JAVASCRIPT

JAVASCRIPT CƠ BẢN – HÀNH TRÌNH CỦA MỘT ANH

bởi | 18 bình luận | 8164 views

Mục lục nội dung

Kiến thức cơ bản về HTML và CSS có thể giúp bạn tạo được website đơn giản. Nhưng nếu bạn mong r website sinh động và phức tạp hơn, bạn cần Javascript. Javascript là ngôn ngữ lập trình đơn giản, như mạnh mẽ và phổ biến cho lập trình web. Các ứng dụng thường thấy ở Javascript có thể kể đến như:

- Tương tác với HTML và thay đổi nội dung và định dạng trên website dễ dàng.
- Tương tác với các hành động của người dùng như nhấn chuột, gõ phím...
- Xử lý và kiểm tra các dữ liệu trên form trước khi gửi về server.
- Tạo và truy xuất thông tin lưu trong cookie trên máy người dùng.
- Đóng vai trò như 1 ngôn ngữ lập trình phía server (sử dụng các framework như Node.js).

Có nhiều phương pháp để học Javascript, và tốt nhất là để người học được tự tay mày mò trong suốt trình tìm hiểu. Bài viết hôm nay sẽ áp dụng phương pháp đó, và tiếp cận nó theo một cách mới để mọ đọc dễ làm quen và hình dung hơn: gamification – trò chơi hóa nội dung bài học.

Hãy tưởng tượng bạn là nhân vật chính trong một game nhập vai, khởi đầu từ con số 0 tròn trĩnh để đ trở thành Anh Hùng trong cõi Javascript. Không gì hứa hẹn một hành trình bằng phẳng cả, nhưng đừn ngần khi định mệnh đã gọi tên!

INTRO CHAPTER!

Hãy tạo nên huyền thoại của riêng bạn về 1 Anh Hùng Javascript!

Javascript có thể được sử dụng dễ dàng với thẻ HTML `script` : chỉ cần đưa các câu lệnh Javascript và cặp thẻ hoặc nhúng 1 file Javascript bên ngoài.

```
<html>
<body>
<script type="text/javascript">
// Gõ code ở đây
</script>
</body>
</html>
```

```
<html>
<body>
<script type="text/javascript" src="đường-dẫn-đến-file-javascript.js"></scr
</body>
</html>
```

Tuy nhiên, trong hành trình Javascript cơ bản này, bạn không cần phải chèn code hay file Javascript và html và chạy file này. Bạn sẽ gõ code trực tiếp trên trình duyệt bằng công cụ Console. Để mở Console, bấm F12 và chọn tab Console ở khung công cụ lập trình được hiển thị, hoặc sử dụng phím tắt nhanh Ctrl+Shift+J (Chrome/Firefox).

Trường hợp nếu bạn không muốn sử dụng Console để thực hành mà muốn viết lên web thì hãy sử dụ phương thức xuất dữ liệu ra trang tên là document.write(). Ví dụ:

```
document.write("Tôi tên là Phúc!"); // Hiển thị chữ Tôi tên là Phúc ở websi
var name = "Phúc";
document.write(name); // Hiển thị chữ Phúc ở website.
```

Xong chưa nào? Hãy sẵn sàng chinh phục những thử thách để viết nên câu chuyện về Anh Hùng Javas của riêng bạn!

WELCOME TO CHAPTER 1!

Chúc mừng bạn đã bắt đầu hành trình Anh Hùng Javascript!

Quest 1: Kiến tạo Anh Hùng – Biến và kiểu dữ liệu

> *Việc lưu trữ dữ liệu là một trong những điều quan trọng nhất khi lập trình. Thông thường, các dữ sẽ được lưu giữ bằng các biến (variable) trong bộ nhớ. Đối với Javascript, bạn cần khai báo biến b khóa* `var` *trước khi gán dữ liệu cho biến đó.*

Mỗi Anh Hùng đều cần có 1 tên gọi để lưu danh sử sách, và tiện khoe với các em gái xinh đẹp trong q rượu :">. Hãy gõ lại đoạn code bên dưới vào Console và nhập tên vào hộp thoại được xổ ra.

NEW SKILL!

Hàm `prompt()` dùng để nhận nhập liệu từ người dùng và trả về dạng chuỗi kí tự.

