**JAVASCRIPT**

**1. Javascript là gì?**

Javascript là một ngôn ngữ lập trình kịch bản dựa vào đối tượng phát triển có sẵn hoặc tự định nghĩa ra, javascript được sử dụng rộng rãi trong các ứng dụng Website. Javascript được hỗ trợ hầu như trên tất cả các trình duyệt như Firefox, Chrome, ... thậm chí các trình duyệt trên thiết bị di động cũng có hỗ trợ.

Nếu bạn đã biết qua về HTML5 thì bạn thấy có các khái niệm như sessionStore hay localStore, đây là hai đối tượng được tạo nên từ Javascript nên rõ ràng trong HTML5 cũng có sử dụng nó. Với những ứng dụng đó thì bạn thấy javascript không thể chết trong các ứng dụng website được.

Hay thậm chí có những ứng dụng Webgame người ta sử dụng javascript để xử lý các thao tác trên Client, nếu không có nó thì thông thường chọn Flash để xây dựng nhưng lại gặp vấn đề load chậm nên có một số người chọn Javascript để làm.

**2. Các thư viện Javascript đình đám hiện nay**

Hiện nay có rất nhiều libraries và framework được viết từ Javascript như:

* **AngularJS**: Một thư viện dùng để xây dựng ứng dụng Single Page
* **NodeJS**: Một thư viện được phát triển phía Server dùng để xây dựng ứng dụng realtime
* **Sencha Touch**: Một Framework dùng để xây dựng ứng dụng Mobile
* **ExtJS**: Một Framework dùng xây dựng ứng dụng quản lý (Web Applications)
* **jQuery**: Một thư viện rất mạnh về hiểu ứng
* **ReactJS**: Một thư viện viết ứng dụng mobie
* Và còn nhiều thư viện khác

**3. Viết chương trình javascript “Hello World”**

<html>

<head>

<title></title>

<script language="javascript">

alert("Hello World!");

console.log("Hello World!");

</script>

</head>

<body>

</body>

</html>

|  |
| --- |
| <html>  <head>  <title></title>  <script language="javascript">  alert("Hello World!");  console.log("Hello World!");  </script>  </head>  <body>    </body>  </html> |

Cách 2

<script language="javascript" src="helloworld.js"></script>

**Tham khảo thêm tại** [**https://github.com/getify/You-Dont-Know-JS**](https://github.com/getify/You-Dont-Know-JS)

**ECMAScript ES6 là gì?**



**1. ECMAScript ES6 là gì?**

ES6 là chữ viết tắt của **ECMAScript 6,** đây được coi là một tập hợp các kỹ thuật nâng cao của Javascript và là phiên bản mới nhất của chuẩn **ECMAScript**. ECMAScript do hiệp hội các nhà sản xuất máy tính Châu Âu đề xuất làm tiêu chuẩn của ngôn ngữ Javascript. Bạn cứ nghĩ xem hiện nay có khá nhiều trình duyệt Browser ra đời và nếu mỗi Browser lại có cách chạy Javascript khác nhau thì các trang web không thể hoạt động trên tất cả các trình duyệt đó được, vì vậy cần có một chuẩn chung để bắt buộc các browser phải phát triển dựa theo chuẩn đó.

ES6 ra đời vào năm 2015 nên cái tên **ES2015** được lấy làm tên chính thức với nhiều tính năng mới, học hỏi các ngôn ngữ cấp cao khác. Hy vọng dần theo thời gian Javascript trở thành một ngôn ngữ lập trình hướng đối tượng.

Phiên bản sắp ra trong năm 2017 đó là phiên bản ES7 cũng đang được nghiên cứu và phát triển, họ cũng nhắm đến các kiến thức mới lạ như *async function, observer*, .. Hy vọng sẽ có nhiều biến động mới.

Hiện nay các JS Framework như NodeJS, Angular2, React Native ... đều sử dụng ES6 nên để học được các framework đó thì ban phải thành thạo Javascript nói chung và ES6 nói riêng.

**2. Các chức năng mới của ES6**

Sau đây là một số chức năng mới thêm vào trong ES6.

