Date: 2021-01-12

1. Section intro.
   * + - * Info. About how JavaScript work behind.
2. How our code is Executed JavaScript Parsers and Engines.
   * + - * A java Script engine is a code where java script code is execute.
         * Java script engines : Googles v8,spiral monkey, java script core etc.
         * Parser:

In engine first parser check the code based on syntax either code is in correct syntax or not. And if there are any mistake then it throw a error and stop the execution.

And if everything Is correct then it parse produce the data structure tree named Abstract syntax tree.

* + - * + Conversion of machine code:

After the parser that is not a machine code. So now code is converted in a machine code.

Now the code is machine code.

* + - * + Run code:

Now this machine code run and produce the Output.

May be Different engine have a different execution pattern.

1. Execution Contexts and the Execution Stack:
   * + - * All JavaScript code need a environment and this environment is called a **Execution Context**.

Execution context means a box or container means it stores a variables, and our piece of code which we used in our code.

1.Global context (Default) .

A code that is not in any function.

Associated with global object.

For browser it is a window object.

In short it contains a variable and function which is not in any function.

Know about a execution stack.

1. Execution Contexts in Detail Creation and Execution Phases and Hoisting:
   * + - * How execution contexts created.

There are a 2 phase

1.creation phase.

A) creation of the variable object:

The argument object is created.it contain all argument which is passed in function.

Code is scanned. For function declaration for each function property is created in variable object. And pointing to the function.

Code is scanned. For variable declaration for each variable property is created in variable object. And set undefined.

Function and a variable are hoisting in JavaScript.

See the detail in 5.

B) Creation of the scope chain

See the detail in a 6.

C) Determine value of ‘this’ object

See the detail in 7.

2.Ececution phase.

The code of the function that generate the current execution context is run line by line.

1. Hoisting phases:
   * + - * If we create a function and we call function before the function definition code then it work fine.
         * But if we use function expression then we must write a function code and then we can call it.
         * If we get value of variable before a declare it return a undefined.
2. Scoping and the Scope Chain:
   * + - * Each new function create new scope.
         * Scope means where to access the variable.
         * **Lexical scoping :** means a function that Is lexically within another function get the access the scope of the outer function.
3. The ‘this’ keyword:
   * + - * **Regular function call** this keyword points to a global object.
         * **Method call** : this object points to the object that is calling a method.
4. ‘this’ keyword in practice:
   * + - * Practical of this keyword.
5. Objects and properties:
   * + - * Syntax:

var object\_name = {

key:value,

} ;

* + - * + Object can contain any data type even we can store array in it.

EX:

var pinfo = {

name : ‘jay’,

bdate : ‘7-5-1999’,

family : [‘abc’,’xyz’,’pqr’]

}

Or

var pifo = new Object();

pinfo.name = ‘jay’;

pinfo[‘age’]=18;

* + - * + for access the value

look take a example of pinfo.

For access a name we can write like..

Console.log(pinfo.name);

Or

Console.log(pingo[‘name’]);

1. Object and Methods:
   * + - * Practical of objects and methods.
2. Loops and iteration:
   * + - * 1) for loop:

Syntax:

for (initialization ; condition ; increment / decrement)

{

//code

}

Example:

for(var i=1; i<= 10 ; i++)

{

Console.log(i);

}

* + - * + 2)while loop:

Syntax:

While(condition)

{

//code

}

Example:

var i=1;

while(i<=10)

{

console.log(i);

I++;

}

* + - * + Continue:

For skip a particular value based on condition.

* + - * + Break :

For break the loop in between.

1. JavaScript Versions ES5, ES6 ES2015 and ES6+ :
   * + - * History of java script.
         * Version release.

HTML

* information about html-5
* styling element
  + internal css =medium priority
  + inline css = highest priority
  + external css = low priority
* calling javascript
  + internal
  + external
* Html noscript element
  + Use when user disable javascript and them we want to display a message..