

Cho người mới

Javascript

Từ cơ bản tới nâng cao

- Lý thuyết kèm bài tập
- Chỉ dẫn từng bước kèm code minh họa dễ hiểu



MỤC LỤC

LỜI NÓI ĐẦU	5
NỘI DUNG CUỐN SÁCH	7
CUỐN SÁCH NÀY DÀNH CHO AI?	9
Yêu cầu trình độ	9
Cách học đúng cách	9
GIỚI THIỆU	11
Lịch sử Javascript	12
Tại sao nên học Javascript?	12
TỔNG QUAN JAVASCRIPT	15
Ưu điểm của Javascript	16
Giới hạn của Javascript	17
Công cụ phát triển	18
Thực thi chương trình Javascript	18
Tạo chương trình Javascript đầu tiên	19
CÚ PHÁP JAVASCRIPT CƠ BẢN	23
Variable - Biến	24
Variable Scope - Phạm vi sử dụng của một biến	25
Sự khác nhau giữa var và let	28
Khái niệm và cơ chế Hoisting	29
Kiểu dữ liệu	31
Toán tử - Operators	35
Toán tử số học	35
Toán tử so sánh	36
Toán tử logic	37
Toán tử gán	38
Toán tử điều kiện rút gọn	39

Làm việc với điều kiện và cấu trúc có điều kiện	40
if...else Statements	40
Switch Statements	41
CÚ PHÁP JAVASCRIPT NÂNG CAO.....	42
Function.....	43
Cách định nghĩa một function	44
Tham số và phạm vi.....	45
Nested scope	46
Pure function và non-pure function.....	47
Loop - vòng lặp	48
Vòng lặp là gì?	49
Tại sao phải dùng vòng lặp.....	49
Vòng lặp for (...).....	50
Vòng lặp while() {...}	51
Vòng lặp do {...} while().....	52
Câu lệnh break và continue trong vòng lặp	53
DỮ LIỆU CÓ CẤU TRÚC.....	56
Object.....	57
Thuộc tính riêng và thuộc tính kế thừa	59
Cách tạo Object.....	60
Truy xuất thông tin Object.....	64
Truy xuất hàng loạt keys của Object	65
Xóa thuộc tính của Object	67
Array	67
Cách khai báo Array	67
Truy cập vào phần tử mảng	69
Các thao tác làm việc với mảng	70
HIGHER-ORDER FUNCTION.....	80
Khái niệm Higher-Order Functions	81

Functional Programming	81
First-Class Functions.....	82
Higher-Order Functions là gì?.....	82
Ví dụ minh họa Higher-Order function.....	84
Tìm hiểu kỹ hơn về Callback.....	87
Promise	92
Async/Await.....	98
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG VỚI JS	102
Nguyên lý lập trình hướng đối tượng (OOP).....	103
Javascript có hướng đối tượng không?	104
Tính kế thừa.....	106
Tính đóng gói.....	108
Tính đa hình và trừu tượng.....	110
CÚ PHÁP ES6.....	113
String.....	115
Function.....	118
Class.....	121
Destructuring.....	124
Object Destructuring.....	125
Array Destructuring	126
Spread operator (...).....	126
Modules.....	129
Export.....	130
Import	131
JAVASCRIPT FRAMEWORK.....	133
BÀI TẬP	136
KẾT NỐI VỚI VNTALKING	145
THÔNG TIN TÁC GIẢ	146
CUỐN SÁCH CỦA VNTALKING	147

LỜI NÓI ĐẦU

Hầu hết mọi người khi bắt đầu học lập trình web đều nhận được lời khuyên là bắt đầu từ HTML. Tuy nhiên, bản thân HTML không có nhiều tương tác và logic để bạn có thể học.

Có thể bạn không biết, phần lớn những tương tác giữa trang web với người dùng như: hiện popup, hiệu ứng chuyển động, slideshow ảnh hay gửi dữ liệu lên server... chủ yếu được thực hiện bằng Javascript.

Để phát triển các ứng dụng web tương tác với người dùng như vậy, bạn cần phải biết Javascript.

Việc tự học Javascript không hề khó, chỉ cần bạn hiểu được tư tưởng ban đầu của nó, thì việc học sẽ cực dễ dàng.

Cuốn sách này sẽ giúp bạn giải đáp những câu hỏi, những trăn trở khi tự mày mò tìm hiểu Javascript, cũng như trang bị những kiến thức từ nền tảng tới nâng cao về javascript, đặc biệt phù hợp với những người chưa có kinh nghiệm lập trình thực tế.

Điểm xuất phát của bạn có thể rơi vào 2 trường hợp sau:

- **Javascript là ngôn ngữ lập trình đầu tiên mà bạn tiếp cận.** Có thể bạn vô tình được nghe tới Javascript hoặc được bạn bè giới thiệu "Javascript là ngôn ngữ lập trình dễ học nhất".
- **Bạn đã từng học và làm việc trên các ngôn ngữ lập trình khác như Java, C#, Python...** Do điều kiện ngoại cảnh như chuyển dự án, vì nghe lời quảng cáo thần thánh hóa của các cao nhân trên mạng về Javascript. Hoặc có thể bạn có duyên với Javascript mà yêu thích ngôn ngữ lập trình này, muốn tìm hiểu và kết thân với nó.

Dù xuất phát điểm như thế nào, cuốn sách này cũng sẽ giúp bạn hiểu cặn kẽ, sử dụng thành thạo Javascript như một công cụ để xây dựng ứng dụng web, làm nền tảng vững chắc để bạn tìm hiểu các framework front-end như VueJS, ReactJS... hoặc chuyển sang cả mảng Back-end với NodeJS.

"Với Javascript, con đường trở thành full stack developer ngắn hơn bao giờ hết"

--VNTALKING --

Sau một thời gian dài chuẩn bị, mình cho ra đời cuốn sách này. Với mục tiêu: *"Mang cả thế giới Javascript vào giường của bạn, nhằm, vào sự nghiệp của bạn"* 😊

Bạn đã sẵn sàng đắm chìm vào thế giới "ma thuật" của Javascript chưa?

Còn chờ gì nữa. Hãy tiếp tục đọc và nghiền ngẫm, bạn sẽ cảm thấy yêu thích cuốn sách này.

Mình đảm bảo!

NỘI DUNG CUỐN SÁCH

Javascript là ngôn ngữ lập trình phổ biến nhất hiện nay. Giờ đây, bạn chỉ cần biết một ngôn ngữ lập trình là có thể chinh chiến từ Front-end tới Back-end.

