



Web Front End cho người mới

Phần 7 – Javascript cơ bản: Variable, Function

Giới thiệu về ECMAScript 6 (ES6) – ECMAScript 2015

- Bổ sung thêm một số tính năng mới so với Javascript phiên bản cũ (ES5): let, const, ...
- Hiện đã được hỗ trợ trên hầu hết các trình duyệt phổ biến.

				
Chrome 58	Edge 14	Firefox 54	Safari 10	Opera 55
Jan 2017	Aug 2016	Mar 2017	Jul 2016	Aug 2018



Sử dụng let và const thay cho var

- Để khai báo biến thì sử dụng **let**.
- Để khai báo hằng số (constant - biến không thay đổi giá trị trong toàn bộ chương trình) thì sử dụng **const**.
- Với những giá trị khó nhớ, khó hiểu hoặc dùng đi dùng lại nhiều lần thì cũng nên dùng **const**. Trong trường hợp này biến const thường được đặt theo kiểu viết hoa toàn bộ, ví dụ:

```
const PRIMARY_COLOR = '#FF7F00';
```



Khác biệt giữa var và let

var không có block scope

```
for (var i = 0; i < 10; i++) {  
    // ...  
}  
alert(i);
```

let có block scope

```
for (let i = 0; i < 10; i++) {  
    // ...  
}  
alert(i);
```


Khác biệt giữa var và let

Khai báo biến bằng **var** có hoisted

```
function sayHi() {  
  phrase = 'Hello';  
  alert(phrase);  
  var phrase;  
}  
sayHi();
```

Khai báo biến bằng **let** không hoisted

```
function sayHi() {  
  phrase = 'Hello';  
  alert(phrase);  
  let phrase;  
}  
sayHi();
```



Template literals (Template strings)

- Khai báo chuỗi sử dụng ký tự *back-tick* ``` thay cho ký tự *ngoặc đơn* hay *ngoặc kép*.
- Khi khai báo chuỗi kiểu này có thể viết chuỗi trong nhiều dòng và có thể dùng dấu *ngoặc đơn* hay *ngoặc kép* trong chuỗi thoải mái không cần *escape character*.
- Có thể dùng các biến, biểu thức ngay trong chuỗi với cú pháp như sau: ``string text ${expression} string text``.



Function default values

```
function sendEmail(from, message = "Không có nội dung") {  
    alert(from + ": " + message);  
}  
  
sendEmail("Robin");
```



Arrow function

- Tạo hàm với cú pháp ngắn gọn hơn.
- Không có **this** và **arguments**.
- Không gọi được với **new**.

```
let sum = function(a, b) {  
  return a + b;  
}
```



```
let sum = (a, b) => a + b;
```


Callback function

```
function ask(question, yes, no) {  
  if (confirm(question)) {  
    yes();  
  } else {  
    no();  
  }  
}
```

```
function showOk() {  
  alert("Bạn đã chọn đồng ý");  
}  
  
function showCancel() {  
  alert("Bạn đã hủy chọn");  
}  
  
ask("Bạn đồng ý hay không?", showOk, showCancel);
```

Bài tập thực hành



Đâu là tên biến hợp lệ?

- | | |
|----------------|--------------|
| 1. \$usd | 9. let |
| 2. test123 | 10. variable |
| 3. userName | 11. Object |
| 4. docker7890- | 12. my-name |
| 5. _self | 13. can't |
| 6. 1a | 14. error |
| 7. string | 15. Person |
| 8. first_name | 16. numBer |

Lệnh sau hiển thị gì ra màn hình?

```
let a = 1;  
let b = 22;  
let name = "John";
```

1. `alert(b + a);`
2. `alert(c);`
3. `alert(a + name);`
4. `alert(a + name + b);`
5. `alert(a + b + name);`
6. `alert(name + a);`
7. `alert(name + a + b);`
8. `alert(name + (a + b));`
9. `alert(`Hello ${name}`);`
10. `alert(`${name} + 1`);`
11. `alert(`${name + 1}`);`
12. `alert(`${name + a}`);`
13. `alert(`a + b = ${a + b}`);`
14. `alert(`a + b = ${1 + 2}`);`
15. `alert(`1 + 2 = ${1 + 2}`);`
16. `alert(`1 + 2 = ${a + b}`);`

Kết quả của những biểu thức sau?

1. `"" + 1 + 0`

2. `"" - 1 + 0`

3. `true + false`

4. `6 / "3"`

5. `"2" * "3"`

6. `4 + 5 + "px"`

7. `"$" + 4 + 5`

8. `"4" - 2`

9. `"4px" - 2`

10. `7 / 0`

11. `"-9" + 5`

12. `"-9" - 5`

13. `null + 1`

14. `undefined + 1`

15. `typeof(7) + 1`

16. `typeof "9" + 5`

Viết Function thực hiện những thao tác sau (gọi từ Console)

1. Thông báo ra màn hình: Xin chào các bạn.
2. Thông báo ra màn hình: Xin chào X (với X là một chuỗi bất kỳ, nhập vào tham số khi gọi hàm).
3. Thông báo ra màn hình: Xin chào "X" (với X là một chuỗi bất kỳ, nhập vào tham số khi gọi hàm).
4. Tính tổng 2 số (được truyền vào khi gọi hàm).
5. Tính bình phương 1 số (được truyền vào khi gọi hàm).
6. Tính thế kỷ của 1 năm (số năm được truyền vào khi gọi hàm, là 1 số nguyên dương).

Bài tập về nhà



Bài tập về nhà

- Ôn lại lý thuyết Javascript cơ bản đến phần **Function**.
- Học và làm bài tập phần **Vòng lặp**.