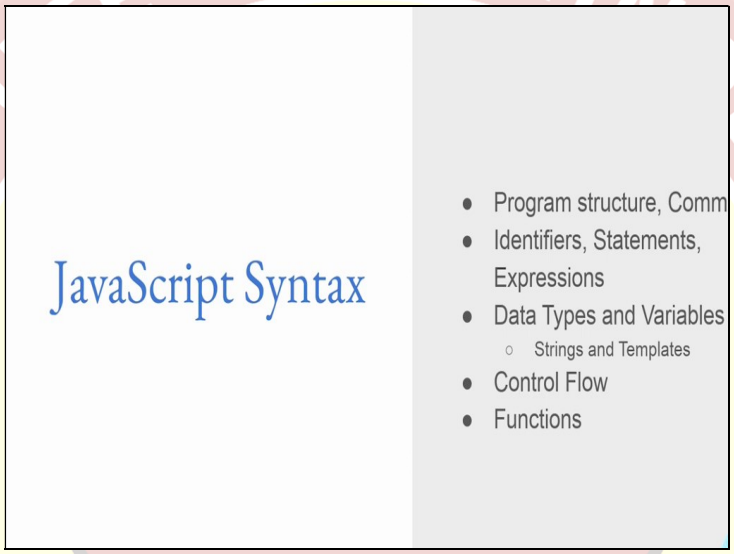
IIT Madras

ONLINE DEGREE

Modern Application Development II
Online Degree Programme
B. Sc in Programming and Data Science
Diploma Level
Prof. Nitin Chandrachoodan
Department of Electrical Engineering
Indian Institute of Technology- Madras

Basics of JavaScript

(Refer Slide Time: 00:17)

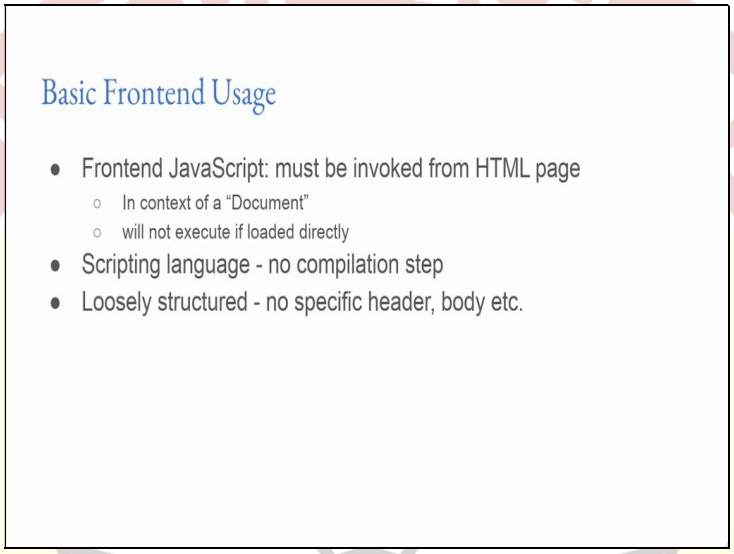


Hello everyone, welcome to modern application development part two. So, now we are going to get into the syntax of javascript and this is I am going to go fairly quickly through most of this what I will be doing is sort of giving a rough presentation of some of the concepts and then switching over to a demonstration of some parts of it in replit. But the reason why I am sort of going a little bit quickly through this is because the assumption here is that most of you have some degree of familiarity with a language like python.

So, as we move forward I will sort of try and make the connection with python as far as I can in most places. If you know a language like C or Java that is even better it would make it even easier to learn javascript. The basic syntax is actually very straightforward and easy. And what we are looking at is not sort of a complete detailed knowledge of the syntax what we are looking at is a working knowledge meaning that enough to sort of get started.

And then as you find that you know maybe how do I implement this or how do I do this particular thing you get back and say fine you know maybe I need to learn a little bit more of the syntax . So, that is the approach that we are going to be taking here and as we move forward you will basically be learning more and more and trying out different instances and ultimately you know like with any other language the only way of learning it is to work with it try programming.

