

Javascript Coding Standards - UCL WAMS

The current state of the Indigo project does not guarantee the below guidelines are being met. We will work on bringing the project up to these standards and require future work on the project adheres to them.

Editor & setup

1. **Do** use the editor config. This will enable you to be consistent with indentation - Indigo includes an editor config so please make sure your IDE is compatible with it.
2. **BONUS MARKS** - install node js and grunt and make sure the grunt default task is running successfully for your project before you begin editing.
3. **Do** minify your js, **Don't** forget to provide an un-minified version too
4. **Do** limit your line length to 80 characters. Place a break after the operator and indent your next line by one tab
5. **Do** use line comments, saving block comments for formal documentation
6. **Do** host JS on UCL CDN e.g. <http://cdn.ucl.ac.uk/>
7. **Do** request a Github repository from UCL-WAMS
8. **Do** commit both minified and un-minified JS to your Github repository
9. **Do** keep the readme up to date

Writing

10. **Do** use the require framework. The preferred area to write your code is in the app relevant to your project <http://requirejs.org/>

11. **Do** declare variables before they are used, the **var** statement should be the first statement in the function body.
12. **Do** minimise use of global variables
13. **Don't** block scope variables. Javascript does not have block scope and this can confuse developers familiar with languages such as C

```
function example() {  
  // Do not do this  
  for (var i = 0; i < 3; i++) {  
    var a = 1;  
    // Do stuff with `a`  
  }  
}
```

```
function example() {  
  // Do this  
  var i, a;  
  for (i = 0; i < 3; i++) {  
    a = 1;  
    // Do stuff with `a`  
  }  
}
```

14. **Don't** use eval(). It has associated performance and security issues. It is also hard to debug.

15. **Don't** pass strings to `setTimeout()` or `setInterval()`, this is known as implied `eval()`;

```
//don't do this
```

```
var timer = setInterval("alert('hello world');", 1000);
```

```
//do this
```

```
function testFunction(){  
    alert('hello world');  
}
```

```
var timer = setInterval(function(){testFunction();}, 1000);
```

16. **Don't** leave uncommented console logs in your code. It will break your script in older versions of IE.
17. **Do** use `===` and `!==` operators, **Don't** use `==` and `!=`

Our conventions

18. **Do** indent the body of your function by one tab
19. **Don't** leave a space between your function name and the left parenthesis. **Don't** leave a space between the right parenthesis first curly brace.

```
function foo () {
```

```
    //this example is wrong
```

```
}
```

```
function foo(){
```

```
    //this example is correct
```

```
}
```

20. **Do** use the 26 characters of the English alphabet (upper and lower case), numbers 0-10 and underscores. **Don't** use characters `$` or `\`.
21. **Do** use `{}` instead of `new Object()`

22. **Do** use [] instead of new Array()

References:

<http://javascript.crockford.com/code.html>

<https://github.com/necolas/idiomatic-css>