Ví dụ làm quen với biến

```
var ten = prompt('Xin chào! Hãy gọi tên Anh Hùng của bạn:');
ten;
```

Hàm `prompt()` là hàm viết sẵn của hệ thống để nhận nhập liệu từ người dùng (ta sẽ tìm hiểu thêm v phần sau). Kết quả của hàm sẽ trả về bất cứ gì người dùng nhập vào ở hộp thoại. Nếu bạn muốn sử d quả đó, bạn cần phải lưu trữ nó bằng 1 **biến**. Ở dòng thứ 1, ta đã khai báo biến `ten` và gán cho nó k trả về của hàm `prompt()`. Sau này, bất cứ khi nào bạn cần, bạn chỉ cần gọi biến cần thiết để lấy dữ li biến đó giữ. Đó là lý do khi bạn gọi lên biến `ten` ở dòng 2, bạn sẽ thấy lại giá trị mà bạn vừa nhập. S chưa 😀

Demo: http://jsfiddle.net/6QhmR/1/

Biến

Đoạn code trên là 1 ví dụ khi sử dụng biến. Hãy nghĩ 1 biến như 1 ngăn tủ được đặt tên, và tên biến là ngăn tủ, giá trị của biến là vật dụng trong ngăn tủ. Bạn có thể chứa bất kì thứ gì bạn muốn trong ngăn và khi cần tìm lại, bạn chỉ cần tra đúng tên ngăn tủ mà bạn cần.

Biến có 2 phần, là tên biến và giá trị của biến. Tên biến rất đơn giản, khi bạn có thể đặt tên biến tùy ý v chữ cái hoa hay thường, các con số và dấu gạch chân (_). Còn với giá trị của biến thì chúng ta cần để ý về kiểu dữ liệu. Ở ví dụ trên, ta đã dùng biến `ten` để chứa 1 hàm prompt và chứa dữ liệu kiểu `strin` Javascript hỗ trợ nhiều kiểu dữ liệu, nhưng ở mức cơ bản bạn sẽ cần nắm vững các kiểu dữ liệu sau:

- **String** dùng để chứa chuỗi kí tự và phải được bao quanh bởi cặp nháy đơn ('...') hay nháy đôi ("...
- **Number** dùng để chứa dữ liệu kiểu số nguyên, số thập phân và không nằm trong cặp nháy đơn nháy đôi.
- **Boolean** là kiểu logic, chỉ có 2 giá trị là đúng (true) hoặc sai (false).
- **Object** là một đối tượng nói chung với các thuộc tính và phương thức riêng. Ta sẽ tìm hiểu về Ob phần 7.
- **Array** là mảng, dùng để chứa tập hợp nhiều biến. Phần 8 sẽ giúp bạn hiểu rõ hơn.

Thao tác với biến

Để chứa đồ trong tủ, bạn cần phải tìm 1 ngăn tủ rỗng, dán tên cho ngăn tủ đó (**khai báo biến**) và đặt đó vào bên trong (**gán giá trị cho biến**).

```
01  // Khai báo với từ khóa var
02  var level;
03        18
04  // Sau khi khai báo, dùng dấu = để gán giá trị
05  // Giá trị kiểu số (Number)
06  level = 1;
07
08  // Khai báo và gán 1 giá trị ngay lúc đó gọi là khởi tạo
09  // Giá trị kiểu boolean
10  var male = true;
11
```

```
// Giá trị kiểu chuỗi kí tự (String)
var phancap = "Anh Hùng";

/* ============================================================
* | HÃY TỰ TAY CODE TRƯỚC KHI KÉO XUỐNG XEM CODE MẪU! |
* ============================================================
* THỬ THÁCH LEVEL 1
* -----------------
* Khởi tạo thêm 3 biến để chứa thông tin về cấp độ (level),
* điểm kinh nghiệm (xp) và điểm kĩ năng (skillPoints).
* Giá trị khởi tạo của từng biến như sau:
* - Cấp độ: 1
* - Điểm kinh nghiệm: 0
* - Điểm kĩ năng: 5
* ============================================================
* | HÃY TỰ TAY CODE TRƯỚC KHI KÉO XUỐNG XEM CODE MẪU! |
* ============================================================
*/
```

Sau khi đặt xong biến, bạn có thể thử in ra bằng cách sử dụng hàm alert() để xem giá trị. Ví dụ:

```
var name = "Thachpham.com";

alert(name);
```

+ expand source

LEVEL UP! Chúc mừng bạn lên cấp 2! **XP +150**

Quest 2: Sức mạnh tính toán – Các toán tử cơ bản

Javascript hỗ trợ đầy đủ các toán tử cơ bản cộng trừ nhân chia. Ngoài ra, bạn sẽ làm quen với 2 toán t ++ (tăng giá trị của biến kiểu Number lên 1.0 đơn vị) và — (giảm giá trị của biến kiểu Number xuống 1 vị). Thứ tự tính toán (trong trường hợp không sử dụng dấu ngoặc tròn để gom nhóm ưu tiên) là nhân trước, cộng trừ sau và từ trái sang phải.

```
var level = 1;
var xp = 0;
var skillPoints = 5;

level = level + 1; // hoặc tương đương là level++;
xp = 150 * (level - 1) + 200; // sử dụng dấu ngoặc tròn để gom nhóm ưu tiên
skillPoints = skillPoints + xp / 10;
```

