GitHub Pages 网站前端开发指南

目录

[1 资料说明 22](#_Toc46394619)

[2 快速笔记 22](#_Toc46394620)

[2.1 站点设置 22](#_Toc46394621)

[2.2 md编辑 23](#_Toc46394622)

[2.3 Git命令 23](#_Toc46394623)

[2.4 Pelican常用命令 24](#_Toc46394624)

[2.5 Hexo常用命令 24](#_Toc46394625)

[2.6 Jekyll 24](#_Toc46394626)

[3 开源许可证 29](#_Toc46394627)

[3.1 可不可以不选择 License ？ 30](#_Toc46394628)

[3.2 具体选择标准 30](#_Toc46394629)

[3.3 MIT 31](#_Toc46394630)

[3.4 GPL 家族 31](#_Toc46394631)

[3.5 总结 33](#_Toc46394632)

[4 Git 34](#_Toc46394633)

[4.1 Git Quick start 34](#_Toc46394634)

[4.1.1 下载Git 34](#_Toc46394635)

[4.1.2 安装Git 34](#_Toc46394636)

[4.1.3 注册GitHub账号 36](#_Toc46394637)

[4.1.4 Git初始化设置 36](#_Toc46394638)

[4.1.5 在GitHub上创建Repository（仓库） 39](#_Toc46394639)

[4.2 Git 安装配置 43](#_Toc46394640)

[4.2.1 Linux 平台上安装 43](#_Toc46394641)

[4.2.1.1 Debian/Ubuntu 43](#_Toc46394642)

[4.2.1.2 Centos/RedHat 44](#_Toc46394643)

[4.2.1.3 源码安装 44](#_Toc46394644)

[4.2.2 Windows 平台上安装 44](#_Toc46394645)

[4.2.3 Mac 平台上安装 45](#_Toc46394646)

[4.2.4 Git 配置 46](#_Toc46394647)

[4.2.4.1 用户信息 47](#_Toc46394648)

[4.2.4.2 文本编辑器 47](#_Toc46394649)

[4.2.4.3 差异分析工具 47](#_Toc46394650)

[4.2.4.4 查看配置信息 47](#_Toc46394651)

[4.2.5 1 篇笔记 写笔记 48](#_Toc46394652)

[4.3 Git 工作流程 49](#_Toc46394653)

[4.4 Git 工作区、暂存区和版本库 51](#_Toc46394654)

[4.5 基本概念 51](#_Toc46394655)

[4.6 Git 创建仓库 52](#_Toc46394656)

[4.7 git init 52](#_Toc46394657)

[4.7.1 使用方法 53](#_Toc46394658)

[4.8 git clone 53](#_Toc46394659)

[4.9 2 篇笔记 写笔记 54](#_Toc46394660)

[4.10 Git 基本操作 55](#_Toc46394661)

[4.11 获取与创建项目命令 55](#_Toc46394662)

[4.11.1 git init 55](#_Toc46394663)

[4.11.2 git clone 55](#_Toc46394664)

[4.12 基本快照 56](#_Toc46394665)

[4.12.1 git add 56](#_Toc46394666)

[4.12.2 git status 58](#_Toc46394667)

[4.12.3 git diff 58](#_Toc46394668)

[4.12.4 git commit 59](#_Toc46394669)

[4.12.5 git reset HEAD 61](#_Toc46394670)

[4.12.6 git rm 62](#_Toc46394671)

[4.12.7 git mv 63](#_Toc46394672)

[4.13 3 篇笔记 写笔记 63](#_Toc46394673)

[4.14 Git 分支管理 66](#_Toc46394674)

[4.15 Git 分支管理 67](#_Toc46394675)

[4.15.1 列出分支 67](#_Toc46394676)

[4.15.2 删除分支 69](#_Toc46394677)

[4.15.3 分支合并 70](#_Toc46394678)

[4.15.4 合并冲突 70](#_Toc46394679)

[4.16 Git 查看提交历史 73](#_Toc46394680)

[4.17 Git 标签 75](#_Toc46394681)

[4.18 1 篇笔记 写笔记 77](#_Toc46394682)

[4.19 新建标签 78](#_Toc46394683)

[4.19.1 2 创建标签 78](#_Toc46394684)

[4.19.2 3 查看已有标签 78](#_Toc46394685)

[4.19.3 4 删除标签 78](#_Toc46394686)

[4.19.4 5 查看此版本所修改的内容 78](#_Toc46394687)

[4.20 Git 远程仓库(Github) 79](#_Toc46394688)

[4.21 添加远程库 79](#_Toc46394689)

[4.22 查看当前的远程库 84](#_Toc46394690)

[4.22.1 实例 84](#_Toc46394691)

[4.23 提取远程仓库 84](#_Toc46394692)

[4.24 推送到远程仓库 86](#_Toc46394693)

[4.25 删除远程仓库 86](#_Toc46394694)

[4.25.1 实例 86](#_Toc46394695)

[4.26 2 篇笔记 写笔记 87](#_Toc46394696)

[4.27 Git 服务器搭建 89](#_Toc46394697)

[4.27.1 1、安装Git 89](#_Toc46394698)

[4.27.2 2、创建证书登录 89](#_Toc46394699)

[4.27.3 3、初始化Git仓库 89](#_Toc46394700)

[4.27.4 4、克隆仓库 90](#_Toc46394701)

[4.28 git 分支查看与切换 91](#_Toc46394702)

[4.29 从仓库中删除敏感数据 91](#_Toc46394703)

[4.30 从仓库的历史记录中删除文件 91](#_Toc46394704)

[4.31 重置存储库 91](#_Toc46394705)

[4.32 静态网站生成器 static site generator 92](#_Toc46394706)

[4.33 emoji-cheat-sheet 92](#_Toc46394707)

[4.33.1 Table of Contents 92](#_Toc46394708)

[4.33.2 Smileys & Emotion 93](#_Toc46394709)

[4.33.2.1 Face Smiling 93](#_Toc46394710)

[4.33.2.2 Face Affection 94](#_Toc46394711)

[4.33.2.3 Face Tongue 94](#_Toc46394712)

[4.33.2.4 Face Hand 94](#_Toc46394713)

[4.33.2.5 Face Neutral Skeptical 94](#_Toc46394714)

[4.33.2.6 Face Sleepy 95](#_Toc46394715)

[4.33.2.7 Face Unwell 95](#_Toc46394716)

[4.33.2.8 Face Hat 96](#_Toc46394717)

[4.33.2.9 Face Glasses 96](#_Toc46394718)

[4.33.2.10 Face Concerned 96](#_Toc46394719)

[4.33.2.11 Face Negative 97](#_Toc46394720)

[4.33.2.12 Face Costume 97](#_Toc46394721)

[4.33.2.13 Cat Face 97](#_Toc46394722)

[4.33.2.14 Monkey Face 98](#_Toc46394723)

[4.33.2.15 Emotion 98](#_Toc46394724)

[4.33.3 People & Body 99](#_Toc46394725)

[4.33.3.1 Hand Fingers Open 99](#_Toc46394726)

[4.33.3.2 Hand Fingers Partial 100](#_Toc46394727)

[4.33.3.3 Hand Single Finger 100](#_Toc46394728)

[4.33.3.4 Hand Fingers Closed 100](#_Toc46394729)

[4.33.3.5 Hands 101](#_Toc46394730)

[4.33.3.6 Hand Prop 101](#_Toc46394731)

[4.33.3.7 Body Parts 101](#_Toc46394732)

[4.33.3.8 Person 102](#_Toc46394733)

[4.33.3.9 Person Gesture 102](#_Toc46394734)

[4.33.3.10 Person Role 103](#_Toc46394735)

[4.33.3.11 Person Fantasy 105](#_Toc46394736)

[4.33.3.12 Person Activity 106](#_Toc46394737)

[4.33.3.13 Person Sport 107](#_Toc46394738)

[4.33.3.14 Person Resting 108](#_Toc46394739)

[4.33.3.15 Family 108](#_Toc46394740)

[4.33.3.16 Person Symbol 109](#_Toc46394741)

[4.33.4 Animals & Nature 109](#_Toc46394742)

[4.33.4.1 Animal Mammal 110](#_Toc46394743)

[4.33.4.2 Animal Bird 111](#_Toc46394744)

[4.33.4.3 Animal Amphibian 111](#_Toc46394745)

[4.33.4.4 Animal Reptile 112](#_Toc46394746)

[4.33.4.5 Animal Marine 112](#_Toc46394747)

[4.33.4.6 Animal Bug 112](#_Toc46394748)

[4.33.4.7 Plant Flower 113](#_Toc46394749)

[4.33.4.8 Plant Other 113](#_Toc46394750)

[4.33.5 Food & Drink 113](#_Toc46394751)

[4.33.5.1 Food Fruit 114](#_Toc46394752)

[4.33.5.2 Food Vegetable 114](#_Toc46394753)

[4.33.5.3 Food Prepared 115](#_Toc46394754)

[4.33.5.4 Food Asian 116](#_Toc46394755)

[4.33.5.5 Food Marine 116](#_Toc46394756)

[4.33.5.6 Food Sweet 116](#_Toc46394757)

[4.33.5.7 Drink 117](#_Toc46394758)

[4.33.5.8 Dishware 117](#_Toc46394759)

[4.33.6 Travel & Places 118](#_Toc46394760)

[4.33.6.1 Place Map 118](#_Toc46394761)

[4.33.6.2 Place Geographic 118](#_Toc46394762)

[4.33.6.3 Place Building 119](#_Toc46394763)

[4.33.6.4 Place Religious 119](#_Toc46394764)

[4.33.6.5 Place Other 120](#_Toc46394765)

[4.33.6.6 Transport Ground 120](#_Toc46394766)

[4.33.6.7 Transport Water 121](#_Toc46394767)

[4.33.6.8 Transport Air 122](#_Toc46394768)

[4.33.6.9 Hotel 122](#_Toc46394769)

[4.33.6.10 Time 122](#_Toc46394770)

[4.33.6.11 Sky & Weather 123](#_Toc46394771)

[4.33.7 Activities 124](#_Toc46394772)

[4.33.7.1 Event 124](#_Toc46394773)

[4.33.7.2 Award Medal 125](#_Toc46394774)

[4.33.7.3 Sport 125](#_Toc46394775)

[4.33.7.4 Game 126](#_Toc46394776)

[4.33.7.5 Arts & Crafts 127](#_Toc46394777)

[4.33.8 Objects 127](#_Toc46394778)

[4.33.8.1 Clothing 128](#_Toc46394779)

[4.33.8.2 Sound 129](#_Toc46394780)

[4.33.8.3 Music 129](#_Toc46394781)

[4.33.8.4 Musical Instrument 129](#_Toc46394782)

[4.33.8.5 Phone 130](#_Toc46394783)

[4.33.8.6 Computer 130](#_Toc46394784)

[4.33.8.7 Light & Video 130](#_Toc46394785)

[4.33.8.8 Book Paper 131](#_Toc46394786)

[4.33.8.9 Money 131](#_Toc46394787)

[4.33.8.10 Mail 132](#_Toc46394788)

[4.33.8.11 Writing 132](#_Toc46394789)

[4.33.8.12 Office 133](#_Toc46394790)

[4.33.8.13 Lock 133](#_Toc46394791)

[4.33.8.14 Tool 134](#_Toc46394792)

[4.33.8.15 Science 134](#_Toc46394793)

[4.33.8.16 Medical 134](#_Toc46394794)

[4.33.8.17 Household 135](#_Toc46394795)

[4.33.8.18 Other Object 135](#_Toc46394796)

[4.33.9 Symbols 135](#_Toc46394797)

[4.33.9.1 Transport Sign 136](#_Toc46394798)

[4.33.9.2 Warning 136](#_Toc46394799)

[4.33.9.3 Arrow 137](#_Toc46394800)

[4.33.9.4 Religion 137](#_Toc46394801)

[4.33.9.5 Zodiac 138](#_Toc46394802)

[4.33.9.6 Av Symbol 138](#_Toc46394803)

[4.33.9.7 Gender 139](#_Toc46394804)

[4.33.9.8 Math 139](#_Toc46394805)

[4.33.9.9 Punctuation 139](#_Toc46394806)

[4.33.9.10 Currency 140](#_Toc46394807)

[4.33.9.11 Other Symbol 140](#_Toc46394808)

[4.33.9.12 Keycap 141](#_Toc46394809)

[4.33.9.13 Alphanum 141](#_Toc46394810)

[4.33.9.14 Geometric 142](#_Toc46394811)

[4.33.10 Flags 143](#_Toc46394812)

[4.33.10.1 Flag 143](#_Toc46394813)

[4.33.10.2 Country Flag 143](#_Toc46394814)

[4.33.10.3 Subdivision Flag 149](#_Toc46394815)

[4.33.11 GitHub Custom Emoji 149](#_Toc46394816)

[4.34 仓库Star的美好方式 149](#_Toc46394817)

[5 Github Desktop 150](#_Toc46394818)

[5.1 GitHub Pages 150](#_Toc46394819)

[5.2 GitHub资源 150](#_Toc46394820)

[6 Python 151](#_Toc46394821)

[6.1 安装 151](#_Toc46394822)

[6.2 解决pip安装时速度慢的问题 152](#_Toc46394823)

[6.3 临时使用： 153](#_Toc46394824)

[6.4 永久修改，一劳永逸： 153](#_Toc46394825)

[7 Ruby 154](#_Toc46394826)

[8 Markdown 155](#_Toc46394827)

[8.1 Markdown简介 155](#_Toc46394828)

[8.2 注释 155](#_Toc46394829)

[8.3 排版 155](#_Toc46394830)

[8.3.1 分割线 155](#_Toc46394831)

[8.3.2 标题 156](#_Toc46394832)

[8.3.3 缩进 157](#_Toc46394833)

[8.3.4 居中 157](#_Toc46394834)

[8.3.5 左对齐 157](#_Toc46394835)

[8.3.6 右对齐 157](#_Toc46394836)

[8.3.7 换行 157](#_Toc46394837)

[8.3.8 列表 158](#_Toc46394838)

[8.3.8.1 无序列表 158](#_Toc46394839)

[8.3.8.2 多级无序列表 158](#_Toc46394840)

[8.3.8.3 有序列表 158](#_Toc46394841)

[8.3.8.4 列表嵌套 159](#_Toc46394842)

[8.3.8.5 复选框列表 159](#_Toc46394843)

[8.4 字体 160](#_Toc46394844)

[8.4.1 字体类型 160](#_Toc46394845)

[8.4.2 大小 160](#_Toc46394846)

[8.4.3 颜色 160](#_Toc46394847)

[8.4.4 加粗 160](#_Toc46394848)

[8.4.5 斜体 160](#_Toc46394849)

[8.4.6 斜体加粗 160](#_Toc46394850)

[8.4.7 删除线 160](#_Toc46394851)

[8.4.8 斜体、粗体、删除线 161](#_Toc46394852)

[8.5 文字高亮 161](#_Toc46394853)

[8.6 超链接 162](#_Toc46394854)

[8.6.1 链接外部URL 162](#_Toc46394855)

[8.6.2 链接本仓库里的URL 163](#_Toc46394856)

[8.6.3 图片链接 163](#_Toc46394857)

[8.6.4 页内锚点 163](#_Toc46394858)

[8.7 引用 164](#_Toc46394859)

[8.8 表格 165](#_Toc46394860)

[8.8.1 定义表格 165](#_Toc46394861)

[8.8.2 对齐 165](#_Toc46394862)

[8.9 代码 166](#_Toc46394863)

[8.9.1 代码高亮 167](#_Toc46394864)

[8.10 流程图 167](#_Toc46394865)

[8.11 多媒体 167](#_Toc46394866)

[8.11.1 图片 167](#_Toc46394867)

[8.11.2 音频 169](#_Toc46394868)

[8.11.2.1 语法格式 169](#_Toc46394869)

[8.11.2.2 音频库 Github 169](#_Toc46394870)

[8.11.3 视频 170](#_Toc46394871)

[8.11.3.1 语法格式 170](#_Toc46394872)

[8.11.4 混合其他语法 170](#_Toc46394873)

[8.11.4.1 使用普通文本的删除线，斜体等效果 170](#_Toc46394874)

[8.11.4.2 表格中嵌入图片（链接） 171](#_Toc46394875)

[8.11.4.3 表情 171](#_Toc46394876)

[8.11.4.4 diff语法 171](#_Toc46394877)

[9 Static site generator 172](#_Toc46394878)

[9.1 概述 172](#_Toc46394879)

[9.2 WEB前端开发工具 177](#_Toc46394880)

[9.2.1 我们从这几个角度去分析一下几个框架的优劣 178](#_Toc46394881)

[9.2.2 Angular 178](#_Toc46394882)

[9.2.3 Vue 179](#_Toc46394883)

[9.2.4 React 179](#_Toc46394884)

[9.3 Python 180](#_Toc46394885)

[9.3.1 Pelican 181](#_Toc46394886)

[9.3.2 Cactus 181](#_Toc46394887)

[9.3.3 MkDocs 182](#_Toc46394888)

[9.4 Node.js 182](#_Toc46394889)

[9.4.1 Hexo 183](#_Toc46394890)

[9.4.2 Metalsmith 183](#_Toc46394891)

[9.4.3 Wintersmith 184](#_Toc46394892)

[9.4.4 基于 Git 制作电子书 GitBook 184](#_Toc46394893)

[9.4.5 Assemble 185](#_Toc46394894)

[9.5 JavaScript 185](#_Toc46394895)

[9.5.1 HubPress 185](#_Toc46394896)

[9.6 Ruby 186](#_Toc46394897)

[9.6.1 Jekyll 186](#_Toc46394898)

[9.6.2 Middleman 187](#_Toc46394899)

[9.6.3 Octopress 187](#_Toc46394900)

[9.7 基于Go 187](#_Toc46394901)

[9.7.1 Hugo 187](#_Toc46394902)

[9.8 React 188](#_Toc46394903)

[9.8.1 React 的渐进式静态网站生成器 React Static 188](#_Toc46394904)

[9.8.2 ReactJS 静态网站生成器 Gatsby 189](#_Toc46394905)

[9.1 Vue.js 190](#_Toc46394906)

[9.1.1 Vue.js 后端渲染开源库 Nuxt.js 190](#_Toc46394907)

[9.2 其他 190](#_Toc46394908)

[9.2.1 DocPad 190](#_Toc46394909)

[9.2.2 前端 Web 应用程序构建工具 Brunch 191](#_Toc46394910)

[9.2.3 Expose 191](#_Toc46394911)

[9.2.4 Wintersmith 191](#_Toc46394912)

[9.2.5 模块化网站编译器 Phenomic 191](#_Toc46394913)

[9.2.6 Lektor 191](#_Toc46394914)

[10 Pelican 4 193](#_Toc46394915)

[10.1 Pelican简介 194](#_Toc46394916)

[10.2 pelican 的功能 195](#_Toc46394917)

[10.3 Installing Pelican 196](#_Toc46394918)

[10.3.1 安装git,pip 196](#_Toc46394919)

[10.3.2 使用virtualenv工具创建pelican虚拟环境 196](#_Toc46394920)

[10.3.3 安装pelican 196](#_Toc46394921)

[10.3.4 安装markdown,typogrify 197](#_Toc46394922)

[10.4 Create a project 197](#_Toc46394923)

[10.5 pelican站点目录组成 199](#_Toc46394924)

[10.6 Writing content 200](#_Toc46394925)

[10.6.1 Articles and pages 文章和页面 200](#_Toc46394926)

[10.6.2 File metadata 文件中的信息 201](#_Toc46394927)

[1 201](#_Toc46394928)

[2 201](#_Toc46394929)

[3 201](#_Toc46394930)

[4 201](#_Toc46394931)

[5 201](#_Toc46394932)

[6 201](#_Toc46394933)

[7 201](#_Toc46394934)

[8 201](#_Toc46394935)

[9 201](#_Toc46394936)

[10 201](#_Toc46394937)

[Title: My super title 201](#_Toc46394938)

[Date: 2010-12-03 10:20 201](#_Toc46394939)

[Modified: 2010-12-05 19:30 201](#_Toc46394940)

[Category: Python 201](#_Toc46394941)

[Tags: pelican, publishing 201](#_Toc46394942)

[Slug: my-super-post 201](#_Toc46394943)

[Authors: Alexis Metaireau, Conan Doyle 201](#_Toc46394944)

[Summary: Short version for index and feeds 201](#_Toc46394945)

[This is the content of my super blog post. 202](#_Toc46394946)

[10.6.3 Pages 页面 206](#_Toc46394947)

[10.6.4 Linking to internal content 内部链接 206](#_Toc46394948)

[10.6.4.1 Linking to static files 链接到静态文件 207](#_Toc46394949)

[10.6.4.2 Mixed content in the same directory 同一文件夹下的混合内容 208](#_Toc46394950)

[10.6.4.3 Attaching static files 附件 208](#_Toc46394951)

[10.6.4.4 Linking to authors, categories, index and tags 210](#_Toc46394952)

[10.6.4.5 Deprecated internal link syntax 快要过时的内部链接语法 210](#_Toc46394953)

[10.6.5 Importing an existing site 从已存在的站点导入 210](#_Toc46394954)

[10.6.6 Translations 翻译 210](#_Toc46394955)

[10.6.7 Syntax highlighting 语法高亮 212](#_Toc46394956)

[10.6.8 Publishing drafts 草稿 213](#_Toc46394957)

[10.6.9 学习小结 213](#_Toc46394958)

[10.7 Publish Your Site 214](#_Toc46394959)

[10.7.1 Site generation 网站生成 214](#_Toc46394960)

[10.8 Viewing the generated files 预览 215](#_Toc46394961)

[10.9 Deployment 部署 216](#_Toc46394962)

[10.10 github部署 217](#_Toc46394963)

[10.11 Automation 自动化 218](#_Toc46394964)

[10.11.1 Fabric 218](#_Toc46394965)

[10.11.2 Make 219](#_Toc46394966)

[10.12 更新pelican 220](#_Toc46394967)

[10.13 发布博客站点 221](#_Toc46394968)

[生成博客站点 221](#_Toc46394969)

[预览生成的站点 221](#_Toc46394970)

[部署博客站点 222](#_Toc46394971)

[自动化操作 222](#_Toc46394972)

[10.14 Pelican Plugins 223](#_Toc46394973)

[10.14.1 How to use plugins 223](#_Toc46394974)

[10.14.2 Plugin descriptions 223](#_Toc46394975)

[添加评论系统 224](#_Toc46394976)

[10.15 配置文件设置 重点 224](#_Toc46394977)

[10.16 pelican-themes 226](#_Toc46394978)

[10.16.1 pelican-themes 插入主题 227](#_Toc46394979)

[10.16.2 Description 描述 227](#_Toc46394980)

[10.16.2.1 Usage 用法 227](#_Toc46394981)

[10.16.2.2 Optional arguments 参数 227](#_Toc46394982)

[10.16.2.3 Examples 例子 228](#_Toc46394983)

[10.16.2.3.1 Listing the installed themes 228](#_Toc46394984)

[10.16.2.3.2 Installing themes 安装主题 229](#_Toc46394985)

[10.16.2.3.3 Removing themes 移除主题 229](#_Toc46394986)

[10.16.2.3.4 Creating symbolic links 符号安装 229](#_Toc46394987)

[10.16.2.3.5 Doing several things an once 230](#_Toc46394988)

[10.16.3 pelican主题更换 230](#_Toc46394989)

[10.16.4 git提交代码 231](#_Toc46394990)

[10.16.5 学习笔记 231](#_Toc46394991)

[10.17 Pelican参考资源 232](#_Toc46394992)

[10.17.1 Pelican4.1.0博客系统官方文档中文版 232](#_Toc46394993)

[10.18 Pelican Plugins资源 233](#_Toc46394994)

[10.19 Pelican主题资源 239](#_Toc46394995)

[10.19.1 资源一览 239](#_Toc46394996)

[10.19.2 主题gum修改 240](#_Toc46394997)

[10.19.3 主题pelican-alchemy修改 242](#_Toc46394998)

[10.19.4 主题XXXX修改 242](#_Toc46394999)

[11 Hexo 243](#_Toc46395000)

[11.1 什么是 Hexo？ 243](#_Toc46395001)

[11.2 安装 243](#_Toc46395002)

[11.2.1 安装前提 243](#_Toc46395003)

[11.2.2 安装 Git 243](#_Toc46395004)

[11.2.3 安装 Node.js 244](#_Toc46395005)

[11.2.4 安装 Hexo 244](#_Toc46395006)

[11.2.5 进阶安装和使用 245](#_Toc46395007)

[11.3 建站 245](#_Toc46395008)

[11.3.1 Hexo文件夹介绍 245](#_Toc46395009)

[11.3.2 config.yml 246](#_Toc46395010)

[11.3.3 package.json 246](#_Toc46395011)

[11.3.4 scaffolds 247](#_Toc46395012)

[11.3.5 source 247](#_Toc46395013)

[11.3.6 themes 247](#_Toc46395014)

[11.3.7 Hexo常用命令 247](#_Toc46395015)

[11.4 配置 249](#_Toc46395016)

[11.4.1 网站 249](#_Toc46395017)

[11.4.2 网址 249](#_Toc46395018)

[11.4.3 目录 250](#_Toc46395019)

[11.4.4 文章 251](#_Toc46395020)

[11.4.5 分类 & 标签 252](#_Toc46395021)

[11.4.6 日期 / 时间格式 253](#_Toc46395022)

[11.4.7 分页 253](#_Toc46395023)

[11.4.8 扩展 253](#_Toc46395024)

[11.4.8.1 包括或不包括目录和文件 254](#_Toc46395025)

[11.4.8.2 使用代替配置文件 255](#_Toc46395026)

[11.4.8.3 覆盖主题配置 256](#_Toc46395027)

[11.5 Hexo指令 256](#_Toc46395028)

[11.5.1 init 256](#_Toc46395029)

[11.5.2 new 256](#_Toc46395030)

[11.5.3 generate 257](#_Toc46395031)

[11.5.4 publish 258](#_Toc46395032)

[11.5.5 server 258](#_Toc46395033)

[11.5.6 deploy 258](#_Toc46395034)

[11.5.7 render 258](#_Toc46395035)

[11.5.8 migrate 259](#_Toc46395036)

[11.5.9 clean 259](#_Toc46395037)

[11.5.10 list 259](#_Toc46395038)

[11.5.11 version 259](#_Toc46395039)

[11.5.12 选项 259](#_Toc46395040)

[11.5.12.1 安全模式 259](#_Toc46395041)

[11.5.12.2 调试模式 260](#_Toc46395042)

[11.5.12.3 简洁模式 260](#_Toc46395043)

[11.5.12.4 自定义配置文件的路径 260](#_Toc46395044)

[11.5.12.5 显示草稿 260](#_Toc46395045)

[11.5.12.6 自定义 CWD 261](#_Toc46395046)

[11.6 hexo部署到github 261](#_Toc46395047)

[11.6.1.1.1 设置hexo主题 264](#_Toc46395048)

[11.6.1.1.2 新增菜单栏选项 264](#_Toc46395049)

[11.6.1.1.3 新增草稿 264](#_Toc46395050)

[11.6.1.1.4 图片处理 264](#_Toc46395051)

[11.6.1.1.4.1 搜索 265](#_Toc46395052)

[11.6.1.1.4.2 字数 265](#_Toc46395053)

[11.6.1.1.4.3 宠物 265](#_Toc46395054)

[11.6.1.1.4.4 点击弹爱心 266](#_Toc46395055)

[11.7 Hexo迁移 268](#_Toc46395056)

[11.7.1 RSS 268](#_Toc46395057)

[11.7.2 Jekyll 268](#_Toc46395058)

[11.7.3 Octopress 268](#_Toc46395059)

[11.7.4 WordPress 268](#_Toc46395060)

[11.7.5 Joomla 269](#_Toc46395061)

[11.8 内容 269](#_Toc46395062)

[11.8.1.1 布局（Layout） 269](#_Toc46395063)

[11.8.1.2 文件名称 270](#_Toc46395064)

[11.8.1.3 草稿 270](#_Toc46395065)

[11.8.2 模版（Scaffold） 270](#_Toc46395066)

[11.8.2.1 支持的格式 271](#_Toc46395067)

[11.9 Front-matter 271](#_Toc46395068)

[11.9.1 分类和标签 272](#_Toc46395069)

[11.9.2 JSON Front-matter 273](#_Toc46395070)

[11.10 标签插件（Tag Plugins） 273](#_Toc46395071)

[11.10.1 引用块 273](#_Toc46395072)

[11.10.1.1 样例 273](#_Toc46395073)

[11.10.2 代码块 274](#_Toc46395074)

[11.10.2.1 样例 275](#_Toc46395075)

[11.10.3 反引号代码块 276](#_Toc46395076)

[11.10.4 Pull Quote 276](#_Toc46395077)

[11.10.5 jsFiddle 276](#_Toc46395078)

[11.10.6 Gist 276](#_Toc46395079)

[11.10.7 iframe 276](#_Toc46395080)

[11.10.8 Image 276](#_Toc46395081)

[11.10.9 Link 276](#_Toc46395082)

[11.10.10 Include Code 277](#_Toc46395083)

[11.10.10.1 样例 277](#_Toc46395084)

[11.10.11 Youtube 277](#_Toc46395085)

[11.10.12 Vimeo 277](#_Toc46395086)

[11.10.13 引用文章 277](#_Toc46395087)

[11.10.14 引用资源 278](#_Toc46395088)

[11.10.15 Raw 278](#_Toc46395089)

[11.10.16 文章摘要和截断 279](#_Toc46395090)

[11.11 资源文件夹 279](#_Toc46395091)

[11.11.1 文章资源文件夹 280](#_Toc46395092)

[11.11.2 相对路径引用的标签插件 280](#_Toc46395093)

[11.12 数据文件 281](#_Toc46395094)

[11.13 服务器 281](#_Toc46395095)

[11.13.1 hexo-server 281](#_Toc46395096)

[11.13.1.1 静态模式 282](#_Toc46395097)

[11.13.1.2 自定义 IP 282](#_Toc46395098)

[11.13.2 Pow 282](#_Toc46395099)

[11.13.2.1 安装 282](#_Toc46395100)

[11.13.2.2 设置 282](#_Toc46395101)

[11.14 生成文件 283](#_Toc46395102)

[11.14.1.1 监视文件变动 283](#_Toc46395103)

[11.14.1.2 完成后部署 283](#_Toc46395104)

[11.15 将 Hexo 部署到 GitHub Pages 284](#_Toc46395105)

[11.15.1 Project page 285](#_Toc46395106)

[11.15.2 私有 Repository 285](#_Toc46395107)

[11.15.3 有用的参考链接 285](#_Toc46395108)

[11.16 将 Hexo 部署到 GitLab Pages 286](#_Toc46395109)

[11.16.1 Project page 286](#_Toc46395110)

[11.16.2 Useful links 287](#_Toc46395111)

[11.17 部署 287](#_Toc46395112)

[11.17.1 Git 288](#_Toc46395113)

[11.17.1.1 这一切是如何发生的？ 289](#_Toc46395114)

[11.17.2 Heroku 289](#_Toc46395115)

[11.17.3 Netlify 289](#_Toc46395116)

[11.17.4 Rsync 290](#_Toc46395117)

[11.17.5 OpenShift 291](#_Toc46395118)

[11.17.6 FTPSync 291](#_Toc46395119)

[11.17.7 SFTP 292](#_Toc46395120)

[11.17.8 ZEIT Now 293](#_Toc46395121)

[11.17.9 21云盒子 293](#_Toc46395122)

[11.17.10 其他方法 294](#_Toc46395123)

[11.18 永久链接（Permalinks） 294](#_Toc46395124)

[11.18.1.1 变量 294](#_Toc46395125)

[11.18.1.2 示例 295](#_Toc46395126)

[11.18.1.3 多语种支持 295](#_Toc46395127)

[11.19 主题 296](#_Toc46395128)

[11.19.1.1 \_config.yml 296](#_Toc46395129)

[11.19.1.2 languages 296](#_Toc46395130)

[11.19.1.3 layout 296](#_Toc46395131)

[11.19.1.4 scripts 296](#_Toc46395132)

[11.19.1.5 source 297](#_Toc46395133)

[11.19.1.6 发布 297](#_Toc46395134)

[11.20 模版 297](#_Toc46395135)

[11.20.1 布局（Layout） 298](#_Toc46395136)

[11.20.2 局部模版（Partial） 298](#_Toc46395137)

[11.20.2.1 局部变量 299](#_Toc46395138)

[11.20.3 优化 299](#_Toc46395139)

[11.21 Hexo变量 300](#_Toc46395140)