(Refer Slide Time: 01:38)



So, the like I already mentioned the basic front end usage the javascript must be invoked from an html page you do not just write a program and run the program so to say. It must be invoked as a script from an html page only then it has this concept of a document that is it is associated with. And front-end javascript is closely tied to the idea of the dom the document object model which means that if it is not directly invoked from an html page there is no document and basically it cannot execute or rather cannot do anything useful.

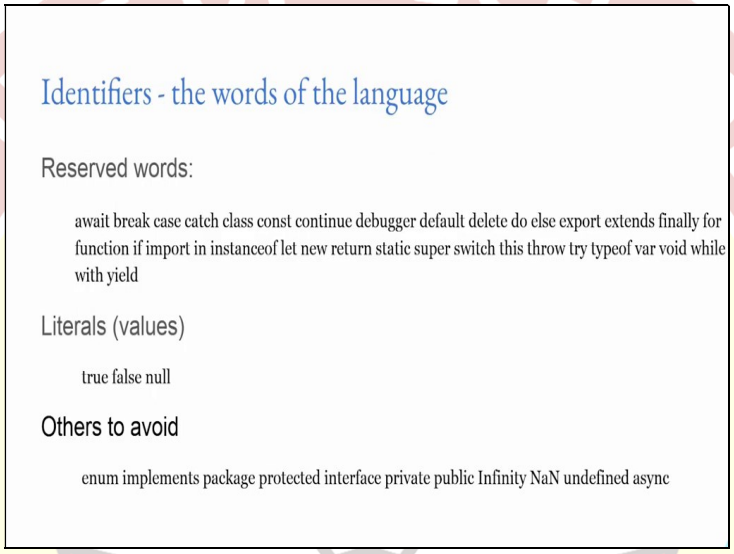
It is a scripting language there is no compilation step. So, whatever you type in and you know you create the script that is pretty much what runs just like python. It is also loosely structured similar to python. What do I mean by this for those of you who are familiar with C you would know that you know the C program needs to start with certain hash includes.

It may not have hash includes but you know if they are there they usually have to be at the top of the program. And then you have the main function the int main which is going

to execute when the program is actually executed. So, that main function has to be there certain other things you know I mean it then has to have the open braces close braces and within that a sequence of statements.

It is a little bit more structured than a python script which could just have a bunch of statements that get executed one after the other. Javascript in that sense is closer to python than to C.

(Refer Slide Time: 03:15)



Now there are certain things called identifiers they have the sort of the words used in the language and you need to be aware of these although you know you do not have to sort of learn this list by heart. There are a number of so called reserved words and you can think of obvious reserve words. So, for example this word if, is something that you could easily understand would be a reserved word.

Function is the equivalent of def in python which is used in order to define a new function. Then you have certain things which we will come to more in Javascript there is this thing called let and const which are used in order to declare variables. And a bunch of other different languages out here there is a while which is used in order to implement while loops. There are certain things like width and so on. Which you may have come across in python but in javascript have a slightly different meaning.

So, it is not that it is a direct one-on-one translation from everything that you have seen in python but the basic thing to keep in mind here is that you know there are a few

reserved words that you do not really need to think too much about to start with you need to know what they are so, that you can use them. The most important thing is if something is called a reserved word; do not use that word as the name of a variable.

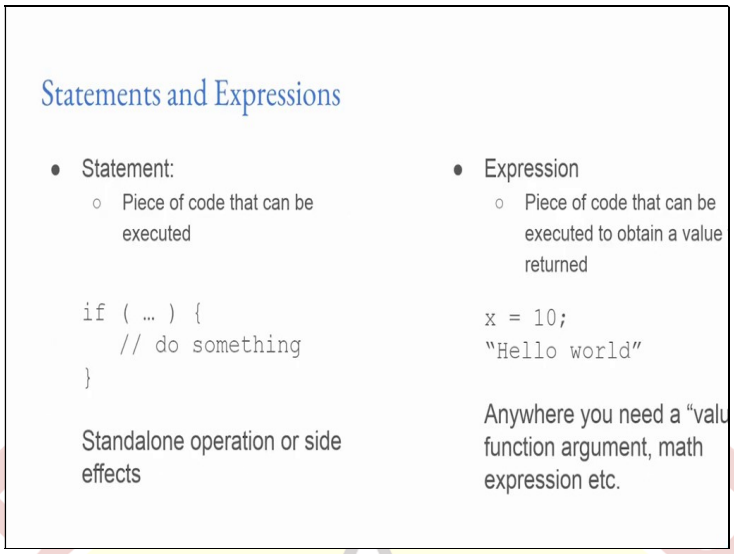
And in general that is liquid advice I mean avoid using the name of anything which you even suspect might be used as part of the language as a variable. So, do not for example try and call a variable delete or del even though del is not a reserved word in this case there could be uses of del that you do not really know what it is going to be used for. So, call it something else you can always give it a different name.

