

CODING STANDARDS

Follow Me Drones

Members

Len Bekker

Devon Petrie

Gilad Tabul

Brendon van Biljoen

Francois Venter

5 Guys 1 Branch

5guys1branch@gmail.com

NAMING CONVENTIONS

Variables, functions, classes, etc. should be named according to their intended purpose/use. The names for the abovementioned items should be descriptive yet concise. Examples will be provided for each subsection to clearly distinguish the correct from the incorrect.

Our Python, JavaScript and TypeScript code comply with Pylint, JSLint and TSLint respectively. Therefore, running either of the three linting methods for the respective piece of code will ensure compliance with our coding standards.

Variables

- Variables should be named according to the camelCase convention, with the first word always beginning on a lowercase letter.
- Variable names, functions, class names, filenames and folder names should not include numbers, symbols, whitespace or non-English characters.

Correct:

- animalAlreadyDetected
- connectedDrone

Incorrect:

- myvar, 5guys, flag, thing
- αδιΔας, פלאפל, 用品, نفط

Functions

- Functions should be named according to the camelCase naming scheme, with the first word always beginning on a lowercase letter.

Correct:

- prepareDrone()
- connectDatabase()

Incorrect:

- doSomething()
- function1()

Classes

- Classes should be named according to the Pascal naming scheme, with the first word always beginning with an uppercase letter.
- Pascal case is a subset of camelCase.

Correct:

- Drone
- DroneController

Incorrect:

- drone_controller
- 5drones1guy

Files & Folders

- Files and folders should always be lowercase letters, with a dash (-) between multiple words.

Correct:

- object-recognition
- interface.py

Incorrect:

- drone_interfaces
- data base

Constants

- Constants should always be uppercase letters, with an underscore (_) between multiple words.
- Constants should be used when the value of a variable will not be modified.

Correct:

- MAX_VELOCITY
- SERVER_PORT

Incorrect:

- THIS-IS-A-CONSTANT
- why_are_we_SHOUTING

FORMATTING

Indentation

- Code blocks in all languages should be indented with 2 spaces, except Python which uses 4 spaces for indentation.

Correct:

All other languages:

```
Function connect() {  
    return true;  
}
```

Python:

```
def connect:  
    return True
```

Incorrect:

All other languages:

```
function connect() {  
    return true;  
}
```

Python:

```
def connect:  
    return False
```

Comments

- Comments are pivotal to a clear understanding of the code and should be used wherever the function name does not give an indication of what the function does.
- Comments should be descriptive, but not describing every single line of code.
- Comments regarding functions should occur above the respective function.
- Every other comment should be placed inline.
- Single line comments should begin with a space and the comment should follow.

Correct:

All other languages:

- `// Navigates to given coordinates`

```
function navigate(latitude, longitude) {  
    return true; // Returns navigation success status  
}
```

- `/*
 * This is a good comment which spans
 * multiple lines!
 */`

Python:

- `# Commenting on commendable things in a comment`

Incorrect:

All other languages:

- `//Navigates`

```
function navigate(latitude, longitude) {  
    return true; //Returns status  
}
```

- `/*
This is a bad comment which spans
multiple lines!
*/`

Python:

- `#Commenting on unnecessary things in a comment`

Spacing

- A single space should be left between operators and operands.
- A single space should follow a comma in a parameter list.

Correct:

```
function delta(oldX, newX) {  
    return newX - oldX;  
}
```

Incorrect:

```
function delta(oldX,newX) {  
    return newX-oldX;  
}
```

GENERAL REMARKS

In JavaScript & TypeScript:

- For string literals, order of precedence is:
 1. Double quotation marks (")
 2. Single apostrophe (')
 3. Grave accent (`)
- Use `const` instead of `let` and `let` instead of `var`, where possible.
- Test equality with triple equals (`===` or `!==`) instead of double equals (`==` or `!=`) to ensure strict equality is enforced, where both the value and the type of the compared variables are the same.
- Ensure that files are encoded using UTF-8.
- Curly braces should begin on the line on which the function, switch statement, class or object is declared.
- String interpolation is preferred over string concatenation.
- Avoid using `eval`, due to known performance issues.
- Avoid using `with`, since it modifies the scope chain and can make debugging difficult.