Trước khi bạn nghĩ tới những điều lớn lao như xây dựng ứng dụng web kiểu như Tiki, Shopee... học các framework Front-end như ReactJS, VueJS, Angular... hay như chuyển sang học NodeJS để làm Back-end cho hệ thống, bạn cần phải nắm vững và sử dụng thành thạo công cụ, đó chính là ngôn ngữ lập trình Javascript.

Cuốn sách này ra đời để giúp bất kỳ ai, từ người chưa từng có kinh nghiệm lập trình tới người đã có kinh nghiệm mà muốn chuyển sang học Javascript.

Trong cuốn sách này, bạn sẽ được học và thực hành:

- Cú pháp Javascript cơ bản
 - Biến - variables
 - Kiểu dữ liệu
 - Toán tử operators
 - String
 - Làm việc với điều kiện và cấu trúc có điều kiện
- Cú pháp Javascript nâng cao
- Dữ liệu có cấu trúc Data structure:
- Higer-Order Function
- Lập trình hướng đối tượng (Object Oriented Programming)
- Làm quen cú pháp và tính năng mới của ES6
- Bài tập thực hành kèm đáp án.

Để đảm bảo các bạn tập trung và hiểu rõ Javascript, mình sẽ không sử dụng bất kỳ thư viện 3rd nào trong cuốn sách này, không trộn lẫn mã nguồn HTML và Javascript.

CUỐN SÁCH NÀY DÀNH CHO AI?

Cuốn sách này rất phù hợp cho những ai yêu thích lập trình, muốn học kiến thức nền tảng để tiếp tục phát triển các ứng dụng web, mobile hay PC bằng Javascript. Nếu bạn có định hướng sự nghiệp thành full stack developer thì cuốn sách này chính là tài liệu đầu tiên mà bạn cần tới.

Đây là cuốn sách "**No Experience Require**", tức là không yêu cầu người có kinh nghiệm lập trình, chưa từng lập trình. Tất cả sẽ được mình hướng dẫn học từ con số 0.

Yêu cầu trình độ

Javascript là một trong bộ ba kỹ thuật để xây dựng trang web gồm: Javascript, HTML, CSS. Do đó, để có thể học Javascript một cách trơn tru nhất, bạn nên biết:

- Kiến thức cơ bản về HTML.
- Biết sử dụng công cụ debug của trình duyệt

Nếu bạn không biết cả hai thứ trên thì sao? Cũng không sao, đọc xong cuốn sách này bạn cũng sẽ biết chúng thôi.

Cách học đúng cách

Cuốn sách này mình chia nhỏ nội dung thành nhiều phần, mỗi phần sẽ giới thiệu một chủ đề riêng biệt, kèm thực hành. Mục đích là để bạn có thể chủ động lịch học, không bị dồn nén quá nhiều, dễ dẫn tới "tẩu hỏa nhập ma", lúc đó lại oán trách mình 😊

Với mỗi phần lý thuyết, mình đều có ví dụ minh họa. Vì vậy, cách học tốt nhất vẫn là vừa học, vừa thực hành. Bạn nên **tự mình gõ lại từng dòng code** và kiểm tra kết quả trên trình duyệt. Đừng copy cả đoạn code trong sách, điều

này sẽ hạn chế khả năng viết code của bạn, cũng như khiến bạn nhiều khi không hiểu vì sao code bị lỗi.

"Nhớ nhé, đọc đến đâu, tự viết code đến đó, tự build và kiểm tra đoạn code đó chạy đúng không"

Ngoài ra, trong cuốn sách này, kiến thức phần sau được xây dựng từ phần trước. Do vậy, bạn đừng đọc lướt mà bỏ sót đoạn nào nhé.

Trong quá trình bạn đọc sách, nếu code của bạn không chạy hoặc chạy không đúng ý muốn mà vất tay lên trán mấy hôm vẫn chưa giải đáp được thì đừng ngần ngại đặt câu hỏi trên Group: **Hỏi đáp lập trình - VNTALKING**

Liên hệ tác giả

Nếu gặp bất kỳ vấn đề gì trong quá trình đọc sách, code bị lỗi hoặc không hiểu, các bạn có thể liên hệ với mình theo bất kỳ kênh nào dưới đây:

- Website: <https://vntalking.com>
- Fanpage: <https://facebook.com/vntalking>
- Group: <https://www.facebook.com/groups/hoidaplaptrinh.vntalking>
- Email: support@vntalking.com
- Github: <https://github.com/vntalking/Book-Javascript>

PHẦN 1

GIỚI THIỆU

Javascript (thường hay viết tắt là JS) là ngôn ngữ lập trình kịch bản (scripting language) cho client-side, sau này còn cho cả server-side (Nodejs)

Javascript được sử dụng chủ yếu để nâng cao sự tương tác của người dùng với trang web. Nói cách khác, bạn có thể làm cho trang web trở nên sinh động và tăng tính tương tác hơn. Trong các ứng dụng web, người ta hay dùng JS để làm các hiệu ứng đặc biệt như sliders, pop-ups, hoặc xác thực dữ liệu các form (form validations) trước khi gửi dữ liệu lên server .v.v...

Ngày nay, Javascript không chỉ giới hạn trong khuôn khổ xây dựng ứng dụng web, mà còn sử dụng rộng rãi trong phát triển ứng dụng, game trên điện thoại hay các ứng dụng dành cho server.

- Web app: ReactJS, VueJS, Angular...
- Mobile app: React Native, Ionic...
- Game: Phaser, Kiwi.js...
- Server app: Nodejs
- Graphic: two.js (2D), three.js (3D)...
- AI: brain.js...

Và còn nhiều nhiều nữa các lĩnh vực mà Javascript có thể làm được. Các bạn cứ bình tĩnh khám phá nhé.

Lịch sử Javascript

Javascript được tạo bởi lập trình viên kỳ cựu Brendan Eich, giới thiệu lần đầu năm 1995, xuất hiện trên trình duyệt Netscape, một trình duyệt phổ biến thời bấy giờ.

Ban đầu, ngôn ngữ lập trình này được gọi là LiveScript, sau này mới đổi tên thành Javascript. Mới đọc tên thì nhiều người sẽ nhầm tưởng Javascript có "họ hàng" với Java. Nhưng thực tế, hai ngôn ngữ này không hề có liên quan gì tới nhau cả, cây gia phả của chúng không hề chung gốc.

Java là ngôn ngữ lập trình hướng đối tượng phức tạp, còn Javascript là một ngôn ngữ kịch bản (scripting language). Cú pháp của Javascript chủ yếu có hơi hướng ảnh hưởng từ ngôn ngữ C.

Tại sao nên học Javascript?

