

# ELEN4010 - CODE STYLING GUIDE

Kayla-Jade Butkow (714227), Jared Ping (704447), Lara Timm (704157) & Matthew van Rooyen (706612)

School of Electrical & Information Engineering, University of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa

## 1 INTRODUCTION

This document presents the code styling guidelines used within the ELEN4010 project, including naming of variables and formatting of code. Two style guides are provided: one for JavaScript and one for HTML.

## 2 JAVASCRIPT

### 2.1 Variables

Use the camelCase convention for names of variables and functions. All names should start with a lower-case letter. Acronyms should be upper-case.

```
firstName = "John";
lastName = "Doe";

index = 0;
counterValue = 0;

elementHTML = document.getElementById("button_1").innerHTML
```

Prefer declaring variables with `let` over `var` for better scoping whenever appropriate.

```
for (let i = 0; i < 10; i++) {
    doSomething();
}
```

### 2.2 Spacing

Do not put spaces around unary operators or before a comma or semi-colon.

```
i++;
f(x, y);
```

Always put spaces around binary (assignment, equality and boolean) and ternary (arithmetic) operators. Always place a space after a comma.

```
y = m * x + c;
f(x, y);
c = !a | b;
```

Tabs are preferred over spaces for indentations within the code base.

```
function toCelcius(fahrenheit) {
    return (5 / 9) * (fahrenheit - 32);
}
```

Place spaces between control statements and their parentheses. Do not place spaces between a function and its parentheses, or between a parenthesis and its content.

```
for (let i = 0; i < 10; i++) {
    doSomething();
}
```

## 2.3 Line Breaking & Braces

Each statement should get its own line.

```
i++;  
j++;  
if (boolCondition)  
    doSomething();
```

An else statement should go on the same line as the preceding close brace if one is present, otherwise should line up with the if statement.

Open braces should be placed on the same line as the preceding code block and close braces should be placed on their own line.

One-line control clauses should not use braces (unless comments are included) or a single statement spans multiple lines.

```
if (boolCondition) {  
    ...  
} else {  
    someFunction(veryLongParam1, veryLongParam2, ...  
    veryLongParam5);  
}  
  
if (boolCondition)  
    doSomething();  
else  
    doADifferentThing();  
  
if (boolCondition)  
    doSomething();  
else {  
    ...  
}
```

## 2.4 Strings

Prefer double quotes over single quotes for strings.

```
console.log("This is how strings should be declared");
```

Use implicit line joining for multi-line strings.

```
console.log("This is how strings should be declared\n"  
    "It's really quite nice to look at\n");
```

# 3 HTML

## 3.1 Nesting and Indentation

When tags contain other tags in their body, the nested tags should be placed on a new line. All new levels of nesting should be indented by another level.

```
<html>  
  <head>  
    <title>Example of good nesting and indentation</title>  
  </head>  
</html>
```

### 3.2 Tags & Attributes

Tag names should all be lower-case.

```
<div>
  <p>Lower-case tags are better</p>
</div>
```

All tags should be closed unless they are non-container tags.

```
<div>Container tags</div>
<p>must be closed</p>
<br>This doesnt have to be closed
```

Use lower-case attribute names.

```
<div class="dropdown">Dropdown menu</div>
```