W04 Readings

**Debugging in the browser**

Inspect the page and go to sources.

Go to the file you want to debug and set breakpoints.

Reload the page to start the debugger if a function is called during page load.

Open the informational dropdowns to the right, e.g. watch, call stack, scope.

Watch shows the current values for any expressions.

The call stack shows the nested calls chain.

Scope shows the current variables.

Click the resume button. The execution will resume and reach another breakpoint.

Clicking Step (the button that looks like an arrow pointing at a dot) will step through all script statements one by one.

Clicking Step over (the button that looks like an arrow going over a dot) will not go into a function and will run the next command.

Step into (the arrow pointing down at a dot) will ignore async actions that executes later, like setTimeout. Step into will go into the code.

Step out (the arrow pointing up above a dot) will execute until the end of the current function. It will stop the execution at the last line of the current function.

**Coding Style**

Curly Braces

Curly braces are not needed when using a single-line construct, and if it’s short enough code.

Don’t split to a separate line without curly braces.

The best variant is to put what the code does onto a separate line within curly braces.

Example:

if (input.value ! = "") {

        displayList(input.value);

}

Line Length

One of the ways to avoid long code is to use backticks (`). It will allow you to split a string up into multiple lines.

Around 80 or 120 should be the maximum line length.

Indents

Horizontal indents should be around 2 or 4 spaces indented from the left.

Vertical indents are empty lines to split code. Code should be split into logical blocks. There shouldn’t be more than 9 blocks of code without using a vertical indent.

Semicolons

Always put a semicolon after every statement.

Nesting Levels

Try to avoid overly nesting your code.

Function Placement

You can declare the functions above the code that uses them or add the code first then the functions they use. You can also declare a function where it’s first used, but the second function placement is preferred.

Automated Linters

Linters can check the style of your code to help you avoid bugs or typos. JSLint, JSHint, and ESLint are some of the well known linting tools.

**Error handling, "try...catch"**

Here is the syntax for the “try…catch” blocks:

try {

// code...

} catch (err) {

// error handling

}

These constructs allows you to catch errors to help prevent the script from dying.