Trước khi bạn quyết định đầu tư học một ngôn ngữ lập trình, đặc biệt với người chưa từng biết một ngôn ngữ lập trình nào, có thể bạn sẽ đắn đo, băn khoăn liệu mình có nên lao đầu vào ngôn ngữ lập trình này không? Liệu tương lai ngôn ngữ này có phát triển hay không?

Dưới đây là một số lý do để bạn bỏ công sức đầu tư học Javascript.

#1- Là ngôn ngữ lập trình phổ biến nhất

Khi bạn định hướng nghề nghiệp trong tương lai, việc chọn một ngôn ngữ lập trình phổ biến để theo đuổi là lựa chọn không hề tồi chút nào. Cũng giống như bạn đi kinh doanh, bán hàng vậy. Không ai dại gì lại đi bán mặt hàng mà thị trường không có nhu cầu sử dụng cả.

Theo một khảo sát mới nhất của Stackoverflow.com (website hỏi đáp dành cho lập trình viên lớn nhất thế giới) cho thấy, Javascript là ngôn ngữ lập trình phổ biến nhất, được rất nhiều lập trình viên chuyên nghiệp tin tưởng.

Không chỉ front-end, ngay cả các dự án back-end cũng ngày càng lựa chọn Javascript nhiều hơn.

#2- Javascript rất dễ học

Với tính mềm dẻo, linh hoạt, Javascript rất dễ học, đặc biệt là cho người mới học lập trình. Javascript biến các chi tiết phức tạp thành các bản tóm tắt, giúp mọi thứ trở nên dễ dàng hơn với người mới.

Không giống như các ngôn ngữ lập trình bậc cao khác, Javascript mang nhiều cảm giác về ngôn ngữ tự nhiên hơn. Tức là bạn nói sao thì viết như vậy.

#3- Tài nguyên học có sẵn rất nhiều

Khi tiếp cận bất kỳ kỹ thuật mới nào, việc quan trọng đầu tiên phải nghĩ tới đó là tài liệu hướng dẫn có đầy đủ không! Mình từng tham dự một dự án mà sử dụng một framework cổ xưa, tài liệu chính chủ còn không có (chắc tác giả cũng bỏ rơi nó luôn), lúc đó mới thấu hiểu nỗi đau khổ khi không có tài liệu.

Với ngôn ngữ Javascript nói chung, các JS frameworks như React, Vue... nói riêng thì đều có tài liệu hướng dẫn rất chi tiết và đầy đủ (cả chính chủ lẫn của cộng đồng). Do đó, bạn sẽ không gặp phải bất kỳ khó khăn nào trong việc tìm kiếm tài liệu hỗ trợ bạn trong việc học.

Ngoài ra, trên internet còn có hàng ngàn tutorial miễn phí để bạn có thể tham khảo. Tuy nhiên, việc dễ dàng tiếp cận hàng ngàn tài liệu cũng khiến bạn dễ bị bối rối, hỗn loạn kiến thức. Đó là lý do bạn tìm tới cuốn sách này.

#4- Một ngôn ngữ cho tất cả

Nếu trước đây, Javascript được sinh ra chỉ để xây dựng các trang web, thì nay đã khác. Javascript giờ đây có thể xây dựng mọi ứng dụng từ client-side tới back-end, các ứng dụng/game mobile, ứng dụng trên PC, kể cả các ứng dụng trên cloud, AI (Trí tuệ nhân tạo)...

Do đó, thay vì bạn phải đầu tư học rất nhiều ngôn ngữ, giờ bạn chỉ cần tập trung học Javascript cho thật tốt là đủ "cân cả bản đồ".

#5- Tiềm năng phát triển sự nghiệp lớn

Với việc ngày càng có nhiều doanh nghiệp và tổ chức chuyển sang sử dụng Javascript cho sản phẩm của mình. Do đó, nhu cầu tuyển dụng lập trình viên Javascript cũng tăng lên rất nhiều.

Theo một khảo sát của Devskiller.com, 70% các công ty công nghệ muốn tuyển một lập trình viên Javascript.

Ở Việt Nam thì sao? Đảo qua một loạt các trang tuyển dụng lớn như Vietnamworks, ITviec... nhu cầu tuyển lập trình viên React, Angular, Vue, NodeJS... rất nhiều, mức lương cũng rất cao (toàn trên 2k\$ cho một senior developer).

Tóm lại, theo đánh giá của mình, việc chọn Javascript là ngôn ngữ lập trình chính cho sự nghiệp là một lựa chọn đáng giá, xứng đáng với mồ hôi nước mắt.

Ok, giờ là lúc chúng ta cùng nhau chinh phục Javascript thôi!

Nội dung chính

- >> Ưu điểm của ngôn ngữ lập trình Javascript
- >> Giới hạn của Javascript
- >> Công cụ để phát triển ứng dụng với JS
- >> Tạo ứng dụng HelloWorld đầu tiên

PHẦN 2

TỔNG QUAN JAVASCRIPT

Rất nhiều bạn mới học sẽ cảm thấy đôi chút thất vọng khi nghe đâu đó có người nói Javascript là ngôn ngữ lập trình dành cho trẻ con. Thực tế thì họ đã nhầm! Để mình chỉ cho bạn thấy.

Đầu tiên, Javascript là một trong những ngôn ngữ lập trình mạnh nhất hiện nay. Nó là một kỹ năng mà mọi lập trình viên cần phải có nếu muốn theo sự nghiệp web development (ít nhất là vậy).

Cái hay của Javascript nằm ở chỗ cách viết đơn giản để giải quyết một vấn đề phức tạp.

Ưu điểm của Javascript

Để nói về ưu điểm của một ngôn ngữ lập trình thì có thể kể hàng chục trang giấy. Bởi vì, mỗi ngôn ngữ được tạo ra, tác giả đều sẽ cố gắng tối ưu, thiết kế sao cho tốt nhất. Không phải ngẫu nhiên mà Javascript được cộng đồng đón nhận rộng rãi đến như vậy.

Tuy vậy, để bạn hiểu được thế mạnh của Javascript, mình sẽ liệt kê một số nét đặc trưng khiến Javascript nổi bật hơn các ngôn ngữ lập trình khác.

#1- Tích hợp sẵn trong hầu hết các trình duyệt

Không giống như nhiều ngôn ngữ phát triển web khác, ví dụ flash, Java... người dùng muốn sử dụng được thì phải cài đặt thêm plugin cho trình duyệt.

Javascript thì khác, hầu hết trình duyệt hiện đại đều đã tích hợp sẵn. Do đó, việc bạn sử dụng JS để phát triển ứng dụng sẽ rất thuận lợi.

