Assignment For Day 2

Subham Dutta
(subhamdutta588@gmail.com)

1. What is lexical structure?

Ans. Lexical structure of a programming language 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.

2. What is Unicode?

Ans. Unicode Standard, is an information technology standard for the consistent encoding, representation, and handling of text expressed in most of the world's writing systems. Javascript is written in unicode, so we can use emoji as variables and even we can write identifiers in any language in javascript.

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

Ans. The keywords present in Javascript are:

1. await: The await operator is used to wait for a promise returned by an async function.
2. do...while: Used to create loop.
3. const: Used to declare constant.
4. break: The break statement terminates the current loop, switch, or label statement and transfers program control to the statement following the terminated statement.
5. catch : It is used to handel error in javascript.
6. class : Classes are a template for creating objects. They encapsulate data with code to work on that data.
7. false: False is a value of boolean function.
8. continue: The continue statement breaks one iteration (in the loop) if a specified condition occurs, and continues with the next iteration in the loop.
9. finally : The Finally is a block of code or statements that will be executed in any case while handling errors using try and catch block. A JavaScript provides try-catch blocks to execute the code which is prone to error and may cause improper behaviour of the program. The finally block is placed after try and catch block and will be executed definitely in case any one of the blocks from them is executed i.e. try or catch. The finally block allows us to define the actions which are compulsory to be performed even after the success or failure of some code.

10. import :The static import statement is used to import read-only live bindings which are exported by another module.
11. new: The new operator creates an instance of a user-defined object type or of one of the built-in object types that has a constructor function.
12. this: In most cases, the value of this is determined by how a function is called (runtime binding). It can't be set by assignment during execution, and it may be different each time the function is called. ES5 introduced the bind() method to set the value of a function's this regardless of how it's called, and ES2015 introduced arrow functions which don't provide their own this binding (it retains the this value of the enclosing lexical context).
13. var: This is used to declare a variable.
14. debugger: The debugger statement stops the execution of program and calls the debugger.
15. default: The default keyword can be used in two situations in JavaScript: within a switch statement, or with an export statement.
16. for:The for statement creates a loop that consists of three optional expressions, enclosed in parentheses and separated by semicolons, followed by a statement (usually a block statement) to be executed in the loop.
17. enum: The enum in JavaScript is used to define set of same type constants in single entity.
18. in: The in operator returns true if the specified property is in the specified object or its prototype chain.
19. null: The value null represents the intentional absence of any object value.
20. return: The return statement ends function execution and specifies a value to be returned to the function caller.
21. throw: The throw statement throws a user-defined exception. Execution of the current function will stop (the statements after throw won't be executed), and control will be passed to the first catch block in the call stack. If no catch block exists among caller functions, the program will terminate.
22. void: The void operator evaluates the given expression and then returns undefined.
23. export: The export statement is used to export functions, objects or primitives from a given file (or module).
24. extend: The extends keyword is used in a class declarations or class expressions to create a class with a child of another class.
25. function: Used to declare a function.
26. instance of: The instanceof operator tests whether an object has in its prototype chain the prototype property of a constructor.
27. try: The try statement defines the code block to run (to try).
28. if....else : The if statement specifies a block of code to be executed if a condition is true and else if the condition is false.
29. interface: An interface describes the shape of an object. (what properties it has, what type of values those properties contain, etc.)
30. privet/public: Class members are public by default, but we can make it private by usin #private keyword.

31. switch: The switch statement is a part of JavaScript's "Conditional" Statements, which are used to perform different actions based on different conditions. Use switch to select one of many blocks of code to be executed. This is the perfect solution for long, nested if/else statements.
32. true: True is a value of boolean function.
33. with: The with statement extends the scope chain for a statement.
34. yield: The yield keyword is used to pause and resume a generator function (function* or legacy generator function).

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

Ans. Shorthand is a process of expressing long expressions in a short form. Example:

```
i + 3= i;
i +=3 ;
```

Both of the above expression results same, but in the second case we are using shorthand operator (+=) which is shortening the expression.

5. What is “use Strict” in JavaScript?

Ans. The “use strict” directive was introduced in ECMAScript version 5. Using “use strict” makes the code strict, that means the code will follow the strict javascript rules.