If you if in doubt then probably modify the name a little bit so that it is not likely to be a reserve word. There are certain things called literals which are for example true and false and null which we will you know you will see a lot of null in Javascript it is basically the equivalent of the none in python and it is sort of the undefined state. Now these are literals they are not really keywords in the sense that they do not start or end the statement they are values that can be used in comparisons and so on for assignments.

And there are a few other words to avoid for example you know package, private, public, undefined and async all of these are words that you will see later as we go through code. And generally yeah you know it is suggested that you avoid them why is does it say avoid because in principle it does not prevent you from. So, even redefining these words you have different meanings.

So, it is always preferable that you sort of keep that in mind and therefore avoid it rather than just sort of flat out saying no you are not allowed to do this.

(Refer Slide Time: 06:15)



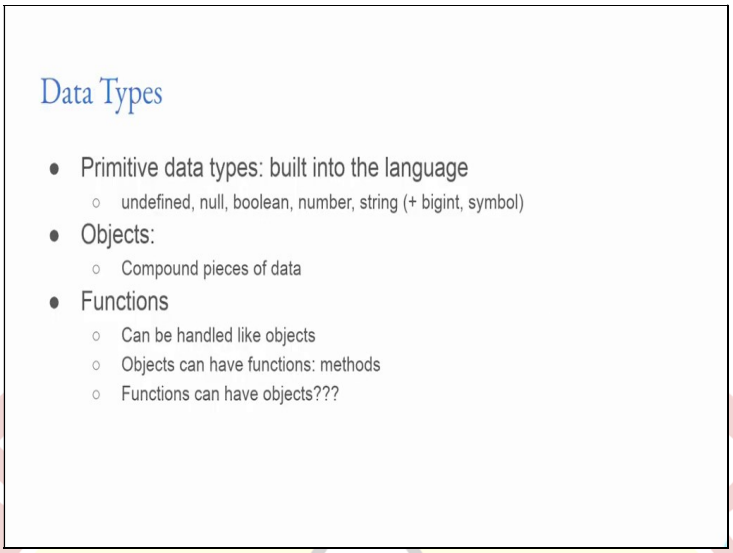
Now in terms of the structure of the language one thing to keep in mind is that its can be broadly looked at in terms of statements and expressions. A statement is a piece of code that can be executed an example is if something do something and that happens within the curly brackets. It could either be a standalone operation or it could have certain side effects. By side effect what I mean is maybe you could have a console.log which basically causes something will get printed in the console but does not directly affect any of the variables or the data present inside the program.

An expression on the other hand is something that can be executed in order to obtain a value to be returned. An example of this is just x equal to 10. So, this is very clearly you are taking the value 10 and assigning it to x. So, there is an expression there, there is a value that is being returned. Now the interesting thing is even a statement like this just hello world is actually an expression in the sense that it is a value that can be used anywhere that a value expression needs to be used.

An example of that is anything that basically goes into a function argument. Let us say that you want to add numbers and you define a function called add of x, y x and y basically have to be expressions which means that they can be evaluated to get certain values. So, the argument for a function for example cannot be an if statement because this by itself does not return a value that can be then passed into the function.

Whereas hello world just standing by itself can be thought of as something that returns itself as the return value and what it means is it can then be passed into a function.

(Refer Slide Time: 07:58)



There are a number of so-called primitive data types in Javascript and these are sort of built into the language. There is something called undefined and null which are important things that you need to keep in mind as we move forward but apart but they are not really data types they are sort of special cases they are what are called non values ok but apart from that there are Boolean true false number.

So, there is no int and float separately unlike in language like C, python of course does not specifically mention whether int or float except for that there are certain things in numpy for example right you would have seen that there is actually a distinction made between ints and float32s and float64s. Javascript again does not have that natively it just sort of calls everything a number.

