Debugging:

* Debugging is used to find and solve the errors. One Error also leads failures of that product.
* Searching for errors in programming is called debugging.
* This is useful to identify the errors which are in the program. After finish that coding part the developer use debugging tool to check the errors.
* Debugging is most important one for identify and solve the errors.
* Debugging is reduced the number of failures and development cost.

ES5, ES6:

* Es5 stands for ECMA Script. ECMA means European Computer Manufacturer's Association.
* ECMA Script is a Standard for scripting languages. This ES have many versions, now we using ES5 and ES6.
* The ES5 is focus on security and library updates. ES5 variable declaration is like, var age=18;
* The ES6 is a updated version of ES5. This has so many updates like arrow functions, binary data and python style generator etc.
* Example for ES5 and ES6 are:

Var age=18; /\*ES5\*/

Let age=18; /\*ES6\*/

Typeless:

* Js is typeless script language. Typeless means it accept all type of data’s.

For example, if you use one variable for accessing string and after some time the same variable will access integer. That is,

**input**

<script>

var a=100;

document.write(a);

a="rupees";

document.write(a);

</script>

**Output** will be like,

100rupees

* One variable is act like both integer and string also. So javascript is called typeless language.

Camelcase, snakeCase:

* Camelcase means first letter must in lower case and remaining letters in upper case.
* Snakecase means one letter and hyphen symbol and after that symbol we have another letter.
* Ex:

Cammelcase: camelCase

Snakecase: total\_result

Typecast:

Javascript is converted automatically from string to integer.

Ex:

<script>

var a=100;

var b="10";

var c=a/b;

document.write(ans=c);

</script>

Output is: 10

* Here one variable is integer and another one is string. but it converts that string into int and perform division operation.

var headCountPerHouse = 300 / "30"

300 + "30" "30030" - Why it does not cast to both integers?

+ use for both string concatenation and addition. if you add one integer with one string then you will get output like string. Why because that + operator is overloaded.

Arrays:

* Array is an object that represents a collection of similar type of elements. the array is have fixed length. The array declaration like:

var a[5];

* Here the array size is 5 and the array index is start from 0 to 4.
* The array size is fixed in the starting stage. So storage size will be waste if that program required less size.
* Insertion and deletion is difficult compare to the link list.

Functions:

* Functions are used to reduce the coding size. Functions have return type and parameters. If function have return type then that function return values.
* Ex: int call() This function has return type.
* If you create one function and use that in multiple times through calling function.

For loop:

* The for loop is used to done some calculation continuously. We run the same code multiple times by using this for loop method.
* For loop have initial value, condition and incremental value.
* Loop reduce the code size and reduce the code complexity.

Json:

* JSON stands for JavaScript Object Notation. Json is a lightweight data-interchange format. JSON is a syntax for storing and exchanging data.
* Json is easy to read and write by humans. It is based on javascript programming language.
* The json is used to exchange data from client to server side.

How to define:

Json object creation will be like, var obj={};

How to access any key in that JSON:

Json have two methods to access that key, which are

1. Dot(.)method
2. Box[] method

dot & box notations:

dot notation:

Ex:

<script>

var clg={"name":"smit","addr":"chennai","pincode":602026};

document.write(clg.name);

document.write(clg.addr);

</script>

Output :

Smit

Chennai

Box notation:

<script>

var clg={"name":"smit","addr":"chennai","pincode":602026};

document.write(clg[‘name’]);

document.write(clg[‘addr’]);

</script>

Output:

Smit

Chennai

How to add a new key:

<script>

var clg={"name":"smit","addr":"chennai","pincode":602026};

document.write(clg.name);

document.write(clg.addr);

clg.another="code";

clg[3]=2126;

document.write(clg.code);

</script>

Output:

Smitchennai2126

\_proto\_ uses:

\_\_proto\_\_ is the actual object. The proto is used to create a new user function. If you create one object then check that one have to many prototypes.