Ngoài ra, Javascript sử dụng toán tử + để nối các chuỗi kí tự.

```
// Nối 3 chuỗi đơn giản
"Javascript" + " " + "căn bản";

// Nối 1 chuỗi và 1 biến String
ten + " sẽ là Anh Hùng Javascript";

// Nối 1 chuỗi với 1 biến Number
"Cấp độ hiện tại của bạn là " + level;

/* ============================================================
* | HÃY TỰ TAY CODE TRƯỚC KHI KÉO XUỐNG XEM CODE MẪU! |
```

```
* ========================================================
* THỬ THÁCH LEVEL 2
* -----------------
* Khởi tạo thêm 3 biến là strength, agility và intel.
* Sử dụng giá trị của điểm kĩ năng (skillPoints) phía trên
* và tính toán giá trị khởi tạo của các biến:
* - strength bằng 60% điểm kĩ năng cộng với 10% của điểm kinh nghiệm.
* - agility bằng 20% điểm kĩ năng cộng với 20% của điểm kinh nghiệm.
* - intel bằng 20% điểm kĩ năng cộng với 1000% của cấp độ.
* Sau khi tính toán xong, hãy xuất các chỉ số theo cú pháp:
* "Chỉ số strength của bạn là " + chỉ số
* ========================================================
* | HÃY TỰ TAY CODE TRƯỚC KHI KÉO XUỐNG XEM CODE MẪU! |
* ========================================================
*/
```

+ expand source

LEVEL UP! Chúc mừng bạn lên cấp 3! **XP +200**

Quest 3: Sức mạnh logic – Các toán tử so sánh

Trên con đường hành hiệp, Anh Hùng sẽ gặp rất nhiều cám dỗ và lựa lọc. Chính vì thế, Anh Hùng cần sánh chính xác và sử dụng sức mạnh logic để nhận ra chân tướng đúng/sai của mọi sự vật, sự việc.

Trong Javascript, kết quả của mọi so sánh đều là 1 biến kiểu boolean: hoặc là true, hoặc là false. Để so giữa các biến hay các biểu thức, bạn có thể sử dụng `===` cho so sánh bằng, `!==` cho so sánh không `<` và `>` cho so sánh hơn kém.

NEW SKILL!

Hàm `console.log()` : có thể được dùng để in giá trị của một biến.

```
// So sánh 2 biến String
var test1 = ("Javascript" === "JAVAscript");
var test2 = ("Javascript" === "Javascript");
var test3 = ("Javascript" !== "javascript");
console.log(test1);
console.log(test2);
console.log(test3);

// So sánh 2 biến Number
test1 = (1 > 2);
test2 = (0 < -5);
test3 = (1.5 === 1.4999);
console.log(test1);
console.log(test2);
console.log(test3);

// So sánh 2 biểu thức Number
test1 = (1 - 2 + 3 - 4) <= 5;
test2 = (1 - 2 + 3 - 4)/5 >= 1;
console.log(test1);
console.log(test2);

// So sánh 1 biến Number và 1 biến String
test3 = 15 === "15";
```

```
25 console.log(test3);
```

LEVEL UP! Chúc mừng bạn lên cấp 4! **XP +300**

Quest 4: Lựa chọn định mệnh – Điều kiện và rẽ nhánh

Chúc mừng bạn đã nắm vững các kĩ năng cơ bản, và đã đến lúc để lựa chọn phân cấp Anh Hùng. Bạn lựa chọn giữa 3 phân cấp với 3 đặc điểm về chỉ số khác nhau.