[11.21.1.1 全局变量 300](#_Toc46395141)

[11.21.1.2 网站变量 301](#_Toc46395142)

[11.21.1.3 页面变量 301](#_Toc46395143)

[11.22 辅助函数（Helpers） 303](#_Toc46395144)

[11.22.1 网址 303](#_Toc46395145)

[11.22.1.1 url\_for 303](#_Toc46395146)

[11.22.1.2 relative\_url 304](#_Toc46395147)

[11.22.1.3 gravatar 304](#_Toc46395148)

[11.22.1.4 full\_url\_for 305](#_Toc46395149)

[11.22.2 HTML 标签 305](#_Toc46395150)

[11.22.2.1 css 305](#_Toc46395151)

[11.22.2.2 js 306](#_Toc46395152)

[11.22.2.3 link\_to 307](#_Toc46395153)

[11.22.2.4 mail\_to 307](#_Toc46395154)

[11.22.2.5 image\_tag 308](#_Toc46395155)

[11.22.2.6 favicon\_tag 308](#_Toc46395156)

[11.22.2.7 feed\_tag 308](#_Toc46395157)

[11.22.3 条件函数 308](#_Toc46395158)

[11.22.3.1 is\_current 309](#_Toc46395159)

[11.22.3.2 is\_home 309](#_Toc46395160)

[11.22.3.3 is\_post 309](#_Toc46395161)

[11.22.3.4 is\_page 309](#_Toc46395162)

[11.22.3.5 is\_archive 309](#_Toc46395163)

[11.22.3.6 is\_year 309](#_Toc46395164)

[11.22.3.7 is\_month 310](#_Toc46395165)

[11.22.3.8 is\_category 310](#_Toc46395166)

[11.22.3.9 is\_tag 310](#_Toc46395167)

[11.22.4 字符串处理 310](#_Toc46395168)

[11.22.4.1 trim 310](#_Toc46395169)

[11.22.4.2 strip\_html 310](#_Toc46395170)

[11.22.4.3 titlecase 311](#_Toc46395171)

[11.22.4.4 markdown 311](#_Toc46395172)

[11.22.4.5 render 311](#_Toc46395173)

[11.22.4.6 word\_wrap 311](#_Toc46395174)

[11.22.4.7 truncate 312](#_Toc46395175)

[11.22.5 模板 312](#_Toc46395176)

[11.22.5.1 partial 312](#_Toc46395177)

[11.22.5.2 fragment\_cache 312](#_Toc46395178)

[11.22.6 日期与时间 313](#_Toc46395179)

[11.22.6.1 date 313](#_Toc46395180)

[11.22.6.2 date\_xml 313](#_Toc46395181)

[11.22.6.3 time 313](#_Toc46395182)

[11.22.6.4 full\_date 314](#_Toc46395183)

[11.22.6.5 moment 314](#_Toc46395184)

[11.22.7 列表 314](#_Toc46395185)

[11.22.7.1 list\_categories 314](#_Toc46395186)

[11.22.7.2 list\_tags 315](#_Toc46395187)

[11.22.7.3 list\_archives 316](#_Toc46395188)

[11.22.7.4 list\_posts 316](#_Toc46395189)

[11.22.7.5 tagcloud 317](#_Toc46395190)

[11.22.8 其他 318](#_Toc46395191)

[11.22.8.1 paginator 318](#_Toc46395192)

[11.22.8.2 search\_form 319](#_Toc46395193)

[11.22.8.3 number\_format 319](#_Toc46395194)

[11.22.8.4 meta\_generator 320](#_Toc46395195)

[11.22.8.5 open\_graph 320](#_Toc46395196)

[11.22.8.6 toc 321](#_Toc46395197)

[11.23 插件 321](#_Toc46395198)

[11.23.1.1 脚本（Scripts） 322](#_Toc46395199)

[11.23.1.2 插件（Packages） 322](#_Toc46395200)

[11.23.1.3 工具 322](#_Toc46395201)

[11.23.1.4 发布 323](#_Toc46395202)

[11.24 国际化（i18n） 323](#_Toc46395203)

[11.24.1.1 语言文件 324](#_Toc46395204)

[11.24.1.2 模板 324](#_Toc46395205)

[11.24.1.3 路径 324](#_Toc46395206)

[11.25 Hexo多语言化 325](#_Toc46395207)

[11.25.1 1.前言 325](#_Toc46395208)

[11.25.2 2.前置条件 325](#_Toc46395209)

[11.25.2.1 2.1 环境准备 325](#_Toc46395210)

[11.25.2.2 2.2 软件安装 325](#_Toc46395211)

[11.25.2.3 2.3 hexo依赖 326](#_Toc46395212)

[11.25.2.4 2.4 npm依赖 326](#_Toc46395213)

[11.25.3 3.折腾过程 326](#_Toc46395214)

[11.25.3.1 3.1 全站配置 326](#_Toc46395215)

[11.25.3.1.1 3.1.1 语系配置 326](#_Toc46395216)

[11.25.3.1.2 3.1.2 主题配置 327](#_Toc46395217)

[11.25.3.1.3 3.1.3 url配置 327](#_Toc46395218)

[11.25.3.1.4 3.1.4 deploy配置 327](#_Toc46395219)

[11.25.3.2 3.2 主题配置 327](#_Toc46395220)

[11.25.3.2.1 3.2.1 主题配置生成 327](#_Toc46395221)

[11.25.3.2.2 3.2.2 多语系配置 327](#_Toc46395222)

[11.25.3.3 3.3 站点文档配置 328](#_Toc46395223)

[11.25.3.3.1 3.3.1 source/\_post目录文档配置 328](#_Toc46395224)

[11.25.3.3.2 3.3.2 source目录文档配置 329](#_Toc46395225)

[11.25.3.3.3 3.3.3 增加多语系切换配置 330](#_Toc46395226)

[11.25.3.4 3.4 查看效果 330](#_Toc46395227)

[11.25.3.4.1 3.4.1 编译静态文件 330](#_Toc46395228)

[11.25.3.4.2 3.4.2 本地测试 330](#_Toc46395229)

[11.25.3.5 3.5 部署到github 332](#_Toc46395230)

[11.25.4 4. 其它功能 332](#_Toc46395231)

[11.25.5 4.1 添加评论功能 332](#_Toc46395232)

[11.25.5.1.1 4.1.1 创建应用 333](#_Toc46395233)

[11.25.5.1.2 4.1.2 查看应用app\_id和app\_key 333](#_Toc46395234)

[11.25.5.1.3 4.1.3 配置应用key 333](#_Toc46395235)

[11.25.5.1.4 4.1.4 文章开启comment或关闭comment 334](#_Toc46395236)

[11.25.5.2 4.2 添加文章访问量及站点访问统计功能 334](#_Toc46395237)

[11.25.5.2.1 4.2.1 修改主题配置 334](#_Toc46395238)

[11.25.5.2.2 4.2.2 增加busuanzi脚本 335](#_Toc46395239)

[11.25.5.2.3 4.2.3 增加访问量统计 335](#_Toc46395240)

[11.25.5.2.4 4.2.4 修改多语系配置 335](#_Toc46395241)

[11.25.5.2.5 4.2.5 查看效果 336](#_Toc46395242)

[11.25.6 5.总结 336](#_Toc46395243)

[11.26 Hexo的多语言配置 336](#_Toc46395244)

[11.26.1 插件安装 336](#_Toc46395245)

[11.26.2 博文配置 337](#_Toc46395246)

[11.26.3 网页配置 337](#_Toc46395247)

[11.26.4 添加切换语言按钮 338](#_Toc46395248)

[11.26.5 部署测试 339](#_Toc46395249)

[11.26.5.1 339](#_Toc46395250)

[11.26.5.2 339](#_Toc46395251)

[11.26.5.3 339](#_Toc46395252)

[11.27 更新 Hexo 官方插件 339](#_Toc46395253)

[11.28 问题解答 340](#_Toc46395254)

[11.28.1 YAML 解析错误 340](#_Toc46395255)

[11.28.2 EMFILE 错误 340](#_Toc46395256)

[11.28.3 Git 部署问题 340](#_Toc46395257)

[11.28.3.1 RPC failed 340](#_Toc46395258)

[11.28.3.2 Error: ENOENT: no such file or directory 341](#_Toc46395259)

[11.28.4 服务器问题 341](#_Toc46395260)

[11.28.5 插件安装问题 341](#_Toc46395261)

[11.28.6 DTrace 错误 （Mac OS X） 341](#_Toc46395262)

[11.28.7 在 Jade 或 Swig 遍历资料 341](#_Toc46395263)

[11.28.8 资料没有更新 342](#_Toc46395264)

[11.28.9 命令没有执行 342](#_Toc46395265)

[11.28.10 泄露（Escape）内容 342](#_Toc46395266)

[11.28.11 ENOSPC 错误 （Linux） 342](#_Toc46395267)

[11.28.12 EMPERM Error (Windows Subsystem for Linux) 343](#_Toc46395268)

[11.28.13 模板渲染错误 343](#_Toc46395269)

[11.28.14 开始之前 344](#_Toc46395270)

[11.28.15 工作流程 344](#_Toc46395271)

[11.28.16 注意事项 344](#_Toc46395272)

[11.29 更新文档 345](#_Toc46395273)

[11.29.1 工作流程 345](#_Toc46395274)

[11.29.2 翻译 345](#_Toc46395275)

[11.30 反馈问题 345](#_Toc46395276)

[11.31 Hexo 巧用 abbrlink 插件实现文章多语言版本 346](#_Toc46395277)

[11.31.1 使用方法 346](#_Toc46395278)

[11.31.2 在首页中隐藏不需要的语言 347](#_Toc46395279)

[11.32 hexo-abbrlink 347](#_Toc46395280)

[11.32.1 How to install 347](#_Toc46395281)

[11.32.2 Sample 348](#_Toc46395282)

[11.32.3 Limitation 349](#_Toc46395283)

[11.32.4 More info 349](#_Toc46395284)

[11.33 hexo-hide-posts 349](#_Toc46395285)

[11.33.1 安装 350](#_Toc46395286)

[11.33.2 使用 350](#_Toc46395287)

[11.33.3 配置 350](#_Toc46395288)

[11.34 设置 RSS 352](#_Toc46395289)

[11.34.1.1 添加「标签」页面 352](#_Toc46395290)

[11.34.1.2 添加「分类」页面 353](#_Toc46395291)

[11.34.1.3 设置字体 353](#_Toc46395292)

[11.34.1.4 设置代码高亮主题 355](#_Toc46395293)

[11.34.1.5 侧边栏社交链接 356](#_Toc46395294)

[11.34.1.6 开启打赏功能 由 habren 贡献 357](#_Toc46395295)

[11.34.1.7 友情链接 由 iamwent 贡献 357](#_Toc46395296)

[11.34.1.8 腾讯公益404页面 由 xirong 贡献 357](#_Toc46395297)

[11.34.1.9 站点建立时间 358](#_Toc46395298)

[11.34.1.10 订阅微信公众号 由 huiwang 贡献 358](#_Toc46395299)

[11.34.1.11 设置「动画效果」 359](#_Toc46395300)

[11.34.1.12 设置「背景动画」 359](#_Toc46395301)

[11.35 第三方服务来 359](#_Toc46395302)

[11.35.1 评论系统 360](#_Toc46395303)

[11.35.1.1 DISQUS 360](#_Toc46395304)

[11.35.1.2 Facebook Comments 由 hydai 贡献 360](#_Toc46395305)

[11.35.1.3 HyperComments 由 ivan-nginx 贡献 360](#_Toc46395306)

[11.35.1.4 网易云跟帖 由 geekrainy 贡献 361](#_Toc46395307)

[11.35.1.5 来必力 由 asmoker 贡献 361](#_Toc46395308)

[11.35.2 数据统计与分析 361](#_Toc46395309)

[11.35.2.1 百度统计 361](#_Toc46395310)

[11.35.2.2 Google 分析 362](#_Toc46395311)

[11.35.2.3 腾讯分析 由 Cissoid 贡献 362](#_Toc46395312)

[11.35.2.4 CNZZ 统计 由 jerrybendy 贡献 362](#_Toc46395313)

[11.35.2.5 不蒜子统计 由 panzhitian 贡献 363](#_Toc46395314)

[11.35.2.6 腾讯移动分析 由 aleonchen 贡献 363](#_Toc46395315)

[11.35.2.7 阅读次数统计（LeanCloud) 由 Doublemine 贡献 363](#_Toc46395316)

[11.35.3 内容分享服务 363](#_Toc46395317)

[11.35.3.1 JiaThis 363](#_Toc46395318)

[11.35.3.2 百度分享 363](#_Toc46395319)

[11.35.3.3 AddThis 由 hackjustu 贡献 364](#_Toc46395320)

[11.35.4 搜索服务 364](#_Toc46395321)

[11.35.4.1 Swiftype 站内搜索 由 lzlun129 贡献 364](#_Toc46395322)

[11.35.4.2 微搜索 由 lzlun129 贡献 365](#_Toc46395323)

[11.35.4.3 Local Search 由 flashlab 贡献 365](#_Toc46395324)

[11.35.4.4 Algolia 365](#_Toc46395325)

[11.35.5 其他服务 366](#_Toc46395326)

[11.35.5.1 MathJax 366](#_Toc46395327)

[11.35.5.2 Facebook SDK 由 changyuheng 贡献 367](#_Toc46395328)

[11.35.5.3 Google Webmaster tools 367](#_Toc46395329)

[11.36 设置 「JavaScript 第三方库」 367](#_Toc46395330)

[11.36.1 如何设置「阅读全文」？ 369](#_Toc46395331)

[11.36.2 Favicon 设置后没有生效？ 370](#_Toc46395332)

[11.36.3 如何更改字体？ 370](#_Toc46395333)

[11.36.4 如何更改内容区域的宽度？ 370](#_Toc46395334)

[11.36.5 为何新增的菜单项显示 Menu.xxx 371](#_Toc46395335)

[11.36.6 标签/分类数量统计不准确？ 371](#_Toc46395336)

[11.36.7 为何语言显示不正确？ 372](#_Toc46395337)

[11.36.8 如何关闭新建页面的评论功能？ 372](#_Toc46395338)

[11.36.9 如何设置页面文章的篇数？ 373](#_Toc46395339)

[11.36.10 如何优化 NexT 主题？ 373](#_Toc46395340)

[11.36.11 Cannot find module XXX 373](#_Toc46395341)

[11.37 Hexo插件 374](#_Toc46395342)

[11.38 Hexo的数据DB扩展查询 376](#_Toc46395343)

[11.39 总结 377](#_Toc46395344)

[11.40 新建仓库 378](#_Toc46395345)

[11.40.1.1 2. 全局安装hexo 378](#_Toc46395346)

[11.40.1.2 3. 初始化项目 378](#_Toc46395347)

[11.40.1.3 4. 部署到github 378](#_Toc46395348)

[11.40.1.4 5. 查看效果 379](#_Toc46395349)

[11.40.1.5 如何创建新文章 379](#_Toc46395350)

[11.40.1.6 更换主题皮肤 379](#_Toc46395351)

[11.40.1.7 部署优化 380](#_Toc46395352)

[11.40.1.8 接入评论系统 - valine 380](#_Toc46395353)

[11.41 Hexo主题开发实践 380](#_Toc46395354)

[11.41.1 前言 380](#_Toc46395355)

[11.41.2 目录结构 381](#_Toc46395356)

[11.41.3 局部模板 381](#_Toc46395357)

[11.41.4 首页 382](#_Toc46395358)

[11.41.5 编写导航栏和底部信息 383](#_Toc46395359)

[11.41.6 添加主题配置 385](#_Toc46395360)

[11.41.7 添加文章列表 386](#_Toc46395361)

[11.41.8 添加样式 387](#_Toc46395362)

[11.41.9 添加分页 389](#_Toc46395363)

[11.41.10 添加文章详情页 390](#_Toc46395364)

[11.41.11 添加归档页 391](#_Toc46395365)

[11.41.12 添加分类页 392](#_Toc46395366)

[11.41.13 添加标签页 393](#_Toc46395367)

[11.41.14 添加自定义页面 394](#_Toc46395368)

[11.42 将Hexo部署到个人网站上 395](#_Toc46395369)

[11.42.1 修改配置 395](#_Toc46395370)

[11.42.2 执行部署命令 396](#_Toc46395371)

[11.43 Hexo资源 396](#_Toc46395372)

[11.44 Hexo主题资源 396](#_Toc46395373)

[11.44.1 主题NexT 396](#_Toc46395374)

[11.44.1.1 安装 NexT 397](#_Toc46395375)

[11.44.1.1.1 下载主题 397](#_Toc46395376)

[11.44.1.1.2 启用主题 397](#_Toc46395377)

[11.44.1.1.3 验证主题 398](#_Toc46395378)

[11.44.1.2 主题设定 399](#_Toc46395379)

[11.44.1.2.1 选择 Scheme 399](#_Toc46395380)

[11.44.1.2.2 设置 语言 399](#_Toc46395381)

[11.44.1.2.3 设置 菜单 400](#_Toc46395382)

[11.44.1.2.4 设置 侧栏 401](#_Toc46395383)

[11.44.1.2.5 设置 头像 402](#_Toc46395384)

[11.44.1.2.6 设置 作者昵称 403](#_Toc46395385)

[11.44.1.2.7 站点描述 403](#_Toc46395386)

[11.44.1.3 NexT标签 403](#_Toc46395387)

[11.44.1.3.1 文本居中的引用 403](#_Toc46395388)

[11.44.1.3.2 突破容器宽度限制的图片 404](#_Toc46395389)

[11.44.1.3.3 Bootstrap Callout 由 ivan-nginx 贡献 405](#_Toc46395390)

[11.44.1.3.3.2 集成常用的第三方服务 407](#_Toc46395391)

[11.44.2 主题hexo icarus 408](#_Toc46395392)

[11.44.2.1 安装icarus 408](#_Toc46395393)

[11.44.2.2 配置icarus 409](#_Toc46395394)

[11.44.2.2.1 配置 icarus 主题 409](#_Toc46395395)

[11.44.2.2.2 配置文章部分 409](#_Toc46395396)

[11.44.2.2.3 顶部图片添加 409](#_Toc46395397)

[11.44.2.2.4 左边文章导航栏开启 410](#_Toc46395398)

[11.44.2.2.5 提取文章目录 410](#_Toc46395399)

[11.44.2.2.6 评论系统开启 410](#_Toc46395400)

[11.44.2.2.7 捐赠收款开启 411](#_Toc46395401)

[11.44.2.2.8 全局搜索开启 411](#_Toc46395402)

[11.44.2.2.9 更多配置请参考官网配置 411](#_Toc46395403)

[11.44.2.3 自定义 icarus 主题 412](#_Toc46395404)

[11.44.2.4 代码高亮 412](#_Toc46395405)

[11.44.2.5 更新logo 412](#_Toc46395406)

[11.44.2.6 页面类型判定 412](#_Toc46395407)

[11.44.2.7 布局 412](#_Toc46395408)

[11.44.2.7.1 文章页面两栏布局 412](#_Toc46395409)

[11.44.2.7.2 调整阅读模式双栏 414](#_Toc46395410)

[11.44.2.7.3 优化文章标题布局 415](#_Toc46395411)

[11.44.2.7.4 优化文章结尾布局 417](#_Toc46395412)

[11.44.2.7.5 优化个人信息布局 418](#_Toc46395413)

[11.44.2.7.6 优化移动端显示 418](#_Toc46395414)

[11.44.2.7.7 目录粘性布局 418](#_Toc46395415)

[11.44.2.8 功能 419](#_Toc46395416)

[11.44.2.8.1 增加版权说明 419](#_Toc46395417)

[11.44.2.8.2 增加文章版权 420](#_Toc46395418)

[11.44.2.8.3 配置 gitalk 评论 421](#_Toc46395419)

[11.44.2.8.4 增加标题自动计数 421](#_Toc46395420)

[11.44.2.8.5 默认显示目录 422](#_Toc46395421)

[11.44.2.8.6 页面footer显示一组icon 422](#_Toc46395422)

[11.44.2.8.7 可以配置文章开头是否显示图片 423](#_Toc46395423)

[11.44.2.8.8 ShareThis 424](#_Toc46395424)

[11.44.2.8.9 总访问量统计 426](#_Toc46395425)

[11.44.2.9 样式 427](#_Toc46395426)

[11.44.2.9.1 修改 logo 和 favicon 427](#_Toc46395427)

[11.44.2.9.2 按钮背景颜色增加渐变 427](#_Toc46395428)

[11.44.2.9.3 card 增加浮动效果 427](#_Toc46395429)

[11.44.2.9.4 更新 428](#_Toc46395430)

[11.44.2.9.5 总结 428](#_Toc46395431)

[11.44.2.10 参考资料 429](#_Toc46395432)

[11.44.2.11 修改要点 429](#_Toc46395433)

[12 Jekyll 429](#_Toc46395434)

[12.1 安装 430](#_Toc46395435)

[12.2 增加新的pem证书，解决问题 430](#_Toc46395436)

[12.3 主题 436](#_Toc46395437)

[12.4 前言 436](#_Toc46395438)

[12.5 对主题的要求 436](#_Toc46395439)

[12.6 推荐主题 436](#_Toc46395440)

[12.6.1 2.1 NexT 437](#_Toc46395441)

[12.6.2 2.2 Hux Blog 437](#_Toc46395442)

[12.6.3 2.3 码志 437](#_Toc46395443)

[12.7 附上一些主题站点 437](#_Toc46395444)

[12.8 Jekyll资源 448](#_Toc46395445)

[12.9 Jekyll主题资源 449](#_Toc46395446)

[12.9.1 NexT 449](#_Toc46395447)

[12.9.2 minimal-mistakes 449](#_Toc46395448)

[12.9.3 码志 449](#_Toc46395449)

[13 MKDocs 449](#_Toc46395450)

[13.1 mkdocs-material介绍 450](#_Toc46395451)

[13.2 安装 450](#_Toc46395452)

[**初始化项目** 451](#_Toc46395453)

[**修改主题** 451](#_Toc46395454)

[13.3 运行 451](#_Toc46395455)

[13.4 mkdocs.yml注意事项 451](#_Toc46395456)

[13.5 添加页面 452](#_Toc46395457)

[13.6 添加扩展 452](#_Toc46395458)

[13.7 主题 453](#_Toc46395459)

[13.8 使用MkDocs 管理博客 454](#_Toc46395460)

[13.8.1 需求 454](#_Toc46395461)

[13.8.2 MkDocs 优点 454](#_Toc46395462)

[13.8.3 方案 454](#_Toc46395463)

[13.8.4 试验环境 454](#_Toc46395464)

[13.9 Cinder 456](#_Toc46395465)

[13.10 MKDocs资源 457](#_Toc46395466)

[13.11 MKDocs主题资源 457](#_Toc46395467)

[14 Vue 457](#_Toc46395468)

[15 React 457](#_Toc46395469)

[16 Angular 457](#_Toc46395470)

[17 Django 458](#_Toc46395471)

[18 附录：资源 458](#_Toc46395472)

# 资料说明

本资料为个人学习笔记，随时更新

资源及参考网址统一收录在附录中

本资料保存网址：https://github.com/SB-FOSS/SBF-Resources

# 快速笔记

## 站点设置

使用 Jekyll 在本地测试 GitHub Pages 站点

https://docs.github.com/cn/github/working-with-github-pages/testing-your-github-pages-site-locally-with-jekyll

https://docs.getpelican.com/en/latest/#

网址

https://sb-talents.github.io

https://sb-talents.github.io/SBT-Japanese/

https://github.com/SB-Talents/SBT-English

https://SB-Talents.github.io/SBT-Business-Competence

https://github.com/SB-Talents/SBT-Japanese

https://github.com/SB-Talents/SBT-Book

https://github.com/SB-Talents/SBT-News

Professional competence knowledge of Smart Bridge Talents

Document template of Smart Bridge Talents

Professional-competence

English learning media of Smart Bridge Talents

Japanese learning media of Smart Bridge Talents

Intelligence-Decision-Supporting-System

Educational-Technology

Adaptive learning

MENUITEMS = [

('Home', '/'),

('Archives', [

('Tags', '/tags.html'),

('Categories', '/categories.html'),

('Chronological', '/archives.html'),

]),

('Social', [

('Email', 'mailto: maurelinus@stoic.edu'),

('Github', 'http://url-to-github-page'),

('Facebook', 'http://url-to-facebook-page'),

]),

]

## md编辑

url: https://sb-talents.github.io

root: /The-Practice-of-Business-Competence/

echo %path%

## Git命令

git clone https://github.com/ppoffice/hexo-theme-icarus.git themes/icarus

git clone https://github.com/mmistakes/minimal-mistakes.git minimal-mistakes

======================================================

配置项：永久

# permalink: :year/:month/:day/:title/

permalink: :lang/:category/:title/

## Pelican常用命令

pelican ../content/ -o ../output -s ../pelicanconf.py

pelican ../content-SB-Talents/ -o ../output-SB-Talents -s ../pelicanconf-SB-Talents.py

pelican ../content-SBT-Business-Competence/ -o ../output-SBT-Business-Competence -s ../pelicanconf-SBT-Business-Competence.py

pelican ../content-SBT-CRI-News/ -o ../output-SBT-CRI-News -s ../pelicanconf-SBT-CRI-News.py

python -m pelican.server

## Hexo常用命令

hexo clean & hexo g & hexo s

https://hexo.io/docs/

https://hexo.io/zh-cn/docs/

http://theme-next.iissnan.com

主题一览

https://github.com/hexojs/hexo/wiki/Themes

## Jekyll

//创建jekyll工程目录

jekyll new myblog

//切换到工程目录，并开启服务

cd myblog

jekyll serve

gem install bundler

bundle install

rake theme:install name="minimal-mistakes"

===========================================

gem

gem sources -l

gem update --system

Hexo

测试

hexo clean & hexo g & hexo s

创建内容

hexo new page "中国" --lang zh-CN

安装Thems NexT

git clone https://github.com/next-theme/hexo-theme-next themes/next

安装Thems icarus

https://github.com/ppoffice/hexo-theme-icarus

git clone https://github.com/ppoffice/hexo-theme-icarus.git themes/icarus

创建项目

hexo init myblog

cd your-hexo-site

插件

npm install hexo-abbrlink --save

配置npm源

npm config set registry https://registry.npm.taobao.org

安装Hexo

npm install -g hexo-cli

hexo -v

替换谷歌字体，解决谷歌被墙的问题

最近由于大家都心知肚明的原因，谷歌被彻底封禁。这次封禁直接导致许多谷歌服务无法使用。包括wordpress中的谷歌字体加载问题。

由于谷歌被封，所以网站在打开的时候，由于加载字体失败，左下角一直显示在加载http://fonts.googleapis.com/。导致网站前台后台打开速度极慢。目前流行的解决办法有两种。

一，代码插件法。这种方法不喜欢，就不多说了。

二，替换法。360已经将谷歌字体放在了自己的服务器，更改代码让网站从360服务器加载字体。

打开wordpress文件： wp-includes/script-loader.php 搜索关键字： fonts.googleapis.com 替换：fonts.googleapis.com替换为fonts.useso.com ，

保存之后，字体加载问题迎刃而解。360总算为广大网友做了件好事情。

我使用的是替换法。效果很明显。

http://www.sb-hrbp.com/Headhunters/assets/img/logo.png

<iframe frameborder="0" width="1100" height="560"

src="https://v.qq.com/txp/iframe/player.html?vid=t30485bq0l5&autoplay=true" allowFullScreen="true" ></iframe>

参数:autoplay: Boorer 是否自动播放

vid:String 视频的vid

https://links.jianshu.com/go?to=https%3A%2F%2Ftgideas.qq.com%2Fdoc%2Ffrontend%2Fcomponent%2Fcommon%2Ftxplayer.html

想要在网页中嵌入优酷、土豆、爱奇艺的视频很简单，但是现在将腾讯视频嵌入网页后却不能正常显示，而腾讯视频现在的资源也越来越多，想直接将腾讯视频嵌入到网页中怎么操作呢？其实只需要一段代码就可以了。

首先，提供一个腾讯视频嵌入网页代码。我们只需替换这段代码的一部分就可以了。

<p style="text-align: center"><iframe class="video\_iframe" style="z-index:1;" src="http://v.qq.com/iframe/player.html?vid=t01662frswa&amp;width=500&amp;height=375&amp;auto=0" allowfullscreen="" frameborder="0" height="375" width="500"></iframe></p>

找到一段腾讯视频，比如地址：

http://v.qq.com/boke/page/d/0/i/d0163kxz8di.html

这是一段《工业机器人工作场景》的视频，将这个网址的蓝色部分，替换到上面给出的代码中的红色部分，就可以了。

最终效果为：

<p style="text-align: center"><iframe class="video\_iframe" style="z-index:1;" src="http://v.qq.com/iframe/player.html?vid=d0163kxz8di&amp;width=500&amp;height=375&amp;auto=0" allowfullscreen="" frameborder="0" height="375" width="500"></iframe></p>

把vid参数从腾讯视频网页查找

想将优酷视频、土豆视频、新浪视频等网站的视频调用到个人网站上面的方法非常简单，这些网站，均提供了直接调用的代码。如下图所示：

1、优酷网

youku.jpg

2、土豆网

tudou.jpg

3、新浪视频

xinlang.jpg

但是腾讯视频则是这样的（并没有提供嵌入网页的代码）：

tengxun.jpg

应该怎么办呢？

可以从微信公众平台中学习一下，因为腾讯视频，可以直接嵌入微信公众平台啊？点击查看：微信公众平台转载别人发的视频的方法

方法一：将视频在微信公众平台中的调用代码提取出来，如下所示：

<iframe class="video\_iframe" style=" z-index:1; " src="http://v.qq.com/iframe/player.html?vid=m0137rrajuc&amp;width=300&amp;height=200&amp;auto=0" allowfullscreen="" frameborder="0" height="200" width="300"></iframe>

打开需要添加的腾讯视频网址，例如：

http://v.qq.com/cover/z/zrxyhghf3n8xhxl/k0015trfczz.html

将上述代码中的vid=后面的内容替换成上述网址中标红的部分，然后将上述网址重新复制出来：

<iframe class="video\_iframe" style=" z-index:1; " src="http://v.qq.com/iframe/player.html?vid=k0015trfczz&amp;width=300&amp;height=200&amp;auto=0" allowfullscreen="" frameborder="0" height="200" width="300"></iframe>

效果参见下方：

其中的视频的宽度（width）和高度（height）可以分别通过修改代码中对应的数值（本视频中width=300，height=200）来进行修改。

不自动播放

<iframe class="video\_iframe" style=" z-index:1; " src="http://v.qq.com/iframe/player.html?vid=k0344pfymze&tiny=0&auto=0&isAutoPlay=false;width=300&amp;height=200&amp;auto=0" allowfullscreen="" frameborder="0" height="200" width="300"></iframe>

# 开源许可证

**Github 项目的开源许可证**

Github 官方专门制作了一个网站 [https://choosealicense.com/](https://link.zhihu.com/?target=https%3A//choosealicense.com/) 帮助大家选择合适的开源，License。中文版也有 [http://choosealicense.online/](https://link.zhihu.com/?target=http%3A//choosealicense.online/)。这里我对这些如果选择 Github 项目的 License ，给出一些具体的操作建议。

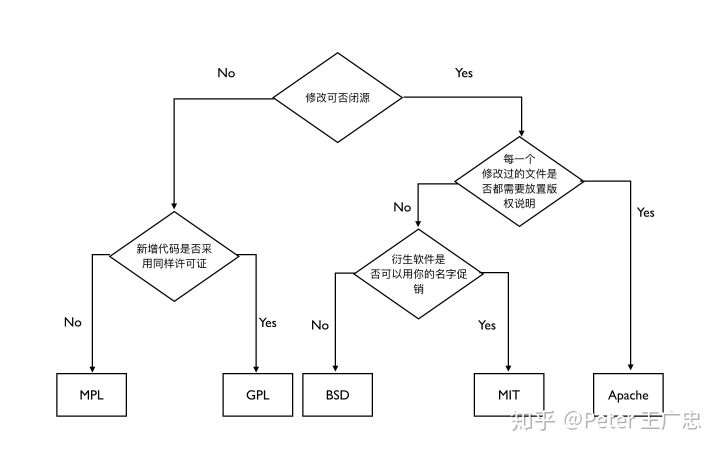
## 可不可以不选择 License ？

没有 License 的内容是默认会被版权保护。所以如果你想要的是让大家都放心使用，就需要选择一个合适的 License ，只有这样才能赋予任何人使用，分享和修改这个软件的权力。

