

If you are working on javascript then you will be following this style guide.

<http://google-styleguide.googlecode.com/svn/trunk/javascriptguide.xml>

if you are working in java we are following the android style guides.

(I looked up oracle style guides but they were very ugly and did not make sense)

<http://source.android.com/source/code-style.html>

Important things to note.

- Due to widescreen computers and the increasing display size we will support lines up to 150 characters but no larger.

- There are spaces between logic and math:

- `if (true && false) int i = 1 + 2; i += 4;`

- There are spaces between the `if`, `for`, and `while` and the opening parenthesis

- `if (true) {`

- `while (false) {`

- `for (int k = 0; k < 10; k++) {`

- There is a space between any closing parenthesis and an open bracket

- `if (true) {`

- `public void testFunction() {`

- `public class TestClass() {`

- As in the examples above all brackets are on the same line as the function, class or logic statement.

- Arrays can be declared in one line if they fit. (notice space before bracket)

- `int[] list = new int[] {1, 2, 3, 4, 5};`

- Do not leave trailing spaces

- `int i = 6;`

- Do not do the above (if you highlight you will see there are spaces after the end of the line)

- `int i = 6;`

- The above does not have trailing spaces after the end of the line

- `int i = 6;`

- `int i = 6;`

- 

- The above is a picture of the two types when highlighted, the correct way to do it is the lower line.

- when doing javadoc comments do not write them all as one line.
  - ```
/**  
 * comment  
 */
```

Is better than

    - ```
/** comment */
```
- Java/javascript naming conventions
  - classes start with a capital letter
    - `class John`
    - `class PeoplePerson`
  - methods start with a lower case letter
    - `public void getTime()`
    - `public void tuple()`
  - private variables start with an m
    - `private long mTime;`
    - `private String mName;`
  - constants are all caps with underscore in between
    - `public static final ERROR_MESSAGE = "oopsie";`
    - `private final long RESPONSE_CODE = 6;`
  - enum instances are considered constants but the type is considered a class