1. **What is lexical structure?**

Lexical structure specifies a set of some basic rules about how code should be written in it.

1. **What is Unicode?**

In Javascript, the identifiers and string literals can be expressed in Unicode via a Unicode escape sequence. The general syntax is \uXXXX, where X denotes four hexadecimal digits. Ex: the letter o is denoted as ‘\u006F’ in Unicode.

1. **Explain all the keywords present in the JavaScript with examples.**

**Keywords** are reserved words that are part of the syntax in the programming language.

Keywords list:

**await break case catch class**

**const continue debugger default delete**

**do else enum export extends**

**false finally for function if**

**implements import in instanceof interface**

**let new null package private**

**protected public return super switch**

**static this throw try true**

**typeof var void while with**

**yield**

**a)goto:** Used to return execution control to a specific location

Ex: var no=0;

Sposition

document.write(" something print here ");

no++;

if(no < 10) goto sposition;

**b)public:** It is an access modifier that can be used with attributes, classes, constructors and methods which make it accessible to other classes.

Ex:

public class Employee {

public String efn = "Joseph";

public String eln = "Doe";

}

class MainClass {

public static void main(String[] args) {

Employee obj = new Employee ();

System.out.println("Name= " + obj.efn + " " + obj.lname);

}

}

**c)do:**  Used to define do-while loop.

Ex: var a=1;

do {

document.write("loop is running for " + a + "times</p>");

a++;

}

while(a <= 10);

**d)Function keyword**

Used to define a function to execute a block of code.

Example:

var func = function(){

return "Hello";

}

alert(func());

**e)For** **keyword**

Used to define a loop, for loop to repeatedly execute a block of code until a condition true.

Example:

**f)if keyword**

Used to define a conditioned construct. if the statement is used to run a block of code if the condition is true.

Example:

var date = new Date();

var day = date.getDay(); // Sunday Saturday : 0 6

if(day==5) {

alert("This is weekend!");

} else {

alert("This is non-weekend!");

**g)Break keyword**

used into a loop to break or stop the execution of the loop.

Example:

for(var a=0; a<=10; a++) {

if(a == 5)

break;

document.write("The loop is running for " + a + " times");

}

for(var a=0; a<=10; a++) {

document.write("The loop is running for " + a + " times");

}

**h)while keyword**

Used for while loop, while loop executes the block of code until the condition is true.

Example

var a=1;

while(a <= 10)

{

document.write("loop is running for " + a + "times</p>");

a++;

}

**i)Var keyword**

Used to declare a variable

Example

var fruits = ["apple", "banana", "orange"];

var age=22;

**j)True keyword**

Used to store or represent primitive data type Boolean ‘true’.

Example

var inp = true;

**k) False keyword**

Used to store or represent primitive data type Boolean ‘false’.

Example

var inp = false;

**l)Void keyword**

used to evaluates an expression and returns undefined. A void operator is frequently used to get the undefined primitive value.

Example

<a href = "javascript:void(0);">

click link

</a>

**m) switch keyword**

Used in a switch-case construct, where switch evaluates an expression.

Example

var date = new Date();

switch(date.getDay()) {

case 6:

alert("This is weekend.");

break;

case 0:

alert("This is weekend.");

default:

alert("Looking for a weekend.");

break;

}

**n)This keyword**

Used to refer to the current object.

Example

class Employee extends Person {

constructor(name, eid, salary) {

super(name);

}

get incsalary() {

return this.salary \* 0.2;

}

}

**o)let keyword:**Used to declare a variable limited to a scope of a block of code, unlike a variable declared by the var keyword.

Example

let var fruits = ["apple", "banana", "orange"];

**p)else keyword**

Used in the if-else statement, the else indicates the block of statements to be executed if the expression evaluates false.

Example

var date = new Date();

var day = date.getDay(); // Sunday - Saturday : 0 - 6

if(day==5) {

alert("This is weekend!");

} else {

alert("This is non-weekend!");

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

**Shorthand:**A shorthand operator is a shorter way to express something that is already available in the JS.

Shorthand operators +=, -=, \*=, /= and \*=

A frequent construct is the following:

x is a variable in the program

x = x + value ; // Add value to the variable x

x = x - value ; // Subtract value to the variable x

x = x \* value ; // Increase the variable x by value times

and so on…

Operator assignment shorthands:

Operator symbol Name of the operator Example Equivalent construct

+= Addition assignment x += 4; x = x + 4;

-= Subtraction assignment x -= 4; x = x - 4;

\*= Multiplication assignment x \*= 4; x = x \* 4;

/= Division assignment x /= 4; x = x / 4;

%= Remainder assignment x %= 4; x = x % 4;

1. **What is “use Strict” in JavaScript?**

The purpose of "use strict" is to indicate that the code should be executed in "strict mode".This makes it easier to write good and secure JS code.