所以，如果你只是想奉献爱心，想让大家无限制的使用自己仓库的代码，选择 MIT 协议即可。MIT License 是一个宽松的 License ，允许别人用你的代码做任何事情，但必须保证你的所有权，并且你无须承担代码使用产生的风险。

## 具体选择标准

开源 License 很多，具体的差别可以看一下下面这个图。



总结一下，MIT 最自由，简直就是没有任何限制，任何人都可以售卖我的软件，甚至可以用我的名字促销。BSD 和 Apache 协议也很自由，跟 MIT 的区别分别是不允许用作者本人名义促销和保护作者版权。GPL 可以说最霸道，对代码的修改部分也必须是 GPL 的，同时基于 GPL 代码而开发的代码也必须按照 GPL 发布，而 MPL ，也就是 Mozilla Public License 就温和一些，如果后续开发的代码中添加了新文件，同时新文件中也没有用到原来的代码，那么新文件可以不必继续沿用 MPL 。

## MIT

## GPL 家族

GPL 是最著名的开源 License，没有之一。目前在 Github 创建项目的时候，一共有12个备选 License ，其中5个都是 GPL 家族的。

人们对 GPL 感情是复杂的。激进的人会比较喜欢 GPL ，因为它要求任何人修改了软件后，修改后的内容也必须开源。但是商业公司有时候会非常有顾虑，因为 GPL 有病毒性，一旦公司使用了 GPL 的代码，那么自己辛辛苦苦做出的修改内容也必须要通过 GPL 开源，让竞争对手也可以直接拿来用，很难形成竞争壁垒。

GPL 协议也发展出了很多分支，其中 GPL v3 是最为激进的，基本上跟原始代码沾点边的代码就必须也得是 GPL 的，例如，最极端的，如果我的代码调用了 GPL 的库，那么我的代码就必须是 GPL 的。这基本意味着如果我是一个商业软件系统，那么我就没有权利使用 GPL v3 的代码了。v3 的背后是 GPL 之父 Richard Stallman 不断在宣传推进，代表了开源激进派的最前沿。而其他的 GPL 版本可以说都是略微温和版的 GPL ，例如 Linux 项目的 GPL v2 ，也是 Richard 自己写的，由于发布的早，所以很多问题他没有考虑都，所以让商业运用有了一些空间。另外还有 GNU Lesser General Public License ，GNU较宽松通用公共许可证 ，看名字就知道是比较温和的了。

这就是 GPL 的基本情况了。关于 GPL 可以专门写一本《GPL 传奇》了，我们这里不展开，大家可以去 [http://gnu.org](https://link.zhihu.com/?target=http%3A//gnu.org) 学习或者听一下 Richard Stallman 的演讲。

## 总结

关于，在 Github 使用开源 license ，还有其他一些要注意的地方，例如 license 要存放到哪个文件中，如果按照协议类似搜索项目等，这些内容可以参考官方文档 [https://help.github.com/articles/licensing-a-repository/](https://link.zhihu.com/?target=https%3A//help.github.com/articles/licensing-a-repository/) 。另外，如果项目内容不是代码，而是书稿或者其他作品，可以参考这里的说明 [http://choosealicense.online/non-software/](https://link.zhihu.com/?target=http%3A//choosealicense.online/non-software/) 使用 CC License 。

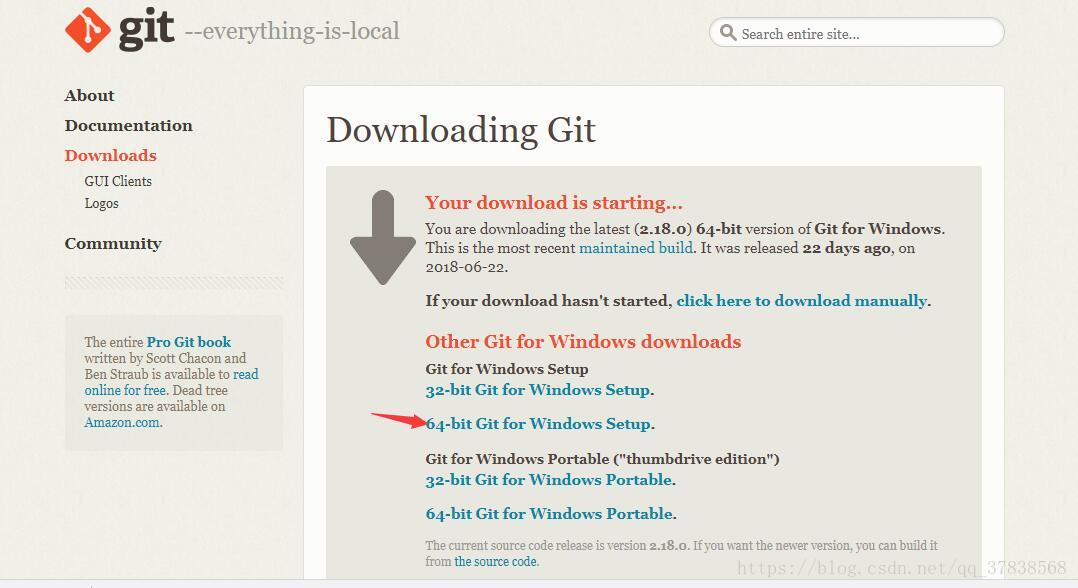
参考：

* [http://www.ruanyifeng.com/blog/2011/05/howtochoosefreesoftware\_licenses.html](https://link.zhihu.com/?target=http%3A//www.ruanyifeng.com/blog/20%253Ci%253E11%253C/i%253E/05/ho%253Ci%253Ew_to%253C/i%253E_choose_free_software_licenses.html)
* [https://help.github.com/articles/licensing-a-repository/](https://link.zhihu.com/?target=https%3A//help.github.com/articles/licensing-a-repository/)

# Git

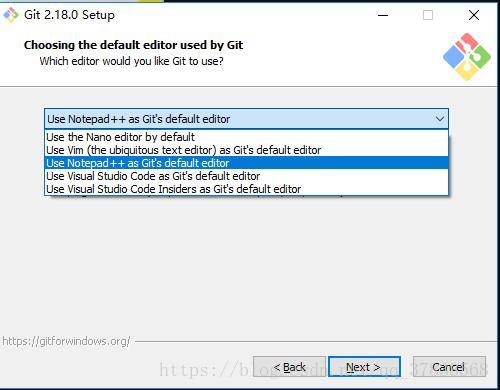
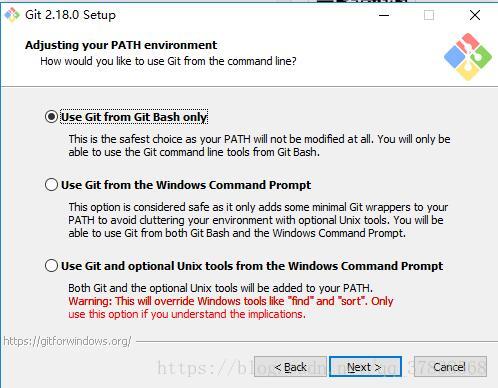
## Git Quick start

### 下载Git

[官网链接](https://git-scm.com/downloads)   
 进去之后会默认弹出下载，没有弹出的话就根据电脑位数选择下载。   


### 安装Git

 下载完成后打开就可以开始安装，更改安装位置后大部分都是默认选择，可以一路next直到完成，下面有几个地方是我安装过程中改过的，供大家参考：

 1.这里选择默认编辑器，我选择了notepad ++；   
    
 2.这里选择git自带的命令行工具（Git Bash）   
    
 后面继续一路next下去就完成安装了，安装完成后在文件夹右键就可以看见右键菜单多了两行。

   
至此Git的安装完成，

### 注册GitHub账号

[注册链接](https://github.com/join)  
 填写用户名、邮箱、密码即可，都是些常规操作，不多赘述。

### Git初始化设置

 1. 打开Git Bash  
   
 这是一个命令行工具，使用的命令与Linux下有些类似（操作技巧：使用TAB键补全命令、按下鼠标滚轮粘贴内容、上下键切换使用过的命令、左右键移动光标等）。  
 2.使用以下命令，设置用户名和邮箱

git config --global user.name '你在GitHub注册的用户名'

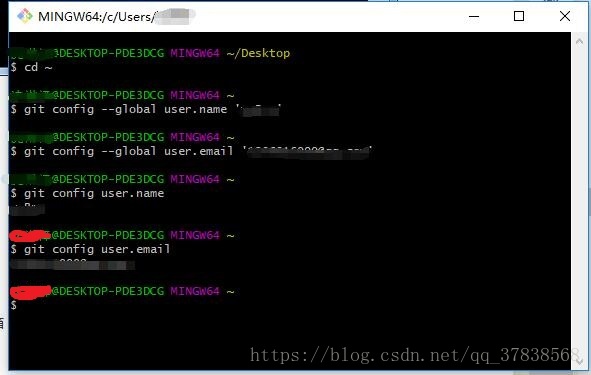
git config --global user.email '你在GitHub注册的邮箱'

 3.使用以下命令，查看设置的用户名和邮箱

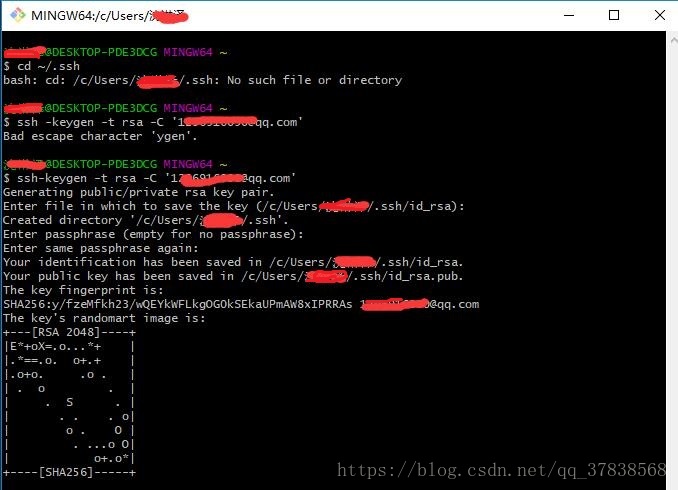
git config user.name

git config user.email

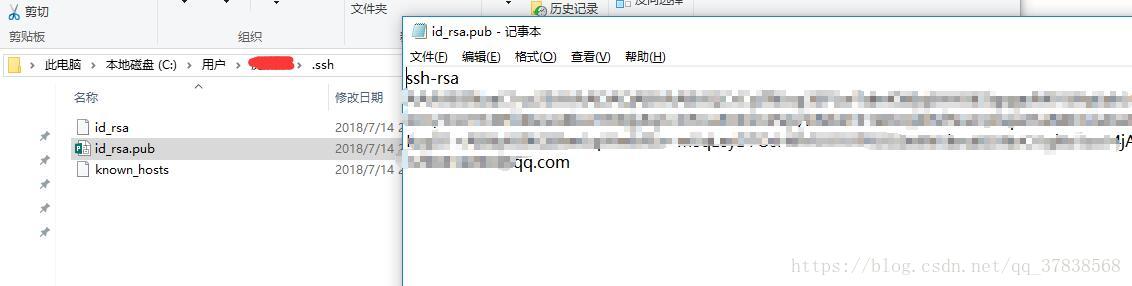
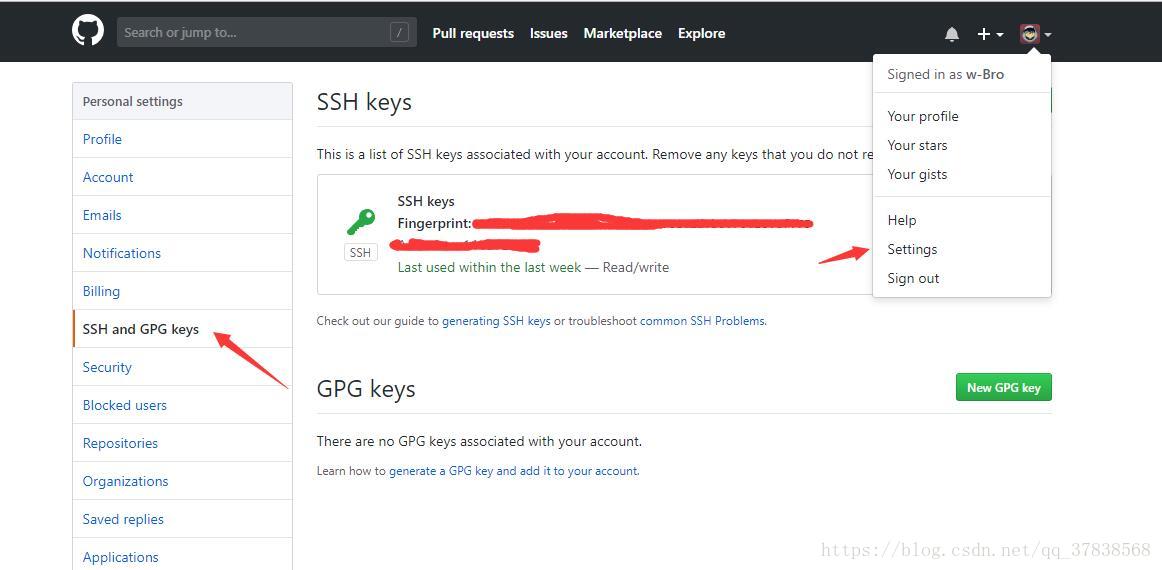
* 1
* 2

  
 4.生成SSH公钥

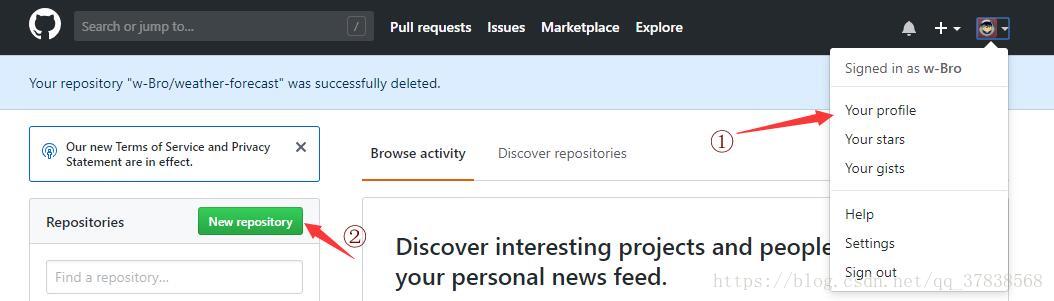
ssh-keygen -t rsa -C '你在GitHub注册的邮箱'

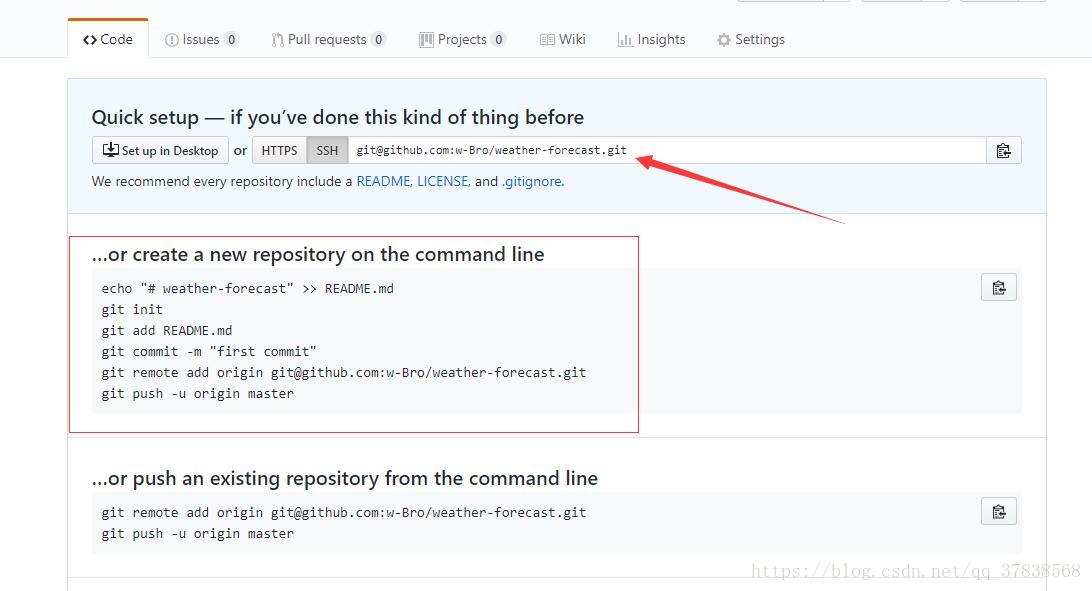
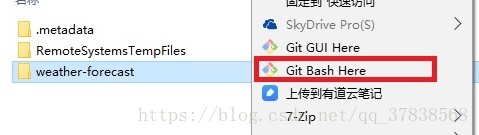
 在Git Bash输入这条命令，回车之后再按三个回车（默认存储位置，密码为空，确认密码），看到下图效果即可。  
  
 5.GitHub配置SSH公钥  
 在此路径下用记事本打开文件（id\_rsa.pub）并全选复制里面的内容

C:\Users\你的用户名\.ssh

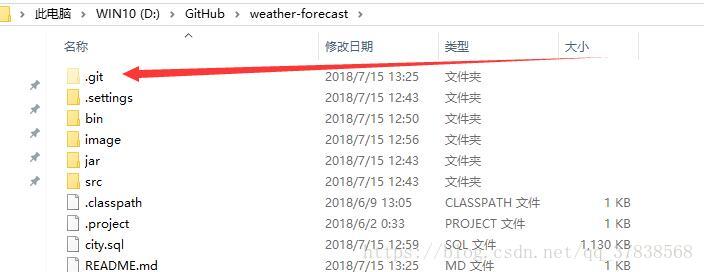
  
在GitHub点击你的头像，选择"Settings"，选择右边的"SSH and GPG Keys",选择New SSH Key，Title随便填，再把复制的公钥粘贴进Key里面，点击Add SSK key即完成。  


### 在GitHub上创建Repository（仓库）

 1.点击头像，选择"Your profile"，再选择右边的New repository  
  
 2.填写仓库名称及其描述(这里以我的小项目为例)  
  
 建议：红框内容最好不要勾选，到时候自己手动新建一个README.md即可。  
创建完成后效果如下：

  
 注意这里箭头指向的链接及红框的内容，后面我们会用到。  
#五、上传项目  
 1.在你想上传的项目文件夹右键，选中Git Bash Here，即会弹出命令行窗口  
  
 2.初始化仓库  
 在命令行中输入初始化命令：

git init

 这一步完成后，打开刚才选中的文件夹会出现一个隐藏文件夹.git  
  
 3.上传文件  
[如想创建README.md](http://xn--README-hn3j520cbtliye.md)，可以自己在文件夹中手动创建，也可以使用命令创建

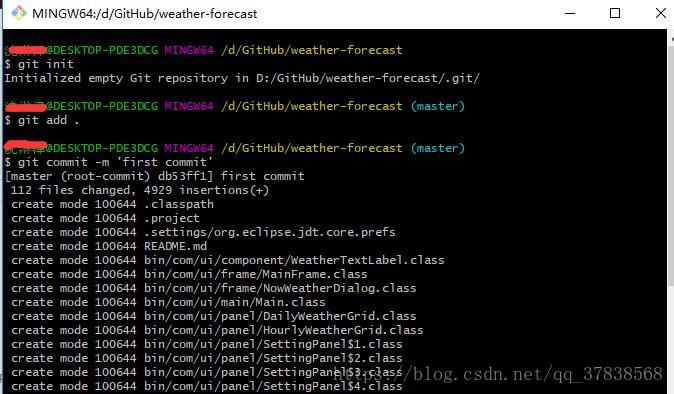
touch README.md

依次使用以下命令将文件夹的文件全部添加到暂存区并提交

git add .

git commit -m 'first commit'

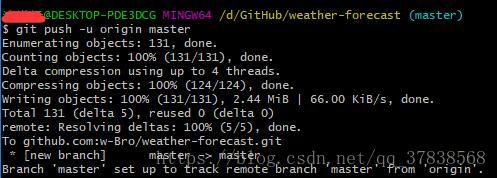
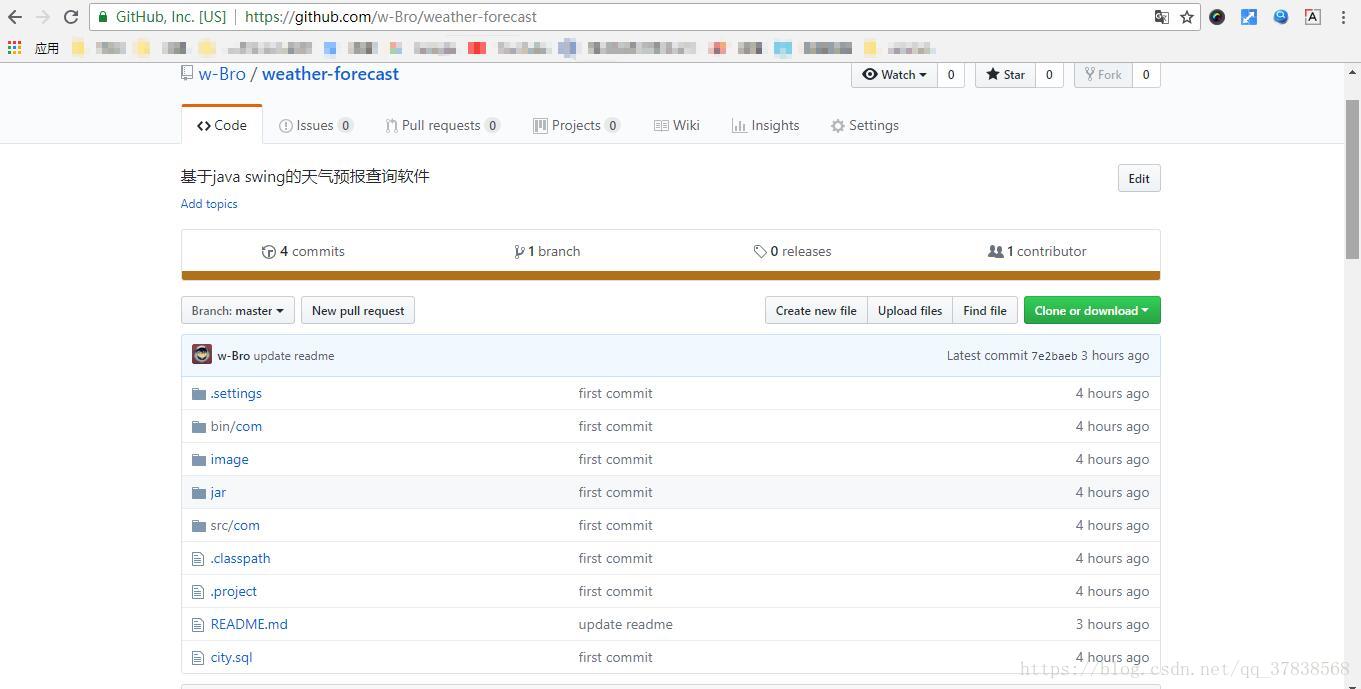
 注意add后面有个点，代表文件夹里面的所有文件，如果要提交某一个文件，将点替换成文件名或文件夹名；commit -m后面的单引号内容为本次提交的注释，可以根据自己的要求填写。

  
 提交完成后，需要将远程仓库与本地仓库同步连接，还记得在GitHub创建仓库时让你注意的链接吗？那就是远程仓库的地址，复制它。（这一步只需要在你第一次提交项目的时候进行，后面如果你还是使用这个仓库提交到同一个远程仓库的话，就不需再进行一次同步连接了）

git remote add origin 粘贴你自己的链接

这里写图片描述  
 最后一步将本地仓库的内容推送到远程仓库：

git push -u origin master

等待上传结束即可，文件多的话会稍微慢点，耐心等待。  
  
至此便完成将自己的项目上传到GitHub上，效果如下：  


## Git 安装配置

在使用Git前我们需要先安装 Git。Git 目前支持 Linux/Unix、Solaris、Mac和 Windows 平台上运行。

Git 各平台安装包下载地址为：<http://git-scm.com/downloads>

### Linux 平台上安装

Git 的工作需要调用 curl，zlib，openssl，expat，libiconv 等库的代码，所以需要先安装这些依赖工具。

在有 yum 的系统上（比如 Fedora）或者有 apt-get 的系统上（比如 Debian 体系），可以用下面的命令安装：

各 Linux 系统可以使用其安装包管理工具（apt-get、yum 等）进行安装：

#### Debian/Ubuntu

Debian/Ubuntu Git 安装命令为：

$ apt-get install libcurl4-gnutls-dev libexpat1-dev gettext \

libz-dev libssl-dev

$ apt-get install git

$ git --version

git version 1.8.1.2

#### Centos/RedHat

如果你使用的系统是 Centos/RedHat 安装命令为：

$ yum install curl-devel expat-devel gettext-devel \

openssl-devel zlib-devel

$ yum -y install git-core

$ git --version

git version 1.7.1

#### 源码安装

我们也可以在官网下载源码包来安装，最新源码包下载地址：<https://git-scm.com/download>

安装指定系统的依赖包：

########## Centos/RedHat ##########

$ yum install curl-devel expat-devel gettext-devel \

openssl-devel zlib-devel

########## Debian/Ubuntu ##########

$ apt-get install libcurl4-gnutls-dev libexpat1-dev gettext \

libz-dev libssl-dev

解压安装下载的源码包：

$ tar -zxf git-1.7.2.2.tar.gz

$ cd git-1.7.2.2

$ make prefix=/usr/local all

$ sudo make prefix=/usr/local install

### Windows 平台上安装

在 Windows 平台上安装 Git 同样轻松，有个叫做 msysGit 的项目提供了安装包，可以到 GitHub 的页面上下载 exe 安装文件并运行：

安装包下载地址：<https://gitforwindows.org/>



完成安装之后，就可以使用命令行的 git 工具（已经自带了 ssh 客户端）了，另外还有一个图形界面的 Git 项目管理工具。

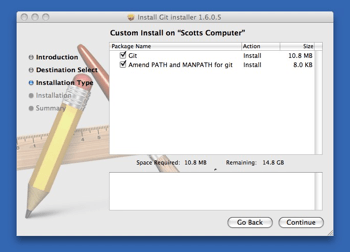
在开始菜单里找到"Git"->"Git Bash"，会弹出 Git 命令窗口，你可以在该窗口进行 Git 操作。

### Mac 平台上安装

在 Mac 平台上安装 Git 最容易的当属使用图形化的 Git 安装工具，下载地址为：

<http://sourceforge.net/projects/git-osx-installer/>

安装界面如下所示：



### Git 配置

Git 提供了一个叫做 git config 的工具，专门用来配置或读取相应的工作环境变量。

这些环境变量，决定了 Git 在各个环节的具体工作方式和行为。这些变量可以存放在以下三个不同的地方：

* /etc/gitconfig 文件：系统中对所有用户都普遍适用的配置。若使用 git config 时用 --system 选项，读写的就是这个文件。
* ~/.gitconfig 文件：用户目录下的配置文件只适用于该用户。若使用 git config 时用 --global 选项，读写的就是这个文件。
* 当前项目的 Git 目录中的配置文件（也就是工作目录中的 .git/config 文件）：这里的配置仅仅针对当前项目有效。每一个级别的配置都会覆盖上层的相同配置，所以 .git/config 里的配置会覆盖 /etc/gitconfig 中的同名变量。

在 Windows 系统上，Git 会找寻用户主目录下的 .gitconfig 文件。主目录即 $HOME 变量指定的目录，一般都是 C:\Documents and Settings\$USER。

此外，Git 还会尝试找寻 /etc/gitconfig 文件，只不过看当初 Git 装在什么目录，就以此作为根目录来定位。

#### 用户信息

配置个人的用户名称和电子邮件地址：

$ git config --global user.name "runoob"

$ git config --global user.email test@runoob.com

如果用了 **--global** 选项，那么更改的配置文件就是位于你用户主目录下的那个，以后你所有的项目都会默认使用这里配置的用户信息。

如果要在某个特定的项目中使用其他名字或者电邮，只要去掉 --global 选项重新配置即可，新的设定保存在当前项目的 .git/config 文件里。

#### 文本编辑器

设置Git默认使用的文本编辑器, 一般可能会是 Vi 或者 Vim。如果你有其他偏好，比如 Emacs 的话，可以重新设置：:

$ git config --global core.editor emacs

#### 差异分析工具

还有一个比较常用的是，在解决合并冲突时使用哪种差异分析工具。比如要改用 vimdiff 的话：

$ git config --global merge.tool vimdiff

Git 可以理解 kdiff3，tkdiff，meld，xxdiff，emerge，vimdiff，gvimdiff，ecmerge，和 opendiff 等合并工具的输出信息。

当然，你也可以指定使用自己开发的工具，具体怎么做可以参阅第七章。

#### 查看配置信息

要检查已有的配置信息，可以使用 git config --list 命令：

$ git config --list

http.postbuffer=2M

user.name=runoob

user.email=test@runoob.com

有时候会看到重复的变量名，那就说明它们来自不同的配置文件（比如 /etc/gitconfig 和 ~/.gitconfig），不过最终 Git 实际采用的是最后一个。

这些配置我们也可以在 **~/.gitconfig** 或 **/etc/gitconfig** 看到，如下所示：

vim ~/.gitconfig

显示内容如下所示：

[http]

postBuffer = 2M

[user]

name = runoob

email = test@runoob.com

也可以直接查阅某个环境变量的设定，只要把特定的名字跟在后面即可，像这样：

$ git config user.name

runoob

[Git 教程](https://www.runoob.com/git/git-tutorial.html)

[Git 工作流程](https://www.runoob.com/git/git-workflow.html)

### 1 篇笔记 写笔记

1. 宋某人c

  sya\*\*\*g@iroogoo.com

43

1、最新git源码下载地址：

<https://github.com/git/git/releases>

<https://www.kernel.org/pub/software/scm/git/>

可以手动下载下来在上传到服务器上面

2 移除旧版本git

centos自带Git，7.x版本自带git 1.8.3.1（应该是，也可能不是）， 安装新版本之前需要使用yum remove git卸载（安装后卸载也可以）。

[root@Git ~]# git --version ## 查看自带的版本

git version 1.8.3.1

[root@Git ~]# yum remove git ## 移除原来的版本

3 安装所需软件包

[root@Git ~]# yum install curl-devel expat-devel gettext-devel openssl-devel zlib-devel

[root@Git ~]# yum install gcc-c++ perl-ExtUtils-MakeMaker

下载&安装

[root@Git ~]# cd /usr/src

[root@Git ~]# wget https://www.kernel.org/pub/software/scm/git/git-2.7.3.tar.gz

5 解压

[root@Git ~]# tar xf git-2.7.3.tar.gz

6 配置编译安装

[root@Git ~]# cd git-2.7.3

[root@Git ~]# make configure

[root@Git ~]# ./configure --prefix=/usr/git ##配置目录

[root@Git ~]# make profix=/usr/git

[root@Git ~]# make install

7 加入环境变量

[root@Git ~]# echo "export PATH=$PATH:/usr/git/bin" >> /etc/profile

[root@Git ~]# source /etc/profile

8 检查版本

[root@Git git-2.7.3]# git --version

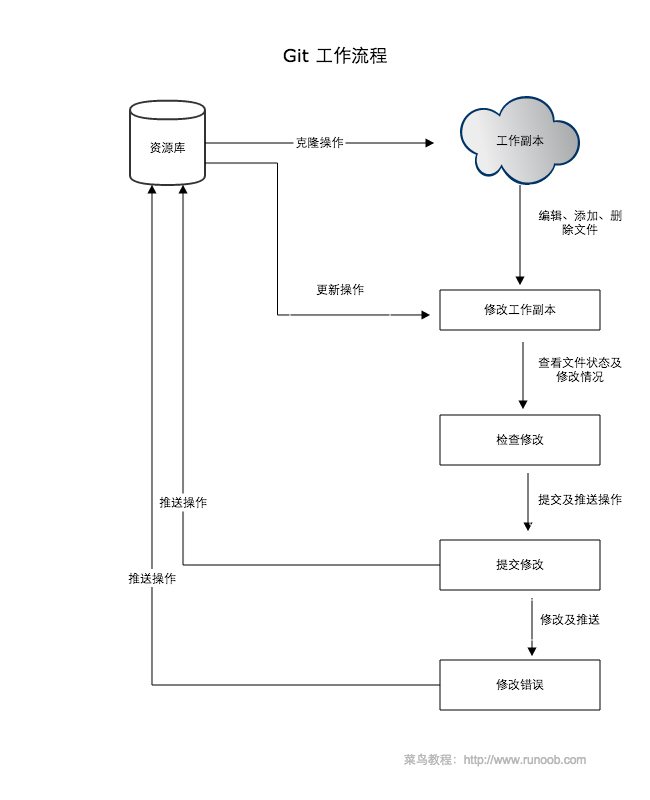
git version 2.7.3

## Git 工作流程

一般工作流程如下：

* 克隆 Git 资源作为工作目录。
* 在克隆的资源上添加或修改文件。
* 如果其他人修改了，你可以更新资源。
* 在提交前查看修改。
* 提交修改。
* 在修改完成后，如果发现错误，可以撤回提交并再次修改并提交。

