Module 3: JavaScript – Cheat Sheet

- **By Harsha Vardhan (UI Expert)**

2.0. Introduction to JavaScript	
What is JavaScript?	• Statement-based programming language. • Creates functionality in the web page. • Makes the web page interactive to the user.
Features:	• Case sensitive language. • It is Object Oriented Programming (OOP) Language; • It is mainly used to manipulate elements that are created by using HTML.
History:	• Developed by Brendan Eich in 1995 and became an ECMA (European Computer Manufacturers Association) standard in 1997. • ECMAScript (ES) is the official name of "JavaScript". • JavaScript 1 / ECMAScript 1: 1997 • JavaScript 2 / ECMAScript 2: 1998 • JavaScript 3 / ECMAScript 3: 1999 • JavaScript 4 / ECMAScript 4: Not released • JavaScript 5 / ECMAScript 5: 2009 • ECMAScript 6 / ECMAScript 2015: 2015 • ECMAScript 7 / ECMAScript 2016: 2016 • ECMAScript 8 / ECMAScript 2017: 2017 • ECMAScript 9 / ECMAScript 2018: 2018 • ECMAScript 10 / ECMAScript 2019: 2019

Concepts:

2.1. Variable	
Syntax:	var variableName = value;
Interpretation:	• Creates a variable to store a value, array, function or object.
Example:	var x = 100;

2.2. Function	

Syntax:	function functionName( )
	{
	}

Interpretation:	• Contains a set of statements to perform some operation. • Can receive parameters and can return one value.

Example:	function sayHello()
	{
	console.log("Hello");
	}

2.3. Arrays	
Syntax:	[value1, value2, …]
Interpretation:	• Array is a collection of multiple values of same / different data type. • You can add / remove / sort or perform any manipulation on arrays dynamically.
Example:	[10, 20, 'abc', true]

2.4. Object [or] Object Literal	

Syntax:	{
	property: value, property: value,
	method: function() { }
	}

Interpretation:	• Stores a set of properties and methods. • Methods can manipulate properties.

Example:	{
	studentRollNo: 123,
	studentName: "Scott",
	getStudentName()
	{
	return this.studentName;
	}
	}

Conversion Functions:

2.5. JSON.stringify( )	
Syntax:	JSON.stringify({ property: value })
Interpretation:	• Converts object into JSON. • JSON data can be portable to other program and also can be stored in browser memory.
Example:	JSON.stringify({ studentRollNo: 123, studentName: "Scott"})

2.6. JSON.parse( )	
Syntax:	JSON.parse('{ "property": "value" }')
Interpretation:	• Converts JSON data into object. • Used to convert JSON data that is received from other programs or from browser's memory.
Example:	JSON.stringify('{ "studentRollNo": 123, "studentName": "Scott"}')

2.7. Object.keys( )	
Syntax:	Object.keys({ property: value })
Interpretation:	• Return list of property names as an array.
Example:	Object.keys({ studentRollNo: 123, studentName: "Scott"})

2.8. typeof	
Syntax:	typeof *value*
Interpretation:	• Returns data type of given value.
Example:	typeof 100

2.9. Number	
Syntax:	Number(*string value*)
Interpretation:	• Converts the given string value into 'number" data type.
Example:	Number('100')

210. String	
Syntax:	String(*number value*)
Interpretation:	• Converts the given number value into 'string" data type.
Example:	String(100)