Coding Standards Document

Table of Contents

[1. Whitespace 1](#__RefHeading___Toc157_3381119126)

[1.1 Indentation 1](#__RefHeading___Toc159_3381119126)

[1.2 Space between operators and variables 1](#__RefHeading___Toc161_3381119126)

[1.3 Newlines 1](#__RefHeading___Toc163_3381119126)

[2. Naming 2](#__RefHeading___Toc165_3381119126)

[2.1 Capitalisation 2](#__RefHeading___Toc167_3381119126)

[2.2 Variable naming 2](#__RefHeading___Toc169_3381119126)

[3. Commenting 2](#__RefHeading___Toc171_3381119126)

[3.1 Headers 2](#__RefHeading___Toc173_3381119126)

[3.2 Line comments 2](#__RefHeading___Toc175_3381119126)

[4. Braces 2](#__RefHeading___Toc177_3381119126)

[4.1 Opening brace (curly bracket) location 2](#__RefHeading___Toc179_3381119126)

[4.2 Closing brace location 3](#__RefHeading___Toc181_3381119126)

[5. Other general coding standards 3](#__RefHeading___Toc183_3381119126)

[5.1 Max line length 3](#__RefHeading___Toc185_3381119126)

[5.2 String quotes 3](#__RefHeading___Toc187_3381119126)

[5.3 Objects per file 3](#__RefHeading___Toc189_3381119126)

[5.4 Semicolons 4](#__RefHeading___Toc191_3381119126)

[6. CSS 4](#__RefHeading___Toc193_3381119126)

[6.1 CSS files 4](#__RefHeading___Toc195_3381119126)

[6.2 Units 4](#__RefHeading___Toc197_3381119126)

# 1. Whitespace

## 1.1 Indentation

All indentations should be 4 spaces, regardless of language (JS, HTML, Angular etc.) allowing for consistency and readibility of conditionals, loops and nested functions in JS, or tag heirarchy in HTML.

## 1.2 Space between operators and variables

There should always be a single space between variables/numbers and operators (e.g. “4 + i” as opposed to “4+i”).

## 1.3 Newlines

There should always be two newlines (an empty line) in the following cases.

* Between variable declarations and functions/conditionals
* Between two different functions
* Between two different conditionals (except for “if...else” statements)
* Between a function and a conditional (or visa versa)

# 2. Naming

## 2.1 Capitalisation

All variable names, regardless of language, should be in lower camel case (unless unavoidable, as with certain pre-defined Angular objects). e.g. “fooBar” as opposed to “FooBar” or “foo-bar”.

## 2.2 Variable naming

All variables should be as descriptive as is necessary to explain their function without a comment. Comments should still be used in the case of any ambiguity. Iterators should be descriptive unless they are a simple integer, in which case they may be “i”, “j” or onwards if necessary.

# 3. Commenting

## 3.1 Headers

All files should contain a header including a brief summary of what the file does, the functions contained within, and who has worked on the file.

## 3.2 Line comments

All comments for individual lines should be located above the line it references, not at the end of the line or below. These comments should be present above any function declarations or complicated evaluations to maintain code readablity.

# 4. Braces

## 4.1 Opening brace (curly bracket) location

An opening brace (“{“) should appear on the same line as a conditional or function declaration, seperated by a space, and the content within should be on a new line following that.

e.g.

“foo(int bar) {

fooBar;“

rather than

“foo(int bar)

{

fooBar;“

## 4.2 Closing brace location

A closing brace should be located on the line following the last line contained within the braces, and followed by a blank line as per 1.3 (unless it’s an else statement).

e.g.

“...{

foo;

}”

rather than

“...{

foo;}”

# 5. Other general coding standards

## 5.1 Max line length

Unless absolutely unavoidable, a line should not exceed 100 characters to avoid scrolling horizontally while reading.

## 5.2 String quotes

All strings should use double quotes (" ") rather than single quotes (' ').

## 5.3 Objects per file

Only one object should be defined per file to maintain abstraction.

## 5.4 Semicolons

Semicolons (“;”) should be used at the end of any JavaScript statement even when not necessary, for consistency with places where it is necessary.

# 6. CSS

## 6.1 CSS files

Each Angular object should have its own CSS file during development in order to maintain readability. (NOTE: Prior to deployment, these may be combined to a single, site-wide CSS file.)

## 6.2 Units

All size, position etc. values should use relative lengths such as %, em, vh or similar rather than absolute lengths like cm, px etc. in order to maintain relative size and position of objects in different size viewports.