* **Block Scoped**: Định nghĩa biến với từ khóa let, cách định nghĩa này thì biến chỉ tồn tại trong phạm vi khối của nó (Block Scope)
* **Destructuring Assignments**: Bạn có thể khởi tạo các biến từ một mảng bằng một dòng code đơn giản.
* **Default Parameters**: Bạn có thể gán giá trị mặc định cho các tham số.
* **Rest Parameter**: Tham số không giới hạn
* **Arrow function**: Bạn có thể tạo hàm bằng cách sử dụng dấu mũi tên =>.
* **Template String:** Tạo template HTML cực kì đơn giản
* **Weak, Set:** các kiểu dữ liệu phức tạp mới
* **Iterables và iterators**
* **Class, import**

**3. Một số phiên bản khác của ES6**

ES6 là phiên bản mới nhất tính đến thời điểm hiện tại là tháng 1 năm 2017, các phiên bản khác của ES6 bao gồm:

* **ECMAScript 5 (December 2009)**: Phiên bản này hầu hết các trình duyệt đều hỗ trợ, nó cải tiến và bổ sung thêm một số thư viện chuẩn, sử dụng chế độ [strict mode](https://freetuts.net/use-strict-la-gi-strict-mode-trong-javascript-407.html) nghiêm ngặt.
* **ECMAScript 5.1 (June 2011)**: ES5 đã được xem như là một chuẩn ISO, phiên bản này đã sửa một số lỗi nhỏ.
* **ECMAScript 6 (June 2015)**: Là phiên bản hiện tại, bổ sung nhiều cú pháp và thư viện.
* **ECMAScript 2016**: Hay còn gọi là ES7, đây là phiên bản tương lai chưa được public, được hứa hẹn có nhiều bổ sung giúp Javascript trở thành ngôn ngữ sử dụng được hầu hết ở mọi môi trường.

# **Mối liên hệ giữa**

# **Javascript và ECMAScript**



### **ECMA International là gì?**

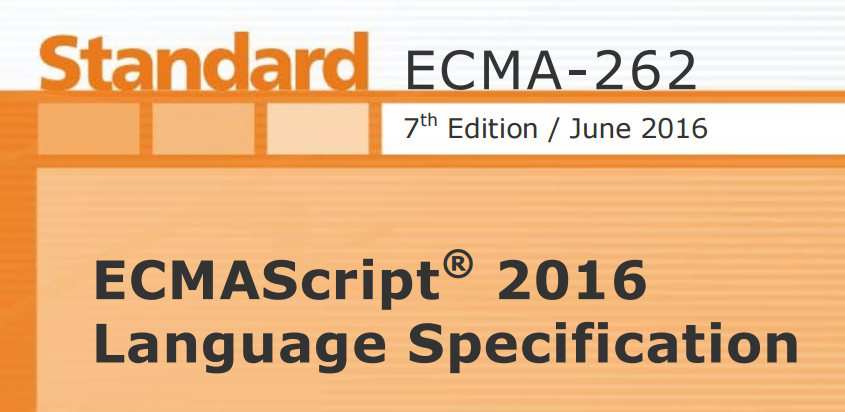


*Là một tổ chức tạo ra tiêu chuẩn (standard) cho các công nghệ.*

Tiêu chuẩn” là gì? Để hiểu khái niệm này, hãy nghĩ về chiếc bàn phím mà các bạn hay sử dụng trên chiếc điện thoại, laptop hay PC. Mặc dù các loại bàn phím này khác nhau về hãng sãn xuất, nhưng chúng vẫn dùng chung một layout (vị trí các phím chữ cái, phím số, phím chức năng, Enter đều nằm trên các vị trí giống nhau). Đó là vì các hãng sản xuất bàn phím đang tuân thủ theo ***QWERTY layout standard***.

*Như vậy tiêu chuẩn có thể được hiểu là một văn-bản được một tổ chức có-uy-tín trong một cộng đồng công bố và đòi hỏi tất cả các sản phẩm liên quan ra đời sau đó cần tuân theo các quy định trong tài liệu này.*

### **ECMA-262 là gì?**



*Đây là một tiêu chuẩn được tổ chức ECMA International công bố, chứa các đặc tả hay quy định cho các ngôn ngữ script nói chung.*

***Nhưng tại sao lại là con số 262?***

*Giả sử bạn đang ở trong một tổ chức khá lớn, tổ chức này viết rất nhiều tiêu chuẩn cho công nghệ nói chung, mỗi lần tổ chức cần viết ra một tài liệu gì đều lập ra một cái lịch biểu, trong lịch biểu đó mỗi task đều được tổ chức đề cho nó 1 mã như một ID để phân biệt, trường hợp này ECMA-262 chính là mã nhiệm vụ tạo tài liệu ECMAScript*