下图展示了 Git 的工作流程：



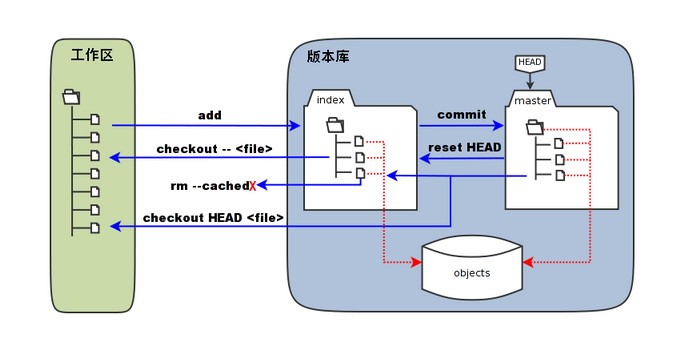
## Git 工作区、暂存区和版本库

## 基本概念

我们先来理解下Git 工作区、暂存区和版本库概念

* **工作区：**就是你在电脑里能看到的目录。
* **暂存区：**英文叫stage, 或index。一般存放在 ".git目录下" 下的index文件（.git/index）中，所以我们把暂存区有时也叫作索引（index）。
* **版本库：**工作区有一个隐藏目录.git，这个不算工作区，而是Git的版本库。

下面这个图展示了工作区、版本库中的暂存区和版本库之间的关系：



图中左侧为工作区，右侧为版本库。在版本库中标记为 "index" 的区域是暂存区（stage, index），标记为 "master" 的是 master 分支所代表的目录树。

图中我们可以看出此时 "HEAD" 实际是指向 master 分支的一个"游标"。所以图示的命令中出现 HEAD 的地方可以用 master 来替换。

图中的 objects 标识的区域为 Git 的对象库，实际位于 ".git/objects" 目录下，里面包含了创建的各种对象及内容。

当对工作区修改（或新增）的文件执行 "git add" 命令时，暂存区的目录树被更新，同时工作区修改（或新增）的文件内容被写入到对象库中的一个新的对象中，而该对象的ID被记录在暂存区的文件索引中。

当执行提交操作（git commit）时，暂存区的目录树写到版本库（对象库）中，master 分支会做相应的更新。即 master 指向的目录树就是提交时暂存区的目录树。

当执行 "git reset HEAD" 命令时，暂存区的目录树会被重写，被 master 分支指向的目录树所替换，但是工作区不受影响。

当执行 "git rm --cached <file>" 命令时，会直接从暂存区删除文件，工作区则不做出改变。

当执行 "git checkout ." 或者 "git checkout -- <file>" 命令时，会用暂存区全部或指定的文件替换工作区的文件。这个操作很危险，会清除工作区中未添加到暂存区的改动。

当执行 "git checkout HEAD ." 或者 "git checkout HEAD <file>" 命令时，会用 HEAD 指向的 master 分支中的全部或者部分文件替换暂存区和以及工作区中的文件。这个命令也是极具危险性的，因为不但会清除工作区中未提交的改动，也会清除暂存区中未提交的改动。

## Git 创建仓库

你可以使用一个已经存在的目录作为Git仓库。

## git init

Git 使用 **git init** 命令来初始化一个 Git 仓库，Git 的很多命令都需要在 Git 的仓库中运行，所以 **git init** 是使用 Git 的第一个命令。

在执行完成 **git init** 命令后，Git 仓库会生成一个 .git 目录，该目录包含了资源的所有元数据，其他的项目目录保持不变（不像 SVN 会在每个子目录生成 .svn 目录，Git 只在仓库的根目录生成 .git 目录）。

### 使用方法

使用当前目录作为Git仓库，我们只需使它初始化。

git init

该命令执行完后会在当前目录生成一个 .git 目录。

使用我们指定目录作为Git仓库。

git init newrepo

初始化后，会在 newrepo 目录下会出现一个名为 .git 的目录，所有 Git 需要的数据和资源都存放在这个目录中。

如果当前目录下有几个文件想要纳入版本控制，需要先用 git add 命令告诉 Git 开始对这些文件进行跟踪，然后提交：

$ git add \*.c

$ git add README

$ git commit -m '初始化项目版本'

以上命令将目录下以 .c 结尾及 README 文件提交到仓库中。

## git clone

我们使用 **git clone** 从现有 Git 仓库中拷贝项目（类似 **svn checkout**）。

克隆仓库的命令格式为：

git clone <repo>

如果我们需要克隆到指定的目录，可以使用以下命令格式：

git clone <repo> <directory>

**参数说明：**

* **repo:**Git 仓库。
* **directory:**本地目录。

比如，要克隆 Ruby 语言的 Git 代码仓库 Grit，可以用下面的命令：

$ git clone git://github.com/schacon/grit.git

执行该命令后，会在当前目录下创建一个名为grit的目录，其中包含一个 .git 的目录，用于保存下载下来的所有版本记录。

如果要自己定义要新建的项目目录名称，可以在上面的命令末尾指定新的名字：

$ git clone git://github.com/schacon/grit.git mygrit

[Git 工作区、暂存区和版本库](https://www.runoob.com/git/git-workspace-index-repo.html)

[Git 基本操作](https://www.runoob.com/git/git-basic-operations.html)

## 2 篇笔记 写笔记

1. CosmosHua

  cos\*\*\*cosmos@163.com

55

几种效果等价的git clone写法：

git clone http://github.com/CosmosHua/locate new

git clone http://github.com/CosmosHua/locate.git new

git clone git://github.com/CosmosHua/locate new

git clone git://github.com/CosmosHua/locate.git new

[CosmosHua](javascript:;)

   CosmosHua

  cos\*\*\*cosmos@163.com

3年前 (2017-07-08)

1. fsliurujie

  fsl\*\*\*ujie@126.com

56

git clone 时，可以所用不同的协议，包括 ssh, git, https 等，其中最常用的是 ssh，因为速度较快，还可以配置公钥免输入密码。各种写法如下：

git clone git@github.com:fsliurujie/test.git --SSH协议

git clone git://github.com/fsliurujie/test.git --GIT协议

git clone https://github.com/fsliurujie/test.git --HTTPS协议

## Git 基本操作

## 获取与创建项目命令

### git init

用 git init 在目录中创建新的 Git 仓库。 你可以在任何时候、任何目录中这么做，完全是本地化的。

在目录中执行 git init，就可以创建一个 Git 仓库了。比如我们创建 runoob 项目：

$ mkdir runoob

$ cd runoob/

$ git init

Initialized empty Git repository in /Users/tianqixin/www/runoob/.git/

# 在 /www/runoob/.git/ 目录初始化空 Git 仓库完毕。

现在你可以看到在你的项目中生成了 .git 这个子目录。 这就是你的 Git 仓库了，所有有关你的此项目的快照数据都存放在这里。

ls -a

.    ..    .git

### git clone

使用 git clone 拷贝一个 Git 仓库到本地，让自己能够查看该项目，或者进行修改。

如果你需要与他人合作一个项目，或者想要复制一个项目，看看代码，你就可以克隆那个项目。 执行命令：

git clone [url]

[url] 为你想要复制的项目，就可以了。

例如我们克隆 Github 上的项目：

$ git clone git@github.com:schacon/simplegit.git

Cloning into 'simplegit'...

remote: Counting objects: 13, done.

remote: Total 13 (delta 0), reused 0 (delta 0), pack-reused 13

Receiving objects: 100% (13/13), done.

Resolving deltas: 100% (2/2), done.

Checking connectivity... done.

克隆完成后，在当前目录下会生成一个 simplegit 目录：

$ cd simplegit/

$ ls

README Rakefile lib

上述操作将复制该项目的全部记录。

$ ls -a

. .. .git README Rakefile lib

$ cd .git

$ ls

HEAD description info packed-refs

branches hooks logs refs

config index objects

默认情况下，Git 会按照你提供的 URL 所指示的项目的名称创建你的本地项目目录。 通常就是该 URL 最后一个 / 之后的项目名称。如果你想要一个不一样的名字， 你可以在该命令后加上你想要的名称。

## 基本快照

Git 的工作就是创建和保存你的项目的快照及与之后的快照进行对比。本章将对有关创建与提交你的项目的快照的命令作介绍。

### git add

git add 命令可将该文件添加到缓存，如我们添加以下两个文件：

$ touch README

$ touch hello.php

$ ls

README        hello.php

$ git status -s

?? README

?? hello.php

$

git status 命令用于查看项目的当前状态。

接下来我们执行 git add 命令来添加文件：

$ git add README hello.php

现在我们再执行 git status，就可以看到这两个文件已经加上去了。

$ git status -s

A README

A hello.php

$

新项目中，添加所有文件很普遍，我们可以使用 **git add .** 命令来添加当前项目的所有文件。

现在我们修改 README 文件：

$ vim README

在 README 添加以下内容：**# Runoob Git 测试**，然后保存退出。

再执行一下 git status：

$ git status -s

AM README

A hello.php

"AM" 状态的意思是，这个文件在我们将它添加到缓存之后又有改动。改动后我们再执行 git add 命令将其添加到缓存中：

$ git add .

$ git status -s

A README

A hello.php

当你要将你的修改包含在即将提交的快照里的时候，需要执行 git add。

### git status

git status 以查看在你上次提交之后是否有修改。

我演示该命令的时候加了 -s 参数，以获得简短的结果输出。如果没加该参数会详细输出内容：

$ git status

On branch master

Initial commit

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

    new file: README

    new file: hello.php

### git diff

执行 git diff 来查看执行 git status 的结果的详细信息。

git diff 命令显示已写入缓存与已修改但尚未写入缓存的改动的区别。git diff 有两个主要的应用场景。

* 尚未缓存的改动：**git diff**
* 查看已缓存的改动： **git diff --cached**
* 查看已缓存的与未缓存的所有改动：**git diff HEAD**
* 显示摘要而非整个 diff：**git diff --stat**

在 hello.php 文件中输入以下内容：

<?php

echo '菜鸟教程：www.runoob.com';

?>

$ git status -s

A README

AM hello.php

$ git diff

diff --git a/hello.php b/hello.php

index e69de29..69b5711 100644

--- a/hello.php

+++ b/hello.php

@@ -0,0 +1,3 @@

+<?php

+echo '菜鸟教程：www.runoob.com';

+?>

git status 显示你上次提交更新后的更改或者写入缓存的改动， 而 git diff 一行一行地显示这些改动具体是啥。

接下来我们来查看下 git diff --cached 的执行效果：

$ git add hello.php

$ git status -s

A README

A hello.php

$ git diff --cached

diff --git a/README b/README

new file mode 100644

index 0000000..8f87495

--- /dev/null

+++ b/README

@@ -0,0 +1 @@

+# Runoob Git 测试

diff --git a/hello.php b/hello.php

new file mode 100644

index 0000000..69b5711

--- /dev/null

+++ b/hello.php

@@ -0,0 +1,3 @@

+<?php

+echo '菜鸟教程：www.runoob.com';

+?>

### git commit

使用 git add 命令将想要快照的内容写入缓存区， 而执行 git commit 将缓存区内容添加到仓库中。

Git 为你的每一个提交都记录你的名字与电子邮箱地址，所以第一步需要配置用户名和邮箱地址。

$ git config --global user.name 'runoob'

$ git config --global user.email test@runoob.com

接下来我们写入缓存，并提交对 hello.php 的所有改动。在首个例子中，我们使用 -m 选项以在命令行中提供提交注释。

$ git add hello.php

$ git status -s

A README

A hello.php

$ git commit -m '第一次版本提交'

[master (root-commit) d32cf1f] 第一次版本提交

2 files changed, 4 insertions(+)

create mode 100644 README

create mode 100644 hello.php

现在我们已经记录了快照。如果我们再执行 git status:

$ git status

# On branch master

nothing to commit (working directory clean)

以上输出说明我们在最近一次提交之后，没有做任何改动，是一个"working directory clean：干净的工作目录"。

如果你没有设置 -m 选项，Git 会尝试为你打开一个编辑器以填写提交信息。 如果 Git 在你对它的配置中找不到相关信息，默认会打开 vim。屏幕会像这样：

# Please enter the commit message for your changes. Lines starting

# with '#' will be ignored, and an empty message aborts the commit.

# On branch master

# Changes to be committed:

# (use "git reset HEAD <file>..." to unstage)

#

# modified: hello.php

#

~

~

".git/COMMIT\_EDITMSG" 9L, 257C

如果你觉得 git add 提交缓存的流程太过繁琐，Git 也允许你用 -a 选项跳过这一步。命令格式如下：

git commit -a

我们先修改 hello.php 文件为以下内容：

<?php

echo '菜鸟教程：www.runoob.com';

echo '菜鸟教程：www.runoob.com';

?>

再执行以下命令：

git commit -am '修改 hello.php 文件'

[master 71ee2cb] 修改 hello.php 文件

1 file changed, 1 insertion(+)

### git reset HEAD

git reset HEAD 命令用于取消已缓存的内容。

我们先改动文件 README 文件，内容如下：

# Runoob Git 测试

# 菜鸟教程

hello.php 文件修改为：

<?php

echo '菜鸟教程：www.runoob.com';

echo '菜鸟教程：www.runoob.com';

echo '菜鸟教程：www.runoob.com';

?>

现在两个文件修改后，都提交到了缓存区，我们现在要取消其中一个的缓存，操作如下：

$ git status -s

M README

M hello.php

$ git add .

$ git status -s

M README

M hello.php

$ git reset HEAD hello.php

Unstaged changes after reset:

M    hello.php

$ git status -s

M README

M hello.php

现在你执行 git commit，只会将 README 文件的改动提交，而 hello.php 是没有的。

$ git commit -m '修改'

[master f50cfda] 修改

1 file changed, 1 insertion(+)

$ git status -s

M hello.php

可以看到 hello.php 文件的修改并未提交。

这时我们可以使用以下命令将 hello.php 的修改提交：

$ git commit -am '修改 hello.php 文件'

[master 760f74d] 修改 hello.php 文件

1 file changed, 1 insertion(+)

$ git status

On branch master

nothing to commit, working directory clean

简而言之，执行 git reset HEAD 以取消之前 git add 添加，但不希望包含在下一提交快照中的缓存。

### git rm

如果只是简单地从工作目录中手工删除文件，运行 git status 时就会在 Changes not staged for commit 的提示。

要从 Git 中移除某个文件，就必须要从已跟踪文件清单中移除，然后提交。可以用以下命令完成此项工作

git rm <file>

如果删除之前修改过并且已经放到暂存区域的话，则必须要用强制删除选项 -f

git rm -f <file>

如果把文件从暂存区域移除，但仍然希望保留在当前工作目录中，换句话说，仅是从跟踪清单中删除，使用 --cached 选项即可

git rm --cached <file>

如我们删除 hello.php文件：

$ git rm hello.php

rm 'hello.php'

$ ls

README

不从工作区中删除文件：

$ git rm --cached README

rm 'README'

$ ls

README

可以递归删除，即如果后面跟的是一个目录做为参数，则会递归删除整个目录中的所有子目录和文件：

git rm –r \*

进入某个目录中，执行此语句，会删除该目录下的所有文件和子目录。

### git mv

git mv 命令用于移动或重命名一个文件、目录、软连接。

我们先把刚移除的 README 添加回来：

$ git add README

然后对其重名:

$ git mv README README.md

$ ls

README.md

[Git 创建仓库](https://www.runoob.com/git/git-create-repository.html)

[Git 分支管理](https://www.runoob.com/git/git-branch.html)

## 3 篇笔记 写笔记

1. hepburn

  158\*\*\*9602@qq.com

15

git commit、git push、git pull、 git fetch、git merge 的含义与区别

* + git commit：是将本地修改过的文件提交到本地库中；
  + git push：是将本地库中的最新信息发送给远程库；
  + git pull：是从远程获取最新版本到本地，并自动merge；
  + git fetch：是从远程获取最新版本到本地，不会自动merge；
  + git merge：是用于从指定的commit(s)合并到当前分支，用来合并两个分支；

$ git merge -b // 指将 b 分支合并到当前分支

git pull 相当于 git fetch + git merge。

[hepburn](https://www.runoob.com/note/35938)

   hepburn

  158\*\*\*9602@qq.com

2年前 (2018-10-15)

1. tuntuntunwu

  wxq\*\*\*567567@126.com

1

**git 基本操作注意点总结:**

git clone <source repository> <destination repository>

复制本地仓库的命令方式。

**<source repository>**：想克隆的本地仓库路径

**<destination repository>**：想克隆去另一个地方的路径。例如 git clone d:/git e:/git11 是将 d:/git 的仓库（即包含隐藏文件 .git 的目录）克隆到 e:/git11 目录下。

注意：

1、**<destination repository>** 目录必须没有在文件系统上创建，或创建了但里面为空，不然会克隆不成功。

2、与从远程拉取仓库不同，路径的最后不用写 **.git** 来表明这是一个仓库。

git status –s

获得简短的状态输出。

git diff

* + git diff：查看工作区与暂存区的不同。
  + git diff –cached [<commit>]：查看暂存区与指定提交版本的不同，版本可缺省（为HEAD）。
  + git diff <commit>：查看工作区与指定提交版本的不同。
  + git diff <commit>..<commit>：查看2个指定提交版本的不同，其中任一可缺省（为HEAD）。
  + git diff <commit>...<commit>：查看2个不同分支指定提交版本的不同，其中任一可缺省（为HEAD），该命令相当于git diff $(git-merge-base A B) B。

git commit –am ""

直接提交全部修改，相当于 add 和 commit 一起执行了。

注意：全部文件为 tracked 才行，你新建了文件为 untracked 时，该命令不会执行。

git checkout

与 git reset 不同，reset 是替换整个目录树，多余的文件将被删除。而 checkout 只是替换指定的文件，对多余的文件保留不做任何处理。

git rm

把文件从工作区和暂存区中删除。使用 —cached 只从暂存区中删除。使用 –rf <directory> 可删除指定目录下的所有文件和子目录。

git mv <source> <destination>

在工作区和暂存区中进行移动或重命名。若 <destination> 不为一个目录名，则执行重命名。如果为一个目录名，则执行移动。

[tuntuntunwu](javascript:;)

   tuntuntunwu

  wxq\*\*\*567567@126.com

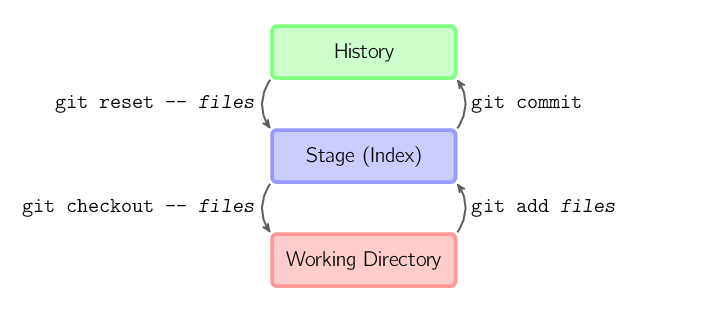
2年前 (2018-11-05)

1. 汪洋

  768\*\*\*667@qq.com

5

[**图解 Git**](https://www.runoob.com/w3cnote/git-graphical.html)



上面的四条命令在工作目录、暂存目录(也叫做索引)和仓库之间复制文件。

* + git add files 把当前文件放入暂存区域。
  + git commit 给暂存区域生成快照并提交。
  + git reset -- files 用来撤销最后一次git add files，你也可以用git reset 撤销所有暂存区域文件。
  + git checkout -- files 把文件从暂存区域复制到工作目录，用来丢弃本地修改。

更多图解参考：<https://www.runoob.com/w3cnote/git-graphical.html>

## Git 分支管理

几乎每一种版本控制系统都以某种形式支持分支。使用分支意味着你可以从开发主线上分离开来，然后在不影响主线的同时继续工作。

有人把 Git 的分支模型称为**必杀技特性**，而正是因为它，将 **Git** 从版本控制系统家族里区分出来。

创建分支命令：

git branch (branchname)

切换分支命令:

git checkout (branchname)

当你切换分支的时候，Git 会用该分支的最后提交的快照替换你的工作目录的内容， 所以多个分支不需要多个目录。

合并分支命令:

git merge

你可以多次合并到统一分支， 也可以选择在合并之后直接删除被并入的分支。

开始前我们先创建一个测试目录：

$ mkdir gitdemo

$ cd gitdemo/

$ git init

Initialized empty Git repository...

$ touch README

$ git add README

$ git commit -m '第一次版本提交'

[master (root-commit) 3b58100] 第一次版本提交

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 README

## Git 分支管理

### 列出分支

列出分支基本命令：

git branch

没有参数时，**git branch** 会列出你在本地的分支。

$ git branch

\* master

此例的意思就是，我们有一个叫做 **master** 的分支，并且该分支是当前分支。

当你执行 **git init** 的时候，默认情况下 Git 就会为你创建 **master** 分支。

如果我们要手动创建一个分支。执行 git branch (branchname) 即可。

$ git branch testing

$ git branch

\* master

testing

现在我们可以看到，有了一个新分支 **testing**。

当你以此方式在上次提交更新之后创建了新分支，如果后来又有更新提交， 然后又切换到了 **testing** 分支，Git 将还原你的工作目录到你创建分支时候的样子。

接下来我们将演示如何切换分支，我们用 git checkout (branch) 切换到我们要修改的分支。

$ ls

README

$ echo 'runoob.com' > test.txt

$ git add .

$ git commit -m 'add test.txt'

[master 3e92c19] add test.txt

1 file changed, 1 insertion(+)

create mode 100644 test.txt

$ ls

README        test.txt

$ git checkout testing

Switched to branch 'testing'

$ ls

README

当我们切换到 **testing** 分支的时候，我们添加的新文件 test.txt 被移除了。切换回 **master** 分支的时候，它们有重新出现了。

$ git checkout master

Switched to branch 'master'

$ ls

README        test.txt

我们也可以使用 git checkout -b (branchname) 命令来创建新分支并立即切换到该分支下，从而在该分支中操作。

$ git checkout -b newtest

Switched to a new branch 'newtest'

$ git rm test.txt

rm 'test.txt'

$ ls

README

$ touch runoob.php

$ git add .

$ git commit -am 'removed test.txt、add runoob.php'

[newtest c1501a2] removed test.txt、add runoob.php

2 files changed, 1 deletion(-)

create mode 100644 runoob.php

delete mode 100644 test.txt

$ ls

README        runoob.php

$ git checkout master

Switched to branch 'master'

$ ls

README        test.txt

如你所见，我们创建了一个分支，在该分支的上移除了一些文件 test.txt，并添加了 runoob.php 文件，然后切换回我们的主分支，删除的 test.txt 文件又回来了，且新增加的 runoob.php 不存在主分支中。

使用分支将工作切分开来，从而让我们能够在不同开发环境中做事，并来回切换。

### 删除分支

删除分支命令：

git branch -d (branchname)

例如我们要删除 testing 分支：

$ git branch

\* master

testing

$ git branch -d testing

Deleted branch testing (was 85fc7e7).

$ git branch

\* master

### 分支合并

一旦某分支有了独立内容，你终究会希望将它合并回到你的主分支。 你可以使用以下命令将任何分支合并到当前分支中去：

git merge

$ git branch

\* master

newtest

$ ls

README        test.txt

$ git merge newtest

Updating 3e92c19..c1501a2

Fast-forward

runoob.php | 0

test.txt | 1 -

2 files changed, 1 deletion(-)

create mode 100644 runoob.php

delete mode 100644 test.txt

$ ls

README        runoob.php

以上实例中我们将 newtest 分支合并到主分支去，test.txt 文件被删除。

合并完后就可以删除分支:

$ git branch -d newtest

Deleted branch newtest (was c1501a2).

删除后， 就只剩下 master 分支了：

$ git branch

\* master

### 合并冲突

合并并不仅仅是简单的文件添加、移除的操作，Git 也会合并修改。

$ git branch

\* master

$ cat runoob.php

首先，我们创建一个叫做 change\_site 的分支，切换过去，我们将 runoob.php 内容改为:

<?php

echo 'runoob';

?>

创建 change\_site 分支：

$ git checkout -b change\_site

Switched to a new branch 'change\_site'

$ vim runoob.php

$ head -3 runoob.php

<?php

echo 'runoob';

?>

$ git commit -am 'changed the runoob.php'

[change\_site 7774248] changed the runoob.php

1 file changed, 3 insertions(+)

将修改的内容提交到 change\_site 分支中。 现在，假如切换回 master 分支我们可以看内容恢复到我们修改前的(空文件，没有代码)，我们再次修改 runoob.php 文件。

$ git checkout master

Switched to branch 'master'

$ cat runoob.php

$ vim runoob.php # 修改内容如下

$ cat runoob.php

<?php

echo 1;

?>

$ git diff

diff --git a/runoob.php b/runoob.php

index e69de29..ac60739 100644

--- a/runoob.php

+++ b/runoob.php

@@ -0,0 +1,3 @@

+<?php

+echo 1;

+?>

$ git commit -am '修改代码'

[master c68142b] 修改代码

1 file changed, 3 insertions(+)

现在这些改变已经记录到我的 "master" 分支了。接下来我们将 "change\_site" 分支合并过来。

$ git merge change\_site

Auto-merging runoob.php

CONFLICT (content): Merge conflict in runoob.php

Automatic merge failed; fix conflicts and then commit the result.

$ cat runoob.php # 代开文件，看到冲突内容

<?php

<<<<<<< HEAD

echo 1;

=======

echo 'runoob';

>>>>>>> change\_site

?>

我们将前一个分支合并到 master 分支，一个合并冲突就出现了，接下来我们需要手动去修改它。

$ vim runoob.php

$ cat runoob.php

<?php

echo 1;

echo 'runoob';

?>

$ git diff

diff --cc runoob.php

index ac60739,b63d7d7..0000000

--- a/runoob.php

+++ b/runoob.php

@@@ -1,3 -1,3 +1,4 @@@

<?php

+echo 1;

+ echo 'runoob';

?>

在 Git 中，我们可以用 git add 要告诉 Git 文件冲突已经解决

$ git status -s

UU runoob.php

$ git add runoob.php

$ git status -s

M runoob.php

$ git commit

[master 88afe0e] Merge branch 'change\_site'

现在我们成功解决了合并中的冲突，并提交了结果。

## Git 查看提交历史

在使用 Git 提交了若干更新之后，又或者克隆了某个项目，想回顾下提交历史，我们可以使用 git log 命令查看。

针对我们前一章节的操作，使用 git log 命令列出历史提交记录如下：

$ git log

commit d5e9fc2c811e0ca2b2d28506ef7dc14171a207d9 (HEAD -> master)

Merge: c68142b 7774248

Author: runoob <test@runoob.com>

Date: Fri May 3 15:55:58 2019 +0800

Merge branch 'change\_site'

commit c68142b562c260c3071754623b08e2657b4c6d5b

Author: runoob <test@runoob.com>

Date: Fri May 3 15:52:12 2019 +0800

修改代码

commit 777424832e714cf65d3be79b50a4717aea51ab69 (change\_site)

Author: runoob <test@runoob.com>

Date: Fri May 3 15:49:26 2019 +0800

changed the runoob.php

commit c1501a244676ff55e7cccac1ecac0e18cbf6cb00

Author: runoob <test@runoob.com>

Date: Fri May 3 15:35:32 2019 +0800

我们可以用 --oneline 选项来查看历史记录的简洁的版本。

$ git log --oneline

$ git log --oneline

d5e9fc2 (HEAD -> master) Merge branch 'change\_site'

c68142b 修改代码

7774248 (change\_site) changed the runoob.php

c1501a2 removed test.txt、add runoob.php

3e92c19 add test.txt

3b58100 第一次版本提交

这告诉我们的是，此项目的开发历史。

我们还可以用 --graph 选项，查看历史中什么时候出现了分支、合并。以下为相同的命令，开启了拓扑图选项：

\* d5e9fc2 (HEAD -> master) Merge branch 'change\_site'

|\

| \* 7774248 (change\_site) changed the runoob.php

\* | c68142b 修改代码

|/

\* c1501a2 removed test.txt、add runoob.php

\* 3e92c19 add test.txt

\* 3b58100 第一次版本提交

现在我们可以更清楚明了地看到何时工作分叉、又何时归并。

你也可以用 **--reverse** 参数来逆向显示所有日志。

$ git log --reverse --oneline

3b58100 第一次版本提交

3e92c19 add test.txt

c1501a2 removed test.txt、add runoob.php

7774248 (change\_site) changed the runoob.php

c68142b 修改代码

d5e9fc2 (HEAD -> master) Merge branch 'change\_site'

如果只想查找指定用户的提交日志可以使用命令：git log --author , 例如，比方说我们要找 Git 源码中 Linus 提交的部分：

$ git log --author=Linus --oneline -5

81b50f3 Move 'builtin-\*' into a 'builtin/' subdirectory

3bb7256 make "index-pack" a built-in

377d027 make "git pack-redundant" a built-in

b532581 make "git unpack-file" a built-in

112dd51 make "mktag" a built-in

如果你要指定日期，可以执行几个选项：--since 和 --before，但是你也可以用 --until 和 --after。

例如，如果我要看 Git 项目中三周前且在四月十八日之后的所有提交，我可以执行这个（我还用了 --no-merges 选项以隐藏合并提交）：

$ git log --oneline --before={3.weeks.ago} --after={2010-04-18} --no-merges

5469e2d Git 1.7.1-rc2

d43427d Documentation/remote-helpers: Fix typos and improve language

272a36b Fixup: Second argument may be any arbitrary string

b6c8d2d Documentation/remote-helpers: Add invocation section

5ce4f4e Documentation/urls: Rewrite to accomodate transport::address

00b84e9 Documentation/remote-helpers: Rewrite description

03aa87e Documentation: Describe other situations where -z affects git diff

77bc694 rebase-interactive: silence warning when no commits rewritten

636db2c t3301: add tests to use --format="%N"

更多 git log 命令可查看：<http://git-scm.com/docs/git-log>

## Git 标签

如果你达到一个重要的阶段，并希望永远记住那个特别的提交快照，你可以使用 git tag 给它打上标签。

比如说，我们想为我们的 runoob 项目发布一个"1.0"版本。 我们可以用 git tag -a v1.0 命令给最新一次提交打上（HEAD）"v1.0"的标签。

-a 选项意为"创建一个带注解的标签"。 不用 -a 选项也可以执行的，但它不会记录这标签是啥时候打的，谁打的，也不会让你添加个标签的注解。 我推荐一直创建带注解的标签。

$ git tag -a v1.0

当你执行 git tag -a 命令时，Git 会打开你的编辑器，让你写一句标签注解，就像你给提交写注解一样。

现在，注意当我们执行 git log --decorate 时，我们可以看到我们的标签了：

\* d5e9fc2 (HEAD -> master) Merge branch 'change\_site'

|\

| \* 7774248 (change\_site) changed the runoob.php

\* | c68142b 修改代码

|/

\* c1501a2 removed test.txt、add runoob.php

\* 3e92c19 add test.txt

\* 3b58100 第一次版本提交

如果我们忘了给某个提交打标签，又将它发布了，我们可以给它追加标签。

例如，假设我们发布了提交 85fc7e7(上面实例最后一行)，但是那时候忘了给它打标签。 我们现在也可以：

$ git tag -a v0.9 85fc7e7

$ git log --oneline --decorate --graph

\* d5e9fc2 (HEAD -> master) Merge branch 'change\_site'

|\

| \* 7774248 (change\_site) changed the runoob.php

\* | c68142b 修改代码

|/

\* c1501a2 removed test.txt、add runoob.php

\* 3e92c19 add test.txt

\* 3b58100 (tag: v0.9) 第一次版本提交

如果我们要查看所有标签可以使用以下命令：

$ git tag

v0.9

v1.0

指定标签信息命令：

git tag -a <tagname> -m "runoob.com标签"

PGP签名标签命令：

git tag -s <tagname> -m "runoob.com标签"

[Git 查看提交历史](https://www.runoob.com/git/git-commit-history.html)

[Git 远程仓库(Github)](https://www.runoob.com/git/git-remote-repo.html)

