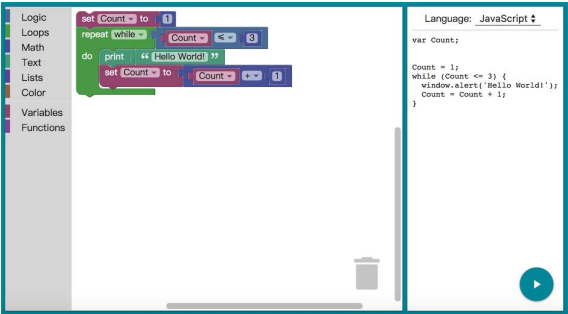




# Blockly - 来自Google的可视化编程工具

☐ 2017年08月11日 ☐ 6204 ☐ 声明



Google Blockly 是一款基于Web的、开源的、可视化程序编辑器。你可以通过拖拽 块 的形式快速构建程序，而这些所拖拽的每个 块 就是组成程序的基本单元。可视化编程完成，Blockly 直接支持 JavaScript、Python、PHP、Lua、Dart 语言源码的导出。此外，还可以将Blockly 编辑器快速集成到Web、Android或iOS环境中。

因近期参与一个机器人项目，可视化编程是项目需求之一。故以Web版本（HTML 和 JavaScrip）为例，整理一下 Blockly 的使用方法及各功能点。

- 1. Blockly 介绍
  - 1.1 构建Blockly应用
  - 1.2 Blockly与其它方案的比较
- 2. 使用 Blockly
  - 2.1 概述
  - 2.2 获取源码
  - 2.3 注入 Blockly
  - 2.4 配置
  - 2.5 代码生成
  - 2.6 “块”的导入、导出
  - 2.7 云存储

## 1. Blockly 介绍

Blockly 是一个向Web或Andorid / iOS应用添加可视化代码编辑器的库，Blockly使用相互关联的、图形化的块来表示代码中的概念，如：如变量、逻辑表达式、循环等。这样，用户就可以应用编程原理，而不必担心具体的语法、或命令行。

### 1.1 构建Blockly应用

对于用户来说，Blockly 只是用更直观的可视化的方式来生成代码。而对于开发都而言，Blockly只是一个文本框，其包含了语法正确的、用户生成的代码。

Blockly可以将“块”导出为代码，其支持以下主流语言：

- JavaScript
- Python
- PHP
- Lua
- Dart

构建一个Blockly应用一般包括以下步骤：

- 集成Blockly编辑器 - 最简单的Blockly编辑器包含了一个“工具箱”来存储的块（block）类型，和一个用于安装块的“工作区”。详细集成方法请参考[使用 Blockly](#)，或官方文档[Web](#)和[Android](#)
- 创建应用块 - 集成Blockly后，就需要创建一些用户代码块，并将其添加到Blockly“工具箱”。创建自定义块参考官方文档[Create Custom Blocks Overview](#)

搜索

### 文章分类

相关

### 阅读排行

[Sequelize 中文API文档 - 1. 快速入门、Seq... \(98911\)](#)

[Sequelize 中文API文档 - 2. Model 的定... \(97371\)](#)

[解决类似 /usr/lib64/libstdc++.so.... \(32075\)](#)

[Sequelize 中文API文档 - 4. 查询与原始查询 \(31611\)](#)

[Sequelize 中文API文档 - 3. 模型（表）之间的... \(28588\)](#)

[HTTP请求方法： GET、HEAD、 POST、 PUT、 DE... \(13256\)](#)

[Sequelize 中文API文档 - 5. 实例的使用、In... \(11693\)](#)

[\[ES6\] Promise对象Promise.all\(\)方法... \(11566\)](#)

[bluebird与原生Promise对象及bluebird... \(10957\)](#)

[MQTT协议 - MQTT协议简介及协议原理 \(10941\)](#)

### 最新文章

[Blockly 创建自定义块-Blockly 开发者工具](#)

Blockly Developer Tools 是一个基于Web的开发者工具，它可以通过配置自动化...

[Blockly 创建自定义块-概述](#)

将 Blockly 集成到应用中后，或多或少总是需要创建一些“块”。本篇...

[Sequelize 嵌套模型查询及嵌套模型字段排序](#)

Sequelize 的“嵌套模型查询”最终

3. 构建应用的其余部分 - Blockly只是解决一代码生成的部分，而应用的核心是如果使用代码，这部分还需要开发者自行实现

## 1.2 Blockly与其它方案的比较

Blockly 正在被越来越多可视化编程环境所使用。使用Blockly做为可视化编程方案具体有以下几点优势：

- 代码可导出 - 用户可基于“块”提取出通用编程语言，并可平滑过渡到基于文本的编程。
- 开源 - Blockly 开放所有源码，你可以复制、修改、并将其应用到你的网站或Andorid等应用中
- 可扩展 - 你可以按需要调整 Blockly，包括根据你的API添加新自定义“块”、移不需要的块和功能等。
- 高可用 - Blockly 不是玩具，你可以用它来实现复杂的编程任务
- 国际化 - Blockly 已被翻译40+种语言

尽管有以上优势，但 Blockly 不可能成为所有应用的解决方案。以下是一些其它可视化编程方案，可按自己需要选用：

- [Scratch Blocks](#): MIT设计和实现的一个Blockly代码库，Scratch Blocks提供了一个简化的编程模型，非常适合于初学者
- [Droplet](#):支持Pencil Code的图形化编程编辑器，它的显著特点是能够从代码转换成块。
- [Snap](#):一个从无到有的图形化编程语言，它不是一个库，而是一个集成执行环境的完整应用