#2- Một ngôn ngữ lập trình vô cùng linh hoạt

Rất nhiều lập trình viên thích trường phái functional programming. Functional Programming một phương pháp lập trình dựa trên các hàm toán học (function), tránh việc thay đổi giá trị của dữ liệu. Nó có nhiều lợi ích như : các khối xử lý độc lập dễ tái sử dụng, thuận lợi cho việc thay đổi logic hoặc tìm lỗi chương trình.

Javascript là ngôn ngữ sinh ra là để dành cho functional programming.

Hai trong số tính năng nổi bật nhất của Javascript là cho phép gán một hàm cho bất kỳ biến nào và tạo một hàm chấp nhận tham số là một hàm khác.

#3- Khả năng tự detect trình duyệt và hệ điều hành

Đôi khi, trong một số ứng dụng, bạn gặp vấn đề và cần phải viết mã nguồn tương thích với từng trình duyệt web hoặc hệ điều hành. Javascript được thiết kế để có thể tự nhận biết được chạy trên trình duyệt gì, hệ điều hành nào.

Điều này, cho phép bạn dễ dàng điều chỉnh mã nguồn để ứng dụng đáp ứng và tương thích với mọi hệ điều hành.

#4- Hỗ trợ cả lập trình hướng đối tượng (OOP)

Lập trình hướng đối tượng cũng là một trường phái lập trình rất phổ biến, khi tất cả mọi thứ của ứng dụng đều xoay quanh đối tượng (Object - Class).

Javascript cung cấp rất nhiều công cụ để bạn làm việc với đối tượng, đồng thời nó cũng dễ học, dễ sử dụng.

Nói một cách chính xác hơn, có thể coi Javascript là ngôn ngữ dựa trên đối tượng, vì lý do:

- Không hỗ trợ đầy đủ các đặc điểm của OOP như: đa hình, kế thừa
- Có sẵn kiểu dữ liệu đối tượng. Ví dụ: JavaScript có sẵn đối tượng window...

Trong cuốn sách này, chúng ta cũng sẽ tìm hiểu cách [lập trình hướng đối tượng bằng Javascript](#). Đừng bỏ qua nhé.

#5- Học một ngôn ngữ dùng mọi nơi

Phần này mình chỉ nhắc lại thôi. Nếu trước đây, Javascript được tạo ra chỉ để phát triển các ứng dụng front-end chạy trên trình duyệt thì giờ đây mọi chuyện đã khác. Có thể bạn chưa biết, trước đây một full stack developer cần phải học rất nhiều ngôn ngữ lập trình. Có thể kể tên nhẹ nhàng như: front-end thì có javascript, back-end thì có PHP, JAVA, Ruby, Golang..., ứng dụng mobile thì có Java, Kotlin, Swift... rất nhiều ngôn ngữ phải học.

Nhưng ngày nay, bạn chỉ cần học duy nhất Javascript là đủ. Biết Javascript, bạn có thể xây dựng các ứng dụng web (cái này tất nhiên rồi), xây dựng ứng dụng phía back-end (nhờ Node.JS), xây dựng ứng dụng mobile (React Native, Ionic...)



Đây chỉ là bản demo - phần này còn nữa
Đặt sách [tại đây](#) để đọc bản đầy đủ nhé

Tạo chương trình Javascript đầu tiên

Trong toàn bộ nội dung cuốn sách này, chúng ta sẽ chỉ sử dụng Javascript là ngôn ngữ xây dựng ứng dụng client (front-end) trên trình duyệt. Còn Javascript để tạo ứng dụng trên server (sử dụng NodeJS) sẽ đề cập trong cuốn sách khác. Mặc dù, cơ bản cú pháp là như nhau nhưng môi trường để chạy JS sẽ có khác nhau đôi chút.

Để chạy một chương trình Javascript, bạn cần đưa mã JS vào trong một trang HTML. Có hai cách để thêm mã vào trang HTML:

- Cách 1: Viết mã lệnh JavaScript trực tiếp vào trang web.

- Cách 2: Viết mã lệnh JavaScript vào tập tin JavaScript (*tập có phần đuôi là .js*)

Với các ứng dụng web thực tế, chủ yếu sử dụng cách 2 để thêm JS. Tuy nhiên, mình vẫn sẽ giới thiệu cả hai cách trong cuốn sách này.

Cách 1: Viết mã lệnh trực tiếp vào trang web

Bạn đặt toàn bộ mã Javascript vào trong thẻ `<script>`. Điều này giúp trình duyệt phân biệt mã JS với phần còn lại. Ngoài ra, vì có các ngôn ngữ kịch bản cho client khác nữa (ví dụ: VBScript...), do đó, bạn nên chỉ định ngôn ngữ kịch bản mà bạn sử dụng trong thẻ `<script>`, kiểu như sau:

```
<script type="text/javascript">
```

Bạn tạo mới một tập html đặt tên là `index.html`, có nội dung như sau:

```
<html lang='en'>
  <head>
    <meta charset='UTF-8' />
    <title>Sách học lập trình Javascript - by VNTALKING</title>
  </head>
  <body>
    <h2>Xin chào độc giả VNTALKING</h2>
    <span id='content'></span>
  </body>
  <script type='text/javascript'>
    document.getElementById("content").innerHTML = "Tài liệu học Javascript t
oàn
    tập";
  </script>
</html>;
```

Bạn lưu lại và mở tập lên bằng trình duyệt. Đây là kết quả thu được.



Xin chào độc giả VNTALKING

Tài liệu học Javascript toàn tập

Hình 2.1: Chương trình hello world bằng javascript

Cách 2: Viết mã lệnh JavaScript vào tập tin JavaScript

Với cách viết Javascript trực tiếp vào trong HTML sẽ khiến cho mã nguồn trở nên "rối rắm", đặc biệt khi dự án ngày càng trở nên phức tạp. Đặc biệt là trong trường hợp chúng ta muốn xây dựng các plugin/module bằng javascript, tách biệt với HTML.

Chúng ta sẽ tách mã JS ra khỏi HTML, và chỉ nhúng đường dẫn tới tệp .js trong HTML mà thôi.

```
<html lang='en'>
  <head>
    <meta charset='UTF-8' />
    <title>Sách học lập trình Javascript - by VNTALKING</title>
  </head>
  <body>
    <h2>Xin chào độc giả VNTALKING</h2>
    <span id='content'></span>
  </body>
  <script src='src/index.js'></script>
</html>;
```

Còn nội dung tệp *index.js* như sau:

```
document.getElementById("content").innerHTML =
"Tài liệu học Javascript toàn tập";
```

Cách 3 (Optional): Sử dụng IDE online.