## 1 篇笔记 写笔记

1. 宋某人c

  sya\*\*\*g@iroogoo.com

35

1、标签介绍

发布一个版本时，我们通常先在版本库中打一个标签（tag），这样就唯一确定了打标签时刻的版本。将来无论什么时候，取某个标签的版本，就是把那个打标签的时刻的历史版本取出来。

所以，标签也是版本库的一个快照。

Git 的标签虽然是版本库的快照，但其实它就是指向某个 commit 的指针（跟分支很像对不对？但是分支可以移动，标签不能移动），所以，创建和删除标签都是瞬间完成的。

Git有commit，为什么还要引入tag？

"请把上周一的那个版本打包发布，commit号是6a5819e…"

"一串乱七八糟的数字不好找！"

如果换一个办法：

"请把上周一的那个版本打包发布，版本号是v1.2"

"好的，按照tag v1.2查找commit就行！"

所以，tag就是一个让人容易记住的有意义的名字，它跟某个commit绑在一起。

同大多数 VCS 一样，Git 也可以对某一时间点上的版本打上标签。人们在发布某个软件版本（比如 v1.0 等等）的时候，经常这么做。

本节我们一起来学习如何列出所有可用的标签，如何新建标签，以及各种不同类型标签之间的差别。

## 新建标签

Git 使用的标签有两种类型：**轻量级的（lightweight）**和**含附注的（annotated）**。

轻量级标签就像是个不会变化的分支，实际上它就是个指向特定提交对象的引用。

而含附注标签，实际上是存储在仓库中的一个独立对象，它有自身的校验和信息，包含着标签的名字，电子邮件地址和日期，以及标签说明，标签本身也允许使用 GNU Privacy Guard (GPG) 来签署或验证。

一般我们都建议使用含附注型的标签，以便保留相关信息；

当然，如果只是临时性加注标签，或者不需要旁注额外信息，用轻量级标签也没问题。

### 2 创建标签

[root@Git git]# git tag v1.0

### 3 查看已有标签

[root@Git git]# git tag

v1.0

[root@Git git]# git tag v1.1

[root@Git git]# git tag

v1.0

v1.1

### 4 删除标签

[root@Git git]# git tag -d v1.1

Deleted tag ‘v1.1’ (was 91388f0)

[root@Git git]# git tag

v1.0

### 5 查看此版本所修改的内容

[root@Git git]# git show v1.0

commit 91388f0883903ac9014e006611944f6688170ef4

Author: "syaving" <"819044347@qq.com">

Date: Fri Dec 16 02:32:05 2016 +0800

commit dir

diff –git a/readme b/readme

index 7a3d711..bfecb47 100644

— a/readme

+++ b/readme

@@ -1,2 +1,3 @@

text

hello git

+use commit

[root@Git git]# git log –oneline

91388f0 commit dir

e435fe8 add readme

2525062 add readme

## Git 远程仓库(Github)

Git 并不像 SVN 那样有个中心服务器。

目前我们使用到的 Git 命令都是在本地执行，如果你想通过 Git 分享你的代码或者与其他开发人员合作。 你就需要将数据放到一台其他开发人员能够连接的服务器上。

本例使用了 Github 作为远程仓库，你可以先阅读我们的 [Github 简明教程。](https://www.runoob.com/w3cnote/git-guide.html)

## 添加远程库

要添加一个新的远程仓库，可以指定一个简单的名字，以便将来引用,命令格式如下：

git remote add [shortname] [url]

本例以 Github 为例作为远程仓库，如果你没有 Github 可以在官网 <https://github.com/>注册。

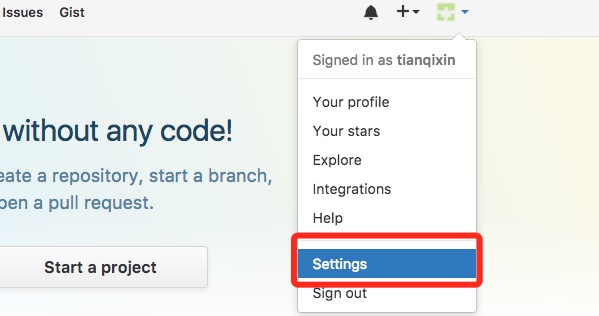
由于你的本地 Git 仓库和 GitHub 仓库之间的传输是通过SSH加密的，所以我们需要配置验证信息：

使用以下命令生成 SSH Key：

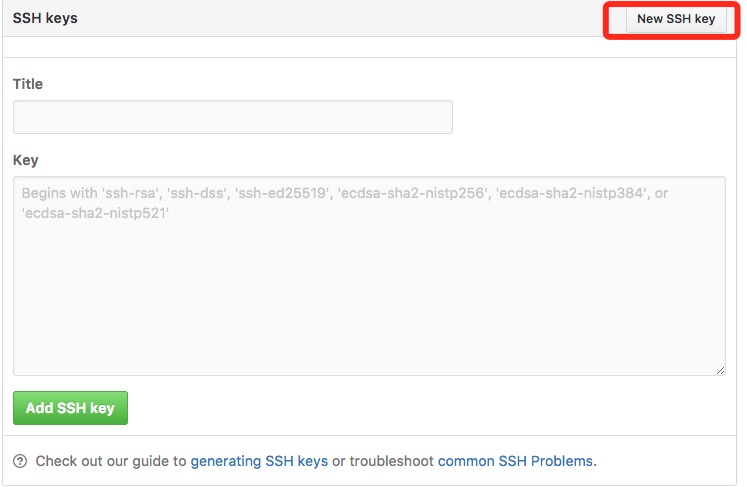
$ ssh-keygen -t rsa -C "youremail@example.com"

后面的 **your\_email@youremail.com** 改为你在 Github 上注册的邮箱，之后会要求确认路径和输入密码，我们这使用默认的一路回车就行。成功的话会在 ~/ 下生成 .ssh 文件夹，进去，打开 **id\_rsa.pub**，复制里面的 **key**。

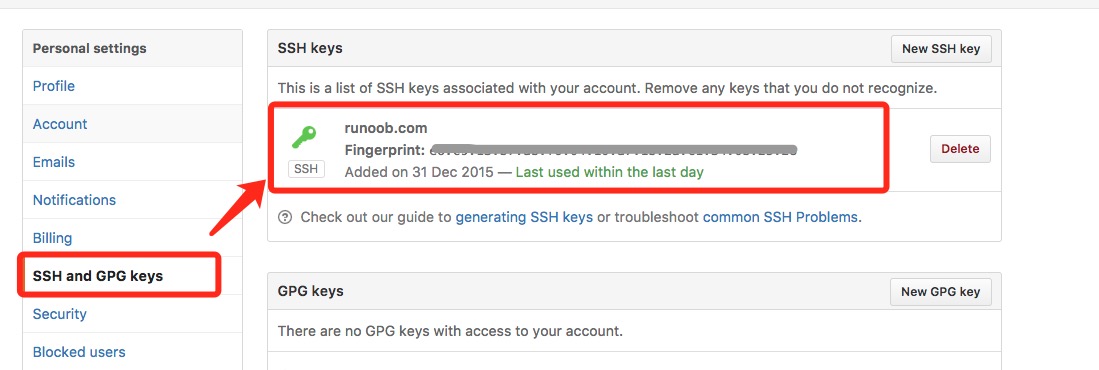
回到 github 上，进入 Account => Settings（账户配置）。



左边选择 **SSH and GPG keys**，然后点击 **New SSH key** 按钮,title 设置标题，可以随便填，粘贴在你电脑上生成的 key。



添加成功后界面如下所示



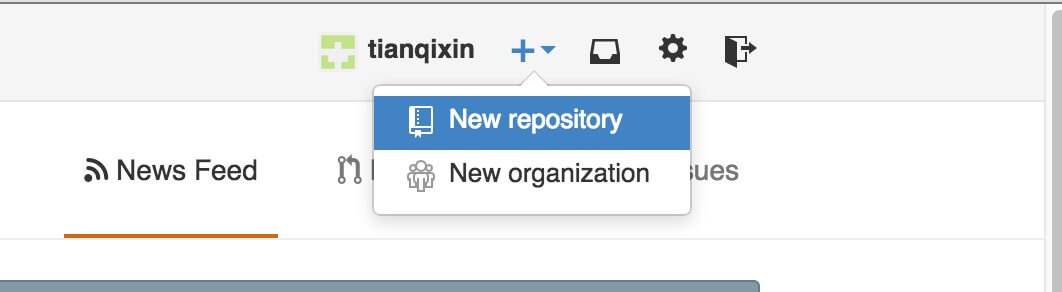
为了验证是否成功，输入以下命令：

$ ssh -T git@github.com

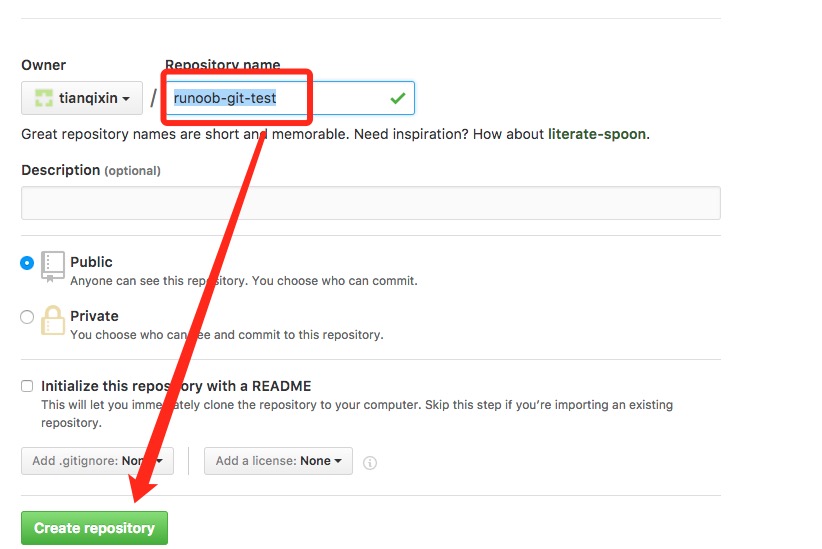
Hi tianqixin! You've successfully authenticated, but GitHub does not provide shell access.

以下命令说明我们已成功连上 Github。

之后登录后点击" New repository " 如下图所示：



之后在在Repository name 填入 runoob-git-test(远程仓库名) ，其他保持默认设置，点击"Create repository"按钮，就成功地创建了一个新的Git仓库：



创建成功后，显示如下信息：



以上信息告诉我们可以从这个仓库克隆出新的仓库，也可以把本地仓库的内容推送到GitHub仓库。

现在，我们根据 GitHub 的提示，在本地的仓库下运行命令：

$ mkdir runoob-git-test # 创建测试目录

$ cd runoob-git-test/ # 进入测试目录

$ echo "# 菜鸟教程 Git 测试" >> README.md # 创建 README.md 文件并写入内容

$ ls # 查看目录下的文件

README

$ git init # 初始化

$ git add README.md # 添加文件

$ git commit -m "添加 README.md 文件" # 提交并备注信息

[master (root-commit) 0205aab] 添加 README.md 文件

1 file changed, 1 insertion(+)

create mode 100644 README.md

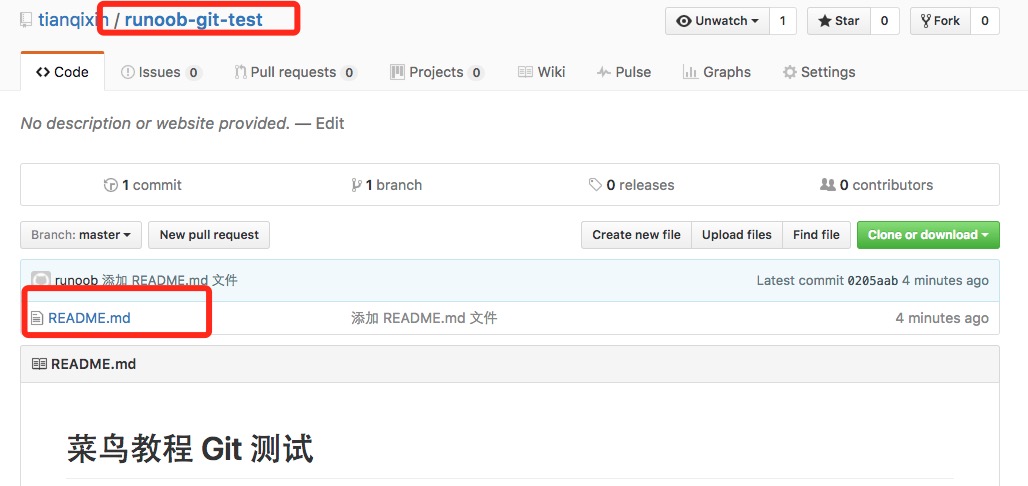
# 提交到 Github

$ git remote add origin git@github.com:tianqixin/runoob-git-test.git

$ git push -u origin master

以下命令请根据你在Github成功创建新仓库的地方复制，而不是根据我提供的命令，因为我们的Github用户名不一样，仓库名也不一样。

接下来我们返回 Github 创建的仓库，就可以看到文件已上传到 Github上：



## 查看当前的远程库

要查看当前配置有哪些远程仓库，可以用命令：

git remote

### 实例

$ git remote

origin

$ git remote -v

origin    git@github.com:tianqixin/runoob-git-test.git (fetch)

origin    git@github.com:tianqixin/runoob-git-test.git (push)

执行时加上 -v 参数，你还可以看到每个别名的实际链接地址。

## 提取远程仓库

Git 有两个命令用来提取远程仓库的更新。

1、从远程仓库下载新分支与数据：

git fetch

该命令执行完后需要执行git merge 远程分支到你所在的分支。

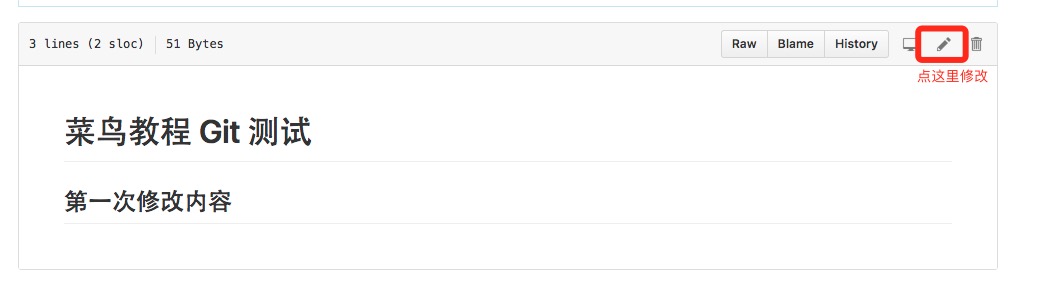
2、从远端仓库提取数据并尝试合并到当前分支：

git merge

该命令就是在执行 git fetch 之后紧接着执行 git merge 远程分支到你所在的任意分支。

假设你配置好了一个远程仓库，并且你想要提取更新的数据，你可以首先执行 **git fetch [alias]** 告诉 Git 去获取它有你没有的数据，然后你可以执行 **git merge [alias]/[branch]** 以将服务器上的任何更新（假设有人这时候推送到服务器了）合并到你的当前分支。

接下来我们在 Github 上点击" README.md" 并在线修改它:



然后我们在本地更新修改。

$ git fetch origin

remote: Counting objects: 3, done.

remote: Compressing objects: 100% (2/2), done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), done.

From github.com:tianqixin/runoob-git-test

0205aab..febd8ed master -> origin/master

以上信息"0205aab..febd8ed master -> origin/master" 说明 master 分支已被更新，我们可以使用以下命令将更新同步到本地：

$ git merge origin/master

Updating 0205aab..febd8ed

Fast-forward

README.md | 1 +

1 file changed, 1 insertion(+)

查看 README.md 文件内容：

$ cat README.md

# 菜鸟教程 Git 测试

## 第一次修改内容

## 推送到远程仓库

推送你的新分支与数据到某个远端仓库命令:

git push [alias] [branch]

以上命令将你的 [branch] 分支推送成为 [alias] 远程仓库上的 [branch] 分支，实例如下。

$ touch runoob-test.txt # 添加文件

$ git add runoob-test.txt

$ git commit -m "添加到远程"

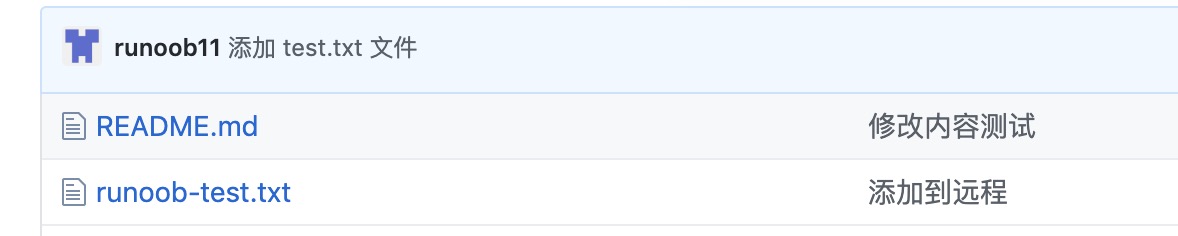
master 69e702d] 添加到远程

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 runoob-test.txt

$ git push origin master # 推送到 Github

重新回到我们的 Github 仓库，可以看到文件已经提交上来了：



## 删除远程仓库

删除远程仓库你可以使用命令：

git remote rm [别名]

### 实例

$ git remote -v

origin    git@github.com:tianqixin/runoob-git-test.git (fetch)

origin    git@github.com:tianqixin/runoob-git-test.git (push)

# 添加仓库 origin2

$ git remote add origin2 git@github.com:tianqixin/runoob-git-test.git

$ git remote -v

origin    git@github.com:tianqixin/runoob-git-test.git (fetch)

origin    git@github.com:tianqixin/runoob-git-test.git (push)

origin2    git@github.com:tianqixin/runoob-git-test.git (fetch)

origin2    git@github.com:tianqixin/runoob-git-test.git (push)

# 删除仓库 origin2

$ git remote rm origin2

$ git remote -v

origin    git@github.com:tianqixin/runoob-git-test.git (fetch)

origin    git@github.com:tianqixin/runoob-git-test.git (push)

[Git 标签](https://www.runoob.com/git/git-tag.html)

[Git Gitee](https://www.runoob.com/git/git-gitee.html)

## 2 篇笔记 写笔记

1. leon

  105\*\*\*6491@qq.co

[参考地址](https://blog.twofei.com/695/)

54

* + 执行 git fetch origin master 时，它的意思是从名为 **origin** 的远程上拉取名为 **master** 的分支到本地分支 **origin/master** 中。既然是拉取代码，当然需要同时指定远程名与分支名，所以分开写。
  + 执行 git merge origin/master 时，它的意思是合并名为 **origin/master** 的分支到当前所在分支。既然是分支的合并，当然就与远程名没有直接的关系，所以没有出现远程名。需要指定的是被合并的分支。
  + 执行 git push origin master 时，它的意思是推送本地的 **master** 分支到远程 **origin**，涉及到远程以及分支，当然也得分开写了。
  + 还可以一次性拉取多个分支的代码：git fetch origin master stable oldstable；
  + 也还可以一次性合并多个分支的代码：git merge origin/master hotfix-2275 hotfix-2276 hotfix-2290；

[leon](javascript:;)

   leon

  105\*\*\*6491@qq.co

[参考地址](https://blog.twofei.com/695/)

11个月前 (09-08)

1. earhtnorth

  ear\*\*\*orth@163.com

5

ssh 访问 gitHub 出错如下：

$ ssh -T git@github.com

The authenticity of host 'github.com (140.82.118.4)' can't be established.

RSA key fingerprint is SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8.

Are you sure you want to continue connecting (yes/no)?

Host key verification failed.

解决办法：（将GitHub添加到信任主机列表后，可以成功访问）

$ ssh-keyscan -t rsa github.com >> ~/.ssh/known\_hosts

# github.com:22 SSH-2.0-babeld-d45c1532

$ ssh -T git@github.com

Warning: Permanently added the RSA host key for IP address '140.82.118.4' to the list of known hosts.

Hi earthnorth! You've successfully authenticated, but GitHub does not provide shell access.

## Git 服务器搭建

### 1、安装Git

$ yum install curl-devel expat-devel gettext-devel openssl-devel zlib-devel perl-devel

$ yum install git

接下来我们 创建一个git用户组和用户，用来运行git服务：

$ groupadd git

$ useradd git -g git

### 2、创建证书登录

收集所有需要登录的用户的公钥，公钥位于id\_rsa.pub文件中，把我们的公钥导入到/home/git/.ssh/authorized\_keys文件里，一行一个。

如果没有该文件创建它：

$ cd /home/git/

$ mkdir .ssh

$ chmod 755 .ssh

$ touch .ssh/authorized\_keys

$ chmod 644 .ssh/authorized\_keys

### 3、初始化Git仓库

首先我们选定一个目录作为Git仓库，假定是/home/gitrepo/runoob.git，在/home/gitrepo目录下输入命令：

$ cd /home

$ mkdir gitrepo

$ chown git:git gitrepo/

$ cd gitrepo

$ git init --bare runoob.git

Initialized empty Git repository in /home/gitrepo/runoob.git/

以上命令Git创建一个空仓库，服务器上的Git仓库通常都以.git结尾。然后，把仓库所属用户改为git：

$ chown -R git:git runoob.git

### 4、克隆仓库

$ git clone git@192.168.45.4:/home/gitrepo/runoob.git

Cloning into 'runoob'...

warning: You appear to have cloned an empty repository.

Checking connectivity... done.

192.168.45.4 为 Git 所在服务器 ip ，你需要将其修改为你自己的 Git 服务 ip。

这样我们的 Git 服务器安装就完成。

## git 分支查看与切换

git 分支查看与切换

|  |  |
| --- | --- |
| 1  2 | # 1.查看所有分支  > git branch -a |

|  |  |
| --- | --- |
| 1  2  3  4  5 | # 2.查看当前使用分支(结果列表中前面标\*号的表示当前使用分支)  > git branch    # 3.切换分支  > git checkout 分支名 |

## 从仓库中删除敏感数据

https://docs.github.com/cn/github/authenticating-to-github/removing-sensitive-data-from-a-repository

## 从仓库的历史记录中删除文件

https://docs.github.com/cn/github/managing-large-files/removing-files-from-a-repositorys-history

## 重置存储库

有时候当不小心上传了一个很大的文件到github上时，后面在git clone的时候需要等很长的时间（这是因为git克隆代码库时会将所有历史的文件都保存下来），即使你删除了这个文件，但是为了实现正常的分支切换，这个文件的信息依然保存在代码库里。或者有时将一些私密的文件不小心上传了（会泄露个人隐私），所以为了解决这个问题，可以考虑重置代码库，即将所有的历史提交信息删除。

尝试 运行 git checkout --orphan latest\_branch （创建分支）

添加所有文件git add -A

提交更改git commit -am "commit message"（提交）

删除分支git branch -D master（删除主分支）

将当前分支重命名git branch -m master

最后，强制更新存储库。git push -f origin master

## 静态网站生成器 static site generator

github 官方教程给出的是 jekyll，国内搜建站教程也是满眼的 jekll. jekll 是 ruby 写的。搭建本地环境还需要搭建完整的 ruby。这个东西一点都不熟悉，插件下载还要找代理，所有操作都要按照网上的教程按部就班，跟瞎子摸象一样。

github 是能够使用其他静态生成器的，比如 pelican 。由 python 编写，用户基数也很大，拥有丰富的插件和主题。

## emoji-cheat-sheet

This cheat sheet is automatically generated from [GitHub Emoji API](https://api.github.com/emojis) and [Unicode Full Emoji List](https://unicode.org/emoji/charts/full-emoji-list.html).

### Table of Contents

* [Smileys & Emotion](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion)
* [People & Body](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body)
* [Animals & Nature](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature)
* [Food & Drink](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink)
* [Travel & Places](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places)
* [Activities](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities)
* [Objects](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects)
* [Symbols](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols)
* [Flags](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags)
* [GitHub Custom Emoji](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#github-custom-emoji)

### Smileys & Emotion

* [Face Smiling](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#face-smiling)
* [Face Affection](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#face-affection)
* [Face Tongue](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#face-tongue)
* [Face Hand](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#face-hand)
* [Face Neutral Skeptical](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#face-neutral-skeptical)
* [Face Sleepy](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#face-sleepy)
* [Face Unwell](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#face-unwell)
* [Face Hat](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#face-hat)
* [Face Glasses](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#face-glasses)
* [Face Concerned](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#face-concerned)
* [Face Negative](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#face-negative)
* [Face Costume](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#face-costume)
* [Cat Face](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#cat-face)
* [Monkey Face](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#monkey-face)
* [Emotion](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#emotion)

#### Face Smiling

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😀 | :grinning: | 😃 | :smiley: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😄 | :smile: | 😁 | :grin: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😆 | :laughing: :satisfied: | 😅 | :sweat\_smile: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🤣 | :rofl: | 😂 | :joy: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🙂 | :slightly\_smiling\_face: | 🙃 | :upside\_down\_face: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😉 | :wink: | 😊 | :blush: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😇 | :innocent: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Face Affection

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🥰 | :smiling\_face\_with\_three\_hearts: | 😍 | :heart\_eyes: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🤩 | :star\_struck: | 😘 | :kissing\_heart: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😗 | :kissing: | ☺️ | :relaxed: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😚 | :kissing\_closed\_eyes: | 😙 | :kissing\_smiling\_eyes: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Face Tongue

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😋 | :yum: | 😛 | :stuck\_out\_tongue: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😜 | :stuck\_out\_tongue\_winking\_eye: | 🤪 | :zany\_face: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😝 | :stuck\_out\_tongue\_closed\_eyes: | 🤑 | :money\_mouth\_face: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Face Hand

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🤗 | :hugs: | 🤭 | :hand\_over\_mouth: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🤫 | :shushing\_face: | 🤔 | :thinking: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Face Neutral Skeptical

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🤐 | :zipper\_mouth\_face: | 🤨 | :raised\_eyebrow: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😐 | :neutral\_face: | 😑 | :expressionless: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😶 | :no\_mouth: | 😏 | :smirk: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😒 | :unamused: | 🙄 | :roll\_eyes: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😬 | :grimacing: | 🤥 | :lying\_face: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Face Sleepy

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😌 | :relieved: | 😔 | :pensive: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😪 | :sleepy: | 🤤 | :drooling\_face: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😴 | :sleeping: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Face Unwell

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😷 | :mask: | 🤒 | :face\_with\_thermometer: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🤕 | :face\_with\_head\_bandage: | 🤢 | :nauseated\_face: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🤮 | :vomiting\_face: | 🤧 | :sneezing\_face: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🥵 | :hot\_face: | 🥶 | :cold\_face: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🥴 | :woozy\_face: | 😵 | :dizzy\_face: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🤯 | :exploding\_head: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Face Hat

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🤠 | :cowboy\_hat\_face: | 🥳 | :partying\_face: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Face Glasses

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😎 | :sunglasses: | 🤓 | :nerd\_face: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🧐 | :monocle\_face: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Face Concerned

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😕 | :confused: | 😟 | :worried: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🙁 | :slightly\_frowning\_face: | ☹️ | :frowning\_face: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😮 | :open\_mouth: | 😯 | :hushed: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😲 | :astonished: | 😳 | :flushed: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🥺 | :pleading\_face: | 😦 | :frowning: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😧 | :anguished: | 😨 | :fearful: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😰 | :cold\_sweat: | 😥 | :disappointed\_relieved: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😢 | :cry: | 😭 | :sob: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😱 | :scream: | 😖 | :confounded: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😣 | :persevere: | 😞 | :disappointed: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😓 | :sweat: | 😩 | :weary: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😫 | :tired\_face: | 🥱 | :yawning\_face: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Face Negative

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😤 | :triumph: | 😡 | :pout: :rage: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😠 | :angry: | 🤬 | :cursing\_face: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😈 | :smiling\_imp: | 👿 | :imp: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 💀 | :skull: | ☠️ | :skull\_and\_crossbones: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Face Costume

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 💩 | :hankey: :poop: :shit: | 🤡 | :clown\_face: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 👹 | :japanese\_ogre: | 👺 | :japanese\_goblin: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 👻 | :ghost: | 👽 | :alien: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 👾 | :space\_invader: | 🤖 | :robot: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Cat Face

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😺 | :smiley\_cat: | 😸 | :smile\_cat: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😹 | :joy\_cat: | 😻 | :heart\_eyes\_cat: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😼 | :smirk\_cat: | 😽 | :kissing\_cat: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🙀 | :scream\_cat: | 😿 | :crying\_cat\_face: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 😾 | :pouting\_cat: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Monkey Face

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🙈 | :see\_no\_evil: | 🙉 | :hear\_no\_evil: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🙊 | :speak\_no\_evil: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Emotion

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 💋 | :kiss: | 💌 | :love\_letter: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 💘 | :cupid: | 💝 | :gift\_heart: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 💖 | :sparkling\_heart: | 💗 | :heartpulse: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 💓 | :heartbeat: | 💞 | :revolving\_hearts: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 💕 | :two\_hearts: | 💟 | :heart\_decoration: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | ❣️ | :heavy\_heart\_exclamation: | 💔 | :broken\_heart: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | ❤️ | :heart: | 🧡 | :orange\_heart: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 💛 | :yellow\_heart: | 💚 | :green\_heart: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 💙 | :blue\_heart: | 💜 | :purple\_heart: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🤎 | :brown\_heart: | 🖤 | :black\_heart: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🤍 | :white\_heart: | 💯 | :100: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 💢 | :anger: | 💥 | :boom: :collision: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 💫 | :dizzy: | 💦 | :sweat\_drops: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 💨 | :dash: | 🕳️ | :hole: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 💣 | :bomb: | 💬 | :speech\_balloon: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 👁️‍🗨️ | :eye\_speech\_bubble: | 🗨️ | :left\_speech\_bubble: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 🗯️ | :right\_anger\_bubble: | 💭 | :thought\_balloon: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#smileys--emotion) | 💤 | :zzz: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

### People & Body

* [Hand Fingers Open](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#hand-fingers-open)
* [Hand Fingers Partial](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#hand-fingers-partial)
* [Hand Single Finger](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#hand-single-finger)
* [Hand Fingers Closed](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#hand-fingers-closed)
* [Hands](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#hands)
* [Hand Prop](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#hand-prop)
* [Body Parts](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#body-parts)
* [Person](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#person)
* [Person Gesture](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#person-gesture)
* [Person Role](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#person-role)
* [Person Fantasy](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#person-fantasy)
* [Person Activity](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#person-activity)
* [Person Sport](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#person-sport)
* [Person Resting](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#person-resting)
* [Family](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#family)
* [Person Symbol](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#person-symbol)

#### Hand Fingers Open

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👋 | :wave: | 🤚 | :raised\_back\_of\_hand: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🖐️ | :raised\_hand\_with\_fingers\_splayed: | ✋ | :hand: :raised\_hand: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🖖 | :vulcan\_salute: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Hand Fingers Partial

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👌 | :ok\_hand: | 🤏 | :pinching\_hand: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | ✌️ | :v: | 🤞 | :crossed\_fingers: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🤟 | :love\_you\_gesture: | 🤘 | :metal: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🤙 | :call\_me\_hand: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Hand Single Finger

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👈 | :point\_left: | 👉 | :point\_right: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👆 | :point\_up\_2: | 🖕 | :fu: :middle\_finger: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👇 | :point\_down: | ☝️ | :point\_up: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Hand Fingers Closed

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👍 | :+1: :thumbsup: | 👎 | :-1: :thumbsdown: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | ✊ | :fist: :fist\_raised: | 👊 | :facepunch: :fist\_oncoming: :punch: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🤛 | :fist\_left: | 🤜 | :fist\_right: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Hands

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👏 | :clap: | 🙌 | :raised\_hands: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👐 | :open\_hands: | 🤲 | :palms\_up\_together: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🤝 | :handshake: | 🙏 | :pray: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Hand Prop

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | ✍️ | :writing\_hand: | 💅 | :nail\_care: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🤳 | :selfie: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Body Parts

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 💪 | :muscle: | 🦾 | :mechanical\_arm: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🦿 | :mechanical\_leg: | 🦵 | :leg: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🦶 | :foot: | 👂 | :ear: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🦻 | :ear\_with\_hearing\_aid: | 👃 | :nose: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧠 | :brain: | 🦷 | :tooth: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🦴 | :bone: | 👀 | :eyes: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👁️ | :eye: | 👅 | :tongue: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👄 | :lips: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Person

