**JavaScript (12-01-2021)**

1)JavaScript Parsers and Engine

* JavaScript engine is a computer program that executes JavaScript code.
* JavaScript engine is parsing our code by parser.
* A Parser basically reads our code line by line and checks it for errors and stops execution if any, if the code is free of error syntax the parser produces a data structure called abstract syntax tree , which is then translated to machine code and thus result to output.

2)Execution Contexts and Execution Stack

* The environment in which our code run is called execution context and is defined as one of the following

1) Global :- Default environment where code is executed first time.

2) Local/Function :- When the code within the function gets executed.

* Global context is used to execute the code that is not part of any function. Whenever a function is called a new execution context is created and is pushed on top of the current context thus creating what is known as Execution Stack.

3)Execution Context Phases

* The execution context is divided into two phases named Creation and Execution Phase.
* Creation phase :- This phase is defined when the function is called before any code is run within .
* In the creation phase first comes the creation of variable object then scope chain and lastly the this is determined and set which together form the properties of execution context object.
* The variable object is created by following these steps in the process.

1)Adding the argument object containing all the objects that were passed in function call.

2)For each function a property is created in Variable Object, pointing to the function (all function will be stored in the object before the code starts execution).

3)Variable declarations are looked up and for each variable a property is set in Variable Object and set to undefined.

* The last two points are generally known as Hosting. Functions and variables are hoisted which means they are available before the execution phase starts but there is a small catch both are hoisted differently , variables are defined only in execution whereas the functions are already defined.

4)Scoping

* Scoping answer the question “where can we access a ceryain variable?”
* Each new function creates a scope which means the space/environment , in which the variables it defines are accessible.
* Lexical scoping means a function that is lexically within another function get access to the scope of the outer function.
* Execution stack is the order in which functions are called.
* Scope chain is the order in which functions are written lexically.

5)’this’ keyword

* 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.

6)Loops and Iteration

* If we want to execute a piece of code multiple time until the condition not return false then at that time we can use loops.

7)Short history of JavaScript

* 1996 :- Changed from LiveScript to JavaScript to attract Java developers . JavaScript has almost nothing to do with java.
* 1997 :- ES1 (ECMA Script 1) became the first version of the JavaScript language standard.

1)ECMAScript :- The language standard

2)JavaScript :- The language in practice

* 2009 :- ES5(ECMAScript 5) was released with lots of new features.
* 2015 :- ES6/Es2015(ECMAScript 2015) was released the biggest update to the language ever.
* 2015 :- Changed to an annual release cycle.