## 2. 使用 Blockly

接下来，我们基于 `HTML` 和 `JavaScript`，来介绍将Blockly做为代码编辑器集成到Web应用中的过程。除Web应用外，Blockly还可以集成到Android或iOS应用中，详细请参考官方文档：

- [Get Started-Android](#)
- [Get Started-iOS](#)

### 2.1 概述

Blockly 被设计的可以很容易地安装到你的Web应用中。用户可以拖动“块”，而Blockly通过“块”生成代码，而应用无需为生成代码做任何事情。对应用来说 Blockly 仅是一个用户指定类型语言（JavaScript, Python, PHP, Lua, Dart 或其它）的文本输入框。

Blockly 是一个完全客户端应用，它无需服务端的任何支持（除非你要使用云存储等服务端功能），且没有第三方的依赖（除非你想重新编译内核），一切都是开源的。

### 2.2 获取源码

Blockly 源码托管在 GitHub，可以通过 GitHub 下载或在线查看源码：

- [下载Zip包](#)
- [下载Tar包](#)
- [GitHub在线查看](#)

下载源码并解压后，可以在浏览器打开 `demos/fixed/index.html` 文件，验证 Blockly 的块是否可以拖动等。

### 2.3 注入 Blockly

安装 Blockly 并验证其可用后，就可以引入 Blockly。如，将 Blockly 在Web页面的一个固定尺寸的 `div`：

- [固定尺寸的 Blockly](#)

更高级的用法可以让 Blockly 调整大小，以填满页面：

- [可调尺寸的 Blockly](#)

### 2.4 配置

上面的示例的 `Blockly.inject` 行中，第二个参数是一个键/值对字典。其用于配置Blockly，可用的配置项有：

- `collapse` - boolean。允许“块”折叠或展开。如果工具箱有类别，默认为 `true`；其它情况为 `false`
- `comments` - boolean。允许“块”有注释。如果工具箱有类别，默认为 `true`；其它情况为 `false`
- `css` - boolean。如果设置 `false`，则不注入 CSS；默认为 `true`
- `disable` - boolean。使“块”不可用。如果工具箱有类别，默认为 `true`；其它情况为 `false`
- `grid` - object。配置一个网格，使块可以捕获到。见[Grid](#)
- `horizontalLayout` - boolean。设置 `true` 则工具箱使用水平布局；`false` 则使用垂直布局。默认为 `false`

会生成SQL“连接查询”语句，即生成'inner joi...

[gyp 文件输入格式参考](#)

GYP（node-gyp）基于 .gyp 文件构建...

[Pomelo 前端服务器与客户端的通讯](#)

处理和响应客户端请求是 Pomelo 核心任务之一，客户端请求处理过程中会涉及很多组件，包括

交流群：564850876



- `maxBlocks` - number。最大可创建的“块”数量。默认为 `Infinity`
- `media` - string。Blockly 媒体文件目录路径。默认为 `"https://blockly-demo.appspot.com/static/media/"`
- `oneBasedIndex` - boolean。设置为 `true` 则字符串操作索引会从 `1` 开始；`false` 则从 `0` 开始。默认为 `true`
- `readOnly` - boolean。设置为 `true`，则禁止用户编辑。影响“工具箱”和“垃圾桶”。默认为 `false`
- `rtl` - boolean。设置为 `true`，则镜像化编辑器。默认为 `false`。见 [RTL Demo](#)
- `scrollbars` - boolean。设置工作区是否可滚动。如果工具箱有类别，默认为 `true`；其它情况为 `false`
- `sounds` - boolean。设置为 `false`，则点击或删除时不会播放声音。默认为 `true`
- `toolbox` - XML节点或string。用户可用“分类”和“块”的结构树。
- `toolboxPosition` - string。设置为 `start`，则工具箱在上部（水平显示时）或左侧（垂直显示时）或右侧（垂直LTR显示时）。设置为 `end`，则在相对侧。默认为 `start`
- `trashcan` - boolean。显示或隐藏“垃圾桶”。如果工具箱有类别，默认为 `true`；其它情况为 `false`
- `zoom` - object。工作区缩放配置。见 [Zoom](#)

在以上配置中，最重要的选项是 `toolbox`。它是一个XML节点树，用于指定工具箱中有哪些可用的“块”、块如何分布、及是否有类别。

- 更多信息参考[工具箱配置](#)

另外，除 Blockly 的默认“块”外，定义块需要通过调用你Web应用的API来构建。

- 更多信息参考[创建自定义块](#)

## 2.5 代码生成

Blockly 是编程语言，也就不能“运行”Blockly 程序。但是你可以将 Blockly 转为用户所需要的JavaScript、Python、PHP、Dart或其它语言

- 更多信息参考[代码生成](#)

## 2.6 “块”的导入、导出

如果你需要将“块”导出，以在其它应用中使用或在以后还原。可以调用以下XML：

```
var xml = Blockly.Xml.workspaceToDom(workspace);
var xml_text = Blockly.Xml.domToText(xml);
```

调用后，会生成一个最小化的包含用户“块”的XML。如果想使导出的XML更可读，那么可以使用 `Blockly.Xml.domToPrettyText` 来代替上面方法。

恢复已导出的XML，像下面这样即可：

```
var xml = Blockly.Xml.textToDom(xml_text);
Blockly.Xml.domToWorkspace(xml, workspace);
```

## 2.7 云存储

Blockly 带有一个可选的云存储功能。它允许用户保存、加载、共享和发布程序。如果你的项目是在云端托管，那么你可以利用此服务的优势。

- 更多信息参考[云存储](#)

下一篇：[Blockly 的配置](#)

上一篇：[高性能分布式队列系统 Beanstalkd 介绍及使用](#)