

# 构建

Blockly由一百多个JavaScript文件组成。对于最终用户来说，通过Internet加载它们是一种缓慢的体验。为了使Blockly加载更快，可以将其压缩为六个文件，总共约720kb。

Blockly附带了源代码和压缩文件，因此除非您要侵入core, blocks, generators, 或 msg目录，否则不需要构建Blockly。

Blockly压缩非常简单。假设您已安装Python 2.x，只需在Blockly的根目录中运行build.py即可：

```
python build.py
```

此命令大约需要20秒钟，并使用Google的在线Closure Compiler重建几个关键文件：

- core/ 目录的内容和closure-library/ 的相关位变为blockly\_compressed.js
- 生成一个名为blockly\_uncompressed.js的备用文件（请参见下文）
- blocks/ 目录的内容变为blocks\_compressed.js
- generators/ 目录的内容变为javascript\_compressed.js, python\_compressed.js, php\_compressed.js, lua\_compressed和dart\_compressed.js
- 对messages.js文件的任何更改都将映射到msg/json 文件，然后重新生成msg/js 文件

## 构建脚本参数

您可能不希望在开发过程中构建所有Blockly。您可以使用0个或多个可选参数来更改要构建的文件：

- 没有参数的build.py会构建所有东西
- build.py -core 生成blockly\_compressed, blockly\_uncompressed和blocks\_compressed
- ld.py -generators 生成每个<language>\_compressed.js文件
- build.py -langfiles 生成每个msg/js/<LANG>.js文件

注意：截至2019年7月，参数前面必须有一个“-”。不推荐使用不带-的旧参数样式。

## 压缩模式

作为压缩的结果，可以为Blockly加载少量HTTP请求：

```
<script src="blockly_compressed.js"></script>
<script src="blocks_compressed.js"></script>
<script src="javascript_compressed.js"></script>
<script src="msg/js/en.js"></script>
```

请记住，作为开发人员，您可能比用户拥有更好的Internet连接。压缩代码给用户带来了巨大的变化。

## 未压缩模式

也就是说，压缩代码对于开发人员而言简直是糟糕透顶。JavaScript开发人员不应容忍在编辑代码和查看结果之间进行构建的步骤。而且，任何JavaScript开发人员都不必调试压缩的代码。进行Blockly开发时，请使用非压缩模式。让我们拆开上面列出的四个脚本，找到它们的未压缩版本：

## Core

```
<script src="blockly_compressed.js"></script>
```

这很容易。只需将“压缩”更改为“未压缩”：

```
<script src="blockly_uncompressed.js"></script>
```

blockly\_uncompressed.js 文件由构建脚本生成，并加载所有必需的core/ 和closure-library/ 文件

## Blocks

```
<script src="blocks_compressed.js"></script>
```

如果有人要破解默认块，请使用原始源文件替换压缩的块文件：

```
<script src="blocks/logic.js"></script>
<script src="blocks/loops.js"></script>
<script src="blocks/math.js"></script>
<script src="blocks/text.js"></script>
<script src="blocks/lists.js"></script>
<script src="blocks/colour.js"></script>
<script src="blocks/variables.js"></script>
<script src="blocks/procedures.js"></script>
```

## Generators

```
<script src="javascript_compressed.js"></script>
```

如果有人要破解默认块，则将压缩的块文件替换为原始源文件（根据需要将“javascript”更改为“python”，“php”，“lua”或“dart”）：

```
<script src="generators/javascript.js"></script>
<script src="generators/javascript/logic.js"></script>
<script src="generators/javascript/loops.js"></script>
<script src="generators/javascript/math.js"></script>
<script src="generators/javascript/text.js"></script>
```

```
<script src="generators/javascript/lists.js"></script>
<script src="generators/javascript/colour.js"></script>
<script src="generators/javascript/variables.js"></script>
<script src="generators/javascript/procedures.js"></script>
```

## Language

```
<script src="msg/js/en.js"></script>
```

用messages.js替换语言文件（en，ru，vi等）：

```
<script src="msg/messages.js"></script>
```

## Playground

当入侵Blockly的核心时，Playground是一个非常有用的工具。只需将您的浏览器指向tests/playground.html。作为预览，这是演示服务器上的操场。Playground的特点：

- 所有代码均未压缩，可以快速开发
- 所有默认块（某些不推荐使用的块除外）
- 所有语言生成器（JavaScript，Python，PHP，Lua和Dart）
- 将程序导入和导出为XML
- 在LTR和RTL布局之间切换
- 在工具箱布局之间切换
- 渲染器的压力测试
- 在控制台中记录所有事件

在Google，几乎Blockly的所有开发都是在Playground上进行的。仅在将新版本推向生产之前完成构建。