## ANSWER 4-

ECMAScript 6, also known as ES6 or ES2015, introduced several major features that enhanced JavaScript. Here are some of the key features in simple words:

Block-Scoped Variables (let and const): ES6 introduced the let and const keywords for declaring variables. They have block-level scope, meaning they are limited to the block of code they are defined in (e.g., within curly braces {}). let allows reassignment of values, while const creates read-only variables that cannot be reassigned.

Arrow Functions: Arrow functions provide a shorter syntax for writing functions. They use an arrow (=>) instead of the function keyword, making the code more concise. Arrow functions also have lexical scoping for this keyword, which helps avoid issues with traditional function scoping.

Classes: ES6 introduced a class syntax that allows defining classes in JavaScript. Classes provide a way to create objects with predefined properties and methods, following the object-oriented programming (OOP) paradigm. The class syntax simplifies the creation and inheritance of objects.

Template Literals: Template literals provide an improved way to work with strings. They allow embedding expressions and variables directly within backtick ( ) delimited strings, using placeholders (${}) to interpolate values. This simplifies string concatenation and improves code readability.

Destructuring Assignment: Destructuring assignment provides a convenient way to extract values from arrays or objects and assign them to variables. It allows you to unpack values from complex data structures quickly and easily.

Spread Operator: The spread operator (...) allows you to expand elements from arrays, objects, or iterables. It simplifies the process of combining arrays, cloning objects, and passing multiple arguments to functions.

Modules: ES6 introduced native support for modules in JavaScript. Modules are reusable pieces of code that can be imported and exported between files, promoting better code organization, encapsulation, and reusability.

Promises: Promises provide a better way to handle asynchronous operations in JavaScript. They represent the eventual completion (or failure) of an asynchronous operation and allow chaining of actions through .then() and .catch() methods, improving the readability and maintainability of asynchronous code.

These are just a few of the major features introduced in ECMAScript 6. ES6 brought many other enhancements and new syntax elements to JavaScript, making the language more powerful and expressive for developers.