### **Scripting language là gì?**

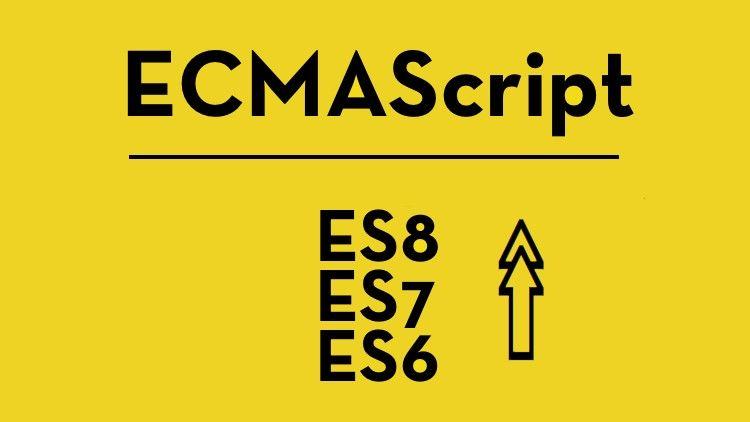
**

*Một loại ngôn ngữ được thiết kế ra để sử dụng lại các “thực thể hoặc hệ thống” đã được định nghĩa sẵn từ đầu.*

*Nôm na là như này, khi bạn có một thư viện đã có sẵn các hàm như “walk”, “run”, “jump”, những hàm này có chức năng di chuyển một đồ vật hoặc di chuyển một nhân vật trong game. Tuy nhiên, các hàm này không thể được thực thi nếu không có chỗ nào gọi đến chúng. Lúc ấy, tập hợp đoạn gọi ra các hàm này được gọi là scripting-language, đây là một loại ngôn ngữ tập trung vào việc sử dụng lại các thư viện đã có trong hệ thống.*

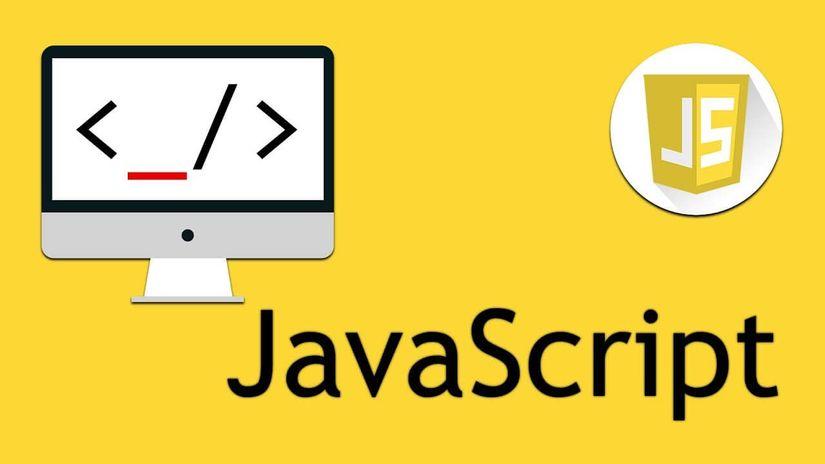
### **ECMAScript là gì?**

*Từ khóa tương đương: ECMAScript specification*

**

*Đặc tả này được định nghĩa trong ECMA-262 nhằm mục đích tạo ra tiêu chuẩn cho ngôn ngữ Script nói chung, nó cung cấp một bộ các quy tắc, đặc tả và hướng dẫn mà bắt buộc các ngôn ngữ script phải theo dõi và cân nhắc trong quá trình được triển khai nếu các tác giả của các ngôn ngữ này muốn được công nhận là phù hợp với tiêu chuẩn ECMAScript.*

### **JavaScript**

*Đây là một ngôn ngữ Script đã được triển khai dựa vào các chỉ dẫn trong tài liệu ECMAScript.*

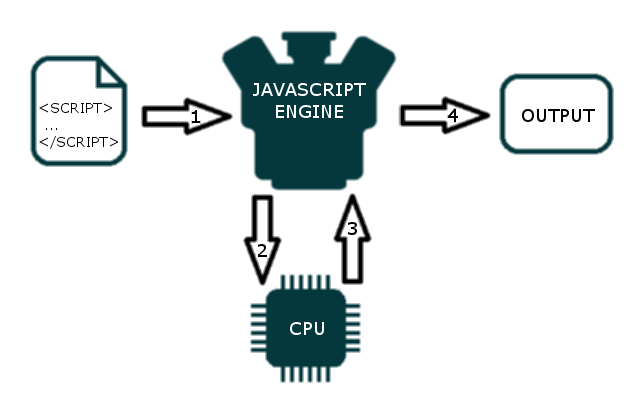
*JavaScript is dialect of the ECMAScript language*