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👶 | :baby: | 🧒 | :child: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👦 | :boy: | 👧 | :girl: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧑 | :adult: | 👱 | :blond\_haired\_person: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👨 | :man: | 🧔 | :bearded\_person: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👨‍🦰 | :red\_haired\_man: | 👨‍🦱 | :curly\_haired\_man: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👨‍🦳 | :white\_haired\_man: | 👨‍🦲 | :bald\_man: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👩 | :woman: | 👩‍🦰 | :red\_haired\_woman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧑‍🦰 | :person\_red\_hair: | 👩‍🦱 | :curly\_haired\_woman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧑‍🦱 | :person\_curly\_hair: | 👩‍🦳 | :white\_haired\_woman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧑‍🦳 | :person\_white\_hair: | 👩‍🦲 | :bald\_woman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧑‍🦲 | :person\_bald: | 👱‍♀️ | :blond\_haired\_woman: :blonde\_woman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👱‍♂️ | :blond\_haired\_man: | 🧓 | :older\_adult: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👴 | :older\_man: | 👵 | :older\_woman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Person Gesture

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🙍 | :frowning\_person: | 🙍‍♂️ | :frowning\_man: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🙍‍♀️ | :frowning\_woman: | 🙎 | :pouting\_face: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🙎‍♂️ | :pouting\_man: | 🙎‍♀️ | :pouting\_woman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🙅 | :no\_good: | 🙅‍♂️ | :ng\_man: :no\_good\_man: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🙅‍♀️ | :ng\_woman: :no\_good\_woman: | 🙆 | :ok\_person: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🙆‍♂️ | :ok\_man: | 🙆‍♀️ | :ok\_woman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 💁 | :information\_desk\_person: :tipping\_hand\_person: | 💁‍♂️ | :sassy\_man: :tipping\_hand\_man: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 💁‍♀️ | :sassy\_woman: :tipping\_hand\_woman: | 🙋 | :raising\_hand: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🙋‍♂️ | :raising\_hand\_man: | 🙋‍♀️ | :raising\_hand\_woman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧏 | :deaf\_person: | 🧏‍♂️ | :deaf\_man: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧏‍♀️ | :deaf\_woman: | 🙇 | :bow: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🙇‍♂️ | :bowing\_man: | 🙇‍♀️ | :bowing\_woman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🤦 | :facepalm: | 🤦‍♂️ | :man\_facepalming: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🤦‍♀️ | :woman\_facepalming: | 🤷 | :shrug: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🤷‍♂️ | :man\_shrugging: | 🤷‍♀️ | :woman\_shrugging: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Person Role

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧑‍⚕️ | :health\_worker: | 👨‍⚕️ | :man\_health\_worker: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👩‍⚕️ | :woman\_health\_worker: | 🧑‍🎓 | :student: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👨‍🎓 | :man\_student: | 👩‍🎓 | :woman\_student: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧑‍🏫 | :teacher: | 👨‍🏫 | :man\_teacher: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👩‍🏫 | :woman\_teacher: | 🧑‍⚖️ | :judge: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👨‍⚖️ | :man\_judge: | 👩‍⚖️ | :woman\_judge: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧑‍🌾 | :farmer: | 👨‍🌾 | :man\_farmer: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👩‍🌾 | :woman\_farmer: | 🧑‍🍳 | :cook: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👨‍🍳 | :man\_cook: | 👩‍🍳 | :woman\_cook: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧑‍🔧 | :mechanic: | 👨‍🔧 | :man\_mechanic: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👩‍🔧 | :woman\_mechanic: | 🧑‍🏭 | :factory\_worker: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👨‍🏭 | :man\_factory\_worker: | 👩‍🏭 | :woman\_factory\_worker: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧑‍💼 | :office\_worker: | 👨‍💼 | :man\_office\_worker: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👩‍💼 | :woman\_office\_worker: | 🧑‍🔬 | :scientist: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👨‍🔬 | :man\_scientist: | 👩‍🔬 | :woman\_scientist: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧑‍💻 | :technologist: | 👨‍💻 | :man\_technologist: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👩‍💻 | :woman\_technologist: | 🧑‍🎤 | :singer: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👨‍🎤 | :man\_singer: | 👩‍🎤 | :woman\_singer: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧑‍🎨 | :artist: | 👨‍🎨 | :man\_artist: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👩‍🎨 | :woman\_artist: | 🧑‍✈️ | :pilot: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👨‍✈️ | :man\_pilot: | 👩‍✈️ | :woman\_pilot: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧑‍🚀 | :astronaut: | 👨‍🚀 | :man\_astronaut: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👩‍🚀 | :woman\_astronaut: | 🧑‍🚒 | :firefighter: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👨‍🚒 | :man\_firefighter: | 👩‍🚒 | :woman\_firefighter: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👮 | :cop: :police\_officer: | 👮‍♂️ | :policeman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👮‍♀️ | :policewoman: | 🕵️ | :detective: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🕵️‍♂️ | :male\_detective: | 🕵️‍♀️ | :female\_detective: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 💂 | :guard: | 💂‍♂️ | :guardsman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 💂‍♀️ | :guardswoman: | 👷 | :construction\_worker: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👷‍♂️ | :construction\_worker\_man: | 👷‍♀️ | :construction\_worker\_woman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🤴 | :prince: | 👸 | :princess: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👳 | :person\_with\_turban: | 👳‍♂️ | :man\_with\_turban: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👳‍♀️ | :woman\_with\_turban: | 👲 | :man\_with\_gua\_pi\_mao: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧕 | :woman\_with\_headscarf: | 🤵 | :man\_in\_tuxedo: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👰 | :bride\_with\_veil: | 🤰 | :pregnant\_woman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🤱 | :breast\_feeding: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Person Fantasy

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👼 | :angel: | 🎅 | :santa: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🤶 | :mrs\_claus: | 🦸 | :superhero: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🦸‍♂️ | :superhero\_man: | 🦸‍♀️ | :superhero\_woman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🦹 | :supervillain: | 🦹‍♂️ | :supervillain\_man: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🦹‍♀️ | :supervillain\_woman: | 🧙 | :mage: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧙‍♂️ | :mage\_man: | 🧙‍♀️ | :mage\_woman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧚 | :fairy: | 🧚‍♂️ | :fairy\_man: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧚‍♀️ | :fairy\_woman: | 🧛 | :vampire: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧛‍♂️ | :vampire\_man: | 🧛‍♀️ | :vampire\_woman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧜 | :merperson: | 🧜‍♂️ | :merman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧜‍♀️ | :mermaid: | 🧝 | :elf: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧝‍♂️ | :elf\_man: | 🧝‍♀️ | :elf\_woman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧞 | :genie: | 🧞‍♂️ | :genie\_man: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧞‍♀️ | :genie\_woman: | 🧟 | :zombie: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧟‍♂️ | :zombie\_man: | 🧟‍♀️ | :zombie\_woman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Person Activity

| **ico** | **shortcode** | **ico** | **shortcode** |
| --- | --- | --- | --- |
| 💆 | :massage: | 💆‍♂️ | :massage\_man: |
| 💆‍♀️ | :massage\_woman: | 💇 | :haircut: |
| 💇‍♂️ | :haircut\_man: | 💇‍♀️ | :haircut\_woman: |
| 🚶 | :walking: | 🚶‍♂️ | :walking\_man: |
| 🚶‍♀️ | :walking\_woman: | 🧍 | :standing\_person: |
| 🧍‍♂️ | :standing\_man: | 🧍‍♀️ | :standing\_woman: |
| 🧎 | :kneeling\_person: | 🧎‍♂️ | :kneeling\_man: |
| 🧎‍♀️ | :kneeling\_woman: | 🧑‍🦯 | :person\_with\_probing\_cane: |
| 👨‍🦯 | :man\_with\_probing\_cane: | 👩‍🦯 | :woman\_with\_probing\_cane: |
| 🧑‍🦼 | :person\_in\_motorized\_wheelchair: | 👨‍🦼 | :man\_in\_motorized\_wheelchair: |
| 👩‍🦼 | :woman\_in\_motorized\_wheelchair: | 🧑‍🦽 | :person\_in\_manual\_wheelchair: |
| 👨‍🦽 | :man\_in\_manual\_wheelchair: | 👩‍🦽 | :woman\_in\_manual\_wheelchair: |
| 🏃 | :runner: :running: | 🏃‍♂️ | :running\_man: |
| 🏃‍♀️ | :running\_woman: | 💃 | :dancer: :woman\_dancing: |
| 🕺 | :man\_dancing: | 🕴️ | :business\_suit\_levitating: |
| 👯 | :dancers: | 👯‍♂️ | :dancing\_men: |
| 👯‍♀️ | :dancing\_women: | 🧖 | :sauna\_person: |
| 🧖‍♂️ | :sauna\_man: | 🧖‍♀️ | :sauna\_woman: |
| 🧗 | :climbing: | 🧗‍♂️ | :climbing\_man: |
| 🧗‍♀️ | :climbing\_woman: |  |  |

#### Person Sport

| **ico** | **shortcode** | **ico** | **shortcode** |
| --- | --- | --- | --- |
| 🤺 | :person\_fencing: | 🏇 | :horse\_racing: |
| ⛷️ | :skier: | 🏂 | :snowboarder: |
| 🏌️ | :golfing: | 🏌️‍♂️ | :golfing\_man: |
| 🏌️‍♀️ | :golfing\_woman: | 🏄 | :surfer: |
| 🏄‍♂️ | :surfing\_man: | 🏄‍♀️ | :surfing\_woman: |
| 🚣 | :rowboat: | 🚣‍♂️ | :rowing\_man: |
| 🚣‍♀️ | :rowing\_woman: | 🏊 | :swimmer: |
| 🏊‍♂️ | :swimming\_man: | 🏊‍♀️ | :swimming\_woman: |
| ⛹️ | :bouncing\_ball\_person: | ⛹️‍♂️ | :basketball\_man: :bouncing\_ball\_man: |
| ⛹️‍♀️ | :basketball\_woman: :bouncing\_ball\_woman: | 🏋️ | :weight\_lifting: |
| 🏋️‍♂️ | :weight\_lifting\_man: | 🏋️‍♀️ | :weight\_lifting\_woman: |
| 🚴 | :bicyclist: | 🚴‍♂️ | :biking\_man: |
| 🚴‍♀️ | :biking\_woman: | 🚵 | :mountain\_bicyclist: |
| 🚵‍♂️ | :mountain\_biking\_man: | 🚵‍♀️ | :mountain\_biking\_woman: |
| 🤸 | :cartwheeling: | 🤸‍♂️ | :man\_cartwheeling: |
| 🤸‍♀️ | :woman\_cartwheeling: | 🤼 | :wrestling: |
| 🤼‍♂️ | :men\_wrestling: | 🤼‍♀️ | :women\_wrestling: |
| 🤽 | :water\_polo: | 🤽‍♂️ | :man\_playing\_water\_polo: |
| 🤽‍♀️ | :woman\_playing\_water\_polo: | 🤾 | :handball\_person: |
| 🤾‍♂️ | :man\_playing\_handball: | 🤾‍♀️ | :woman\_playing\_handball: |
| 🤹 | :juggling\_person: | 🤹‍♂️ | :man\_juggling: |
| 🤹‍♀️ | :woman\_juggling: |  |  |

#### Person Resting

| **ico** | **shortcode** | **ico** | **shortcode** |
| --- | --- | --- | --- |
| 🧘 | :lotus\_position: | 🧘‍♂️ | :lotus\_position\_man: |
| 🧘‍♀️ | :lotus\_position\_woman: | 🛀 | :bath: |
| 🛌 | :sleeping\_bed: |  |  |

#### Family

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🧑‍🤝‍🧑 | :people\_holding\_hands: | 👭 | :two\_women\_holding\_hands: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👫 | :couple: | 👬 | :two\_men\_holding\_hands: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 💏 | :couplekiss: | 👩‍❤️‍💋‍👨 | :couplekiss\_man\_woman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👨‍❤️‍💋‍👨 | :couplekiss\_man\_man: | 👩‍❤️‍💋‍👩 | :couplekiss\_woman\_woman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 💑 | :couple\_with\_heart: | 👩‍❤️‍👨 | :couple\_with\_heart\_woman\_man: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👨‍❤️‍👨 | :couple\_with\_heart\_man\_man: | 👩‍❤️‍👩 | :couple\_with\_heart\_woman\_woman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👪 | :family: | 👨‍👩‍👦 | :family\_man\_woman\_boy: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👨‍👩‍👧 | :family\_man\_woman\_girl: | 👨‍👩‍👧‍👦 | :family\_man\_woman\_girl\_boy: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👨‍👩‍👦‍👦 | :family\_man\_woman\_boy\_boy: | 👨‍👩‍👧‍👧 | :family\_man\_woman\_girl\_girl: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👨‍👨‍👦 | :family\_man\_man\_boy: | 👨‍👨‍👧 | :family\_man\_man\_girl: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👨‍👨‍👧‍👦 | :family\_man\_man\_girl\_boy: | 👨‍👨‍👦‍👦 | :family\_man\_man\_boy\_boy: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👨‍👨‍👧‍👧 | :family\_man\_man\_girl\_girl: | 👩‍👩‍👦 | :family\_woman\_woman\_boy: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👩‍👩‍👧 | :family\_woman\_woman\_girl: | 👩‍👩‍👧‍👦 | :family\_woman\_woman\_girl\_boy: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👩‍👩‍👦‍👦 | :family\_woman\_woman\_boy\_boy: | 👩‍👩‍👧‍👧 | :family\_woman\_woman\_girl\_girl: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👨‍👦 | :family\_man\_boy: | 👨‍👦‍👦 | :family\_man\_boy\_boy: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👨‍👧 | :family\_man\_girl: | 👨‍👧‍👦 | :family\_man\_girl\_boy: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👨‍👧‍👧 | :family\_man\_girl\_girl: | 👩‍👦 | :family\_woman\_boy: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👩‍👦‍👦 | :family\_woman\_boy\_boy: | 👩‍👧 | :family\_woman\_girl: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👩‍👧‍👦 | :family\_woman\_girl\_boy: | 👩‍👧‍👧 | :family\_woman\_girl\_girl: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Person Symbol

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 🗣️ | :speaking\_head: | 👤 | :bust\_in\_silhouette: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#people--body) | 👥 | :busts\_in\_silhouette: | 👣 | :footprints: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

### Animals & Nature

* [Animal Mammal](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animal-mammal)
* [Animal Bird](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animal-bird)
* [Animal Amphibian](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animal-amphibian)
* [Animal Reptile](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animal-reptile)
* [Animal Marine](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animal-marine)
* [Animal Bug](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animal-bug)
* [Plant Flower](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#plant-flower)
* [Plant Other](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#plant-other)

#### Animal Mammal

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐵 | :monkey\_face: | 🐒 | :monkey: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🦍 | :gorilla: | 🦧 | :orangutan: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐶 | :dog: | 🐕 | :dog2: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🦮 | :guide\_dog: | 🐕‍🦺 | :service\_dog: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐩 | :poodle: | 🐺 | :wolf: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🦊 | :fox\_face: | 🦝 | :raccoon: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐱 | :cat: | 🐈 | :cat2: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🦁 | :lion: | 🐯 | :tiger: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐅 | :tiger2: | 🐆 | :leopard: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐴 | :horse: | 🐎 | :racehorse: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🦄 | :unicorn: | 🦓 | :zebra: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🦌 | :deer: | 🐮 | :cow: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐂 | :ox: | 🐃 | :water\_buffalo: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐄 | :cow2: | 🐷 | :pig: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐖 | :pig2: | 🐗 | :boar: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐽 | :pig\_nose: | 🐏 | :ram: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐑 | :sheep: | 🐐 | :goat: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐪 | :dromedary\_camel: | 🐫 | :camel: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🦙 | :llama: | 🦒 | :giraffe: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐘 | :elephant: | 🦏 | :rhinoceros: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🦛 | :hippopotamus: | 🐭 | :mouse: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐁 | :mouse2: | 🐀 | :rat: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐹 | :hamster: | 🐰 | :rabbit: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐇 | :rabbit2: | 🐿️ | :chipmunk: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🦔 | :hedgehog: | 🦇 | :bat: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐻 | :bear: | 🐨 | :koala: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐼 | :panda\_face: | 🦥 | :sloth: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🦦 | :otter: | 🦨 | :skunk: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🦘 | :kangaroo: | 🦡 | :badger: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐾 | :feet: :paw\_prints: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Animal Bird

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🦃 | :turkey: | 🐔 | :chicken: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐓 | :rooster: | 🐣 | :hatching\_chick: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐤 | :baby\_chick: | 🐥 | :hatched\_chick: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐦 | :bird: | 🐧 | :penguin: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🕊️ | :dove: | 🦅 | :eagle: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🦆 | :duck: | 🦢 | :swan: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🦉 | :owl: | 🦩 | :flamingo: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🦚 | :peacock: | 🦜 | :parrot: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Animal Amphibian

|  | **ico** | **shortcode** |  |
| --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐸 | :frog: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Animal Reptile

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐊 | :crocodile: | 🐢 | :turtle: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🦎 | :lizard: | 🐍 | :snake: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐲 | :dragon\_face: | 🐉 | :dragon: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🦕 | :sauropod: | 🦖 | :t-rex: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Animal Marine

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐳 | :whale: | 🐋 | :whale2: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐬 | :dolphin: :flipper: | 🐟 | :fish: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐠 | :tropical\_fish: | 🐡 | :blowfish: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🦈 | :shark: | 🐙 | :octopus: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐚 | :shell: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Animal Bug

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐌 | :snail: | 🦋 | :butterfly: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐛 | :bug: | 🐜 | :ant: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🐝 | :bee: :honeybee: | 🐞 | :beetle: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🦗 | :cricket: | 🕷️ | :spider: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🕸️ | :spider\_web: | 🦂 | :scorpion: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🦟 | :mosquito: | 🦠 | :microbe: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Plant Flower

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 💐 | :bouquet: | 🌸 | :cherry\_blossom: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 💮 | :white\_flower: | 🏵️ | :rosette: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🌹 | :rose: | 🥀 | :wilted\_flower: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🌺 | :hibiscus: | 🌻 | :sunflower: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🌼 | :blossom: | 🌷 | :tulip: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Plant Other

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🌱 | :seedling: | 🌲 | :evergreen\_tree: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🌳 | :deciduous\_tree: | 🌴 | :palm\_tree: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🌵 | :cactus: | 🌾 | :ear\_of\_rice: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🌿 | :herb: | ☘️ | :shamrock: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🍀 | :four\_leaf\_clover: | 🍁 | :maple\_leaf: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#animals--nature) | 🍂 | :fallen\_leaf: | 🍃 | :leaves: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

### Food & Drink

* [Food Fruit](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food-fruit)
* [Food Vegetable](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food-vegetable)
* [Food Prepared](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food-prepared)
* [Food Asian](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food-asian)
* [Food Marine](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food-marine)
* [Food Sweet](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food-sweet)
* [Drink](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#drink)
* [Dishware](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#dishware)

#### Food Fruit

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍇 | :grapes: | 🍈 | :melon: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍉 | :watermelon: | 🍊 | :mandarin: :orange: :tangerine: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍋 | :lemon: | 🍌 | :banana: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍍 | :pineapple: | 🥭 | :mango: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍎 | :apple: | 🍏 | :green\_apple: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍐 | :pear: | 🍑 | :peach: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍒 | :cherries: | 🍓 | :strawberry: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🥝 | :kiwi\_fruit: | 🍅 | :tomato: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🥥 | :coconut: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Food Vegetable

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🥑 | :avocado: | 🍆 | :eggplant: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🥔 | :potato: | 🥕 | :carrot: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🌽 | :corn: | 🌶️ | :hot\_pepper: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🥒 | :cucumber: | 🥬 | :leafy\_green: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🥦 | :broccoli: | 🧄 | :garlic: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🧅 | :onion: | 🍄 | :mushroom: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🥜 | :peanuts: | 🌰 | :chestnut: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Food Prepared

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍞 | :bread: | 🥐 | :croissant: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🥖 | :baguette\_bread: | 🥨 | :pretzel: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🥯 | :bagel: | 🥞 | :pancakes: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🧇 | :waffle: | 🧀 | :cheese: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍖 | :meat\_on\_bone: | 🍗 | :poultry\_leg: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🥩 | :cut\_of\_meat: | 🥓 | :bacon: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍔 | :hamburger: | 🍟 | :fries: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍕 | :pizza: | 🌭 | :hotdog: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🥪 | :sandwich: | 🌮 | :taco: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🌯 | :burrito: | 🥙 | :stuffed\_flatbread: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🧆 | :falafel: | 🥚 | :egg: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍳 | :fried\_egg: | 🥘 | :shallow\_pan\_of\_food: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍲 | :stew: | 🥣 | :bowl\_with\_spoon: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🥗 | :green\_salad: | 🍿 | :popcorn: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🧈 | :butter: | 🧂 | :salt: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🥫 | :canned\_food: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Food Asian

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍱 | :bento: | 🍘 | :rice\_cracker: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍙 | :rice\_ball: | 🍚 | :rice: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍛 | :curry: | 🍜 | :ramen: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍝 | :spaghetti: | 🍠 | :sweet\_potato: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍢 | :oden: | 🍣 | :sushi: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍤 | :fried\_shrimp: | 🍥 | :fish\_cake: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🥮 | :moon\_cake: | 🍡 | :dango: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🥟 | :dumpling: | 🥠 | :fortune\_cookie: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🥡 | :takeout\_box: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Food Marine

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🦀 | :crab: | 🦞 | :lobster: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🦐 | :shrimp: | 🦑 | :squid: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🦪 | :oyster: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Food Sweet

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍦 | :icecream: | 🍧 | :shaved\_ice: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍨 | :ice\_cream: | 🍩 | :doughnut: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍪 | :cookie: | 🎂 | :birthday: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍰 | :cake: | 🧁 | :cupcake: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🥧 | :pie: | 🍫 | :chocolate\_bar: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍬 | :candy: | 🍭 | :lollipop: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍮 | :custard: | 🍯 | :honey\_pot: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Drink

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍼 | :baby\_bottle: | 🥛 | :milk\_glass: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | ☕ | :coffee: | 🍵 | :tea: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍶 | :sake: | 🍾 | :champagne: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍷 | :wine\_glass: | 🍸 | :cocktail: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍹 | :tropical\_drink: | 🍺 | :beer: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍻 | :beers: | 🥂 | :clinking\_glasses: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🥃 | :tumbler\_glass: | 🥤 | :cup\_with\_straw: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🧃 | :beverage\_box: | 🧉 | :mate: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🧊 | :ice\_cube: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Dishware

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🥢 | :chopsticks: | 🍽️ | :plate\_with\_cutlery: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🍴 | :fork\_and\_knife: | 🥄 | :spoon: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#food--drink) | 🔪 | :hocho: :knife: | 🏺 | :amphora: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

### Travel & Places

* [Place Map](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#place-map)
* [Place Geographic](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#place-geographic)
* [Place Building](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#place-building)
* [Place Religious](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#place-religious)
* [Place Other](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#place-other)
* [Transport Ground](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#transport-ground)
* [Transport Water](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#transport-water)
* [Transport Air](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#transport-air)
* [Hotel](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#hotel)
* [Time](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#time)
* [Sky & Weather](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#sky--weather)

#### Place Map

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌍 | :earth\_africa: | 🌎 | :earth\_americas: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌏 | :earth\_asia: | 🌐 | :globe\_with\_meridians: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🗺️ | :world\_map: | 🗾 | :japan: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🧭 | :compass: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Place Geographic

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🏔️ | :mountain\_snow: | ⛰️ | :mountain: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌋 | :volcano: | 🗻 | :mount\_fuji: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🏕️ | :camping: | 🏖️ | :beach\_umbrella: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🏜️ | :desert: | 🏝️ | :desert\_island: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🏞️ | :national\_park: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Place Building

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🏟️ | :stadium: | 🏛️ | :classical\_building: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🏗️ | :building\_construction: | 🧱 | :bricks: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🏘️ | :houses: | 🏚️ | :derelict\_house: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🏠 | :house: | 🏡 | :house\_with\_garden: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🏢 | :office: | 🏣 | :post\_office: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🏤 | :european\_post\_office: | 🏥 | :hospital: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🏦 | :bank: | 🏨 | :hotel: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🏩 | :love\_hotel: | 🏪 | :convenience\_store: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🏫 | :school: | 🏬 | :department\_store: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🏭 | :factory: | 🏯 | :japanese\_castle: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🏰 | :european\_castle: | 💒 | :wedding: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🗼 | :tokyo\_tower: | 🗽 | :statue\_of\_liberty: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Place Religious

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | ⛪ | :church: | 🕌 | :mosque: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🛕 | :hindu\_temple: | 🕍 | :synagogue: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | ⛩️ | :shinto\_shrine: | 🕋 | :kaaba: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Place Other

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | ⛲ | :fountain: | ⛺ | :tent: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌁 | :foggy: | 🌃 | :night\_with\_stars: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🏙️ | :cityscape: | 🌄 | :sunrise\_over\_mountains: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌅 | :sunrise: | 🌆 | :city\_sunset: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌇 | :city\_sunrise: | 🌉 | :bridge\_at\_night: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | ♨️ | :hotsprings: | 🎠 | :carousel\_horse: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🎡 | :ferris\_wheel: | 🎢 | :roller\_coaster: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 💈 | :barber: | 🎪 | :circus\_tent: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Transport Ground

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🚂 | :steam\_locomotive: | 🚃 | :railway\_car: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🚄 | :bullettrain\_side: | 🚅 | :bullettrain\_front: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🚆 | :train2: | 🚇 | :metro: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🚈 | :light\_rail: | 🚉 | :station: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🚊 | :tram: | 🚝 | :monorail: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🚞 | :mountain\_railway: | 🚋 | :train: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🚌 | :bus: | 🚍 | :oncoming\_bus: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🚎 | :trolleybus: | 🚐 | :minibus: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🚑 | :ambulance: | 🚒 | :fire\_engine: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🚓 | :police\_car: | 🚔 | :oncoming\_police\_car: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🚕 | :taxi: | 🚖 | :oncoming\_taxi: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🚗 | :car: :red\_car: | 🚘 | :oncoming\_automobile: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🚙 | :blue\_car: | 🚚 | :truck: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🚛 | :articulated\_lorry: | 🚜 | :tractor: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🏎️ | :racing\_car: | 🏍️ | :motorcycle: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🛵 | :motor\_scooter: | 🦽 | :manual\_wheelchair: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🦼 | :motorized\_wheelchair: | 🛺 | :auto\_rickshaw: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🚲 | :bike: | 🛴 | :kick\_scooter: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🛹 | :skateboard: | 🚏 | :busstop: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🛣️ | :motorway: | 🛤️ | :railway\_track: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🛢️ | :oil\_drum: | ⛽ | :fuelpump: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🚨 | :rotating\_light: | 🚥 | :traffic\_light: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🚦 | :vertical\_traffic\_light: | 🛑 | :stop\_sign: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🚧 | :construction: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Transport Water

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | ⚓ | :anchor: | ⛵ | :boat: :sailboat: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🛶 | :canoe: | 🚤 | :speedboat: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🛳️ | :passenger\_ship: | ⛴️ | :ferry: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🛥️ | :motor\_boat: | 🚢 | :ship: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Transport Air

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | ✈️ | :airplane: | 🛩️ | :small\_airplane: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🛫 | :flight\_departure: | 🛬 | :flight\_arrival: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🪂 | :parachute: | 💺 | :seat: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🚁 | :helicopter: | 🚟 | :suspension\_railway: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🚠 | :mountain\_cableway: | 🚡 | :aerial\_tramway: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🛰️ | :artificial\_satellite: | 🚀 | :rocket: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🛸 | :flying\_saucer: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Hotel

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🛎️ | :bellhop\_bell: | 🧳 | :luggage: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Time

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | ⌛ | :hourglass: | ⏳ | :hourglass\_flowing\_sand: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | ⌚ | :watch: | ⏰ | :alarm\_clock: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | ⏱️ | :stopwatch: | ⏲️ | :timer\_clock: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🕰️ | :mantelpiece\_clock: | 🕛 | :clock12: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🕧 | :clock1230: | 🕐 | :clock1: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🕜 | :clock130: | 🕑 | :clock2: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🕝 | :clock230: | 🕒 | :clock3: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🕞 | :clock330: | 🕓 | :clock4: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🕟 | :clock430: | 🕔 | :clock5: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🕠 | :clock530: | 🕕 | :clock6: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🕡 | :clock630: | 🕖 | :clock7: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🕢 | :clock730: | 🕗 | :clock8: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🕣 | :clock830: | 🕘 | :clock9: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🕤 | :clock930: | 🕙 | :clock10: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🕥 | :clock1030: | 🕚 | :clock11: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🕦 | :clock1130: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Sky & Weather

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌑 | :new\_moon: | 🌒 | :waxing\_crescent\_moon: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌓 | :first\_quarter\_moon: | 🌔 | :moon: :waxing\_gibbous\_moon: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌕 | :full\_moon: | 🌖 | :waning\_gibbous\_moon: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌗 | :last\_quarter\_moon: | 🌘 | :waning\_crescent\_moon: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌙 | :crescent\_moon: | 🌚 | :new\_moon\_with\_face: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌛 | :first\_quarter\_moon\_with\_face: | 🌜 | :last\_quarter\_moon\_with\_face: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌡️ | :thermometer: | ☀️ | :sunny: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌝 | :full\_moon\_with\_face: | 🌞 | :sun\_with\_face: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🪐 | :ringed\_planet: | ⭐ | :star: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌟 | :star2: | 🌠 | :stars: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌌 | :milky\_way: | ☁️ | :cloud: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | ⛅ | :partly\_sunny: | ⛈️ | :cloud\_with\_lightning\_and\_rain: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌤️ | :sun\_behind\_small\_cloud: | 🌥️ | :sun\_behind\_large\_cloud: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌦️ | :sun\_behind\_rain\_cloud: | 🌧️ | :cloud\_with\_rain: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌨️ | :cloud\_with\_snow: | 🌩️ | :cloud\_with\_lightning: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌪️ | :tornado: | 🌫️ | :fog: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌬️ | :wind\_face: | 🌀 | :cyclone: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌈 | :rainbow: | 🌂 | :closed\_umbrella: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | ☂️ | :open\_umbrella: | ☔ | :umbrella: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | ⛱️ | :parasol\_on\_ground: | ⚡ | :zap: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | ❄️ | :snowflake: | ☃️ | :snowman\_with\_snow: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | ⛄ | :snowman: | ☄️ | :comet: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🔥 | :fire: | 💧 | :droplet: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#travel--places) | 🌊 | :ocean: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

### Activities

* [Event](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#event)
* [Award Medal](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#award-medal)
* [Sport](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#sport)
* [Game](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#game)
* [Arts & Crafts](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#arts--crafts)

#### Event

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🎃 | :jack\_o\_lantern: | 🎄 | :christmas\_tree: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🎆 | :fireworks: | 🎇 | :sparkler: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🧨 | :firecracker: | ✨ | :sparkles: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🎈 | :balloon: | 🎉 | :tada: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🎊 | :confetti\_ball: | 🎋 | :tanabata\_tree: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🎍 | :bamboo: | 🎎 | :dolls: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🎏 | :flags: | 🎐 | :wind\_chime: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🎑 | :rice\_scene: | 🧧 | :red\_envelope: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🎀 | :ribbon: | 🎁 | :gift: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🎗️ | :reminder\_ribbon: | 🎟️ | :tickets: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🎫 | :ticket: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Award Medal

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🎖️ | :medal\_military: | 🏆 | :trophy: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🏅 | :medal\_sports: | 🥇 | :1st\_place\_medal: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🥈 | :2nd\_place\_medal: | 🥉 | :3rd\_place\_medal: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Sport

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | ⚽ | :soccer: | ⚾ | :baseball: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🥎 | :softball: | 🏀 | :basketball: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🏐 | :volleyball: | 🏈 | :football: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🏉 | :rugby\_football: | 🎾 | :tennis: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🥏 | :flying\_disc: | 🎳 | :bowling: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🏏 | :cricket\_game: | 🏑 | :field\_hockey: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🏒 | :ice\_hockey: | 🥍 | :lacrosse: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🏓 | :ping\_pong: | 🏸 | :badminton: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🥊 | :boxing\_glove: | 🥋 | :martial\_arts\_uniform: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🥅 | :goal\_net: | ⛳ | :golf: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | ⛸️ | :ice\_skate: | 🎣 | :fishing\_pole\_and\_fish: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🤿 | :diving\_mask: | 🎽 | :running\_shirt\_with\_sash: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🎿 | :ski: | 🛷 | :sled: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🥌 | :curling\_stone: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Game

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🎯 | :dart: | 🪀 | :yo\_yo: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🪁 | :kite: | 🎱 | :8ball: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🔮 | :crystal\_ball: | 🧿 | :nazar\_amulet: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🎮 | :video\_game: | 🕹️ | :joystick: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🎰 | :slot\_machine: | 🎲 | :game\_die: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🧩 | :jigsaw: | 🧸 | :teddy\_bear: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | ♠️ | :spades: | ♥️ | :hearts: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | ♦️ | :diamonds: | ♣️ | :clubs: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | ♟️ | :chess\_pawn: | 🃏 | :black\_joker: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🀄 | :mahjong: | 🎴 | :flower\_playing\_cards: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Arts & Crafts