Các lựa chọn trong Javascript sẽ dẫn đến các điều kiện rẽ nhánh khác nhau. Để tạo ra các lựa chọn tro Javascript, bạn sẽ sử dụng từ khóa `if` hoặc `if...else` . Các lựa chọn có thể lồng ghép với nhau để các luồng rẽ nhánh phức tạp. Tuy nhiên, cần cẩn trọng vì quá nhiều luồng rẽ nhánh sẽ khiến đoạn cod hiểu và khó bảo trì về sau.

```
01 /* Câu lệnh if thông thường
02 * ------------------------
03 * Biểu thức
04 * Khi biểu thức logic trong ngoặc
05 */
06 if (level < 4) {
07 console.log('Bạn chưa đạt cấp độ tối thiểu để tham gia thử thách.');
08 }
09
10 // Câu lệnh if...else
11 if (level >= 4) {
12 console.log('Bạn đã đủ cấp độ để tham gia thử thách level 4.');
13 } else {
14 console.log('Bạn cần đạt ít nhất level 4 để tham gia thử thách.');
15 }
16
17 // Câu lệnh if...else if...else
18 if (level > 4) {
19 console.log('Bạn dư sức vượt qua thử thách level 4.');
20 } else if (level === 4) {
21 console.log('Bạn đủ sức vượt qua thử thách level 4.');
22 } else {
23 console.log('Về uống sữa thêm cho mau lớn nha cưng!');
24 }
25
26 // Ở phần 1, bạn đã khai báo biến <code>ten</code>
27 // và sử dụng để giữ giá trị về tên Anh Hùng của bạn
28 // Ta sẽ kiểm tra xem bạn có thể gọi lại chính xác không
29 // để thưởng hoặc phạt theo kết quả
30 var kiemTraTen = prompt('Tên Anh Hùng của bạn là gì?');
31 if (kiemTraTen === ten) {
32 console.log('Chính xác! Bạn có trí nhớ rất tốt! intel +5!');
33 intel = intel + 5;
34 } else {
35 console.log('Tệ thật! Tên chính mình mà không nhớ sao?! intel -5!');
36 intel = intel - 5;
37 }
38
39 /* ========================================================
40 * | HÃY TỰ TAY CODE TRƯỚC KHI KÉO XUỐNG XEM CODE MẪU! |
41 * ========================================================
42 * THỬ THÁCH LEVEL 4
43 * -----------------
```

```
* Sử dụng hàm prompt('Bạn muốn chọn phân cấp nào? a: Chiến
* Binh | b: Sát Thủ | c: Thợ Săn') để yêu cầu người
* dùng nhập vào 1 kí tự hoặc là a, hoặc b, hoặc c. Giá trị
* nhập vào sẽ được xử lý như sau:
* 1. Nếu là kí tự a:
* - Cập nhật phân cấp Anh Hùng (biến phancap): Chiến Binh.
* - Điều chỉnh chỉ số: strength +10, agility -10
* 2. Nếu là kí tự b:
* - Cập nhật phân cấp Anh Hùng (biến phancap): Sát Thủ.
* - Điều chỉnh chỉ số: strength -10, agility +5, intel +5
* 3. Nếu là kí tự c:
* - Cập nhật phân cấp Anh Hùng (biến phancap): Thợ Săn.
* - Điều chỉnh chỉ số: strength -10, intel +10
* 4. Nếu không phải 3 trường hợp trên
* - Không cập nhật và điều chỉnh gì cả.
* ========================================================
* | HÃY TỰ TAY CODE TRƯỚC KHI KÉO XUỐNG XEM CODE MẪU! |
* ========================================================
*/
```