*Một ngôn ngữ được coi là “dialect” (biến thể) của ngôn ngữ khác khi nó kế thừa hầu hết các khai báo, cú pháp của ngôn ngữ tiền bối của nó nhưng vẫn có một ít đặc trưng đủ để khiến nó trở nên khác biệt với các ngôn ngữ còn lại.*

***JavaScript chính là một biến thể của ngôn ngữ ECMAScript.***

*Các lập trình viên tạo lên Javascript đã dựa vào tiêu chuẩn của ECMAScript để tạo ra, cập nhật và phát triển ngôn ngữ này. Như vậy, chúng ta có thể đi đến kết luận, nếu chúng ta đọc đặc tả trong tài liệu ECMAScript, chúng ta sẽ biết cách tạo-ra một ngôn ngữ Script. Nhưng nếu chúng ta đọc tài liệu JavaScript, chúng ta sẽ biết được cách để sử dụng một ngôn ngữ script.*

### **JavaScript engine là gì?**

*Từ khóa tương đương: JavaScript interpreter, JavaScript implementation. *

*Một chương trình hiểu được và thực thi được các đoạn code được viết bằng JavaScript.*

*Các JavaScript engine thường được tìm thấy trong các trình duyệt Web như V8 ( của Chrome), SpiderMonkey ( của Firefox), Chakra ( của Edge). Mỗi engine tương tự như một language-module tương ứng với ứng dụng của nó, cho phép hỗ trợ một tập hợp các tập con của ngôn ngữ JavaScript.*

### **Hiệu năng của Javascript trên các trình duyệt?**

*Cũng tương tự như sự khác biệt giữa những người cùng nói và hiểu chung một ngôn ngữ (một số người có thể biết một vài từ mới, một vài cách diễn đạt và một vài quy tắc mà một số khác lại không biết và ngược lại), các trình duyệt cũng như vậy.*

*Dù các JavaScript engines của trình duyệt hiểu được JavaScript, nhưng một số trình duyệt lại có khả năng hiểu biết JavaScript tốt hơn.*

*Liên quan đến sự hỗ trợ của trình duyệt, mọi người thường nói về ECMAScript compatibility , mặc dù JavaScript engines phân tích cú pháp và thực thi JavaScript chứ không phải là ECMAScript.*

*Đoạn này nghe có vẻ không liên quan lắm, tuy nhiên có một lời giải thích hợp lý cho điều này:*

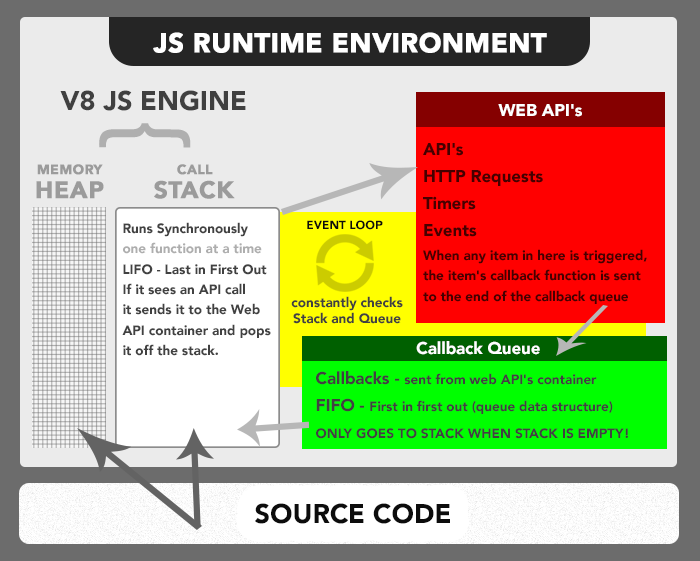
*ECMAScript là một đặc tả chứa một loạt các yêu cầu đòi hỏi các nhà phát hành ngôn ngữ Script phải tuân theo.*

***Việc phát hành một phiên bản mới của ECMAScript không có nghĩa là toàn bộ JavaScript engines của các hãng phần mềm hiện tại bắt buộc phải có các tính năng mới này****. Nó phụ thuộc vào các nhóm hoặc các tổ chức chịu trách nhiệm cho việc JavaScript engines phải là-bản-mới-nhất tương ứng với các đặc tả ECMAScript mới-nhất. Họ không cập nhật toàn bộ các tính năng mới vào một lúc mà chỉ cập nhật dần dần theo kế hoạch*

