**LUFSC JAVASCRIPT ASSIGNMENT DAY-1**

**DATE:-14/07/2020**

**QUESTION - 1**

Explore and explain the various methods in console function Explain them:-

ANSWER:-

**Console in JavaScript**

**Web Console:-**

A Web console is a tool which is mainly used to log information associated with a web page like:*network requests, Javascript, security errors, warnings, CSS etc.* It enables us to interact with a web page by executing Javascript expression in the contents of the page.

**Console Object:-**

In Javascript, the console is an object which provides access to the browser debugging console. We can open a console in web browser by using: *Ctrl + Shift + K* for windows and *Command + Option + K* for Mac. The console object provides us with several different methods, like

* log()
* error()
* warn()
* clear()
* time() and timeEnd()
* table()
* count()
* group() and groupEnd()
* custom console logs

METHODS IN CONSOLE FUNCTION:-

* **Console.log()**

Mainly used to log(print) the output to the console. We can put any type inside the log(), be it a string, array, object, boolean etc

**EXAMPLE:-**

console.log('abc');

console.log(1);

console.log(true);

console.log(null);

console.log(undefined);

console.log([1, 2, 3, 4]); // array inside log

console.log({a:1, b:2, c:3}); // object inside log

* **Console.error()**

Used to log error message to the console. Useful in testing of code. By default the error message will be highlighted with red color.

**EXAMPLE:-**

// console.error() method

console.error('This is a simple error');

* **Console.warn()**

Used to log warning message to the console. By default the warning message will be highlighted with yellow color.

**EXAMPLE:-**

// console.warn() method

console.warn('This is a warning.');

* **Console.clear()**

Used to clear the console. The console will be cleared, in case of Chrome a simple overlayed text will be printed like : ‘Console was cleared’ while in firefox no message is returned.

**EXAMPLE:-**

// console.clear() method

console.clear();

* **Console.time() and Console.timeEnd()**

Whenever we want to know the amount of time spend by a block or a function, we can make use of the time() and timeEnd() methods provided by the javascript console object. They take a label which must be same, and the code inside can be anything( function, object, simple console).

**EXAMPLE:-**

// console.time() and console.timeEnd() method

console.time('abc');

let fun = function(){

console.log('fun is running');

}

let fun2 = function(){

console.log('fun2 is running..');

}

fun(); // calling fun();

fun2(); // calling fun2();

console.timeEnd('abc');

* **Console.table()**

This method allows us to generate a table inside a console. The input must be an array or an object which will be shown as a table.

**EXAMPLE:-**

// console.table() method

console.table({'a':1, 'b':2});

* **Console.count()**

This method is used to count the number that the function hit by this counting method.

**EXAMPLE:-**

// console.count() method

for(let i=0;i<5;i++){

console.count(i); }

* **Console.group() and Console.groupEnd()**

group() and groupEnd() methods of the console object allows us to group contents in a separate block, which will be indented. Just like the time() and the timeEnd() they also accepts label, again of same value.

**EXAMPLE:-**

// console.group() and console.groupEnd() method

console.group('simple');

console.warn('warning!');

console.error('error here');

console.log('vivi vini vici');

console.groupEnd('simple');

console.log('new section');

* **Custom Console Logs**

User can add Styling to the console logs in order to make logs Custom. The Syntax for it is to add the css styling as a parameter to the logs which will replace %c in the logs as shown in the example below

**EXAMPLE:-**

// Custom Console log example

const spacing = '10px';

const styles =

`padding: ${spacing}; background-color: white; color: green; font-style:

italic; border: 1px solid black; font-size: 2em;`;

console.log('%cGeeks for Geeks', styles);