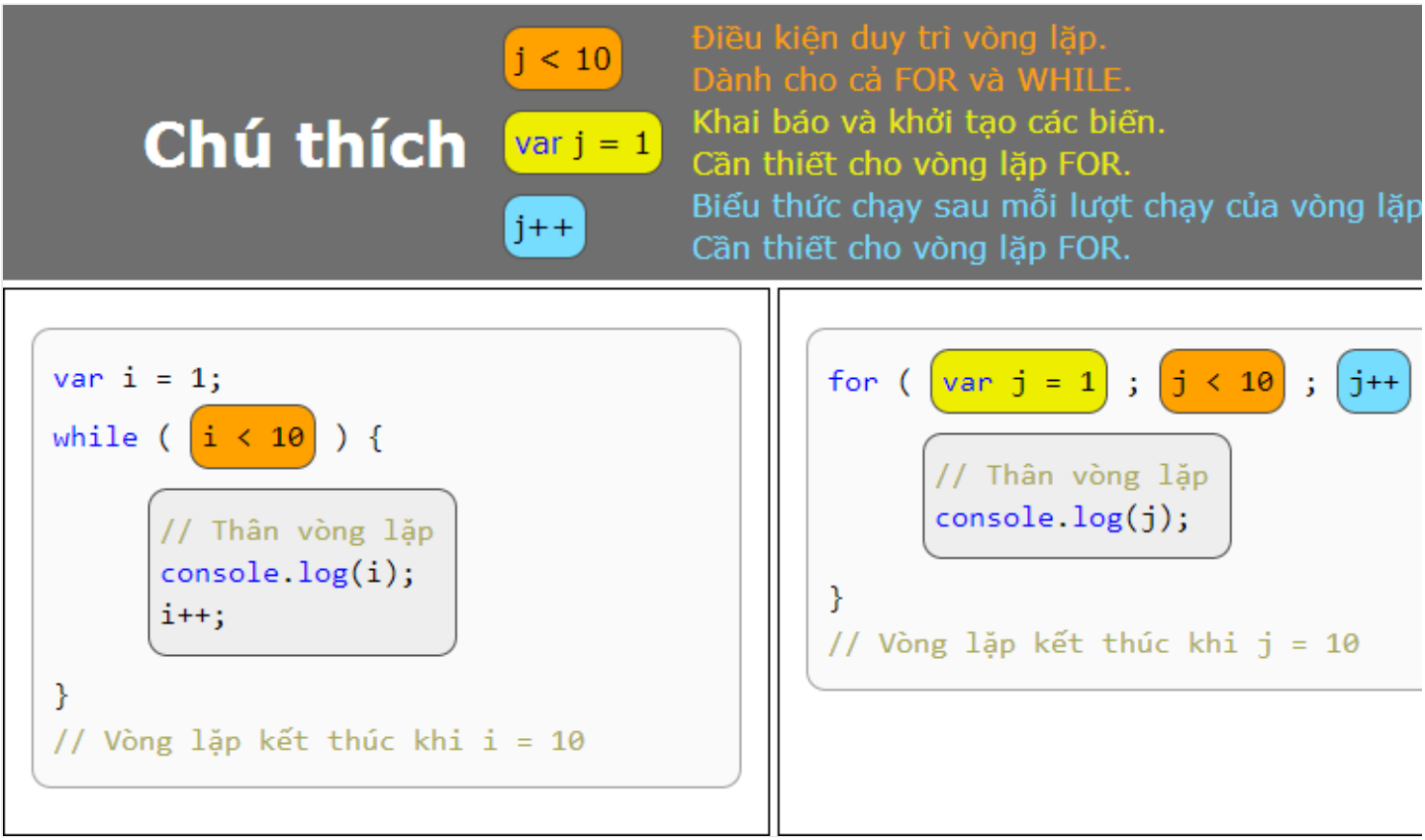
+ expand source

LEVEL UP! Chúc mừng bạn lên cấp 5! **XP +400**

Quest 5: Đấu trường Sinh Tử – Vòng lặp

Anh Hùng nào cũng phải trải qua quá trình tập luyện và đấu tranh gian khổ để đạt đến vinh quang. Q đó đòi hỏi những nỗ lực lặp đi lặp lại đến khi đạt được một ý đồ, một mục đích nào đó.

Việc lặp các thao tác trong lập trình được gọi là vòng lặp, và là một trong những **thành phần quan tr nhất của lập trình**. Vòng lặp sẽ có 1 điều kiện để duy trì và khi điều kiện đó không được đảm bảo thì lặp sẽ kết thúc. Đối với Javascript cơ bản, có 2 cách để tạo vòng lặp: vòng lặp `for` và vòng lặp `while` đều bao gồm 2 thành phần: điều kiện để duy trì vòng lặp, và thân vòng lặp chứa các thao tác sẽ được




```
01  // Ví dụ vòng lặp while
02  var i = 1;
03  while (i < 10) {
04  console.log(i);
05  i++;
06  }
07  console.log(i);
08
09  // Ví dụ vòng lặp for
10  for (var j = 1; j < 10;j++) {
11  console.log(j);
12  }
13  console.log(j);
```

BATTLE! Bạn chạm trán 1 con rồng già, và buộc phải giết nó để vượt qua

NEW SKILL!

Hàm `Math.random()` : dùng để tạo giá trị ngẫu nhiên từ 0 đến 1. Để tạo xác suất ngẫu nhiên theo %, sử dụng (Math.random() * 100).

NEW SKILL!

Hàm `alert()` : dùng để xuất một chuỗi kí tự qua hộp thoại thông báo. Có thể dùng để debug hay xu thông tin cho người dùng.

```
01  /* ==========================================================
02  * Trước khi vào trận chiến, cần xem lại các chỉ số của bạn
03  * và xem xét các thông tin của đối thủ.
04  * ==========================================================*/
05  level = 5; // Đảm bảo bạn sẽ ở cấp độ 5
06  var health=level*100;// Khi lượng máu <= 0 thì bạn sẽ chết
07  console.log(strength); // Quyết định mức sát thương
08  console.log(agility); // Quyết định xác suất né đòn
09  console.log(intel); // Quyết định xác suất chí mạng
```

```
10
11  /* ========================================================
12  * Rồng Già
13  * --------
14  * Chậm chạp và yếu ớt hơn so với đồng loại, nhưng vẫn rất
15  * nguy hiểm với khả năng phun lửa có thể gây sát thương chí
16  * mạng rất cao
17  * ========================================================*/
18  var dragonHealth = 600;
19  var dragonStr = 100;
20  var dragonAgi = 15;
21  var dragonInt = 80;
22
23  /* ========================================================
24  * | HÃY TỰ TAY CODE TRƯỚC KHI KÉO XUỐNG XEM CODE MẪU! |
25  * ========================================================
26  * THỬ THÁCH LEVEL 5
27  * -----------------
28  * Viết trò chơi Diệt Rồng theo các bước sau:
29  * 1. Mỗi lượt chơi, Anh Hùng và Rồng lần lượt tấn công nhau.
30  * => GỢI Ý: Sử dụng vòng lặp while.
31  * 2. Anh Hùng tấn công trước, Rồng tấn công sau.
32  * 3. Trò chơi kết thúc khi một phe hết máu (health < 0 hoặc dragonHealth <
33  * => GỢI Ý: đặt 1 biến boolean để kiểm tra.
34  * 4. Sử dụng hàm (Math.random() * 100) để lấy xác suất ngẫu nhiên theo %.
35  * 5. Rồng né đòn thành công khi % xác suất ngẫu nhiên bé hơn dragonAgi.
36  * => GỢI Ý: so sánh dragonAgi với (Math.random() * 100).
37  * 6. Anh Hùng né đòn thành công khi % xác suất ngẫu nhiên bé hơn agility.
38  * => GỢI Ý: so sánh agility với (Math.random() * 100) (khác ở 5).
39  * 7. Sát thương do Anh Hùng gây ra bằng với chỉ số strength.
40  * => GỢI Ý: khi Anh Hùng đánh trúng, giảm máu Rồng theo giá trị sát thương.
41  * 8. Sát thương do Rồng gây ra bằng với chỉ số dragonStr.
42  * => GỢI Ý: khi Rồng đánh trúng, giảm máu Anh Hùng theo giá trị sát thương.
43  * 9. Anh Hùng gây sát thương chí mạng khi % xác suất ngẫu nhiên lớn hơn int
44  * => GỢI Ý: so sánh intel với 1 giá trị (Math.random() * 100) (khác ở 5 và
45  * 10. Rồng gây sát thương chí mạng khi % xác suất ngẫu nhiên lớn hơn dragon
46  * => GỢI Ý: so sánh dragonInt với 1 giá trị (Math.random() * 100) (khác ở 5
47  * 11. Sát thương chí mạng sẽ gây thêm 200% sát thương.
48  * => GỢI Ý: nếu gây sát thương chí mạng, giảm máu đi 2 lần sát thương.
49  *
50  * ========================================================
51  * | HÃY TỰ TAY CODE TRƯỚC KHI KÉO XUỐNG XEM CODE MẪU! |
52  * ========================================================
53  */
```