### **JavaScript runtime**

*Môi trường mà các đoạn code JavaScript được chạy trong đó và được thông dịch bởi JavaScript engine.*

*Môi trường runtime cung cấp các* ***host-objects*** *(host environment), đây là những objects mà JavaScript sẽ điều khiển và thao tác trên đó. JavaScript runtime chính là “thực thể hoặc hệ thống” đã được đề cập trong phần định nghĩa Scripting Language ở trên.*

**

*Các đoạn code được đưa vào JavaScript engine, JavaScript engine sẽ phân tích các đoạn code này xem chúng tương ứng với những hành động nào được định nghĩa trong JavaScript runtime và thực thi những hành động này.*

*Các ứng dụng nếu muốn hỗ trợ ngôn ngữ JavaScript, chúng phải tạo ra được các “host objects” tại thời điểm runtime. Cụ thể:*

* *Với phía client:*
  + *JavaScript runtime: trình duyệt Web*
  + *Host objects: window, document, các đối tượng chứa trong nó các phương thức để điều khiển logic hiển thị các thành phần trên trình duyệt*
* *Với phía server:*
  + *JavaScript runtime: Node.js*
  + *Host objects: file system, processs, requests trong Node.js*

*Bạn nào đã từng code JavaScript cho các ứng dụng Web chắc chắn ít nhất một lần sử dụng document hoặc window.* ***window và document không phải là một tính năng mặc định được JavaScript cung cấp. Chúng chỉ là các Web APIs được cung cấp bởi trình duyệt và xuất hiện với vai trò là JavaScript-host-environment****. Có một điểm khá thú vị:*

*Các JavaScript runtime khác nhau có thể chia sẻ chung một JavaScript engine.*

*Ví dụ như V8 là JavaScript engine được sử dụng trên cả Google Chrome và Node.js  — Đây là hai môi trường khác hẳn nhau.*

### **ECMAScript 6 là gì?**

*Từ khóa tương đương: ES6, ECMAScript 2015*

*Đây là phiên bản thứ 6 của tiêu chuẩn ECMA-262 với những thay đổi lớn và cản thiện văn bản đặc tả ECMAScript. Phiên bản ECMAScript thay đổi từ ES6 sang ES2015 bởi vì vào năm 2015, tổ chức ECMA International quyết định sẽ hàng năm xuất bản tài liệu này. Theo đó, đặc tả ECMAScript được dựa trên năm mà chúng được công bố.*

### **Babel là gì?**

*Transpiler (transforming + compiler) là một cách gọi các công cụ có tác dụng chuyển đổi code từ một phiên bản A về một phiên bản B đối với từng ngôn ngữ.*

*Babel là một transpiler có tác dụng chuyển đổi code ES6 về code ES5.*

*Babel dành cho các developer muốn đưa các tính năng mới nhất của ES6 vào ứng dụng của mình nhưng vẫn cần chạy các tính năng này trên tất hầu hết các trình duyệt. Babel sẽ giúp developer chuyển đổi code ES6 về ES5, vì phiên bản ES5 đã được hỗ trợ trên hầu hết tất cả các trình duyệt.*

### **Tóm gọn lại:**

#### Câu chuyện con gà hay quả trứng

JavaScript được tạo ra năm 1996, sau đó được gửi tới tổ chức ECMA International năm 1997 để viết ra văn bản ECMAScript. Về mặt logic mà nói, ngay tại thời điểm này, JavaScript vừa thỏa mãn tất cả các đặc tả viết trong văn bản ECMAScript, vừa là một ví dụ cho ngôn ngữ implement đặc tả này tốt nhất.

**Như vậy, ECMAScript được xây dựng dựa trên JavaScript và JavaScript cũng được xây dựng dựa vào ECMAScript??**

*Tham khảo thêm:* [*https://toidicodedao.com/2016/04/05/series-javascript-sida-cung-lam-quen-va-chich-nham-nghich-es6/#more-1882*](https://toidicodedao.com/2016/04/05/series-javascript-sida-cung-lam-quen-va-chich-nham-nghich-es6/#more-1882)

# **Tài liệu học Javascript**

Học theo từng bước qua các bài học dưới link:

<https://www.w3schools.com/js/default.asp>