1. [linguist](https://github.com/github/linguist)

语言识别库，能够自动根据项目的代码来识别你所使用的语言。 在你的项目源代码页面，可以看到一个彩条，点开以后会显示项目中的编程语言比例。linguist主要通过文件的后缀来识别，对于一些通用的扩展名，例如.m文件，linguist通过一些语言的特征片段来做判断。由于编程语言很多，linguist还不能覆盖所有语言的检测。

1. [jquery-pjax](https://github.com/defunkt/jquery-pjax)

pjax是Github的联合创始人之一[defunkt](https://github.com/defunkt)的作品，它使用html的pushState特性与ajax，可以实现页面内容动态局部刷新，当点击项目源代码页面中具体的一个文件或者文件夹时，你将会看到页面的其他部分是不变的，只有定义的页面DOM会刷新，这里使用的就是pjax。

1. [elasticsearch](https://github.com/elasticsearch/elasticsearch)

Eleasticsearch支撑了Github的搜索功能，2年之前Github使用Solor做搜索，随着用户和托管项目的增加，索引的大小超过了solor节点的最大存储空间，也出现了很多的问题，Github团队在思考解决方案时决定使用Elasticsearch做替换。Github最开始使用ES时，使用了44台亚马逊EC2实例，每台实例配备2T的存储，其中8台实例指负责查询请求。目前，Github已经将原有的EC搜索集群迁移到了东海岸的一个数据中心，使用8台物理主机替换了44台EC2。

1. [Rails](https://github.com/rails/rails)

Ruby实现的MVC Web框架。Github的用户界面和功能大部分基于Rails构建，不过需要注意的是现在虽然Rails的项目版本已经发展到了Rails 4，但是Github依旧使用的是自己维护的2.3分支，对于不保持和现有的Rails主版本号一致的原因，Github员工Kneath做了如下的解释：

* 1. . 花更过的时间来升级更新Rails，将会减少为用户构建新特性的时间，我们更关注用户；
  2. . 性能问题是一个很重要的考虑。在过去的几年中，我们极大的减少了响应时间。而升级Rails不仅会带来一个更慢的框架，而且还会引入一个不同的架构——我们需要再根据新的框架特性来定位优化性能。我们对于现有的框架已经做了很多的优化以保持性能稳定，最主要的是：将时间花费在升级上不会让我们的架构更快。
  3. . 过去的三年我们一直在升级这个堆栈，不升级Rails版本我们依然可以使用新的特性。

1. [Redis](https://github.com/antirez/redis)

Redis是K/V存储系统，知名的NoSQL实现之一，在Github，主要使用Redis来进行队列中的异常处理。在Github早期，曾尝试过很多的基于Ruby的队列机制，也曾使用Amazon SQS，但是这些方案都不能在Github快速增长的同时满足稳定性要求，最终Github迁移到了使用Redis的技术方案resque。

1. [sprocket](https://github.com/sstephenson/sprockets)

Sprocket是一个网站资源打包的Ruby库，它不仅能够管理JavaScript和CSS资源，还可以按照pipline的方式来流式预处理CoffeeScript、Sass、SCSS和LESS代码等；

1. [libgit2](https://github.com/libgit2/libgit2)

libgit2是一个可移植、纯C语言实现的Git核心方法类库，提供API重新链入Git方法。Github的背后使用的原生的git来实现commit、push等功能，但是使用libgit2来针对桌面应用调用、Ruby代码中调用等；

1. [rugged](https://github.com/libgit2/rugged)

libgit2的Ruby类库；

1. [bcrypt-ruby](https://github.com/codahale/bcrypt-ruby)

OpenBSD bcypt()密码哈希算法的Ruby实现；

1. [html-pipeline](https://github.com/blog/1311-html-pipeline-chainable-content-filters)

html-pipline是一个gem包，可以将现有Github前端HTML中的一些特性进行流式处理，例如在Github的评论框中，你可以@某一个人、输入emoji的表情、使用markdown的语法来写内容等，但是这些都是由单独的插件来控制的，html-pipeline可以流式的使用相应的插件处理原始内容，例如先将markdown转义成html，继而自动添加emoji表情，然后进行代码的语法高亮等。

1. [gemoji](https://github.com/github/gemoji)

