

时尚指南

TL;DR

遵循Google JavaScript样式指南。 如果不确定使用哪种样式，请遵循现有代码的样式。 如果样式指南说要使用Internet Explorer 10中不存在的功能，请忽略它。

建议

- 对所有控制结构使用大括号
- 缩进而不是制表符
- 使用eslint
- 我们为.eslintrc设置了首选样式的规则
- 使用分号
- 编写适用于Internet Explorer 10的代码
- 在privateVariableNames_和privateFunctionNames_的末尾使用下划线
- 使用TODO创建GitHub问题，并使用TODO（#issueNumber）链接它们
- 用JSDoc注释一切
- 将camelCase用于变量和函数
- 将TitleCase用于类
- 使用ALL_CAPS作为常量
- 使用单引号（编写JSON时除外）
- 换行符为80个字符，使审核更加容易
- 在包装的行上缩进四个空格
 - 如果一行在多个句法级别被打断，则每个级别应在前一行/句法级别上有四个空格缩进
- 以大写字母开头的注释，以句号结尾
- 在for循环中重新声明变量。 也就是说，始终写for（var i = 0; ...）而不是for（i = 0; ...）
 - 不这样做会带来风险，即在将函数重构到更高的位置后，变量将被孤立并成为意外的全局变量

不建议

- 用制表符缩进。
- 使用双引号（编写JSON时除外）。
- 使用snake_case。
- 使用ES6。
 - 并非在我们支持的所有平台上都实现了它。
- 使用let（兼容性表）。
- 使用const（兼容性表）。
- 在publicVariableNames和publicFunctionNames的末尾使用下划线。
- 使用格式错误的JSDoc。
 - 我们的JSDoc将作为文档的一部分自动发布。
- 写入TODO（用户名）。
 - 而是使用TODO创建GitHub问题，并使用TODO（#issueNumber）链接它们。
- 使用string.startsWith。
 - 它在IE中不存在。 请改用goog.string.startsWith。

JSDoc

Blockly团队使用JSDoc注释我们的代码并生成文档。我们期望JSDoc用于类的所有属性（私有和公共）以及所有函数。

JSDoc注释必须以`/*`开头，以`*/`结尾，才能正确解析。

块式使用三个可见性标签。始终使用尽可能受限的范围。

- `public` 表示功能或属性可以由外部开发人员使用。
- `private` 意味着只有拥有的类才应使用该属性或调用该函数。
- `package` 意味着Blockly中的其他类可以使用该属性或调用该函数，但外部开发人员则可以不使用它。

属性

属性注释应包括

- 物业描述
- 方式
- 可见性标签：公共，打包或私有

首选：

```
/**
 * Whether the block may be deleted by the user.
 * @type {boolean}
 * @private
 */
this.deletable_ = true;
```

仅当属性的含义显而易见时才允许。

```
/** @type {boolean} */
this.movable = true;
```

函数

功能注释应包括

- 功能说明
- 每个参数一个`@param`标签，包括
 - 类型
 - 名称
 - 描述
- 可见性标签：公共，打包或私有
- `@return`标记，如果函数将返回一个值，包括
 - 类型

- 返回值的描述（如果有）

例如：

```
/**
 * Obtain a newly created block.
 * @param {!Blockly.Workspace} workspace The block's workspace.
 * @param {?string} prototypeName Name of the language object containing
 *     type-specific functions for this block.
 * @return {!Blockly.Block} The created block.
 * @public
 * @deprecated December 2015
 */
Blockly.Block.obtain = function(workspace, prototypeName) {
  console.warn(
    'Deprecated call to Blockly.Block.obtain, use workspace.newBlock ' +
    'instead.');
```

```
    return workspace.newBlock(prototypeName);
  };
```