Để học Javascript, thay vì sử dụng hai cách trên, mình thường tìm tới các Text Editor Online để viết và chạy thử chương trình Javascript được nhanh và trực quan nhất.

Một số Text Editor Online gợi ý cho bạn:

- [Codesandbox.io](https://codesandbox.io)
- [Playcode.io](https://playcode.io)
- [Jsfiddle.net](https://jsfiddle.net)

Ưu điểm của những trình Text Editor Online này là dễ sử dụng, được tích hợp sẵn mọi thứ, không cần cài đặt thêm gì cả. Việc của bạn chỉ là gõ code và chạy.

Trong cuốn sách này, mình chọn codesandbox.io để chạy các đoạn code minh họa. Lý do đơn giản là bởi vì IDE online này nó thiết kế rất giống với Visual Studio Code, các tính năng cũng tương tự, có thể kết nối tới Github để lưu và lấy code.



Hình 2.2: Cách chạy code Javascript trực tiếp trên các Text Editor Online

Hai đoạn code minh họa trên, mình có tạo trên codesandbox, bạn có thể tham khảo:

- <https://codesandbox.io/s/js-hello-world-cach-1-ijw66>
- <https://codesandbox.io/s/kind--b3upj>

Đọc đến đây, mình hi vọng bạn đã hiểu được phần nào về ngôn ngữ lập trình Javascript, biết cách tạo một ứng dụng đơn giản với Javascript. Phần tiếp theo, chúng ta sẽ cùng nhau tìm hiểu các cú pháp từ cơ bản tới nâng cao của Javascript nhé.

>> Định nghĩa và cách sử dụng biến - variable

>> Tìm hiểu kiểu dữ liệu

>> Các toán tử hay dùng trong Javascript

>> Làm việc với câu lệnh điều kiện

PHẦN 3

CÚ PHÁP JAVASCRIPT CƠ BẢN

Theo các hiểu thông thường, một chương trình máy tính là một tập danh sách các hướng dẫn (có thể gọi là statements) để máy tính thực thi. Và Javascript cũng vậy, chỉ khác một điều là các tập lệnh này cho trình duyệt thực hiện thay vì máy tính.

Sau này, khi tìm hiểu các ngôn ngữ lập trình khác, bạn cũng sẽ thấy về cơ bản thì chúng cũng na ná như nhau mà thôi. Tất cả cũng chỉ xoay quanh về biến, hàm, kiểu dữ liệu, các toán tử tính toán và so sánh, mảng, vòng lặp.v.v... Có chăng, chỉ khác nhau đôi chút về cách viết mà thôi, hay còn gọi là cú pháp (syntax).

Đó là lý do tại sao người ta vẫn truyền tai nhau rằng: chỉ cần bạn nắm vững một ngôn ngữ lập trình, sau đó muốn chuyển sang một ngôn ngữ khác sẽ rất dễ dàng.

Chúng ta sẽ cùng nhau tìm hiểu cú pháp cơ bản của Javascript ngay bây giờ.

Variable - Biến

Hiểu đơn giản, variable - biến là tên nơi lưu trữ dữ liệu. Chúng ta sử dụng biến để lưu trữ giá trị (name = "anh sơn") hoặc biểu thức (sum = x + y).

Trước khi có thể sử dụng được biến, bạn cần phải khai báo nó. Bạn có thể sử dụng từ khóa *var* hoặc *let* để khai báo một biến.

Đoạn code dưới đây, chúng ta khai báo một biến có tên là message:

```
let message;
```

Giờ bạn có thể đưa thông tin vào biến bằng cách sử dụng toán tử "="

```
let message;  
message = "Xin chào độc giả VNTALKING quý mến";
```

Giờ đây thì giá trị của string được lưu trữ trong bộ nhớ và liên kết với biến. Bạn có thể sử dụng nó thông qua tên biến.

```
let message;  
message = "Xin chào độc giả VNTALKING quý mến";  
  
alert(message);
```

Nếu bạn không thích dài dòng, bạn có thể gán giá trị ngay khi khai báo biến.

```
let message = "Xin chào độc giả VNTALKING quý mến";  
alert(message);
```

Ngoài ra, bạn cũng có thể khai báo nhiều biến trên cùng một dòng cũng được.

```
let user = "Son Duong", age = 25, message = "Xin chào";
```

Nhìn có vẻ ngắn gọn hơn đấy, nhưng mình không khuyến khích cách viết này. Nên viết khai báo mỗi biến trên một dòng để dễ đọc hơn.

```
let user = "Son Duong";  
let age = 25;  
let message = "Xin chào";
```

Hoặc viết thế này cũng được:

```
let user = "Son Duong",
```

```
age = 25,  
message = "Xin chào";
```

Như mình đã đề cập ở trên, bạn có thể sử dụng *var* để khai báo biến.

```
var message = "Xin chào độc giả VNTALKING quý mến";
```

Vậy sự khác nhau giữa *var* và *let* như thế nào? Để thấy được sự khác biệt giữa *var* và *let*, chúng ta cần hiểu một khái niệm khác nữa, đó là phạm vi của một biến: global và local variables. Sau khi tìm hiểu xong khái niệm này, mình chỉ quay trở lại vấn đề của *var* và *let* nhé.

Bạn có thể tham khảo code online tại đây: <https://codesandbox.io/s/variable-4vq2v>



Đây chỉ là bản demo - phần này còn nữa
Đặt sách [tại đây](#) để đọc bản đầy đủ nhé

Nội dung chính

>> Làm việc với function, định nghĩa và cách sử dụng

>> Tìm hiểu chi tiết các loại vòng lặp trong JS

>> Sự khác nhau giữa câu lệnh break và continue

PHẦN 4

CÚ PHÁP JAVASCRIPT NÂNG CAO

Sau khi hoàn thành xong phần 3 - cú pháp cơ bản của Javascript, bạn đã hiểu phần nào về ngôn ngữ lập trình đầy mê hoặc này rồi đúng không? 😊

Nhưng Javascript không chỉ dừng lại có thế, nó còn rất nhiều điều hay ho phía trước nữa. Đảm bảo bạn sẽ cảm thấy thích thú hơn nữa cho mà xem.

