1. Coding Standard
   1. Python (extracted from [Google Python Style Guide](https://google.github.io/styleguide/pyguide.html)):
      1. Avoid global variables.
      2. Do not terminate your lines with semicolons and do not use semicolons to put two commands on the same line
      3. Maximum line length is 80 characters. Do not use backslash line continuation. Make use of Python’s implicit line joining inside parentheses, brackets and braces. If necessary you can add an extra pair of parentheses around a statement.
      4. Use parentheses sparingly. Do not use them in return statements or conditional statements unless using parentheses for implied line continuation. It is fine to use parentheses around tuples.
      5. Indent your code blocks with 4 spaces (no tabs).
      6. Two blank lines between top-level definitions, be they function or class definitions. One blank line between method definitions and between the class line and the first method. Use single blank lines within a function or method as judged appropriately to make groupings of statements easier to read.
      7. No whitespace inside parentheses, brackets or braces. No whitespace before a comma, semicolon, or colon. Do use whitespace after a comma, semicolon, or colon except at the end of a line. No whitespace before the open paren/bracket that starts an argument list, indexing or slicing. Surround binary operators with a single space on either side for assignment, comparisons, Booleans, and arithmetic. Do not use spaces around the ‘=’ when used to indicate a keyword argument or a default parameter value. Do not use spaces to vertically align tokens on consecutive lines.
      8. Generally only one statement per line.
      9. Naming: module\_name, package\_name, ClassName, method\_name, ExceptionName, function\_name, GLOBAL\_CONSTANT\_NAME, global\_var\_name, instance\_var\_name, function\_parameter\_name, local\_var\_name.
      10. Imports should be on separate lines.
      11. Even a file meant to be used as a script should be importable and a mere import should not have the side effect of executing the script’s main functionality. The main functionality should be in a main() function.
   2. JavaScript (extracted from [Google JavaScript Style Guide](https://google.github.io/styleguide/jsguide.html)):
      1. Braces are used for all control structures even if only a single statement. The first statement of a non-empty block must begin on its own line.
      2. Non-empty blocks use Kernighan and Ritchie (K&R) style:
         1. No line break before the opening brace.
         2. Line break after the opening brace.
         3. Line break before the closing brace.
         4. Line break after the closing brace if that brace terminates a statement or the body of a function or class statement, or a class method. Specifically, there is no line break after the brace if it is followed by else, catch, while, or a comma, semicolon, or right parenthesis.
      3. An empty block or block-like construct may be closed immediately after it is opened, with no characters, space, or line break in between (e.g. {}), unless it is a part of a multi-block statement (one that directly contains multiple blocks: if/else or try/catch/finally).
      4. Each time a new block or block-like construct is opened, the indent increases by 2 spaces. When the block ends, the indent returns to the previous indent level. The indent level applies to both code and comments throughout the block.
      5. Any array literal may be optionally formatted as if it were a block-like construct.
      6. Any object literal may optionally be formatted as if it were a block-like construct.
      7. Class literals (whether declarations or expressions) are indented as blocks. Do not add semicolons after methods, or after the closing brace of a class declaration (statements -- such as assignments -- that contain class expressions are still terminated with a semicolon).
      8. When declaring an anonymous function in the list of arguments for a function call, the body of the function is indented 2 spaces more than the preceding indentation depth.
      9. One statement per line followed by a line break and terminated with a semicolon. Do not rely on automatic semicolon insertion.
      10. Column limit is 80 characters.
      11. Line-wrapping:
          1. Prefer to break at a higher syntactic level.
          2. When a line is broken at an operator the break comes after the symbol. This does not apply to the “dot” (.), which is not actually an operator.
          3. A method or constructor name stays attached to the open parenthesis that follows it.
          4. A comma stays attached to the token that precedes it.
          5. When line-wrapping, each line after the first (each continuation line) is indented at least 4 spaces from the original line, unless it falls under the rules of block indentation.
      12. Vertical whitespace. A single blank line appears:
          1. Between consecutive methods in a class or object literal.
             1. Exception: A blank line between two consecutive properties definitions in an object literal (with no other code between them) is optional. Such blank lines are used as needed to create logical groupings of fields.
          2. Within method bodies, sparingly to create logical groupings of statements. Blank lines at the start or end of a function body are not allowed.
      13. Horizontal whitespace: no whitespace at the end of a line. A single space can appear in the following places:
          1. Separating any reserved word from an open parenthesis that follows it on a line.
          2. Separating any reserved word from a closing curly brace that precedes it on a line.
          3. Before any open curly brace with two exceptions:
             1. Before an object literal that is the first argument of a function or the first element in an array literal.
             2. In a template expansion, as it is forbidden by the language.
          4. On both sides of any binary or ternary operator.
          5. After a comma, semicolon. Spaces are never allowed before these characters.
          6. After the colon in an object literal.
          7. On both sides of the double slash that begins an end-of-line comment.
          8. After an open-JSDoc comment character and on both sides of close characters.
      14. Vertical token alignment: do not use spaces to vertically align tokens on consecutive lines.
      15. Do not use line continuations for string literals. Use concatenation instead.
      16. It is rarely correct to do nothing in response to a caught exception. When it truly is appropriate to take no action whatsoever in a catch block, the reason this is justified is explained in a comment.
      17. Switch-case:
          1. Fall-throughs are always commented.
          2. Default case is present even if it contains no code.
      18. Naming:
          1. Package names: lowerCamelCase
          2. Class names: UpperCamelCase. Typically nouns or noun phrases.
          3. Method names: lowerCamelCase. Typically verbs or verb phrases
          4. Enum names: UpperCamelCase. Individual items within the enum are named in CONSTANT\_CASE
          5. Constant names: CONSTANT\_CASE
          6. Non-constant field names: lowerCamelCase. Typically nouns or noun phrases.
          7. Parameter names: lowerCamelCase. One-character parameter names should not be used in methods.
          8. Local variable names: lowerCamelCase
          9. Template parameter names: concise single-word or single-letter identifiers in all-caps, such as TYPE or THIS or T.