Code Style Guides

Or: How I Learned to Stop Worrying and Love Spaces
Who?

- David M Sibley
- New MyUW Full Stack Developer
- Previous 7 years at US Geological Survey
- As Technical Lead, I had many opinions
- Coming in to MyUW, I decided to shove those to the back and learn
Tabs vs Spaces

TECHNICAL

Proficient in **Linux** (preferred) and **Windows** development environments.

**Java** - Servlets/JSP, Tomcat, Guava, StAX, RabbitMQ, RxJava, JasperReports

**Javascript** - Node, Webpack, VueJS, AngularJS, Bootstrap, Material Design, HTML5 Canvas

**Database** - Oracle, PostgreSQL, CouchDB, Liquibase, MyBatis

**Geospatial & Statistical** - Geoserver, Openlayers, Leaflet, R, Python

**Continuous Integration & Delivery** - git, SVN, CVS, Jenkins, Maven, JUnit, Jasmine, Docker

ICE BREAKERS

**Agile or Waterfall?**
I run 3 week iterations with a Trello board and retrospectives... for my personal life.

**Tabs or Spaces?**
Tabs, for the same reason you use atomic operations on multi-threaded variables.

**Current Topic:**
Freakout
99% In

References provided enthusiastically upon request
MyUW Values

- Open Source
- Small Directed Commits
- Pull Request Review
Which fights are worth it?

- As Technical Lead, I had many opinions
  - Tabs
  - No extraneous commas
  - One return statement
  - Semicolons at the end of every statement

- Coming in to MyUW, I decided to shove those to the back and learn
  - There is no one true way to write code
What is going on here?

- You don’t want to scroll through a **thousand** line diff for a **fifty** line change
The Crux

Style differences are the bottleneck for productivity of a team.

Not style merits
MyUW Plan

- Arbitrarily pick ONE overarching style
- Give team members the tools to succeed
MyUW Implementation

- Google Style for Javascript
- ESLint
  - Husky Plugin to Lint on Commit
There’s a lot of rules!
Let someone else figure this out.
Google Style

- I don't want to look through this every time I write code.

4.2.4 Function expressions

When declaring an anonymous function in the list of arguments for a function call, the body of the function is indented two spaces more than the preceding indentation depth.

Example:

```java
prefix.something.reallyLongFunctionName('whatever', (a1, a2) => {
    // Indent the function body +2 relative to indentation depth
    // of the 'prefix' statement one line above.
    if (a1.equals(a2)) {
        someOtherLongFunctionName(a1);
    } else {
        andNowForSomethingCompletelyDifferent(a2.parrot);
    }
});

some.reallyLongFunctionCall(arg1, arg2, arg3)
    .thenWrapped()
    .then((result) => {
        // Indent the function body +2 relative to the indentation depth
        // of the '.then()' call.
        if (result) {
            result.use();
        }
    });
```

4.2.5 Switch statements

As with any other block, the contents of a switch block are indented +2.
ESLint

Automated Style Checking

ESLint
The pluggable linting utility for JavaScript and JSX

Welcome
ESLint is an open source project originally created by Nicholas C. Zakas in June 2013. Its goal is to provide a pluggable linting utility for JavaScript.

Latest News
- ESLint v4.0.0 released 11 June 2017
Google Style

- Two Spaces per Indentation
- Don’t Pad Functions with Whitespace
- Trailing & Extraneous Commas

- 80 Characters per Line
### ESLint Report
2925 problems (2925 errors, 0 warnings) - Generated on Wed Jun 21 2017 09:16:44 GMT-0500 (CDT)

<table>
<thead>
<tr>
<th>File Path</th>
<th>Problems</th>
<th>Errors</th>
<th>Warnings</th>
</tr>
</thead>
<tbody>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/docs/configuration.md</td>
<td>11 problems (11 errors, 0 warnings)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/docs/Google-analytics.md</td>
<td>15 problems (15 errors, 0 warnings)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/docs/injecting-dependencies.md</td>
<td>29 problems (29 errors, 0 warnings)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/docs/theming.md</td>
<td>1 problem (1 error, 0 warnings)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/tools/uw-frame-static/build.js</td>
<td>1 problem (1 error, 0 warnings)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/uw-frame-components/config.js</td>
<td>90 problems (90 errors, 0 warnings)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/uw-frame-components/js/app-config.js</td>
<td>74 problems (74 errors, 0 warnings)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/uw-frame-components/js/config.js</td>
<td>2 problems (2 errors, 0 warnings)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/uw-frame-components/js/frame-config.js</td>
<td>859 problems (859 errors, 0 warnings)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/uw-frame-components/js/override.js</td>
<td>5 problems (5 errors, 0 warnings)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/uw-frame-components/js/karma.conf.js</td>
<td>18 problems (18 errors, 0 warnings)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/uw-frame-components/main.js</td>
<td>2 problems (2 errors, 0 warnings)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/uw-frame-components/my-app/app-config.js</td>
<td>3 problems (3 errors, 0 warnings)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/uw-frame-components/my-app/main.js</td>
<td>2 problems (2 errors, 0 warnings)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/uw-frame-components/portal/about/controllers.js</td>
<td>12 problems (12 errors, 0 warnings)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/uw-frame-components/portal/about/route.js</td>
<td>13 problems (13 errors, 0 warnings)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/uw-frame-components/portal/about/services.js</td>
<td>13 problems (13 errors, 0 warnings)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/uw-frame-components/portal/features/controllers.js</td>
<td>25 problems (25 errors, 0 warnings)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/uw-frame-components/portal/features/directives.js</td>
<td>12 problems (12 errors, 0 warnings)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/uw-frame-components/portal/features/route.js</td>
<td>15 problems (15 errors, 0 warnings)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/uw-frame-components/portal/features/services.js</td>
<td>114 problems (114 errors, 0 warnings)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[+] /home/sibley/data/git-repos/uw-frame/uw-frame-components/portal/main.js</td>
<td>93 problems (93 errors, 0 warnings)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ESLint

- Run Autofix
- Warn down large scale offenses
- Left with ~300 miscellaneous errors
  - Over 20 files? I can handle that.
ESLint

- Add rules back in one by one until you’re adhering to the standard
ESLint

- Add rules back in one by one until you’re adhering to the standard
ESLint

- Add rules back in one by one until you're adhering to the standard.
Lessons Learned

● Commit by Rule, not Structure
● Don’t make exceptions, unless style would change functionality.
● Save Line Length for last.
Thanks!