# Building Frontends

Thierry Sans

# Recipes to become a good front-end developer

- Load Javascript code efficiently
- Ensure the DOM is loaded with `window.onload`
- Write good Javascript code
- Encapsulate Javascript in closures
- Create a Frontend API

# The problem with Javascript interpreters

- ✓ **Good Javascript** is interpreted by browsers in a consistent way

- ◉ **Bad javascript** code is loosely interpreted by browsers in an inconsistent way

# Solution 1: using **`strict mode`**

➡ Force the browser to validate Javascript against the standard

✓ Dynamically raises errors (or warnings) in the console when the code is not compliant with the standard

```
"use strict";
var doSomething = function() {
    // this runs in strict mode
}
```

# Solution 2 : using `JSHint`

➡ Analyze Javascript source code with JSHint

✓ Statically finds bugs and reports them in the terminal

```
$ npm install -g jshint
$ jshint js
```

# Problem with scoping

- In the browser, all Javascript files share the same execution environment i.e they share the same scope
  - variable (and function) naming conflicts
  - strict mode applied to all

# Scoping problem with variable names

file1.js

```
let doSomething = function() {
    // first declaration of doSomething
}
```

file2.js

```
let doSomething = function() {
    // shadowing doSomething from file 1
}
```

# Scoping problem with strict mode

file1.js

```
"use strict";
let doSomething = function() {
    // strict mode applies
}
```

file2.js

```
let doSomethingElse = function() {
    // strict mode applies too
}
```

# Solution : encapsulate Javascript in **a closure**

```
(function() {
    "use strict";
    let private = function() {
        // private is not available from outside
    }
}());
```

# Solution : encapsulate and export the **namespace**

```
let $ = (function() {
    "use strict";

    let export = {};

    var private = function(){
        // private is not available from outside
    }

    export.public = function(){
        // public is available from outside
    }

    return export;
}());
```

# Structuring the frontend