Assignment For Day 2 😊 🥇 🌱

1. What is lexical structure?

A programming language's lexical structure specifies **a set of some basic rules** about how code should be written in it. Rules like what variable names look like, the delimiter characters for comments, and how one program statement is separated from the next. It is the lowest-level syntax of a language.

For e.g.

```
// Valid but not recommended way of declaring an identifier

let $nameofperson = "Meghana";

console.log($nameofperson);



// Recommended way of declaring an identifier



let nameOfPerson = "Poorva";

console.log(nameOfPerson);



//Javascript is case Sensetive language



const pi = 3.14;

const PI = 3.1415;
```

```
console.log(pi);

console.log(PI);



// Javascript internally convert code into Unicode UTF-16, it supports
unicode character



    var Omega = "\u03A9";

    console.log(Omega);



// Comments



// Single line comment is given by "//"



/* Multi line Comment is given

 as this*/
```

Output :

Meghana

Poorva

3.14

3.1415

Ω

2. What is Unicode?

While a JavaScript source file can have any kind of encoding, JavaScript will then convert it internally to UTF-16 before executing it.JavaScript strings are all UTF-16 sequences, as the ECMAScript standard says:

A unicode sequence can be added inside any string using the format \uXXXX:

A emojis can be used as an identifier in javascript.

Consider the following example :

```
const s1 = "\u00E9";

console.log(s1);

const s2 = "\u0065\u0301";

console.log(s2);

/*  Eventhough by just looking both the symbols appears to be same but

 there unicode characters are different moreever their length are also not
same

console.log(s1.length == s2.length); */

console.log(s1.length);

console.log(s2.length);

console.log(s1 === s2);

console.log(s1 == s2);
```

Output:

é

é

false

1

2

false

3. Explain all the keywords present in the JavaScript with examples.

“Keywords are those words in javascript whose inter meaning or functionality is already known to the javascript engine. These words can not be used as an identifier. These words also known as reserved words “

Following table shows most common keywords:

abstract	arguments	boolean	break
byte	case	catch	char
const	continue	debugger	default
delete	do	double	else
eval	false	final	finally
float	for	function	goto
if	implements	in	instanceof
int	interface	let	long
native	new	null	package

private	protected	public	return
short	static	switch	synchronized
this	throw	throws	transient
true	try	typeof	var
void	volatile	while	with
yield			

Some other reserved keywords are :

class ,enum ,export,, extends ,import ,super

Consider some examples in program :

```
        // true or false keyword



        let nameOne = "Rahul";

        var nameTwo = "Ritesh";



        console.log(nameOne === nameTwo); //false

        console.log(nameOne == nameTwo); //false



        var nameTwo = "Rahul";

        console.log(nameOne === nameTwo); //true
```

```
console.log(nameOne == nameTwo); //true


// typeof keyword


console.log(typeof " Gulam"); //string
console.log(typeof 1); //number


// for keyword


for (let i = 0; i < 5; i++) {
    console.log(i);
}


/* output :
0
1
2
3
4 */


//function and return keyword


function squareOfNum(num) {
```

```
        let ans = num * num;

        return ans;

    }

    result = squareOfNum(5); //function invoked

    console.log(result); // 25



    // if else



    if (5 === "5") {

        console.log("number and strings are equal");

    } else {

        console.log("numbers and strings are not equal");

    }



    //output: numbers and strings are not equal
```

4. What are shorthand operators, explain with a suitable example?

JavaScript shorthands not only speed up the coding process but also make scripts shorter, therefore lead to faster page loads. Shorthand codes are just as valid as their longhand versions; they essentially stand for the same thing–only in a more compact format. They are one of the simplest code optimization techniques.

Consider following examples :

```
let number1 = 1000000000000;

let number2 = 10e11;



console.log(number1);

console.log(number2);



/* Output :

1000000000000

1000000000000 */



/* Shorthand */

i++;

j--;



/* Longhand */



i = i + 1;

j = j - 1;
```

```
/* Shorthand */

var i,j = 5,k = "Good morning", l,   m = false;

/* Longhand */

var i;

var j = 5;

var k = "Good morning";

var l;

var m = false;



/* Shorthand */

var message = age >= 18 ? "Allowed" : "Denied";



/* Longhand */

if (age >= 18) {

    var message = "Allowed";

} else {

    var message = "Denied";
```

5. What is “use Strict” in JavaScript?

   “use strict” defines that javascript code must be executed in “ strict mode”

   With strict mode, we can not, for example, use undeclared variables.

   Without use strict mode

```
    myName = "Gulam";

    console.log(myName);



    // Output : Gulam
```

   With use strict mode :

```
    "use strict";

    myName = "Gulam";

    console.log(myName);



    /* Output : myName is not defined

    i.e. it causes error because it is mandatory to define a
variable before assigned it with any value */
```

Thank You Mohd Gulam Ansari