

Chromium Group Code Style Guide (for Android Java)

Naming

- When naming a variable/class/method/field, use a meaningful name that easily identifies its purpose and details.

Classes & Interfaces

- Written in UpperCase (capitalize the first letter of every word)
- Example: ClassName

Methods/Functions

- Written in lowerCase (the first letter of the first word is lowercase, the first letters of the following words are capitalized)
- Example: public void setName() {}

Fields

- Written in lowerCase (the first letter of the first word is lowercase, the first letters of the following words are capitalized)
- Examples:

```
public int publicField;
private int privateField;
private static SingletonClass singleton;
```
- Static final fields should be written in all uppercase and separate words with an underscore
- Example: public static final int THIS_INT = 0;

Variables & Parameters

- Written in lowerCase (the first letter of the first word is lowercase, the first letters of the following words are capitalized)
- Avoid using single character variables (ex: i, j, k) except for inside loops
- Example:

```
public void setValue(int sum) {
    for (int i = 0; i<10; i++) {
        sum++;
    }
}
```
- Example: int numVar = 0;
String studentName = "";

XML File

- Written in all lowercase
- Separate words with an underscore

- Example: activity_login.xml, edit_game_screen.xml

XML Attribute

- Written in all lowercase
- Separate words with an underscore
- Example: android: id="@+id/user_name"

Treat Acronyms as Words

- If using an acronym in a name, treat it as a word and follow correct naming convention
- Example:


```
int userId;
class Url;
```

Brace Style

Classes, Methods/Functions, Loops, Conditional statements

- Use braces for all methods/functions, classes, loops, and conditional statements
- Start the first brace on the same line as the attribute that requires the brace
- Close the brace on an empty line after the code for the attribute is finished, all closing braces should have their own lines and be indented to align with the line where the opening brace is
- Example:

```
public Class {
    public int doStuff() {
        //do stuff
    }
}

public int getInt() {
    //do stuff
}

public int loop(int var) {
    for (int i = 0; i<10; i++) {
        var++;
    }
}

public int ifStatement(boolean bool) {
    if (bool) {
        //do stuff
    }
}
```

Declarations

Fields & Variables

- Single declaration per line
- Example:

```
public String firstName;  
public String lastName;
```

Access Level Modifiers

- Explicitly define access level for methods and variables
- Example: public, protected, private, no modifier

Comments

TODO Comments

- Use TODO comments for code that is temporary, a short term solutions or code that will be improved/edited in the near future
- Example: //TODO: Remove this code after rewriting an improved method

Annotations

- Include a comment of the top of every class to describe and provide a quick overview of the code and methods included
- Include comments explaining the code's logic so other team member's/viewers can easily understand the details and function of the code

Exceptions

Try and Catch Blocks

- Include try and catch blocks to catch exceptions when appropriate
- Example:

```
void setServerPort(String value) {  
    try {  
        serverPort = Integer.parseInt(value);  
    } catch (NumberFormatException e) {  
        serverPort = 80; // default port for server  
    }  
}
```

XML Strings

- All strings that can be viewed in the UI should be in strings.xml and accessed using strings.xml

Citations

Code Style Guide is inspired by:

<https://medium.com/@aliasadi/android-java-style-guide-515d99d888a5>