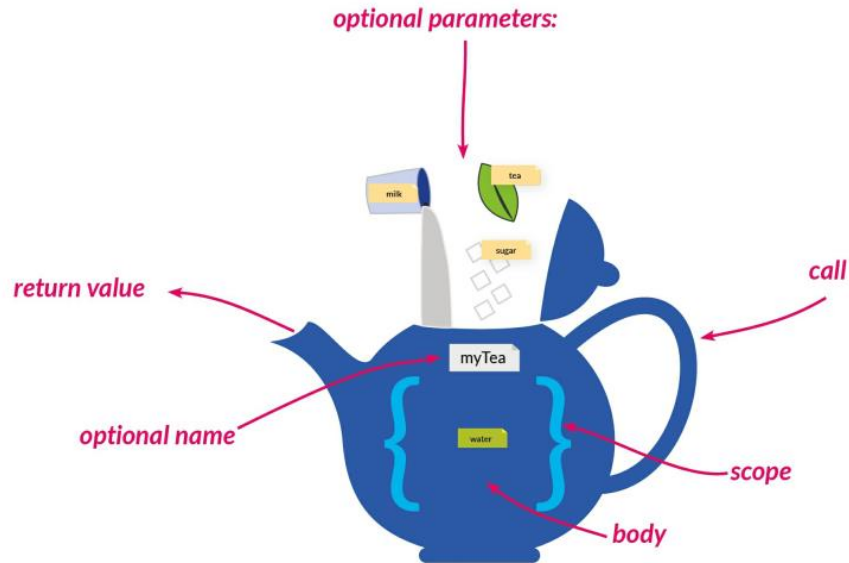
Trong phần cú pháp nâng cao, chúng ta sẽ tập trung tìm hiểu hai khái niệm rất phổ biến: *function* và vòng lặp (*loop*).

Function

Function hay còn gọi là hàm có thể được coi là một chương trình con, thực hiện một nhiệm vụ nào đó.

Cũng tương tự một chương trình hoàn chỉnh, một function gồm một tập hợp nhiều khối lệnh bên trong, gọi là "body" của function. Ngoài ra, function có thể nhận giá trị truyền vào qua các tham số và trả về kết quả sau khi kết thúc.

Có thể hình dung function như một cái máy chế biến, nhận nguyên liệu và trả về thành phẩm sau khi đã chế biến xong.



Hình 4.1: Minh họa cơ chế hoạt động của function

Function sinh ra là để giải quyết bài toán tái sử dụng mã nguồn. Người ta có thể gọi function ra để thực hiện những công việc giống nhau ở mọi nơi, thay vì phải viết lại toàn bộ code thực hiện nhiệm vụ đó, tránh trùng lặp code.



Đây chỉ là bản demo - phần này còn nữa
Đặt sách [tại đây](#) để đọc bản đầy đủ nhé

Cách định nghĩa một function

Chúng ta có nhiều cách để định nghĩa một function. Nhưng nhìn chung là phải sử dụng từ khoá *function* để định nghĩa một function. Từ phiên bản ES6, chúng ta có cách khác nữa để định nghĩa một function, đó là Arrow function. Phần 8 của cuốn sách, mình sẽ giới thiệu kỹ hơn cách này.

Cách 1: Định nghĩa kiểu truyền thống

```
function functionName([param1], [param2],...){
  // body hàm
  statement1;
  statement2;
  ...
}
```


Cách 2: Định nghĩa function rồi gán cho biến

Vì function trong JS được coi là một first-class Object nên function cũng có thể được gán cho một biến và có thể truyền *function* vào một *function* khác với vai trò như một tham số (sau này bạn sẽ gặp nhiều cái này với tên gọi "thân thương" là callback).

```
var functionName = function([param1], [param2],...){
  statement1;
  statement2;
  ...
};
```

Dưới đây là một số ví dụ:

```
// Định nghĩa hàm
function sayHello () {
  console.log("Xin chào các bạn độc giả VNTALKING")
}

var sayHi = function() {
  console.log("Xin chào các bạn độc giả VNTALKING")
}

const square = function (number) {
  return number * number;
}

// gọi hàm
sayHello();
sayHi();
console.log(square(5));
```



Đây chỉ là bản demo - phần này còn nữa
Đặt sách [tại đây](#) để đọc bản đầy đủ nhé

>> Khái niệm, các thao tác làm việc với Object

>> Cách khai báo và làm việc với mảng dữ liệu. Tìm hiểu những hàm built-in hữu ích cho Array.

PHẦN 5

DỮ LIỆU CÓ CẤU TRÚC

Giống với các ngôn ngữ lập trình bậc cao khác, Javascript cũng hỗ trợ các loại dữ liệu có cấu trúc, là những dữ liệu kết hợp các kiểu dữ liệu khác.

Trong phần này, chúng ta sẽ cùng tìm hiểu hai loại dữ liệu có cấu trúc phổ biến nhất:

- Object
- Array

Có một quan điểm cho rằng: chỉ cần bạn hiểu rõ và làm việc thành thục với Object là bạn đã làm chủ được Javascript.

Mình cũng hoàn toàn đồng ý với quan điểm đó. Bởi vì, khi bạn làm việc với Javascript, bạn sẽ làm việc rất nhiều với Object. Sau này, khi bạn học xong cuốn sách này, đi làm dự án, hãy thử ngẫm lại và cảm nhận xem quan điểm trên có đúng không nhé!?

Object

Object hay còn gọi là đối tượng, là một kiểu dữ liệu rất đặc biệt, nói không quá khi nó là kiểu dữ liệu quan trọng nhất, tạo thành nền tảng của Javascript hiện đại.

Kiểu dữ liệu Object là sự kết hợp của nhiều kiểu dữ liệu, cả nguyên thủy (primitive data-types) như: *Number, String, Boolean, null, undefined...* và cả Object khác trong đó (Object lồng nhau).

Thật khó để đưa ra một định nghĩa chính xác về kiểu dữ liệu Object. Mình xin định nghĩa một cách dễ hiểu nhất như sau: Kiểu dữ liệu Object trong Javascript là một tập hợp các kiểu dữ liệu liên quan không theo thứ tự dưới dạng cặp “*key: value*”. Key ở đây có thể là biến hoặc hàm và được gọi là thuộc tính hoặc method, tùy ngữ cảnh của Object đó.

Một Object được định nghĩa bằng dấu ngoặc nhọn {...} và chứa bên trong danh sách các thuộc tính.

Thuộc tính là một cặp “*key:value*”, trong đó key phải là một *string* hoặc *number* (hay còn gọi là tên thuộc tính), còn value thì có thể là bất kì thứ gì (có thể là giá trị, hàm hoặc một object khác...)

Đọc định nghĩa xong lại cảm thấy khó hiểu hơn phải không? Haha, trước mình cũng thế 😊

Để dễ hiểu hơn định nghĩa này, chúng ta cùng xem xét ví dụ sau nhé.

```
let website = {  
  name : "VNTALKING.COM",  
  location : "Hà Nội - Việt Nam",  
  established : "2018"  
}
```

Trong ví dụ trên, “name”, “location”, “established” đều là “key”.

