinguish between

that and a variable that has never been defined, in which case if you try to use

it, you're going to get a reference error. And it will be a real error that will

pop out in your browser console.