+ expand source

BATTLE WON! Chúc mừng bạn đã đạt danh hiệu "Dũng sĩ diệt Rồng"!

LEVEL UP! Chúc mừng bạn lên cấp 6! **XP +600**

Quest 6: Tạo và gọi Hàm

Hàm là 1 đoạn code riêng biệt có thể sử dụng nhiều lần. Hàm có thể nhận **tham số** từ biến và **trả kết** để gán cho biến.

Trước khi sử dụng, hàm cần được khởi tạo với từ khóa `function tênHàm(thamSố1, thamSố2)` . Mọ
mà hàm cần sử dụng từ bên ngoài hàm được truyền vào giữa hai ngoặc, và được gọi là tham số. Để bà
trả về 1 giá trị nào đó, bạn cần sử dụng từ khóa `return tênBiếnMuốnTrảVề;` . `return` sẽ kết thúc
ngay lập tức, nên nếu có câu lệnh nào phía sau, câu lệnh đó sẽ không được thực thi.

Một lưu ý khi sử dụng hàm là đừng ôm đồm quá nhiều xử lý trong một hàm. 1 hàm xử lý chuỗi không
phải biết đầy đủ cắt chuỗi, đảo chuỗi, nhân đôi chuỗi.., mà chỉ cần thực hiện tốt 1 chức năng thôi. Đây
thói quen lập trình tốt, sẽ giúp code đơn giản, dễ hiểu và dễ bảo trì qua thời gian.

```
// Lấy ví dụ trò chơi diệt Rồng ở phần 5
// Tạo hàm gây sát thương để giảm máu
// Tham số truyền vào: biến Number satThuong
// Kết quả trả về: biến Number luongMauHienTai
function gaySatThuong(luongMauHienTai, satThuong) {
luongMauHienTai = luongMauHienTai - satThuong;
return luongMauHienTai;
}

// Để sử dụng hàm, bạn cần phải gọi hàm
var luongMau = 100;
var satThuong = 49;
var luongMauConLai = gaySatThuong(luongMau, satThuong);
console.log(luongMauConLai);

/* ============================================================
* | HÃY TỰ TAY CODE TRƯỚC KHI KÉO XUỐNG XEM CODE MẪU! |
* ============================================================
* THỬ THÁCH LEVEL 6
* -----------------
* Viết lại trò chơi Diệt Rồng, có sử dụng hàm để:
* 1. Tính toán khả năng né đòn.
* 2. Tính toán lượng máu còn lại sau khi bị sát thương.
*
* ============================================================
* | HÃY TỰ TAY CODE TRƯỚC KHI KÉO XUỐNG XEM CODE MẪU! |
* ============================================================
*/
```

LEVEL UP! Chúc mừng bạn lên cấp 7! **XP +700**

Quest 7: Đối tượng

Đối tượng Javascript cũng giống như một vật thể thực ngoài đời: có các tính chất (thuộc tính) và khả r
(phương thức/hàm). Hiểu một cách khác, đối tượng trong Javascript là tập hợp của các thuộc tính và c
phương thức (về bản chất cũng là hàm) bên trong.

