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 <u>consistent way</u>

◉ **Bad javascript** code is loosely interpreted by browsers in an <u>inconsistent way</u>

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";
let 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 myScript.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() {
    // conflicting doSomething from file 1

}
```

Scoping problem with strict mode

```
---------------------------------------------
                                      file1.js
"use strict";
let doSomething = function() {
    // strict mode applies

}


---------------------------------------------
let doSomethingElse = function() {   file2.js
    // strict mode applies too

}
```

Solution : encapsulate Javascript in **a closure**

```javascript
(function() {
    "use strict";

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

Solution : encapsulate and export the **namespace**

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

    let module = {};

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

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

    return module;
}());
```

Structuring the frontend

Frontend API

Backend
Web API

Push & Pull

UI events

UI update

Controller