

Good JavaScript Principles

- Ensure that you have a good understanding of your goals before starting.
- Have a habit of writing clean JavaScript code. Browsers will intemperately badly written JavaScript as long as the syntax is correct. However it makes understanding and working on it at a later date much more difficult.
- Add script tags to the html right before the end of the body tag. This way, the page will load completely before downloading the script files. This will make the page load seem smoother to the user.
- Wherever possible, try not to mix languages. Browsers will intemperately alternating lines of JavaScript and jQuery, but it makes understanding it at a later date more difficult.
- Don't reinvent the wheel. There are lots of libraries, frameworks and plugins out there. Using these, you can achieve the same thing in a fraction of the time, compared with coding yourself from scratch.
- DRY (Don't Repeat Yourself).
- Named items, such as variables and functions should have short, easily understandable names. It should be as easy as possible to understand what the variable represents and what the function does.
- Global variables are bad because any JavaScript files declared further down the document object model that contain the same name will overwrite the global variable declared above it.
- Ensure that any data exchanged between the user and the code is of the correct type. Use "===" wherever possible.
- Quotes around a number make that number a string.
- Add spaces around operators and after commas. It looks better on the page and helps with reading later on.
- Comment on your work. It will help when you are anybody else looks at it in future.
- Name using camelCase. JavaScript uses it, so I should too.
- Indent with a tab or 4 spaces.
- Don't forget the semicolon.
- Don't include JavaScript directly in the HTML. It should be in its own file where possible.
- Minify as much as possible. It will reduce loading times; especially on large projects.