在2013年的QCon北京前夜：Github Drink Up活动中，来自Github的工程师Tim在现场的活动中谈到了他们的一个文化：使用emoji。他解释道：“很多情感使用文字不能做出形象的表达，但是使用emoji表情却能够起到不一样的效果”。在Github现有评论框或其他内容中，都可以看到emoji的身影，所使用的就是gemoji这个gem包。

1. [jekyll](https://github.com/jekyll/jekyll)

Jekyll是一个静态博客生成的程序，Github中项目的Page页面，默认选型使用的就是jekyll。

1. [gollum](https://github.com/gollum/gollum)

Gollum是一套基于git的wiki系统，Github项目的wiki系统背后使用的就是这套开源框架；

1. [octokit.rb](https://github.com/octokit/octokit.rb)

Github API的官方Ruby SDK；

1. [Hubot](https://github.com/github/hubot)

Hubot是Github自行开发的一个聊天机器人，当然它已经超过了聊天机器人的范畴，Github作为一个异步办公的团队，日常的协作、沟通很大部分依赖于聊天室，通过Hubot，Github的员工可以在聊天室中给机器人定制一些特定的回复、3D打印模型，甚至通过hubot来部署生成环境的代码、获取服务状态等，在2013年的QCon北京中，Giuthub的工程曾针对如何使用Hubot做运维进行过分享：[《ChatOps at GitHub》](http://www.qconbeijing.com/2013/speaker.php?id=151)。

1. [d3](https://github.com/mbostock/d3)

d3是使用JavaScript实现的数据可视化框架，使用HTML、SVG和CSS等，在d3的基础之上发展处诸如crossfilter、NVD3.js等一系列扩展或者简化框架，并且形成了一个良好的社区。作者[mbostock](https://avatars1.githubusercontent.com/u/230541?s=140)目前供职于NYTimes，d3是他的博士论文项目，目前Github使用d3来展示托管项目提交历史、记录等的可视化效果图。

1. [plax](https://github.com/cameronmcefee/plax)

plax是控制视差元素的JavaScript类库，你可以在404、505等页面看到它的实现效果。

1. [ace](https://github.com/ajaxorg/ace)

Ace是一个使用Javascript开发的代码编辑器，具备语法高亮、快捷键绑定等特性， Github使用Ace实现[基于web的代码编辑功能](https://github.com/blog/1557-github-flow-in-the-browser)。

1. [zepto](https://github.com/madrobby/zepto)

Zepo是一个JavaScript框架，其特点是兼容现有jQuery API的同时，自身体积十分小；

1. [zeroclipborad](https://github.com/zeroclipboard/zeroclipboard)

Github的“点击复制到粘贴板”的功能就是使用的zeroclipboard，zeroclipboard使用一个不可见的Adobe Flash动画来实现复制粘贴，并提供Javascript的API接口以供调用。

1. [charlock\_holmes](https://github.com/brianmario/charlock_holmes)

charlock\_holmes用来检测字符编码格式，并可以自动将字符编码转化成UTF-8。

1. [puppet](https://github.com/puppetlabs/puppet)

服务器运维工具，可以进行自动化部署、集群管理等。

1. [moment](https://github.com/moment/moment)

moment是一个日期框架，用于解析、验证、格式化日期等，其中一个常用的功能是将原始的Javascript时间类型转化成方便阅读的时间说明格式，例如：”2小时之前“、”3天之前“这种形式。

1. [bower](https://github.com/bower/bower)

前端资源包管理工具，可以通过bower install <package>的形式将常用的前端资源下载到本地的项目目录中，例如：bower install bootstrap将会自动下载bootstrap的项目资源到本地的项目目录中，不需要自己手动来下载、移动资源文件，并且通过配置文件可以方便分享给同事、简化项目初始化等；

1. [resque](https://github.com/github/resque)

Resque是Github Enterprise中使用的一个基于Redis的后台作业控制系统，提供[可视化的界面](https://github.com/blog/542-introducing-resque)，可以方便的监控后台作业的运行状态和监控情况。

另外，Github还发布了“[支撑Github Windows客户端的开源项目](https://github.com/showcases/projects-that-power-github-for-windows)”和“[支撑Github Mac客户端的开源项目](https://github.com/showcases/projects-that-power-github-for-mac)”两个Showcase。

GitHub主要使用Rails实现，