Variables and Datatypes

Primitive Javascript Data Types

Number- floating point numbers, for decimals and integers

String- sequence of characters, used for text

Boolean- logical data type that can only be true or fale

Undefined- data type of a variable which does not have a value yet

Null- also means ‘non existent’

Variable Mutation and Type Coercion

Type coercion- javacript automatically converts numbers to strings.

When you mix different data types it automatically converts them all to the same type

Template site🡪Slashpage

Operators

Difference between == and ===

== performs type coercion

Boolean Logic and Switch Statements

And && (Both Need To Be True)

Or || (Only one variable has to be true)

! Not – inverts true/ false

Section 3

What happens to our code

Javascript is always hosted by browser in this course.

Can be hosted by Server- Node JS

Host is javascript engine that executes our code

First step- Parser (checks for correct syntax)

Second step- Abstract syntrax tree which is converted to machine code

Machine code (set of instructions that can be executed by compuers processor)

Then the code runs

Execution Contexts

Enviorment – execution contents

Global Execution Context- for code that is not inside any function

Execution Context becomes active context for which code is executed.

1. Creation Phase
   1. Creation of the variable object
   2. Creaton of the scope chain
   3. Determine value of the ‘this’ variable
2. Execution Phase
   1. The code of the function that generated the current execution context is ran line by line

Scoping

Each new function creates a scope: the space/ enviorment in which the variales it defines are accessible

Only way to create a new scope is to write a new function

Lexical scoping- a function lexically within another function gets access to the scope of the outer (parent) function.

The global scope will not have access to local variables unless they are returned.

The ‘This’ Variable

Regular function call – this keyword points a the global object

Method call- the this variable points to the object that is calling the method.

\*The this keyword is not assigned a value until a function where it is defined is actually called.