Có thể xem đối tượng như 1 biến đặc biệt trong Javascript, được khai báo trong cặp ngoặc nhọn `{..`
dụng dấu `.` để truy xuất các thuộc tính và phương thức bên trong.

```
// Cú pháp khai báo 1 đối tượng
var RongGia = {
// Khai báo các thuộc tính
name: 'Rồng Già',
```

```
phanLop: 'Rồng',
age: 6969,
health: 600,
strength: 100,
agility: 15,
intel: 80,

// Khai báo các phương thức
bay: function() {
console.log('Flappy Dragon!');
},
phunLửa: function() {
console.log('Rồng phun lửa');
},
ngủ: function() {
console.log('Zzz... Zzz...');
}
};

// Sử dụng các thuộc tính như biến thông thường
console.log(RongGia.name);
console.log(RongGia.phanLop);
console.log(RongGia.age);

// Sử dụng phương thức như gọi hàm bình thường
RongGia.bay();
RongGia.phunLửa();
RongGia.ngủ();

// Bạn cũng có thể thêm các thuộc tính và phương thức
// sau khi đã khởi tạo đối tượng
RongGia.level = 10;
RongGia.chết = function() {
console.log('Rồng lên bàn thờ!');
};

/* ========================================================
* | HÃY TỰ TAY CODE TRƯỚC KHI KÉO XUỐNG XEM CODE MẪU! |
* ========================================================
* THỬ THÁCH LEVEL 7
* -----------------
* Khai báo một đối tượng AnhHung để chứa các thông tin về
* Anh Hùng của bạn:
* 1. Các thuộc tính: tên, phân lớp, cấp độ, các chỉ số..
* 2. Các phương thức ví dụ như đi, chạy, nhảy, ngủ..
*
* ========================================================
* | HÃY TỰ TAY CODE TRƯỚC KHI KÉO XUỐNG XEM CODE MẪU! |
* ========================================================
*/
```

LEVEL UP! Chúc mừng bạn lên cấp 8! **XP +800**

Quest 8: Mảng

Mảng là tập hợp nhiều phần tử, với mỗi phần từ là 1 biến được sắp xếp theo thứ tự và có đánh số chỉ (đánh số từ 0) cho từng biến để tiện truy xuất. Mỗi biến trong mảng có thể mang bất kì kiểu dữ liệu nà

Number, String đến cả đối tượng và mảng khác. Các biến trong mảng được đánh số tuần tự tăng dần nên biến cuối cùng trong mảng sẽ có số chỉ mục bằng tổng số lượng biến trừ đi 1.

Mảng được khai báo bên trong cặp ngoặc vuông `var tênMảng = [biến1, biến2];` . Các biến bên cũng được truy xuất bằng ngoặc vuông theo cú pháp `tênBiến[sốChỉMục]` . Số lượng các phần tử bê mảng có thể được truy xuất nhanh bằng thuộc tính `tênBiến.length` (vì bản chất Mảng Javascript cí đối tượng). Ngoài ra, để thêm phần tử vào 1 mảng đã được khởi tạo, bạn cần sử dụng phương thức `tênBiến.push()` .

```
// Khai báo và khởi tạo mảng ví dụ
var phanlopAnhHung = ['Chiến Binh','Sát Thủ','Thợ Săn'];
console.log(phanlopAnhHung.length);
console.log(phanlopAnhHung[0]);
console.log(phanlopAnhHung[1]);
console.log(phanlopAnhHung[phanlopAnhHung.length - 1]);
// hàm console.log() rất hữu ích để xem toàn bộ phần tử trong mảng
console.log(phanlopAnhHung);

// Mảng rỗng
var danhHieu = [];

// Thêm phần tử mới
danhHieu.push('Dũng sĩ diệt rồng');
console.log(danhHieu);
phanlopAnhHung.push('Anh Hùng');
console.log(phanlopAnhHung.length);
console.log(phanlopAnhHung[phanlopAnhHung.length - 1]);

// Truy xuất từng phần tử trong mảng với vòng lặp
var soPhanTu = phanlopAnhHung.length;
for (var i=0; i < soPhanTu; i++) {
console.log(phanlopAnhHung[i ]);
}

/* ========================================================
* | HÃY TỰ TAY CODE TRƯỚC KHI KÉO XUỐNG XEM CODE MẪU! |
* ========================================================
* THỬ THÁCH LEVEL 8
* -----------------
* Hãy sử dụng vòng lặp và hàm prompt() để yêu cầu người dùng
* nhập vào câu trả lời cho 3 câu hỏi sau:
* 1. Bạn có người yêu chưa?
* 2. Bạn có thích ăn rau dền không?
* 3. Người yêu bạn có thích ăn rau dền không?
* Sau đó, lưu tất cả câu trả lời vào 1 mảng và xuất ngược lại
* từng câu trả lời bằng hàm alert().
*
* ========================================================
* | HÃY TỰ TAY CODE TRƯỚC KHI KÉO XUỐNG XEM CODE MẪU! |
* ========================================================
*/
```

+ expand source

CHAPTER 1 COMPLETE! Chúc mừng bạn đã trở thành Tân Anh Hùng Javascript!

Lời Kết

Con đường cách mạng, còn lắm gian truân! Vì thế, Javascript dù rất đơn giản để học, nhưng để sử dụr thục phải trải qua thử nghiệm và thực hành nhiều. Với kiến thức cơ bản mà bài viết này cung cấp, mìn rằng bạn sẽ có nền tảng tốt để học nhanh và phát triển nhanh kĩ năng Javascript của bản thân. Nếu có mắc, ý kiến hay cần trợ giúp gì, hãy cùng thảo luận ở phần bình luận bên dưới nhé.

