

# Auto-Garçon Style Guide

Authors:

# Contents

<b>1</b>	<b>General Style</b>	<b>2</b>
1.1	File Naming . . . . .	2
1.2	Syntax . . . . .	2
1.3	Indentation . . . . .	3
1.4	README . . . . .	3
<b>2</b>	<b>Web UI: JavaScript</b>	<b>4</b>
<b>3</b>	<b>Android: Java</b>	<b>5</b>
3.1	Documentation . . . . .	5
<b>4</b>	<b>Database: SQL</b>	<b>6</b>
4.1	Naming Conventions . . . . .	6
4.1.1	Tables . . . . .	6
4.1.2	Attributes . . . . .	6
4.1.3	Stored procedures . . . . .	6
4.2	Querying the Database . . . . .	7
4.2.1	Space . . . . .	7
4.2.2	Line spacing . . . . .	7
4.3	Creating a table . . . . .	7
4.4	Documentation . . . . .	7
4.4.1	Queries . . . . .	7
4.4.2	Tables . . . . .	7
4.5	Reference . . . . .	7
<b>5</b>	<b>Alexa: NodeJS</b>	<b>8</b>
5.1	Variables . . . . .	8
5.2	Naming Conventions . . . . .	8
5.3	Conditionals . . . . .	8
5.4	Functions . . . . .	8
5.5	Comments . . . . .	8
5.6	Other . . . . .	8

# Chapter 1

## General Style

### 1.1 File Naming

The file naming convention requires PascalCasing (?) Where every word in file name is capitalizd. The file name should not include any spaces, hyphens, underscores, or other special characters. Avoid creating generic filenames such as main.txt

Allowed:

- RestaurantSite.html
- DatabaseAPI.api

NOT allowed:

- Two Words.txt
- Using\_Underscores.java
- Super-hyphenated.js

Test files should be setup in their own separate

figure out  
how testing  
should hap-  
pen

### 1.2 Syntax

Although Javascript is a flexbile language. It is important for the readability of the code to include uniform syntax. This includes

- Including semicolons after every line of coding
- Including curly braces whenever defining new scope. Specifically in the following way:  
function(){ ← the curly bracket is on the same line as the function header

```
...  
Code  
...  
}
```

- Inline comments are to be written above the line of code that it is referring to. An example would be:  
\\Here is a comment for x = 5\*x  
x = 5\*x

## 1.3 Indentation

Indentation is required for the readability of code. Indents should be done through tabs equivalent to 4 spaces.

## 1.4 README

Every group is responsible to include a README.txt within their working directory. The format for the README.txt file is as follows:

Date Created:

Last Updated:

Group: [Web UI/Database/Android/Alexa]

Execution:

`./script arguments`

Description: This script does the following

Files:

-Example file.txt: Describe what the file does

-

-

Folders:

+exampleDirectory

-File within Directory

+anotherDirectory

-File here

-

-

Include instructions on how to do the tab settings for some popular editing tools to help people out.

## Chapter 2

# Web UI: JavaScript

## Chapter 3

# Android: Java

### 3.1 Documentation

The code documentation should be set up in the standard Javadocs format so that it would be able to compile. The format is shown below:

# Chapter 4

## Database: SQL

### 4.1 Naming Conventions

In general, every name should be unique and should not match with any of the reserved key words in SQL. Names should be limited to 30 characters. Names must begin with a letter and can only contain letters and underscores. Names should only have a single underscore. Names should avoid using abbreviations and should be commonly understood.

#### 4.1.1 Tables

When creating tables for our database, use a collective name such as staff or people, or the plural form of a noun.

#### 4.1.2 Attributes

The attributes of each table should be a singular noun that does not have the same name as a table. There are a collection of suffixes that have universal meaning that should be used if they are applicable.

- `_id`—a unique identifier such as a column that is a primary key.
- `_status`—flag value or some other status of any type such as `publication_status`.
- `_total`—the total or sum of a collection of values.
- `_num`—denotes the field contains any kind of number.
- `_name`—signifies a name such as `first_name`.
- `_seq`—contains a contiguous sequence of values.
- `_date`—denotes a column that contains the date of something.
- `_tally`—a count.
- `_size`—the size of something such as a file size or clothing.
- `_addr`—an address for the record could be physical or intangible such as `ip_addr`.

#### 4.1.3 Stored procedures

Stored procedures must contain a verb.

## 4.2 Querying the Database

Querying the database must use uppercase for all reserved keywords. The query should be aligned to the right of a key word. Every keyword should appear on a separate line except for the AS keyword.

### 4.2.1 Space

Spaces should be included between keywords, before and after the equals sign =, after commas and surround apostrophes where not within parentheses.

### 4.2.2 Line spacing

Newlines should be applied before an AND or OR, after a semicolon to separate queries, after each keyword definition, after a comma when separating multiple columns into logical groups.

## 4.3 Creating a table

The data types that could be used should be.

## 4.4 Documentation

### 4.4.1 Queries

Each query should have a comment briefly describing the query in the C style comment block `/**/`

### 4.4.2 Tables

Tables should have the following documentation

## 4.5 Reference

This style guide is modelled after the style guide provided by Simon Holywell and can be referenced by going to <https://www.sqlstyle.guide>



# Chapter 5

## Alexa: NodeJS

### 5.1 Variables

- Each variable must have its own type declaration.
- String variables must use single quotes.
- NO TERNARY OPERATORS.

### 5.2 Naming Conventions

- Variables and functions must use camel case.
- All letters in constants must be capitalized.

### 5.3 Conditionals

- All conditionals should be as simple as possible.
- All equality logical comparisons must use ===/!==.
- NO TERNARY OPERATORS.

### 5.4 Functions

- Avoid nested conditionals.

### 5.5 Comments

- Always comment unclear code.

### 5.6 Other

- ALWAYS USE SEMICOLONS.
- MUST USE ON LINE CURLY BRACES TO DEFINE SCOPE.