<https://www.javatpoint.com/es5-vs-es6>

ECMAScript is a trademarked scripting language specification that is defined by ECMA International. It was created to standardize JavaScript.

ES5 is an abbreviation of **ECMAScript 5** and also known as **ECMAScript** 2009. The sixth edition of the **ECMAScript** standard is **ES6** or **ECMAScript 6**. It is also known as ECMAScript 2015

|  |  |  |
| --- | --- | --- |
| **Based on** | **ES5** | **ES6** |
| **Definition** | ES5 is the fifth edition of the ECMAScript (a trademarked scripting language specification defined by ECMA International) | ES6 is the sixth edition of the ECMAScript (a trademarked scripting language specification defined by ECMA International). |
| **Release** | It was introduced in 2009. | It was introduced in 2015. |
| **Data-types** | ES5 supports primitive data types that are **string, number, boolean, null,** and **undefined**. | In ES6, there are some additions to JavaScript data types. It introduced a new primitive data type **'symbol'** for supporting unique values. |
| **Defining Variables** | In ES5, we could only define the variables by using the **var** keyword. | In ES6, there are two new ways to define variables that are **let** and **const**. |
| **Performance** | As ES5 is prior to ES6, there is a non-presence of some features, so it has a lower performance than ES6. | Because of new features and the shorthand storage implementation ES6 has a higher performance than ES5. |
| **Support** | A wide range of communities supports it. | It also has a lot of community support, but it is lesser than ES5. |
| **Object Manipulation** | ES5 is time-consuming than ES6. | Due to destructuring and speed operators, object manipulation can be processed more smoothly in ES6. |
| **Arrow Functions** | In ES5, both **function** and **return** keywords are used to define a function. | An arrow function is a new feature introduced in ES6 by which we don't require the **function** keyword to define the function. |
| **Loops** | In ES5, there is a use of **for** loop to iterate over elements. | ES6 introduced the concept of **for...of** loop to perform an iteration over the values of the iterable objects. |