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🎭 | :performing\_arts: | 🖼️ | :framed\_picture: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🎨 | :art: | 🧵 | :thread: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#activities) | 🧶 | :yarn: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

### Objects

* [Clothing](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#clothing)
* [Sound](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#sound)
* [Music](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#music)
* [Musical Instrument](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#musical-instrument)
* [Phone](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#phone)
* [Computer](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#computer)
* [Light & Video](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#light--video)
* [Book Paper](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#book-paper)
* [Money](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#money)
* [Mail](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#mail)
* [Writing](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#writing)
* [Office](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#office)
* [Lock](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#lock)
* [Tool](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#tool)
* [Science](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#science)
* [Medical](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#medical)
* [Household](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#household)
* [Other Object](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#other-object)

#### Clothing

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 👓 | :eyeglasses: | 🕶️ | :dark\_sunglasses: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🥽 | :goggles: | 🥼 | :lab\_coat: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🦺 | :safety\_vest: | 👔 | :necktie: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 👕 | :shirt: :tshirt: | 👖 | :jeans: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🧣 | :scarf: | 🧤 | :gloves: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🧥 | :coat: | 🧦 | :socks: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 👗 | :dress: | 👘 | :kimono: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🥻 | :sari: | 🩱 | :one\_piece\_swimsuit: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🩲 | :swim\_brief: | 🩳 | :shorts: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 👙 | :bikini: | 👚 | :womans\_clothes: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 👛 | :purse: | 👜 | :handbag: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 👝 | :pouch: | 🛍️ | :shopping: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🎒 | :school\_satchel: | 👞 | :mans\_shoe: :shoe: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 👟 | :athletic\_shoe: | 🥾 | :hiking\_boot: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🥿 | :flat\_shoe: | 👠 | :high\_heel: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 👡 | :sandal: | 🩰 | :ballet\_shoes: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 👢 | :boot: | 👑 | :crown: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 👒 | :womans\_hat: | 🎩 | :tophat: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🎓 | :mortar\_board: | 🧢 | :billed\_cap: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | ⛑️ | :rescue\_worker\_helmet: | 📿 | :prayer\_beads: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 💄 | :lipstick: | 💍 | :ring: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 💎 | :gem: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Sound

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🔇 | :mute: | 🔈 | :speaker: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🔉 | :sound: | 🔊 | :loud\_sound: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📢 | :loudspeaker: | 📣 | :mega: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📯 | :postal\_horn: | 🔔 | :bell: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🔕 | :no\_bell: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Music

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🎼 | :musical\_score: | 🎵 | :musical\_note: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🎶 | :notes: | 🎙️ | :studio\_microphone: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🎚️ | :level\_slider: | 🎛️ | :control\_knobs: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🎤 | :microphone: | 🎧 | :headphones: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📻 | :radio: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Musical Instrument

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🎷 | :saxophone: | 🎸 | :guitar: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🎹 | :musical\_keyboard: | 🎺 | :trumpet: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🎻 | :violin: | 🪕 | :banjo: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🥁 | :drum: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Phone

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📱 | :iphone: | 📲 | :calling: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | ☎️ | :phone: :telephone: | 📞 | :telephone\_receiver: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📟 | :pager: | 📠 | :fax: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Computer

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🔋 | :battery: | 🔌 | :electric\_plug: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 💻 | :computer: | 🖥️ | :desktop\_computer: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🖨️ | :printer: | ⌨️ | :keyboard: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🖱️ | :computer\_mouse: | 🖲️ | :trackball: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 💽 | :minidisc: | 💾 | :floppy\_disk: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 💿 | :cd: | 📀 | :dvd: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🧮 | :abacus: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Light & Video

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🎥 | :movie\_camera: | 🎞️ | :film\_strip: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📽️ | :film\_projector: | 🎬 | :clapper: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📺 | :tv: | 📷 | :camera: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📸 | :camera\_flash: | 📹 | :video\_camera: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📼 | :vhs: | 🔍 | :mag: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🔎 | :mag\_right: | 🕯️ | :candle: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 💡 | :bulb: | 🔦 | :flashlight: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🏮 | :izakaya\_lantern: :lantern: | 🪔 | :diya\_lamp: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Book Paper

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📔 | :notebook\_with\_decorative\_cover: | 📕 | :closed\_book: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📖 | :book: :open\_book: | 📗 | :green\_book: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📘 | :blue\_book: | 📙 | :orange\_book: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📚 | :books: | 📓 | :notebook: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📒 | :ledger: | 📃 | :page\_with\_curl: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📜 | :scroll: | 📄 | :page\_facing\_up: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📰 | :newspaper: | 🗞️ | :newspaper\_roll: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📑 | :bookmark\_tabs: | 🔖 | :bookmark: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🏷️ | :label: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Money

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 💰 | :moneybag: | 💴 | :yen: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 💵 | :dollar: | 💶 | :euro: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 💷 | :pound: | 💸 | :money\_with\_wings: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 💳 | :credit\_card: | 🧾 | :receipt: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 💹 | :chart: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Mail

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | ✉️ | :email: :envelope: | 📧 | :e-mail: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📨 | :incoming\_envelope: | 📩 | :envelope\_with\_arrow: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📤 | :outbox\_tray: | 📥 | :inbox\_tray: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📦 | :package: | 📫 | :mailbox: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📪 | :mailbox\_closed: | 📬 | :mailbox\_with\_mail: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📭 | :mailbox\_with\_no\_mail: | 📮 | :postbox: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🗳️ | :ballot\_box: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Writing

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | ✏️ | :pencil2: | ✒️ | :black\_nib: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🖋️ | :fountain\_pen: | 🖊️ | :pen: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🖌️ | :paintbrush: | 🖍️ | :crayon: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📝 | :memo: :pencil: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Office

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 💼 | :briefcase: | 📁 | :file\_folder: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📂 | :open\_file\_folder: | 🗂️ | :card\_index\_dividers: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📅 | :date: | 📆 | :calendar: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🗒️ | :spiral\_notepad: | 🗓️ | :spiral\_calendar: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📇 | :card\_index: | 📈 | :chart\_with\_upwards\_trend: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📉 | :chart\_with\_downwards\_trend: | 📊 | :bar\_chart: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📋 | :clipboard: | 📌 | :pushpin: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📍 | :round\_pushpin: | 📎 | :paperclip: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🖇️ | :paperclips: | 📏 | :straight\_ruler: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📐 | :triangular\_ruler: | ✂️ | :scissors: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🗃️ | :card\_file\_box: | 🗄️ | :file\_cabinet: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🗑️ | :wastebasket: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Lock

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🔒 | :lock: | 🔓 | :unlock: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🔏 | :lock\_with\_ink\_pen: | 🔐 | :closed\_lock\_with\_key: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🔑 | :key: | 🗝️ | :old\_key: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Tool

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🔨 | :hammer: | 🪓 | :axe: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | ⛏️ | :pick: | ⚒️ | :hammer\_and\_pick: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🛠️ | :hammer\_and\_wrench: | 🗡️ | :dagger: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | ⚔️ | :crossed\_swords: | 🔫 | :gun: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🏹 | :bow\_and\_arrow: | 🛡️ | :shield: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🔧 | :wrench: | 🔩 | :nut\_and\_bolt: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | ⚙️ | :gear: | 🗜️ | :clamp: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | ⚖️ | :balance\_scale: | 🦯 | :probing\_cane: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🔗 | :link: | ⛓️ | :chains: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🧰 | :toolbox: | 🧲 | :magnet: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Science

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | ⚗️ | :alembic: | 🧪 | :test\_tube: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🧫 | :petri\_dish: | 🧬 | :dna: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🔬 | :microscope: | 🔭 | :telescope: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 📡 | :satellite: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Medical

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 💉 | :syringe: | 🩸 | :drop\_of\_blood: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 💊 | :pill: | 🩹 | :adhesive\_bandage: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🩺 | :stethoscope: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Household

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🚪 | :door: | 🛏️ | :bed: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🛋️ | :couch\_and\_lamp: | 🪑 | :chair: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🚽 | :toilet: | 🚿 | :shower: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🛁 | :bathtub: | 🪒 | :razor: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🧴 | :lotion\_bottle: | 🧷 | :safety\_pin: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🧹 | :broom: | 🧺 | :basket: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🧻 | :roll\_of\_paper: | 🧼 | :soap: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🧽 | :sponge: | 🧯 | :fire\_extinguisher: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🛒 | :shopping\_cart: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Other Object

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | 🚬 | :smoking: | ⚰️ | :coffin: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#objects) | ⚱️ | :funeral\_urn: | 🗿 | :moyai: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

### Symbols

* [Transport Sign](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#transport-sign)
* [Warning](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#warning)
* [Arrow](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#arrow)
* [Religion](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#religion)
* [Zodiac](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#zodiac)
* [Av Symbol](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#av-symbol)
* [Gender](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#gender)
* [Math](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#math)
* [Punctuation](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#punctuation)
* [Currency](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#currency)
* [Other Symbol](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#other-symbol)
* [Keycap](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#keycap)
* [Alphanum](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#alphanum)
* [Geometric](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#geometric)

#### Transport Sign

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🏧 | :atm: | 🚮 | :put\_litter\_in\_its\_place: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🚰 | :potable\_water: | ♿ | :wheelchair: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🚹 | :mens: | 🚺 | :womens: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🚻 | :restroom: | 🚼 | :baby\_symbol: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🚾 | :wc: | 🛂 | :passport\_control: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🛃 | :customs: | 🛄 | :baggage\_claim: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🛅 | :left\_luggage: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Warning

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ⚠️ | :warning: | 🚸 | :children\_crossing: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ⛔ | :no\_entry: | 🚫 | :no\_entry\_sign: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🚳 | :no\_bicycles: | 🚭 | :no\_smoking: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🚯 | :do\_not\_litter: | 🚱 | :non-potable\_water: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🚷 | :no\_pedestrians: | 📵 | :no\_mobile\_phones: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🔞 | :underage: | ☢️ | :radioactive: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ☣️ | :biohazard: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Arrow

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ⬆️ | :arrow\_up: | ↗️ | :arrow\_upper\_right: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ➡️ | :arrow\_right: | ↘️ | :arrow\_lower\_right: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ⬇️ | :arrow\_down: | ↙️ | :arrow\_lower\_left: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ⬅️ | :arrow\_left: | ↖️ | :arrow\_upper\_left: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ↕️ | :arrow\_up\_down: | ↔️ | :left\_right\_arrow: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ↩️ | :leftwards\_arrow\_with\_hook: | ↪️ | :arrow\_right\_hook: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ⤴️ | :arrow\_heading\_up: | ⤵️ | :arrow\_heading\_down: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🔃 | :arrows\_clockwise: | 🔄 | :arrows\_counterclockwise: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🔙 | :back: | 🔚 | :end: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🔛 | :on: | 🔜 | :soon: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🔝 | :top: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Religion

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🛐 | :place\_of\_worship: | ⚛️ | :atom\_symbol: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🕉️ | :om: | ✡️ | :star\_of\_david: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ☸️ | :wheel\_of\_dharma: | ☯️ | :yin\_yang: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ✝️ | :latin\_cross: | ☦️ | :orthodox\_cross: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ☪️ | :star\_and\_crescent: | ☮️ | :peace\_symbol: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🕎 | :menorah: | 🔯 | :six\_pointed\_star: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Zodiac

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ♈ | :aries: | ♉ | :taurus: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ♊ | :gemini: | ♋ | :cancer: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ♌ | :leo: | ♍ | :virgo: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ♎ | :libra: | ♏ | :scorpius: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ♐ | :sagittarius: | ♑ | :capricorn: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ♒ | :aquarius: | ♓ | :pisces: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ⛎ | :ophiuchus: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Av Symbol

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🔀 | :twisted\_rightwards\_arrows: | 🔁 | :repeat: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🔂 | :repeat\_one: | ▶️ | :arrow\_forward: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ⏩ | :fast\_forward: | ⏭️ | :next\_track\_button: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ⏯️ | :play\_or\_pause\_button: | ◀️ | :arrow\_backward: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ⏪ | :rewind: | ⏮️ | :previous\_track\_button: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🔼 | :arrow\_up\_small: | ⏫ | :arrow\_double\_up: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🔽 | :arrow\_down\_small: | ⏬ | :arrow\_double\_down: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ⏸️ | :pause\_button: | ⏹️ | :stop\_button: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ⏺️ | :record\_button: | ⏏️ | :eject\_button: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🎦 | :cinema: | 🔅 | :low\_brightness: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🔆 | :high\_brightness: | 📶 | :signal\_strength: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 📳 | :vibration\_mode: | 📴 | :mobile\_phone\_off: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Gender

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ♀️ | :female\_sign: | ♂️ | :male\_sign: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Math

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ✖️ | :heavy\_multiplication\_x: | ➕ | :heavy\_plus\_sign: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ➖ | :heavy\_minus\_sign: | ➗ | :heavy\_division\_sign: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ♾️ | :infinity: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Punctuation

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ‼️ | :bangbang: | ⁉️ | :interrobang: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ❓ | :question: | ❔ | :grey\_question: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ❕ | :grey\_exclamation: | ❗ | :exclamation: :heavy\_exclamation\_mark: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 〰️ | :wavy\_dash: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Currency

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 💱 | :currency\_exchange: | 💲 | :heavy\_dollar\_sign: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Other Symbol

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ⚕️ | :medical\_symbol: | ♻️ | :recycle: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ⚜️ | :fleur\_de\_lis: | 🔱 | :trident: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 📛 | :name\_badge: | 🔰 | :beginner: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ⭕ | :o: | ✅ | :white\_check\_mark: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ☑️ | :ballot\_box\_with\_check: | ✔️ | :heavy\_check\_mark: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ❌ | :x: | ❎ | :negative\_squared\_cross\_mark: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ➰ | :curly\_loop: | ➿ | :loop: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 〽️ | :part\_alternation\_mark: | ✳️ | :eight\_spoked\_asterisk: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ✴️ | :eight\_pointed\_black\_star: | ❇️ | :sparkle: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ©️ | :copyright: | ®️ | :registered: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ™️ | :tm: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Keycap

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | #️⃣ | :hash: | \*️⃣ | :asterisk: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 0️⃣ | :zero: | 1️⃣ | :one: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 2️⃣ | :two: | 3️⃣ | :three: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 4️⃣ | :four: | 5️⃣ | :five: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 6️⃣ | :six: | 7️⃣ | :seven: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 8️⃣ | :eight: | 9️⃣ | :nine: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🔟 | :keycap\_ten: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Alphanum

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🔠 | :capital\_abcd: | 🔡 | :abcd: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🔢 | :1234: | 🔣 | :symbols: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🔤 | :abc: | 🅰️ | :a: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🆎 | :ab: | 🅱️ | :b: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🆑 | :cl: | 🆒 | :cool: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🆓 | :free: | ℹ️ | :information\_source: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🆔 | :id: | Ⓜ️ | :m: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🆕 | :new: | 🆖 | :ng: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🅾️ | :o2: | 🆗 | :ok: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🅿️ | :parking: | 🆘 | :sos: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🆙 | :up: | 🆚 | :vs: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🈁 | :koko: | 🈂️ | :sa: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🈷️ | :u6708: | 🈶 | :u6709: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🈯 | :u6307: | 🉐 | :ideograph\_advantage: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🈹 | :u5272: | 🈚 | :u7121: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🈲 | :u7981: | 🉑 | :accept: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🈸 | :u7533: | 🈴 | :u5408: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🈳 | :u7a7a: | ㊗️ | :congratulations: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ㊙️ | :secret: | 🈺 | :u55b6: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🈵 | :u6e80: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Geometric

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🔴 | :red\_circle: | 🟠 | :orange\_circle: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🟡 | :yellow\_circle: | 🟢 | :green\_circle: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🔵 | :large\_blue\_circle: | 🟣 | :purple\_circle: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🟤 | :brown\_circle: | ⚫ | :black\_circle: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ⚪ | :white\_circle: | 🟥 | :red\_square: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🟧 | :orange\_square: | 🟨 | :yellow\_square: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🟩 | :green\_square: | 🟦 | :blue\_square: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🟪 | :purple\_square: | 🟫 | :brown\_square: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ⬛ | :black\_large\_square: | ⬜ | :white\_large\_square: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ◼️ | :black\_medium\_square: | ◻️ | :white\_medium\_square: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ◾ | :black\_medium\_small\_square: | ◽ | :white\_medium\_small\_square: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | ▪️ | :black\_small\_square: | ▫️ | :white\_small\_square: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🔶 | :large\_orange\_diamond: | 🔷 | :large\_blue\_diamond: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🔸 | :small\_orange\_diamond: | 🔹 | :small\_blue\_diamond: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🔺 | :small\_red\_triangle: | 🔻 | :small\_red\_triangle\_down: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 💠 | :diamond\_shape\_with\_a\_dot\_inside: | 🔘 | :radio\_button: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#symbols) | 🔳 | :white\_square\_button: | 🔲 | :black\_square\_button: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

### Flags

* [Flag](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flag)
* [Country Flag](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#country-flag)
* [Subdivision Flag](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#subdivision-flag)

#### Flag

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🏁 | :checkered\_flag: | 🚩 | :triangular\_flag\_on\_post: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🎌 | :crossed\_flags: | 🏴 | :black\_flag: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🏳️ | :white\_flag: | 🏳️‍🌈 | :rainbow\_flag: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🏴‍☠️ | :pirate\_flag: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Country Flag

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇦🇨 | :ascension\_island: | 🇦🇩 | :andorra: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇦🇪 | :united\_arab\_emirates: | 🇦🇫 | :afghanistan: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇦🇬 | :antigua\_barbuda: | 🇦🇮 | :anguilla: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇦🇱 | :albania: | 🇦🇲 | :armenia: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇦🇴 | :angola: | 🇦🇶 | :antarctica: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇦🇷 | :argentina: | 🇦🇸 | :american\_samoa: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇦🇹 | :austria: | 🇦🇺 | :australia: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇦🇼 | :aruba: | 🇦🇽 | :aland\_islands: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇦🇿 | :azerbaijan: | 🇧🇦 | :bosnia\_herzegovina: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇧🇧 | :barbados: | 🇧🇩 | :bangladesh: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇧🇪 | :belgium: | 🇧🇫 | :burkina\_faso: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇧🇬 | :bulgaria: | 🇧🇭 | :bahrain: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇧🇮 | :burundi: | 🇧🇯 | :benin: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇧🇱 | :st\_barthelemy: | 🇧🇲 | :bermuda: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇧🇳 | :brunei: | 🇧🇴 | :bolivia: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇧🇶 | :caribbean\_netherlands: | 🇧🇷 | :brazil: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇧🇸 | :bahamas: | 🇧🇹 | :bhutan: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇧🇻 | :bouvet\_island: | 🇧🇼 | :botswana: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇧🇾 | :belarus: | 🇧🇿 | :belize: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇨🇦 | :canada: | 🇨🇨 | :cocos\_islands: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇨🇩 | :congo\_kinshasa: | 🇨🇫 | :central\_african\_republic: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇨🇬 | :congo\_brazzaville: | 🇨🇭 | :switzerland: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇨🇮 | :cote\_divoire: | 🇨🇰 | :cook\_islands: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇨🇱 | :chile: | 🇨🇲 | :cameroon: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇨🇳 | :cn: | 🇨🇴 | :colombia: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇨🇵 | :clipperton\_island: | 🇨🇷 | :costa\_rica: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇨🇺 | :cuba: | 🇨🇻 | :cape\_verde: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇨🇼 | :curacao: | 🇨🇽 | :christmas\_island: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇨🇾 | :cyprus: | 🇨🇿 | :czech\_republic: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇩🇪 | :de: | 🇩🇬 | :diego\_garcia: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇩🇯 | :djibouti: | 🇩🇰 | :denmark: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇩🇲 | :dominica: | 🇩🇴 | :dominican\_republic: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇩🇿 | :algeria: | 🇪🇦 | :ceuta\_melilla: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇪🇨 | :ecuador: | 🇪🇪 | :estonia: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇪🇬 | :egypt: | 🇪🇭 | :western\_sahara: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇪🇷 | :eritrea: | 🇪🇸 | :es: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇪🇹 | :ethiopia: | 🇪🇺 | :eu: :european\_union: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇫🇮 | :finland: | 🇫🇯 | :fiji: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇫🇰 | :falkland\_islands: | 🇫🇲 | :micronesia: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇫🇴 | :faroe\_islands: | 🇫🇷 | :fr: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇬🇦 | :gabon: | 🇬🇧 | :gb: :uk: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇬🇩 | :grenada: | 🇬🇪 | :georgia: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇬🇫 | :french\_guiana: | 🇬🇬 | :guernsey: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇬🇭 | :ghana: | 🇬🇮 | :gibraltar: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇬🇱 | :greenland: | 🇬🇲 | :gambia: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇬🇳 | :guinea: | 🇬🇵 | :guadeloupe: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇬🇶 | :equatorial\_guinea: | 🇬🇷 | :greece: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇬🇸 | :south\_georgia\_south\_sandwich\_islands: | 🇬🇹 | :guatemala: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇬🇺 | :guam: | 🇬🇼 | :guinea\_bissau: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇬🇾 | :guyana: | 🇭🇰 | :hong\_kong: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇭🇲 | :heard\_mcdonald\_islands: | 🇭🇳 | :honduras: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇭🇷 | :croatia: | 🇭🇹 | :haiti: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇭🇺 | :hungary: | 🇮🇨 | :canary\_islands: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇮🇩 | :indonesia: | 🇮🇪 | :ireland: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇮🇱 | :israel: | 🇮🇲 | :isle\_of\_man: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇮🇳 | :india: | 🇮🇴 | :british\_indian\_ocean\_territory: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇮🇶 | :iraq: | 🇮🇷 | :iran: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇮🇸 | :iceland: | 🇮🇹 | :it: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇯🇪 | :jersey: | 🇯🇲 | :jamaica: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇯🇴 | :jordan: | 🇯🇵 | :jp: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇰🇪 | :kenya: | 🇰🇬 | :kyrgyzstan: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇰🇭 | :cambodia: | 🇰🇮 | :kiribati: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇰🇲 | :comoros: | 🇰🇳 | :st\_kitts\_nevis: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇰🇵 | :north\_korea: | 🇰🇷 | :kr: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇰🇼 | :kuwait: | 🇰🇾 | :cayman\_islands: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇰🇿 | :kazakhstan: | 🇱🇦 | :laos: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇱🇧 | :lebanon: | 🇱🇨 | :st\_lucia: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇱🇮 | :liechtenstein: | 🇱🇰 | :sri\_lanka: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇱🇷 | :liberia: | 🇱🇸 | :lesotho: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇱🇹 | :lithuania: | 🇱🇺 | :luxembourg: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇱🇻 | :latvia: | 🇱🇾 | :libya: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇲🇦 | :morocco: | 🇲🇨 | :monaco: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇲🇩 | :moldova: | 🇲🇪 | :montenegro: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇲🇫 | :st\_martin: | 🇲🇬 | :madagascar: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇲🇭 | :marshall\_islands: | 🇲🇰 | :macedonia: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇲🇱 | :mali: | 🇲🇲 | :myanmar: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇲🇳 | :mongolia: | 🇲🇴 | :macau: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇲🇵 | :northern\_mariana\_islands: | 🇲🇶 | :martinique: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇲🇷 | :mauritania: | 🇲🇸 | :montserrat: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇲🇹 | :malta: | 🇲🇺 | :mauritius: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇲🇻 | :maldives: | 🇲🇼 | :malawi: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇲🇽 | :mexico: | 🇲🇾 | :malaysia: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇲🇿 | :mozambique: | 🇳🇦 | :namibia: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇳🇨 | :new\_caledonia: | 🇳🇪 | :niger: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇳🇫 | :norfolk\_island: | 🇳🇬 | :nigeria: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇳🇮 | :nicaragua: | 🇳🇱 | :netherlands: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇳🇴 | :norway: | 🇳🇵 | :nepal: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇳🇷 | :nauru: | 🇳🇺 | :niue: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇳🇿 | :new\_zealand: | 🇴🇲 | :oman: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇵🇦 | :panama: | 🇵🇪 | :peru: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇵🇫 | :french\_polynesia: | 🇵🇬 | :papua\_new\_guinea: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇵🇭 | :philippines: | 🇵🇰 | :pakistan: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇵🇱 | :poland: | 🇵🇲 | :st\_pierre\_miquelon: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇵🇳 | :pitcairn\_islands: | 🇵🇷 | :puerto\_rico: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇵🇸 | :palestinian\_territories: | 🇵🇹 | :portugal: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇵🇼 | :palau: | 🇵🇾 | :paraguay: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇶🇦 | :qatar: | 🇷🇪 | :reunion: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇷🇴 | :romania: | 🇷🇸 | :serbia: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇷🇺 | :ru: | 🇷🇼 | :rwanda: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇸🇦 | :saudi\_arabia: | 🇸🇧 | :solomon\_islands: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇸🇨 | :seychelles: | 🇸🇩 | :sudan: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇸🇪 | :sweden: | 🇸🇬 | :singapore: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇸🇭 | :st\_helena: | 🇸🇮 | :slovenia: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇸🇯 | :svalbard\_jan\_mayen: | 🇸🇰 | :slovakia: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇸🇱 | :sierra\_leone: | 🇸🇲 | :san\_marino: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇸🇳 | :senegal: | 🇸🇴 | :somalia: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇸🇷 | :suriname: | 🇸🇸 | :south\_sudan: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇸🇹 | :sao\_tome\_principe: | 🇸🇻 | :el\_salvador: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇸🇽 | :sint\_maarten: | 🇸🇾 | :syria: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇸🇿 | :swaziland: | 🇹🇦 | :tristan\_da\_cunha: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇹🇨 | :turks\_caicos\_islands: | 🇹🇩 | :chad: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇹🇫 | :french\_southern\_territories: | 🇹🇬 | :togo: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇹🇭 | :thailand: | 🇹🇯 | :tajikistan: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇹🇰 | :tokelau: | 🇹🇱 | :timor\_leste: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇹🇲 | :turkmenistan: | 🇹🇳 | :tunisia: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇹🇴 | :tonga: | 🇹🇷 | :tr: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇹🇹 | :trinidad\_tobago: | 🇹🇻 | :tuvalu: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇹🇼 | :taiwan: | 🇹🇿 | :tanzania: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇺🇦 | :ukraine: | 🇺🇬 | :uganda: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇺🇲 | :us\_outlying\_islands: | 🇺🇳 | :united\_nations: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇺🇸 | :us: | 🇺🇾 | :uruguay: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇺🇿 | :uzbekistan: | 🇻🇦 | :vatican\_city: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇻🇨 | :st\_vincent\_grenadines: | 🇻🇪 | :venezuela: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇻🇬 | :british\_virgin\_islands: | 🇻🇮 | :us\_virgin\_islands: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇻🇳 | :vietnam: | 🇻🇺 | :vanuatu: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇼🇫 | :wallis\_futuna: | 🇼🇸 | :samoa: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇽🇰 | :kosovo: | 🇾🇪 | :yemen: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇾🇹 | :mayotte: | 🇿🇦 | :south\_africa: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🇿🇲 | :zambia: | 🇿🇼 | :zimbabwe: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

#### Subdivision Flag

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🏴󠁧󠁢󠁥󠁮󠁧󠁿 | :england: | 🏴󠁧󠁢󠁳󠁣󠁴󠁿 | :scotland: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#flags) | 🏴󠁧󠁢󠁷󠁬󠁳󠁿 | :wales: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

### GitHub Custom Emoji

|  | **ico** | **shortcode** | **ico** | **shortcode** |  |
| --- | --- | --- | --- | --- | --- |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#github-custom-emoji) | :atom: | :atom: | :basecamp: | :basecamp: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#github-custom-emoji) | :basecampy: | :basecampy: | :bowtie: | :bowtie: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#github-custom-emoji) | :electron: | :electron: | :feelsgood: | :feelsgood: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#github-custom-emoji) | :finnadie: | :finnadie: | :goberserk: | :goberserk: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#github-custom-emoji) | :godmode: | :godmode: | :hurtrealbad: | :hurtrealbad: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#github-custom-emoji) | :neckbeard: | :neckbeard: | :octocat: | :octocat: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#github-custom-emoji) | :rage1: | :rage1: | :rage2: | :rage2: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#github-custom-emoji) | :rage3: | :rage3: | :rage4: | :rage4: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#github-custom-emoji) | :shipit: | :shipit: | :suspect: | :suspect: | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |
| [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#github-custom-emoji) | :trollface: | :trollface: |  |  | [top](https://github.com/ikatyang/emoji-cheat-sheet/blob/master/README.md#table-of-contents) |

## 仓库Star的美好方式

参数

| **参数** | **必传** | **类型** | **备注** |
| --- | --- | --- | --- |
| user | true | String | 用户名 |
| repo | true | String | 仓库名字 |
| type | true | String | star |
| count | true | String | 数量 |

<iframe style="margin-left: 2px; margin-bottom:-5px;" frameborder="0" scrolling="0" width="100px" height="20px" src="[https://ghbtns.com/github-btn.html?user=fandongtongxue&repo=fandongtongxue.github.io&type=star&count=true](https://link.jianshu.com?t=https:/ghbtns.com/github-btn.html?user=fandongtongxue&repo=fandongtongxue.github.io&type=star&count=true)" >

</iframe>

# Github Desktop

## GitHub Pages

## GitHub资源

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

[github pages](http://pages.github.com/)

[git](http://git-scm.com/)

https://docs.github.com/cn/github/working-with-github-pages/getting-started-with-github-pages

# Python

## 安装

请点击下载以下内容，本文会一步一步介绍如何安装，不着急，慢慢来。

1. 安装python，这个就不用多说了吧~
2. 安装git，简单。git bash使用教程请参见[官方文档](http://www.git-scm.com/book/zh)~
3. 将python安装文件夹中scripts和make.exe加入环境变量。
4. 安装pip，具体可以看[官网介绍](http://www.pip-installer.org/en/latest/installing.html)，或者[Windows环境在python3装pip](http://tech.crandom.com/2011/10/21/python3_pip.html)。
5. 安装pelican和markdown，至于为什么使用markdown语法，这就看个人喜好呢，下载的markdownpad就是window下markdown的编辑器呢，非常简单，linux等其它操作系统编辑器请自行google。你也可以选用REST语法，具体请参见[markdown教程](http://daringfireball.net/projects/markdown/syntax)和[REST教程](https://beinggeekbook.readthedocs.org/en/latest/rst.html)。
6. pip install pelican
7. pip install markdown

date

网上搜到的pip国内镜像大部分是豆瓣的

http://pypi.douban.com/simple/

但是根本不全，很多包没有

所以推荐清华大学的

https://pypi.tuna.tsinghua.edu.cn/simple

临时使用

可以在使用pip的时候加参数-i https://pypi.tuna.tsinghua.edu.cn/simple

例如：pip install -i https://pypi.tuna.tsinghua.edu.cn/simple gevent，这样就会从清华这边的镜像去安装gevent库。

永久修改

Linux下，修改 ~/.pip/pip.conf (没有就创建一个)， 修改 index-url至tuna，内容如下：

[global]

index-url = https://pypi.tuna.tsinghua.edu.cn/simple

1

2

windows下，直接在user目录中创建一个pip目录，如：C:\Users\xx\pip，新建文件pip.ini，内容如下

[global]

index-url = https://pypi.tuna.tsinghua.edu.cn/simple

## 解决pip安装时速度慢的问题

新版ubuntu要求使用https源，要注意。

清华：<https://pypi.tuna.tsinghua.edu.cn/simple>

阿里云：<http://mirrors.aliyun.com/pypi/simple/>

中国科技大学 <https://pypi.mirrors.ustc.edu.cn/simple/>

华中理工大学：<http://pypi.hustunique.com/>

山东理工大学：<http://pypi.sdutlinux.org/>

豆瓣：<http://pypi.douban.com/simple/>

## 临时使用：

可以在使用pip的时候加参数-i <https://pypi.tuna.tsinghua.edu.cn/simple>

例如：pip install -i <https://pypi.tuna.tsinghua.edu.cn/simple> pyspider，这样就会从清华这边的镜像去安装pyspider库。

## 永久修改，一劳永逸：

Linux下，修改 ~/.pip/pip.conf (没有就创建一个文件夹及文件。文件夹要加“.”，表示是隐藏文件夹)

内容如下：

[global]   
index-url = <https://pypi.tuna.tsinghua.edu.cn/simple>   
[install]   
trusted-host=mirrors.aliyun.com   
windows下，直接在user目录中创建一个pip目录，如：C:\Users\xx\pip，新建文件pip.ini。内容同上。

# Ruby

# Markdown

## Markdown简介

Markdown是一种纯文本格式的标记语言。通过简单的标记语法，它可以使普通文本内容具有一定的格式。

**优点：**  
1、因为是纯文本，所以只要支持Markdown的地方都能获得一样的编辑效果，可以让作者摆脱排版的困扰，专心写作。

2、操作简单。比如:WYSIWYG编辑时标记个标题，先选中内容，再点击导航栏的标题按钮，选择几级标题。要三个步骤。而Markdown只需要在标题内容前加#即可

**缺点：**

1、需要记一些语法（当然，是很简单。五分钟学会）。  
2、有些平台不支持Markdown编辑模式。

## 注释

[comment]: <> (This is a comment, it will not be included)

[comment]: <> (in the output file unless you use it in)

[comment]: <> (a reference style link.)

