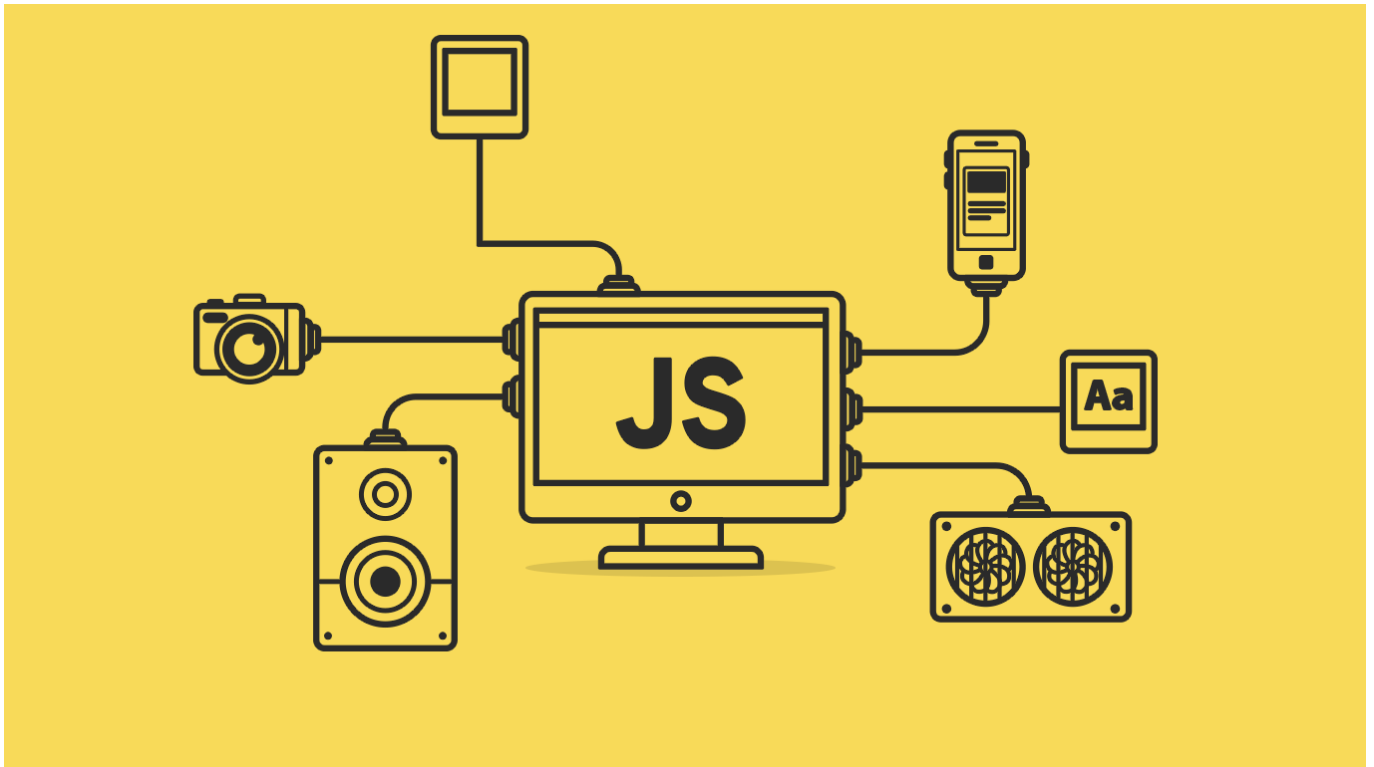


# ES6 và một số concepts trong Javascript

---



Source: <https://medium.com/swlh/introduction-to-javascript-basics-cf901c05ca47>

1. Overview + strict mode
2. Scope
3. Hoisting
4. IIFE
5. Closures
6. Function
7. Object
8. this
9. call, bind, apply
10. Iteration protocols

ES6 features: <http://es6-features.org/>

## Strict mode (ES5)

- Introduced in ECMAScript 5
- A flag to tell that you'd like to use modern code (from ES5 upward)

### It helps:

- Eliminates some JavaScript silent errors by changing them to throw errors.
- Fixes mistakes that make it difficult for JavaScript engines to perform optimizations: strict mode code can sometimes be made to run faster than identical code that's not strict mode.
- Prohibits some syntax likely to be defined in future versions of ECMAScript.

## Invoking strict mode

invoke for the **whole script** by adding 'use strict' on top of the file

```
// Whole-script strict mode syntax
'use strict';

const message = "Hi! I'm a strict mode script!";

function sayHello() {}
```

invoke for **current function** by adding 'use strict' on top of the function

```
function strict() {
  // Function-level strict mode syntax
  'use strict';

  function nested() {
    return 'And so am I!';
  }

  return "Hi! I'm a strict mode function! " + nested();
}

function notStrict() {
  return "I'm not strict.";
}
```

for **classes**, **es6 modules**, it auto turn of strict mode by default

## Strict mode giúp mình tránh được lỗi ở một số tình huống

Impossible to accidentally create global variables

```
// in non-strict mode
channelName = 'Easy Frontend'; // accidentally create global variables
console.log(channelName); // 'Easy Frontend'
console.log(window.channelName); // 'Easy Frontend' --> browser only
console.log(global.channelName); // 'Easy Frontend' --> nodejs server only
```

```
'use strict';
channelName = 'Easy Frontend';
// ReferenceError: channelName is not defined
```

Turn silently fail to throw an exception

```
// in non-strict mode
undefined = 1; // no error
```

```
'use strict'
undefined = 1;
// TypeError: Cannot assign to read only property 'undefined' of object
```

Throw error when you attempt to delete undeletable properties

```
// in non-strict mode
delete Object.prototype; // no effect and no error too 😂
```

```
'use strict';
delete Object.prototype;
// TypeError: Cannot delete property 'prototype' of function Object() {
  [native code] }
```

## Octal literals are not allowed in strict mode

```
// in non-strict mode
console.log(015 + 20); // 33 🤔
// because 015 is understand in octal
// 015 = 5 * 8^0 + 1 * 8^1 = 13
```

```
'use strict';
console.log(015 + 20);
// SyntaxError: Octal literals are not allowed in strict mode.

// correct way: use 0o (zero and small 'o' letter)
console.log(0o15 + 20);
```

## Tham khảo

- [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Strict\\_mode](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Strict_mode)
- <https://exploringjs.com/es6/index.html>

---

## Khoá học Javascript cho người mới bắt đầu 2021 🎉

- Tác giả: **Hậu Nguyễn** - Founder Easy Frontend
- Khoá học chỉ được published trên Udemy, không thông qua trung gian.
- Khoá học không bán dạng videos upload trên Google Drive hay bất cứ hình thức nào tương tự.
- Khoá học có nhóm discord để hỗ trợ trong quá trình học tập.

☎️ Liên hệ tác giả để được hỗ trợ:

- ✅ Facebook: <https://www.facebook.com/nvhauesmn/>
- ✅ Fanpage: <https://www.facebook.com/learn.easyfrontend>
- ✅ Youtube Channel: <https://www.youtube.com/easyfrontend>