

coding standards in JavaScript

“writing clean code is a matter of professionalism”

rule #1

String Concatenation

Always use a space before and after the operators to improve readability

```
var string = 'Foo' + _bar;  
string += _baz();
```

SPACES SPACES

rule #3

Functions

```
function funStuff(field, settings) {  
  settings = settings || Class.settings;  
  alert("This JS file does fun message popups");  
  return field;  
}
```

```
value = funStuff(field, settings);
```

- 1 Functions and methods should be named in lowerCamelCase
- 2 There should be no space between the function name and the following left parenthesis
- 3 Define optional arguments (using default values) at the end of the function signature
- 4 Always attempt to return a meaningful value from a function if one is appropriate
- 5 Functions should be called with no spaces between the function name, the opening parenthesis, and the first parameter; spaces between commas and each parameter, and no space between the last parameter, the closing parenthesis, and the semicolon

rule #2

Semi-colons

All statements should be followed by ; except for the following:

```
for   function   if   switch   try   while
```



- 1 Variables should be declared with var before they are used and should only be declared once
- 2 Variables should not be defined in the global scope; try to define them in a local function scope at all costs
- 3 All variables should be declared at the beginning of a function
- 4 Arrays should be formatted with a space separating each element and assignment operator
- 5 Note there is no comma at the end of the last array element

rule #4

Variables and Arrays

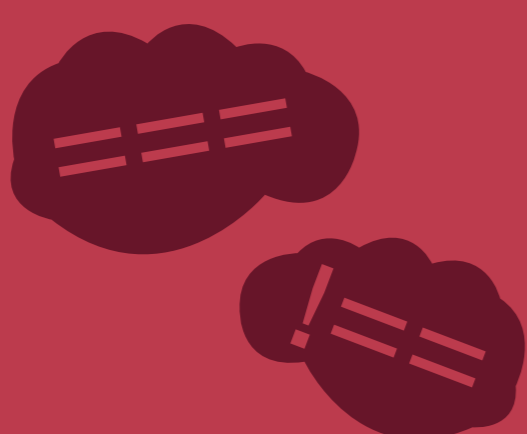
```
var yourArray = ['hello', 'world'];  
  
function fooBar() {  
  var myArray = ['Pakistan', 'Sri Lanka'];  
  var ourArray = myArray.concat(yourArray);  
  return ourArray;  
}
```

rule #5

True or false comparisons

When comparing to any of the following values, use the === or !== operators (not == or !=)

```
0  
"  
undefined  
null  
false  
true
```



rule #6

eval() is evil