Còn “VNTALKING.COM”, “Hà Nội – Việt Nam”, “2018” là các value, giá trị tương ứng.

Tóm lại:

- Mỗi cặp “*key:value*” này được gọi là thuộc tính (properties) của Object.
- Với thuộc tính nào mà value lại là một hàm thì lúc này người ta gọi đây là method của Object đó.

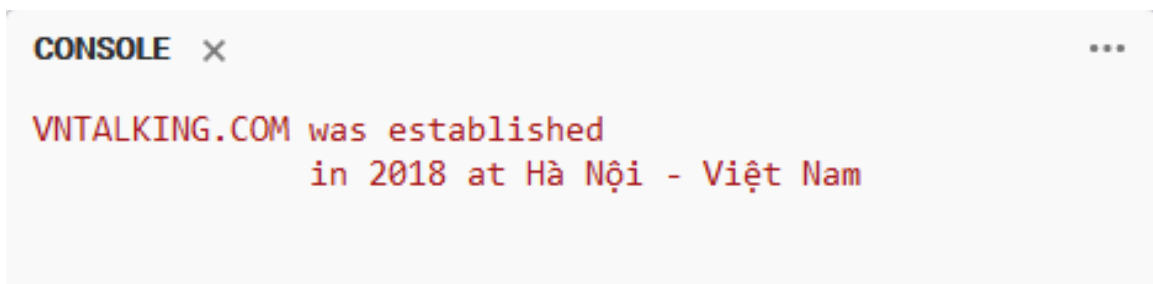
Nếu Object trên mà có method thì sẽ như thế nào? Cùng xem ví dụ sau đây:

```
let website = {  
  name : "VNTALKING.COM", ← Đây gọi là thuộc tính  
  location : "Hà Nội - Việt Nam",  
  established : "2018",  
  displayInfo : function(){  
    console.log(` ${website.name} was established  
      in ${website.established} at ${website.location}`);  
  }  
}
```

↑
↓ Đây gọi là method

```
website.displayInfo();
```

Kết quả khi chạy đoạn code trên:



```
CONSOLE x ...  
  
VNTALKING.COM was established  
  in 2018 at Hà Nội - Việt Nam
```

Chúng ta có thể hiểu nôm na như sau: Object để mô tả đặc điểm – hành vi của một đối tượng, sự vật nào đó. Ví dụ: chúng ta dùng Object để mô tả một ai đó, trong đó thuộc tính là đặc điểm về chiều cao, cân nặng, màu da... còn method là hành vi của người đó như cách người đi lại, cách ăn uống, cách người đó thể hiện bản thân...

Đến đây, bạn đã hiểu rõ định nghĩa về Object rồi đúng không!?



Đây chỉ là bản demo - phần này còn nữa

Đặt sách [tại đây](#) để đọc bản đầy đủ nhé

>> Tìm hiểu khái niệm và cách hoạt động của higher-order function

>> Callback và ứng dụng

>> Sử dụng Promise

>> Viết code đẹp hơn với Async/Await

PHẦN 6

HIGHER-ORDER FUNCTION

Khi bạn đọc đến dòng này, chứng tỏ bạn cũng khá kiên trì đọc sách đấy ^_^.
Với 5 phần đầu của cuốn sách, bạn đã bước đầu hiểu và làm chủ những kỹ thuật cơ bản của ngôn ngữ lập trình Javascript, là nền tảng để tiếp tục nghiên cứu những phần nâng cao về Javascript, những điều mà sẽ khiến bạn thêm yêu mến ngôn ngữ lập trình này.

Phần này, chúng ta sẽ cùng nhau tìm hiểu một khái niệm vô cùng quan trọng và ảnh hưởng rất nhiều tới tư duy lập trình khi làm việc với Javascript. Đó là khái niệm Higher-Order Functions.

Với việc hỗ trợ Higher-Order Functions khiến cho Javascript phù hợp với triết lý lập trình hướng hàm (Functional Programming). Tại sao lại như vậy? Mình sẽ giải thích ngay ở dưới đây.



Đây chỉ là bản demo - phần này còn nữa
Đặt sách [tại đây](#) để đọc bản đầy đủ nhé

Higher-Order Functions là gì?

Chúng ta hiểu đơn giản như sau, Higher-Order Functions là một hàm:

- Chấp nhận truyền một hàm qua tham số

- Hoặc hàm đó có thể return về một hàm.

Ví dụ:

```
// phần định nghĩa hàm higher-order
function higherOrder (number, sayLog) {
  let result = number * 10;
  sayLog(result)
}

function sayLog(x) {
  console.log(x);
}

// Cách sử dụng
higherOrder(10, sayLog)
//==>kết quả: 100
```

Hàm này gọi là higher-order function

Hàm này gọi là callback function

Hoặc chúng ta có thể truyền thẳng hàm ẩn danh (anonymous function) làm callback cũng được. Ví dụ như dưới đây:

```
// phần định nghĩa hàm higher-order
function higherOrder (number, callback) {
  let result = number * 10;
  callback(result)
}

//Cách sử dụng
higherOrder(10, function(result) {
  console.log(result)
});
//==>kết quả: 100
```



Đây chỉ là bản demo - phần này còn nữa
Đặt sách [tại đây](#) để đọc bản đầy đủ nhé

>> Tìm hiểu 4 nguyên lý của lập trình OOP

>> Tìm câu trả lời cho câu hỏi: JS có phải ngôn ngữ lập trình OOP không?

>> Tìm hiểu xem JS hỗ trợ các nguyên lý OOP như thế nào?

PHẦN 7

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG VỚI JS

Từ đầu cuốn sách đến giờ, bạn có nhận ra rằng mọi thứ trong Javascript đều là đối tượng, từ các biến, function, Array... (trừ duy nhất các dữ liệu kiểu nguyên thủy).

Như mình từng đề cập, chỉ cần bạn nắm chắc được Object là bạn sẽ học Javascript rất nhanh chóng và cảm thấy nó đơn giản hơn bao giờ hết.

Trước đây, mọi người nói tới Javascript là nghĩ ngay đây là ngôn ngữ lập trình kịch bản, kiểu script, viết là chạy theo từng block riêng lẻ, chỉ dùng để xử lý logic, tạo hiệu ứng cho các website. Nhưng hiện nay, Javascript đã khác trước rất nhiều, với sự hỗ trợ làm việc mạnh mẽ với Object, liệu Javascript có thực sự phù hợp với triết lý lập trình hướng đối tượng (OOP) hay không? Phần 7 của cuốn sách, chúng ta cùng nhau tìm hiểu chủ đề này.