Đánh giá nội dung này

JAVASCRIPT JAVASCRIPT CĂN BẢN NEWBIE

18 bình luận

bài trước
SỬ DỤNG GOOGLE TRANSLATOR TOOLKIT ĐỂ VIỆT HÓA WORDPRESS

b
TẢI THEME SWIFT PREMIUM M

CÓ THỂ BẠN QUAN TÂM

18

GIỚI THIỆU SERIE HỌC HTML [HỌC HTML] CHÈN TẬP TIN KỸ [HỌC HTML] THẺ TẠO LI

0

Article Rating

Login

Tham gia thảo luận

18 COMMENTS mới nh

6 WEBSITES HAY ĐỂ HỌC JAVASCRIPT MIỄN PHÍ BẰNG TIẾNG ANH – Blogs of Digitech Solutio

5 tháng trước

[...] dù blog mình đã có bài Học Javascript căn bản khá chi tiết và vui nhộn, nhưng nhiêu đó thật sự đủ để bạn có thể [...]

0 Trả lời

Công cụ bí mật để tạo landing page sát thủ – Thông Thiên Phong

8 tháng trước

18

[...] bản từ ThachPham.com: Vào ngay >> 👉 Học Javascript cơ bản từ ThachPham.com: Từ từ rồi vào xong hai cái trên đã >> 👉 Học PHP cơ bản cho WordPress: Vào thôi [...]

0 Trả lời

6 WEBSITES HAY ĐỂ HỌC JAVASCRIPT MIỄN PHÍ BẰNG TIẾNG ANH – Bloghocit

1 năm trước

[...] dù blog mình đã có bài Học Javascript căn bản khá chi tiết và vui nhộn, nhưng nhiêu đó thật sự đủ để bạn có thể [...]

0 Trả lời

Bài 26: KẾT THÚC SERIE CSS CƠ BẢN – tutorialspoint.vn

2 năm trước

[...] nếu có thời gian, mình khuyến khích bạn học thêm Javascript cơ bản và jQuery để biết cách áp c nó vào website để làm các hiệu ứng/sự [...]

0 Trả lời

20 IT Blogger Việt bạn không nên bỏ qua - updated 2017 - ITviec blog

2 năm trước

[...] JavaScript cơ bản – Hành trình của một anh hùng [...]

0 Trả lời

19 IT Blogger Việt bạn không nên bỏ qua - ITviec blog

2 năm trước

[...] JavaScript cơ bản – Hành trình của một anh hùng [...]

0 Trả lời

Hữu

2 năm trước

hay quá! đang định viết code mà không biết viết sao

0 Trả lời

18

nhattnlc

2 năm trước

cái console.log có tác dụng in ra đâu vậy mn ?

0 Trả lời

Thạch Phạm Admin

Reply to nhattnlc 2 năm trước

Nó in ra bảng console của trình duyệt thôi bạn.

+ 2 − Trả lời

Quyết

3 năm trước

Kiến thức rất bổ ích. Xin cảm ơn Ad !

+ 0 − Trả lời

Tuyen

3 năm trước

hay qua

+ 0 − Trả lời

Trung kiên

3 năm trước

cho e hỏi em có đoạn code như sau để chạy slide ảnh:và cho em hỏi làm như thế nào để khi đặt chu
các ảnh trong mảng dừng lại ,và di chuột sang vùng khác các ảnh lại chạy bình thường ạ?

```
var count = 0;
var mangAnh2 = [];
mangAnh2[0]=”banhngon.jpg”;
mangAnh2[1]=”banhngon1.jpg”;
mangAnh2[2]=”banhngon2.jpg”;
function next2(){
count++;
if(count > 2){
count = 0;
}document.images[‘slide2’].src = mangAnh2[count];
}
setInterval(“next2()”,1000);
```

+ 0 − Trả lời

Tony

3 năm trước

Đọc xong em hoa hết cả mắt 😕

+ 0 − Trả lời

Vũ Min

3 năm trước

Sao em thấy khó thế nhỉ ._.

+ 0 − Trả lời

dũng

3 năm trước

ad ơi em có 1 biến là x2 = flaker.mX vậy hiểu dấu chấm trong này là như thế nào ạ ?

+ 0 − Trả lời

phong

3 năm trước

cam? on ban da chia se kien thuc nay cho moi nguoi. mong ban co the them nung bai viet hay va y hon nua~!

+ 0 − Trả lời

Duy

3 năm trước

phần Quest 6, 7 không có Expand source hả anh Phạm

+ 0 − Trả lời

> **TUi**
>
> Reply to Duy 3 năm trước
>
> ko
>
> + 0 − Trả lời

18

Bạn bè & Đối tác

Trần Ngọc Chính Lập trình cho trẻ em – Sumato Đỗ Trung Quân Trung Đức Blog Quản Trị Hệ Thống Mạng Lại Văn Đức Ninh Đôn Đinh Trang Canh Me

LÊN ĐẦU TRANG