It also has something else called bigint which is used for storing large integers and then of course you know you have strings that can be used for actually storing and manipulating strings. There are something called objects and this is something that we will need to spend some time on later we will sort of be looking at in a little bit more detail because it is the basis of how classes and so on are implemented in javascript.

But for now one way to look at it is just that an object is a compound piece of data it has multiple things sort of put together inside it. Now a function is interesting a function basically intuitively we know what a function is like I said add of x, y is something that

will take two arguments x and y and it will return some value. Now the interesting thing about functions in Javascript and this is also true of functions in python.

Although we do not really use it much there is that a function can be handled like an object meaning that I can take the name of a function and assign it into a variable and that that variable basically can be used as the function. Now that can be done in python as well. But in javascript you will find that it is actually used in that sense a bit more you have this concept of taking a function defining it as an anonymous object and then assigning it to a variable that happens a lot in javascript.

Now they can be handled like objects and of course the interesting thing is you can have you know objects can themselves have functions inside them. For those of you who are familiar with object oriented object oriented programming you would know that you know this is basically called the method of the object of the class the interesting thing in javascript is a function can have an object associated with it.

So, now let us take a you know slightly debate from this go over to the browser and take a look at a demo of you know some of the basics of what I have been talking about.
(Video Start: 10:50)

So, this is an example of a basic project that I have created in repliy. In fact if you just go to replit and you say create repl and you choose let us say html, css, javascript and you just give it any name that you want. By default it will create something that looks like this which basically has this set of files it has an index.html it has a script.js here I have added a lot of material the script.js by default it will be empty.

And there will be a style.css which once again in this case is just empty. Now what that means is what you have on the left hand side over here is a set of the files that you have created. You can add your own files you can put images in there you can create sub folders hopefully most of you are familiar with this and have used it at some point either for your flask applications or in some other context otherwise it is quite easy to get used to any of it.

Now the interesting thing is let us start with this index.html. So, it is a you know well defined html file it has a doctype defined at the at the beginning it has this html slash html closing tag head slash head and body slash body. Now in the head all of this the meta material and so on comes by default this is basically what is provided by replit including the link to the style sheet.

So, I have not touched this part at all what has been done is I have added certain entries out here. So, for example this div that is present over here another div that is present over here and lines 14 and 15 which are two more scripts that have been added. All of these things are parts that are not really there in a default replit that will get created I have added them and I will be sort of demonstrating them as we move forward it sort of makes it easier for me to have these things ready which is why I have kept it like this.

Now what is the bottom line? What you can see is that you know now if I go look at script.js well it has just one line over here same console.log hello console this is sort of the equivalent of the hello world program it is the first thing that we want to output from our script. You will notice right now that everything else over here has just basically been commented out I will get to those later.

There are also a couple of other scripts that are also all commented out but let us focus on script.js for now. So, script.js has just this one line console.log ("Hello console!") and this index.html by itself has well it has a couple of empty div's which are not expected to show anything and it has this one thing saying hello world. So, if I look just at the html file what I can expect is inside the body there is some text hello world and there is also some script that is going to get loaded.

And this is it this line 13 over here which basically says script source equal to script dot js slash script is responsible once we run this program for actually loading up that script executing what is inside it and generating output if needed. So, let us think about what is likely to happen the first of course is that you know what we expect is the html file should at least display this.

So, this should show up in the browser that should happen on our right out here right. On the right hand side where we have this https js hello world and so on, in this right hand

side pane we expect to see the html output. Why is it not there because I have not run the program yet. And down here on the right hand side you can see there are two things one is called the console and the other is called the shell.

The shell is an actual linux terminal you can actually go type ls see what files are there and so on. But now we are more interested in the console this is the actual javascript console corresponding to this application that is going to run . So, let us try and run it all I have to do is click on run and it takes a moment and you can see that you know this got updated I can see the hello world out here on the right hand side.

Where did that come from my html directly this has no javascript involved with it at all but the javascript effect came down here in the console. I can see that hello console got printed down there where did that come from script.js. So, this line console.log ("Hello console!") is an example of a statement it is just you know do this and it by itself does not have any return value associated with it.

(Video End: 15:41)