Để hiểu được trọn vẹn phần này, mình khuyên bạn nên đọc lại và hiểu thật rõ về Object. Mời bạn ôn lại [kiến thức về Object](#)



Đây chỉ là bản demo - phần này còn nữa

Đặt sách [tại đây](#) để đọc bản đầy đủ nhé

>> Giới thiệu và thực hành làm quen với cú pháp và tính năng mới trong ES6.

>> Những điểm mới trong ES6 so với bản ES5: String, Function, Class, Destructuring, Module, Spread operator

PHẦN 8

CÚ PHÁP ES6

ES6 là chữ viết tắt của ECMAScript 6 hay ECMAScript 2015, là một tập các quy tắc lập trình không chỉ dành riêng cho Javascript mà nhiều ngôn ngữ lập trình khác nữa như ActionScript. Nhưng có lẽ Javascript là ngôn ngữ lập trình phổ biến nhất nên nhiều người mặc định Javascript và tiêu chuẩn ECMAScript là một cặp đi liền với nhau.

ES6 là phiên bản tiếp theo của ECMAScript, được xuất bản năm 2015, hiện nay ECMAScript đã ra đến phiên bản ES11, nhưng ES6 là phiên bản mà có rất nhiều cải tiến nổi bật, bổ sung rất nhiều tính năng hay hơn hẳn so với bản trước đó và hiện cũng được sử dụng rất nhiều trong các dự án hiện đại, đó cũng là lý do mình đưa vào trong nội dung cuốn sách này.

Hiện tại, ES6 đã được hỗ trợ trên hầu hết các trình duyệt hiện đại như Chrome, Firefox, Edge... và cả trên các phiên bản Node.JS mới nhất.

Không giống với cú pháp mà chúng ta đã học trong các phần trước của cuốn sách, với ES6 bạn sẽ có phần bỏ ngữ khi đọc mã nguồn của ai đó mà nhìn toàn ký tự “suy ra”, “dấu 3 chấm”... kiểu như sau:

```
var pairs = evens.map(v => ({even: v, odd: v + 1}));
// Statement bodies
nums.forEach(v => {
  if (v % 5 === 0)
    fives.push(v);
```



```
});
```

```
// Hoặc  
const arr1 = [1,2,3]  
const arr2 = [4,5,6]  
const arr3 = [...arr1, ...arr2]
```

Trong chương này của cuốn sách, mình không thể giới thiệu hết tất cả cú pháp của ES6, nhưng sẽ giới thiệu những cú pháp, tính năng phổ biến nhất của ES6 để bạn có thể sử dụng và đọc hiểu mã nguồn Javascript được viết theo chuẩn ES6.

OK, giờ còn chờ gì nữa, bắt đầu thôi nhĩ!



Đây chỉ là bản demo - phần này còn nữa
Đặt sách [tại đây](#) để đọc bản đầy đủ nhé

PHẦN 9

JAVASCRIPT FRAMEWORK

Như bạn biết, Javascript chỉ là ngôn ngữ lập trình, việc hiểu và vận dụng tốt các kiến thức về Javascript là nền tảng để bạn bước vào các dự án thực tế.

Thông thường, các dự án thực tế ít ai xây dựng một ứng dụng mà chỉ sử dụng thuần Javascript, bởi vì nó sẽ tốn rất nhiều công sức và thời gian. Thay vì đó, họ sẽ sử dụng các Javascript framework, điều này giúp họ tăng tốc độ phát triển dự án, nhanh chóng đưa sản phẩm ra thị trường.

Tất nhiên, để bạn có thể học và làm chủ một Javascript framework bất kỳ, trước hết bạn phải nắm vững kiến thức nền tảng ngôn ngữ lập trình Javascript, đó cũng là lý do tại sao phần này mình để ở cuối cuốn sách này.



Đây chỉ là bản demo - phần này còn nữa

Đặt sách [tại đây](#) để đọc bản đầy đủ nhé

>> Cùng thực hành viết code Javascript cùng với 5 bài tập . Sau khi viết code xong thì chạy unit test để kiểm tra kết quả.

PHẦN 10

BÀI TẬP

Xin chúc mừng (^_^) vì bạn đã đến được tận đây. Chúc mừng vì sự kiên trì và nỗ lực của bạn.

Sau khi xong phần lý thuyết, để có thể nắm chắc và nhớ lâu thì không thể thiếu phần thực hành. Bạn làm càng nhiều, va chạm với nhiều loại yêu cầu, càng giúp bạn nhanh nhạy và nhớ được lý thuyết lâu hơn, giống như chiến binh càng dạn dày trận mạc sẽ càng trở lên tinh nhuệ hơn.

Phần này, chúng ta sẽ cùng nhau làm bài tập để kiểm tra xem bạn đã nắm được kiến thức đến đâu, cũng như rèn luyện kỹ năng viết code của mình.

Các bài tập sẽ được sắp xếp từ dễ đến khó. Các bạn cố gắng tự thực hiện sau đó mới tham khảo đáp án. Nên nhớ, đáp án chỉ mang tính chất tham khảo, bạn làm theo cách nào cũng được, miễn là chương trình chạy đúng kết quả là được.



Đây chỉ là bản demo - phần này còn nữa

Đặt sách [tại đây](#) để đọc bản đầy đủ nhé

Cám ơn bạn đã quan tâm tới sách "**JAVASCRIPT TỪ CƠ BẢN TỚI NÂNG CAO**" của VNTALKING.

Mời bạn truy cập [Đặt sách "Javascript từ cơ bản tới nâng cao"](#) để đặt mua bản sách đầy đủ nhé. Đặc biệt ưu đãi cho các bạn học sinh - sinh viên.

PHỤ LỤC

CUỐN SÁCH CỦA VNTALKING

Đến thời điểm hiện tại, VNTALKING đã hoàn thành 3 dự án sách học lập trình. Sách học Javascript này là cuốn sách thứ 3 mà VNTALKING thực hiện.

Nếu bạn muốn tìm hiểu nhiều hơn về back-end hoặc front-end, mời bạn tham khảo cuốn sách:



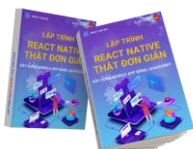
Lập trình Node.JS thật đơn giản

[Đọc sách](#)



Lập trình React thật đơn giản

[Đọc sách](#)



Lập trình React Native thật đơn giản

[Đọc sách](#)

Hi vọng trong thời gian tới, VNTALKING sẽ tiếp tục nhận được sự ủng hộ của độc giả. Thành công của bạn chính là động lực để VNTALKING cho ra nhiều cuốn sách với chất lượng tốt hơn nữa, đáp ứng nhu cầu học lập trình của mọi người.