[//]: <> (This is also a comment.)

[//]: # (This may be the most platform independent comment)

[^\_^]:

## 排版

### 分割线

三个或者三个以上的 - 或者 \* 都可以。

示例：

---

----

\*\*\*

\*\*\*\*\*

效果如下：  
可以看到，显示效果是一样的。

### 标题

在想要设置为标题的文字前面加#来表示  
一个#是一级标题，二个#是二级标题，以此类推。支持六级标题。

注：标准语法一般在#后跟个空格再写文字，貌似简书不加空格也行。

示例：

# 这是一级标题

## 这是二级标题

### 这是三级标题

#### 这是四级标题

##### 这是五级标题

###### 这是六级标题

效果如下：

**这是一级标题**

**这是二级标题**

**这是三级标题**

**这是四级标题**

**这是五级标题**

**这是六级标题**

### 缩进

两种方案

* 手动输入空格 （&nbsp；）。注意！此时的分号为英文分号，但是不推荐使用此方法，太麻烦！
* 使用全角空格(切换快捷键shift+空格)。即：在全角输入状态下直接使用空格键就ok了

### 居中

<center>

居中渲染

</center>

**注意： 内部文本换行时要用<br>,两个回车无效**

### 左对齐

<p align="left">月是故乡明</p>

### 右对齐

<p align="right">月是故乡明</p>

### 换行

两种方案

* 两个回车即可
* 使用< br >

### 列表

#### 无序列表

语法：  
无序列表用 - + \* 任何一种都可以

- 列表内容

+ 列表内容

\* 列表内容

注意：- + \* 跟内容之间都要有一个空格

效果如下：

* 列表内容
* 列表内容
* 列表内容

#### 多级无序列表

* 编程语言
  + 脚本语言
    - Python

#### 有序列表

语法：

数字加点

1. 列表内容

2. 列表内容

3. 列表内容

注意：序号跟内容之间要有空格

效果如下：

1. 列表内容
2. 列表内容
3. 列表内容

#### 列表嵌套

**上一级和下一级之间敲三个空格即可**

* 一级无序列表内容
  + 二级无序列表内容
  + 二级无序列表内容
  + 二级无序列表内容
* 一级无序列表内容
  + 二级有序列表内容
  + 二级有序列表内容
  + 二级有序列表内容

1. 一级有序列表内容
   * 二级无序列表内容
   * 二级无序列表内容
   * 二级无序列表内容
2. 一级有序列表内容
   * 二级有序列表内容
   * 二级有序列表内容
   * 二级有序列表内容

#### 复选框列表

* [x] 需求分析
* [x] 系统设计
* [x] 详细设计
* [ ] 编码
* [ ] 测试
* [ ] 交付

您可以使用这个功能来标注某个项目各项任务的完成情况。

Tip:

在GitHub的**issue**中使用该语法是可以实时点击复选框来勾选或解除勾选的，而无需修改issue原文。

## 字体

### 字体类型

< font face=“微软雅黑”>微软雅黑

### 大小

< font size=5> Hello

### 颜色

<font color=red>color </font>

### 加粗

要加粗的文字左右分别用两个\*号包起来

\*\* 内容 \*\* 　(\*与内容之间没有空格)

### 斜体

要倾斜的文字左右分别用一个\*号包起来

\*内容 \* 　　 (\*与内容之间没有空格)

### 斜体加粗

要倾斜和加粗的文字左右分别用三个\*号包起来

### 删除线

要加删除线的文字左右分别用两个~~号包起来

示例：

\*\*这是加粗的文字\*\*

\*这是倾斜的文字\*`

\*\*\*这是斜体加粗的文字\*\*\*

~~这是加删除线的文字~~

效果如下：

**这是加粗的文字**  
*这是倾斜的文字*  
***这是斜体加粗的文字***

### 斜体、粗体、删除线

| **语法** | **效果** |
| --- | --- |
| \*斜体1\* | *斜体1* |
| \_斜体2\_ | *斜体2* |
| \*\*粗体1\*\* | **粗体1** |
| \_\_粗体2\_\_ | **粗体2** |
| 这是一个 ~~删除线~~ | 这是一个 |
| \*\*\*斜粗体1\*\*\* | ***斜粗体1*** |
| \_\_\_斜粗体2\_\_\_ | ***斜粗体2*** |
| \*\*\*~~斜粗体删除线1~~\*\*\* |  |
| ~~\*\*\*斜粗体删除线2\*\*\*~~ |  |

作者：LucasLight  
链接：https://www.jianshu.com/p/40ba812dd973  
来源：简书  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

## 文字高亮

文字高亮功能能使行内部分文字高亮，使用一对反引号。 一般我们会在一些关键位置加亮提示用户，也可以做网站的tag标签  
语法：

`linux` `网络编程` `socket` `epoll`

效果：linux 网络编程 socket epoll

## 超链接

### 链接外部URL

语法：

[超链接名](超链接地址 "超链接title")

title可加可不加

示例：

[简书](http://jianshu.com)

[百度](http://baidu.com)

效果如下：

[简书](https://www.jianshu.com/u/1f5ac0cf6a8b)  
[百度](https://links.jianshu.com/go?to=http%3A%2F%2Fbaidu.com)

注：Markdown本身语法不支持链接在新页面中打开，貌似简书做了处理，是可以的。别的平台可能就不行了，如果想要在新页面中打开的话可以用html语言的a标签代替。

<a href="超链接地址" title="这是超bai链接显示的信息框" target="\_blank">超链接名</a>

示例

<a href="https://www.jianshu.com/u/1f5ac0cf6a8b" target="\_blank">简书</a>

| **#** | **语法** | **效果** |
| --- | --- | --- |
| 1 | [我的博客](http://blog.csdn.net/guodongxiaren "悬停显示") | [我的博客](http://blog.csdn.net/guodongxiaren) |
| 2 | [我的知乎][zhihu] | [我的知乎](https://www.zhihu.com/people/jellywong) |

语法2由两部分组成：

* 第一部分使用两个中括号，[ ]里的标识符（本例中zhihu），可以是数字，字母等的组合，标识符上下对应就行了
* 第二部分标记实际URL。

使用URL标识符能达到复用的目的，一般把全文所有的URL标识符统一放在文章末尾，这样看起来比较干净。

### 链接本仓库里的URL

| **语法** | **效果** |
| --- | --- |
| [我的简介](/example/profile.md) | [我的简介](/example/profile.md) |
| [Book](./Book) | [Book](/Book) |

### 图片链接

给图片加链接的本质是混合图片显示语法和普通的链接语法。普通的链接中[ ]内部是链接要显示的文本，而图片链接[ ]里面则是要显示的图片。  
直接混合两种语法当然可以，但是十分啰嗦，为此我们可以使用URL标识符的形式。

| **#** | **语法** | **效果** |
| --- | --- | --- |
| 1 | [[图片上传失败...(image-ea9c0f-1531809400967)] | [[图片上传失败...(image-562f47-1531809400967)] |
| 2 | [[图片上传失败...(image-2b77ee-1531809400967)]][zhihu] | [[图片上传失败...(image-d8a872-1531809400967)]](https://www.zhihu.com/people/jellywong) |
| 3 | [![csdn-logo]][csdn] | [![csdn-logo]](http://blog.csdn.net/guodongxiaren) |

因为图片本身和链接本身都支持URL标识符的形式，所以图片链接也可以很简洁（见例3）。  
注意，此时鼠标悬停时显示的文字是图片的title，而非链接本身的title了。

本文URL标识符都放置于文末

### 页内锚点

其实呢，每一个标题都是一个锚点，和HTML的锚点（#）类似，比如我们

| **语法** | **效果** |
| --- | --- |
| [回到顶部](#readme) | [回到顶部](#readme) |

不过要注意，标题中的英文字母都被转化为**小写字母**了。

以前GitHub对中文支持的不好，所以中文标题不能正确识别为锚点，但是现在已经没问题啦！

## 引用

在引用的文字前加>即可。引用也可以嵌套，如加两个>>三个>>>  
n个...  
貌似可以一直加下去，但没神马卵用

示例：

>这是引用的内容

>>这是引用的内容

>>>>>>>>>>这是引用的内容

效果如下：

这是引用的内容

这是引用的内容

这是引用的内容

## 表格

### 定义表格

语法：

表头|表头|表头

---|:--:|---:

内容|内容|内容

内容|内容|内容

第二行分割表头和内容。

- 有一个就行，为了对齐，多加了几个

文字默认居左

-两边加：表示文字居中

-右边加：表示文字居右

注：原生的语法两边都要用 | 包起来。此处省略

示例：

姓名|技能|排行

--|:--:|--:

刘备|哭|大哥

关羽|打|二哥

张飞|骂|三弟

效果如下：

| **姓名** | **技能** | **排行** |
| --- | --- | --- |
| 刘备 | 哭 | 大哥 |
| 关羽 | 打 | 二哥 |
| 张飞 | 骂 | 三弟 |
|  |  |  |

### 对齐

表格可以指定对齐方式

| **左对齐** | **居中** | **右对齐** |
| --- | --- | --- |
| col 3 is | some wordy text | $1600 |
| col 2 is | centered | $12 |
| zebra stripes | are neat | $1 |

## 代码

语法：  
单行代码：代码之间分别用一个反引号包起来

`代码内容`

代码块：代码之间分别用三个反引号包起来，且两边的反引号单独占一行

(```)

代码...

代码...

代码...

(```)

注：为了防止转译，前后三个反引号处加了小括号，实际是没有的。这里只是用来演示，实际中去掉两边小括号即可。

示例：

单行代码

`create database hero;`

代码块

(```)

function fun(){

echo "这是一句非常牛逼的代码";

}

fun();

(```)

效果如下：

单行代码

create database hero;

代码块

function fun(){

echo "这是一句非常牛逼的代码";

}

fun();

### 代码高亮

在三个反引号后面加上编程语言的名字，另起一行开始写代码，最后一行再加上三个反引号。

public static void main(String[]args){} //Java

int main(int argc, char \*argv[]) //C

echo "hello GitHub" #Bash

document.getElementById("myH1").innerHTML="Welcome to my Homepage"; //javascipt

string &operator+(const string& A,const string& B) //cpp

## 流程图

```flow

st=>start: 开始

op=>operation: My Operation

cond=>condition: Yes or No?

e=>end

st->op->cond

cond(yes)->e

cond(no)->op

&```

效果如下：  
简书不支持流程图，所以截了个图

流程图.png

## 多媒体

### 图片

语法：

![图片alt](图片地址 "图片title")

图片alt就是显示在图片下面的文字，相当于对图片内容的解释。

图片title是图片的标题，当鼠标移到图片上时显示的内容。title可加可不加

示例：

![blockchain](https://ss0.bdstatic.com/70cFvHSh\_Q1YnxGkpoWK1HF6hhy/it/

u=702257389,1274025419&fm=27&gp=0.jpg "区块链")

如果引用本仓库中的图片，直接使用**相对路径**就可了，如果引用其他github仓库中的图片要注意格式，即：仓库地址/raw/分支名/图片路径，如：

<https://github.com/guodongxiaren/ImageCache/raw/master/Logo/foryou.gif>

| **#** | **语法** | **效果** |
| --- | --- | --- |
| 1 | ![baidu](http://upload-images.jianshu.io/upload\_images/6153330-305abf60f5e71ed5.gif?imageMogr2/auto-orient/strip "百度logo") | baidu |
| 2 | ![][foryou] |  |

注意例2的写法使用了**URL标识符**的形式，在[链接](#%E9%93%BE%E6%8E%A5)一节有介绍。

在文末有foryou的定义：

[foryou]:https://github.com/guodongxiaren/ImageCache/raw/master/Logo/foryou.gif

### 音频

#### 语法格式

<audio id="audio" controls="" preload="none">

<source id="mp3" src="http://oht4nlntk.bkt.clouddn.com/Music\_iP%E8%B5%B5%E9%9C%B2%20-%20%E7%A6%BB%E6%AD%8C%20%28Live%29.mp3">

</audio>

#### 音频库 Github

相对路径

<audio id="audio" controls="" preload="none">

<source id="mp3" src="./listening/n2-1991.mp3">

</audio>

绝对路径

<audio id="audio" controls="" preload="none">

<source id="mp3" src="https://sb-talents.github.io/The-Practice-of-Japanese/listening/n2-1991.mp3">

</audio>

备注

Bilibili失败

### 视频

#### 语法格式

<video id="video" controls="" preload="none" poster="http://om2bks7xs.bkt.clouddn.com/2017-08-26-Markdown-Advance-Video.jpg">  
<source id="mp4" src="http://om2bks7xs.bkt.clouddn.com/2017-08-26-Markdown-Advance-Video.mp4" type="video/mp4">  
</video>

视频库 哔哩哔哩

<iframe src="//player.bilibili.com/player.html?aid=67086362&bvid=BV1C4411C7sf&cid=116329440&page=1" scrolling="no" border="0" frameborder="no" framespacing="0" allowfullscreen="true"> </iframe>

<iframe src="https://www.bilibili.com/html/html5player.html?aid=11200262&cid=18527724&as\_wide=1" frameborder="0" width="640" height="430" allowfullscreen></iframe>

作者：qddaia  
链接：https://www.jianshu.com/p/457c11fea5e7  
来源：简书  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

### 混合其他语法

表格单元中的内容可以和其他大多数GFM语法配合使用，如：

#### 使用普通文本的删除线，斜体等效果

| **名字** | **描述** |
| --- | --- |
| Help | help window. |
| Close | Closes a window |

#### 表格中嵌入图片（链接）

其实前面介绍图片显示、图片链接的时候为了清晰就是放在在表格中显示的。

| **图片** | **描述** |
| --- | --- |
| baidu  baidu | 百度 |

#### 表情

Github的Markdown语法支持添加emoji表情，输入不同的符号码（两个冒号包围的字符）可以显示出不同的表情。

比如:blush:，可以显示:blush:。

具体每一个表情的符号码，可以查询GitHub的官方网页<http://www.emoji-cheat-sheet.com>。

但是这个网页每次都打开**奇慢**。。所以我整理到了本repo中，大家可以直接在此查看[emoji](emoji.md)。

#### diff语法

版本控制的系统中都少不了diff的功能，即展示一个文件内容的增加与删除。  
GFM中可以显示的展示diff效果。使用绿色表示新增，红色表示删除。

其语法与代码高亮类似，只是在三个反引号后面写diff，  
并且其内容中，以 +开头表示新增，-开头表示删除。

效果如下：

+ 鸟宿池边树，僧敲月下门

- 鸟宿池边树，僧推月下门

# Static site generator

## 概述

静态网站和动态网站相比有如下好处：

* 省钱。静态网站占用的系统资源少。如果挂到github pages上，只要注册一个域名就可以了。
* 速度快。不经过php解析器，不用数据库，速度自然比动态网站快
* 安全。由于静态网站的简洁，免疫很多web攻击方式。
* 服务器端配置简单。只需要一个web server（apache、nginx）。
* 非常容易维护。

静态网站的缺点是功能弱，和用户的交互能力不强

静态网站生成工具能从简单的纯文本文件生成一个网站/博客。常用文本格式有reStructuredText和Markdown。

      如今有大量的服务器端语言，数据库管理系统和内容管理系统，那么为什么许多网站会选用静态网站？

原因有很多，比如：

1.HTML 文件；

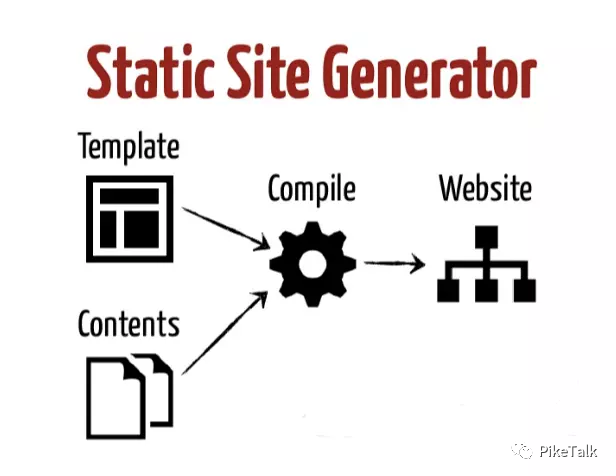
2.没有服务端处理或者数据库交互；

3.比动态网站更安全；

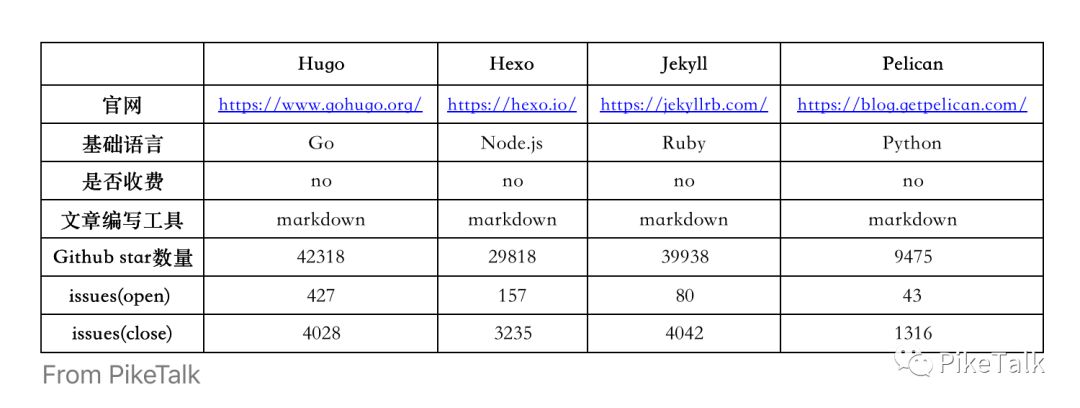
4.利于使用 CDN 进行扩展；

5.缓存会带来比动态网页更高的效率；

6.请求超快速。



本次我们将介绍四种比较流行的静态网站工具：Hugo、Hexo、Jekyll和Pelican，了解它们的优缺点，并根据不同的标准对它们进行比较。



**Hugo**

        Hugo是基于Go的静态网站生成器。被称为“世界上构建网站最快的框架”。普及速度很快。可以在几分钟内安装好Hugo，并在可以以秒为单位的时间内创建一个静态网站。

优势：

1.免费开源；

2.速度非常快，对构建速度做了优化；

3.动态 API 请求的内容；

4.无限制内容类型；

5.预制的Go 模版和模式；

6.对i18n有完整内置支持；

7.支持动态的API；

8.功能强大的内容模型。

劣势：

1.主题使用 Go 模版，所以需要熟悉 Go；

2.没有内置默认主题；

3.缺少扩展性和插件（因为 Go 是编译型语言）。

**Hexo**

        Hexo是基于NodeJs的开源静态生成器，MIT开源协议。借助Node.js平台，以效率和性能闻名，可在几秒钟内生成数百个静态文件，但是没有Hugo快。

优势：

1.相当快

2.在 GitHub Pages 部署简单；

3.强大的Markdown支持

4.高可扩展性

5.免费且开源的主题

6.有很多免费的插件

7.中文支持

8.中文社区

劣势（也是优势）：

没有英文

**Jekyll**

        Jekyll用Ruby语言，由GitHub构建和支持的。



优势：

1.免费且开源；

2.RubyGems 支持构建主题为 gems 方便分发；

3.简单便捷使用；

4.强大的 GitHub Pages 支持；

5.开箱即用的合适的默认极简主题。

劣势：

1.网站内容不断增加后，构建速度会明显变慢

2.自Jekyll 3起没有内置的分页功能

3.它不支持在标题或YAML中使用变量

4.许多插件已过时

5.gem依赖项可能会导致不兼容

6.Github的页面支持杰奇开箱即用，但只有一套Github的安全插件，可以使用

7.没有对livereload的内置支持

**Pelican**

        Pelican是一款基于Python编写的纯静态博客生成器，支持使用restructuredText和Markdown格式编写的文章，可以说是一款比较成熟的静态博客系统。

优势：

1.免费且开源；

2.100% Python

3.是目前最活躍的Python Blog 之一

4.快，安全。

当然，类似的工具还有很多。比如：

**JavaScript：**Next.js & Gatsby（适用于 React）、Nuxt & VuePress（适用于 Vue）、GitBook、Metalsmith、Harp、Spike.

**Python：**Pelican、MkDocs、Cactus.

**Ruby：**Middleman、Nanoc、Octopress.

**Go：**InkPaper.

**.NET：**Wyam、pretzel。

## WEB前端开发工具

目前vue，angular，react这三个前端框架非常流行，但是在很多场景下，我们在选择技术路线的时候总是很纠结，不知道该选择哪一种，这个问题的是本质是对框架的优劣认识不清晰。在这里不详细对比技术细节，因为技术细节差异不是我们选择框架的首要因素。我们该怎样选择一个框架，我觉得应该从以下几个角度：

**1.这个框架的使用场景。**比如是否同时适用于Web端和原生App或者快速搭建一个小型项目等等。

**2.团队当下的技术能力。**学习新框架的时间成本，后期团队维护的成本。

**3.框架能解决哪些问题。**优劣势是什么。

**4.框架的生态系统。**是否有繁荣的生态系统供我们学习使用。

**5.跨平台性。**是否需要同时支持移动端和pc端

### ****我们从这几个角度去分析一下几个框架的优劣****

先说一下这三个框架的共同之处，首先这三个框架都有很好的性能(这里的angular指的是2.0+)，都支持数据绑定，组件等基本功能。

### ****Angular****

这是一个给开发者一整套解决方案的框架，相对于vue和react，angular不需要搭配其他库，就可以构建出一个大型项目，但它并不太适合开发小型应用。下面是angular的一些特点~

1.**这是一个完整的框架拥有良好的项目结构**，通常情况下，我们编写的js代码是没有正规的项目结构的，是因为在小型项目中对结构的要求很低，但是在大型项目中则完全不同，比，如webapp中ionic框架就是用的Angular作为内核个人觉得也是看中了angular的这个特点。但是会丢失一些灵活性。

**2.拥有自己的构建工具**，在Angular-CLI 使用打包编译，生成组件等都有相应的命令行，非常方便快捷，虽然vue和React也有构建工具，但是局限性很大，需要配合其他构建工具，个人觉得Angular-CLI足够强大，这也是一套完整的框架的带来的红利，不必在选择库上浪费时间。

**3.体积较大**。虽然在angular2+之后 使用了AOT和 tree-shaking，但是相对于其他轻量级框架来说还是略显臃肿。加载较慢

**4.学习成本较高**。需要很多基础概念和使用较复杂的api接口，入门相对困难。而且angular2.0+用的是ts语言，需要对ts语言有一定程度的了解。而且从angular1.x升级到2.x的时候框架几乎是重写了一遍，导致之前用angular1写的程序维护起来比较困难。如果是新入门学习angular，推荐从2开始学起。

**5.跨平台优势**。有ionic等使用angular作为内核的框架，如果是用angular开发pc端+移动端的跨平台开发，组件服务指令都可以复用，这对开发这来说是非常不错的，react有React Native同样也是跨平台非常不错。vue有与阿里合作Weex，但是目前来说跟前两个还有很大差距。

**6.生态系统庞大。**angular和react都有庞大的生态系统，vue相对较差

**总结**：它是一个成熟完善的框架，而React和Vue只是一个UI组件库。angular适用于大型项目，所以会有一些代价，比如学习成本高，如果你只是想用到组件，数据绑定等基础功能去开发一个小型应用，那么最好不要选择angular~

### ****Vue****

一句话说明Vue的特点，个人总结：灵活，构建项目可大可小，学习成本低，性能好，适合开发小型应用。这里不是说它不能构建大型应用，只不过个人觉得你如果想开发一个尽可能的小和快的应用，我建议你使用vue~

**1.灵活性**。它从不限制你用什么样的代码组织结构，更加随意。

**2.实用性。**它虽然很轻量，但是却拥有很强大的实用性，数据绑定，计算属性侦听器，组件等常用功能不次于angular。在很多方面比angular容易上手更

3**.体积小。**vue相对于angular体积小了很多。

**4.学习成本较低。**你只需要有良好的 HTML 和 JavaScript 基础就可以通过官网上的阅读指南快速投入开发，相比于react和angular都有很大的优势，angular需要学习各种各样的api，理解各种基础概念，还有学习ts语言才能进行开发；react需要知道 JSX 和 ES2015还需要学习构建系统等

**5.跨平台优势较差。**与其他两个框架相比，跨平台优势较差，虽然与阿里合作weex但是差距还很大。

总结：轻量，学习成本低等等，优点很多。缺点就是他的生态社区跟angular和react目前还差很远~

### ****React****

个人觉得React和Vue有很多相似之处，1.两个框架都专注于视图层，其他功能如路由和全局状态管理交给相关的库，这跟angular有很大不同 2. 都使用 "Virtual DOM" 3.提供了响应式和组件化的视图组件，React 的优势在于生态系统比较繁荣

1.**灵活性。**这点跟vue很像，React可以与已知的库或框架很好地配合。

2.**生态圈强大。**因为react把路由库和状态管理库交给社区维护，虽然相对来说这些方面不如vue和angular的官方发布稳定，但是造就了生态圈的繁荣。

3.**跨平台优势**，React Native 能使你用相同的组件模型编写有本地渲染能力的 APP (iOS 和 Android)。能同时跨多平台开发，类似于ionic。

**4.学习成本一般，**，在你开始学 React 前，你需要知道 JSX 和 ES2015，相对于vue是学习难度高，相对于angular来说比较好学，如果要构架大型应用，它的生态是相对复杂，个人觉得没有angular官方发布的文档清晰。

总结：react和vue一样只关注视图层，只是一个UI组件库，这跟angular有本质的区别，如果react想开发大型应用需要配合第三方库，他的跨平台优势和生态优势大于vue。

假如你想快速开发一个简单学习成本低的小应用，可以考虑Vue~

假如你想开发一个大型应用，可以考虑angular~

假如你想开发一个跨平台应用，可以考虑react~

## Python

### Pelican

**Pelican**是使用Python编写的静态网站生成工具。它支持用reStructuredText, Markdown, 和AsciiDoc创作网站内容。Pelican支持Jinja模版引擎，结果是，它支持很多自定义主题。

Pelican 是一个法国人用 python 写的用于生成静态页面的程序，支持：

* 博客文章和页面
* 使用外部服务 Disqus 实现的评论功能
* 支持主题
* 可对文章生成 PDF 文档
* 支持多语言发布文章
* Atom/RSS feeds
* 代码着色
* 使用 LESS CSS (optional)
* 可导入 WordPress, Dotclear 或者 RSS feeds
* 集成外部功能 Twitter, Google Analytics, etc. (optional)

开始使用Pelican：http://docs.getpelican.com/en/3.6.3/install.html

### Cactus

Cactus是使用Python和Django模版系统制作的静态网站生成工具。

Cactus 是一个简单而强大的静态网页生成器程序，它使用 Python 和 Django 的模板系统。它的本地开发和在S3 上的部署都非常的简单。

因为目前的动态网站大部分都可以使用 JavaScript 来完成，这样实际上网页完全可以是静态的，而且静态网页速度非常快并且容易管理。所以才有了这个项目。

作者开发 Cactus 的目的是为了给设计师们提供一个标准而简单的系统，让他们能够快速的构建和部署一个速度很快的网站。

开始使用Cactus：https://github.com/koenbok/Cactus/

### MkDocs

MkDocs 可以同时编译多个markdown文件，形成书籍一样的文件。有多种主题供你选择，很适合项目使用。

MkDocs 是快速，简单和华丽的静态网站生成器，可以构建项目文档。文档源文件在 Markdown 编写，使用单个 YAML 配置文件配置。

## Node.js

### Hexo

**Hexo**是用Node.js编写的博客框架。这个静态网站生成工具非常快，使用它构建一个完整的网站只需要几秒钟。Hexo支持所有的GitHub Markdown特性，并支持大多数Octopress插件。

从其他博客平台迁移到hexo非常容易。

Hexo 是一个基于nodejs 的静态博客网站生成器，作者是来自台湾的 Tommy Chen。

特点：

* 不可思议的快速 ─ 只要一眨眼静态文件即生成完成
* 支持 Markdown
* 仅需一道指令即可部署到 GitHub Pages 和 Heroku
* 已移植 Octopress 插件
* 高扩展性、自订性
* 兼容于 Windows, Mac & Linux

[Hexo的文档]https://hexo.io/docs/

### Metalsmith

Metalsmith是简单、高效、pluggable静态网站生成工具，它使用nodejs编写。Metalsmith和其他工具的最大区别是它的所有东西都由插件处理，并且插件可以重用。只要决定网站的功能，然后找到相关插件，组合到一起，ok，ready to go!

Metalsmith也可以生成PDF、电子书、文档等等。

开始使用Metalsmith：http://www.metalsmith.io/

### Wintersmith

Wintersmith是极简的、可扩展的静态网站生成工具，它使用Nodejs编写。它同样支持插件。Wintersmith的项目基于目录结构，可以方便的移植旧站点。

开始使用Wintersmith：

https://github.com/jnordberg/wintersmith#quick-start

### ****基于 Git 制作电子书 GitBook****

使用GitBook生成的电子书

GitBook支持输出多种文档格式：

* 静态站点：GitBook默认输出该种格式，生成的静态站点可直接托管搭载Github Pages服务上；
* PDF：需要安装gitbook-pdf依赖；
* eBook：需要安装ebook-convert；
* 单HTML网页：支持将内容输出为单页的HTML，不过一般用在将电子书格式转换为PDF或eBook的中间过程；
* JSON：一般用于电子书的调试或元数据提取。

### Assemble

Assemble 是一个使用 Node.js，Grunt.js，Gulp，Yeoman 等来实现的静态网页生成系统。已被 Zurb Foundation, Zurb Ink, H5BP/Effeckt, Less.js / lesscss.org, Topcoat, Web Experience Toolkit 等数百个项目用来生成项目网站、主题、组件、文档、博客和 github 页面。

## JavaScript

### HubPress

HubPress 是一个由  JavaScript 编写的静态网站生成器，使博客维护更加简单。

主要特性：

* 提供 WYSIWYG 编辑器撰写博客
* 支持 AsciiDoc 标记功能，将内容按照用户需求呈现
* 管理控制台可以自定义博客内容的许多方面
* Disqus 整合博客评论
* 利用Google Analytics 集成来跟踪访问者活动
* 附带多种主题，随时可以使用

HubPress是开源的web应用，使用它可以允许你创建一个基于GitHub Pages的博客。HubPress的使用非常简单，你只需要fork这个项目到你的github，然后修改配置文件就可以了。

开始使用HubPress：https://github.com/HubPress/hubpress.io

## Ruby

### Jekyll

**Jekyll**做为GitHub Pages的构建工具（Ruby语言），使它成为最流行的静态网站生成工具。Jekyll的流行也因为它非常简单，只需要基础的web开发基础。你可以使用它轻易的把文本转换为自定义的网站/博客。

如果你有wordpress或其他博客站点，你可以导入到Jekyll中。Jekyll支持插件、标签等等。

Github Pages：https://pages.github.com 开始使用Jekyll：http://jekyllrb.com/docs/quickstart/

### Middleman

Middleman －中间人，又一个使用Ruby编写的静态网站生成工具。它提供怎么使用和自定义的文档，方便你自定义你的网站。

Middleman is a static site generator using all the shortcuts and tools in modern web development.

开始使用Middleman：https://middlemanapp.com/basics/install/

### Octopress

**Octopress**是基于Jekyll的博客生成工具，它简化了Jekyll的操作，可以让你更舒服的创作。Octopress的一大优势是它插件很多，并且兼容Jekyll的官方插件。

Octopress支持内建的社交平台（Twitter, Google+），Disqus评论和Google Analytics。

Octopress的文档：http://octopress.org/docs/

## 基于Go

### Hugo

[Hugo]http://gohugo.io/是另一个流行的静态网站生成工具，它是使用go语言编写，并且使用Markdown语法。官网对它的描述：This application does not depend on administrative privileges, databases, interpreters, or external libraries, and still works like a charm. Websites or blogs built with Hugo can be hosted on any web host including GitHub Pages, S3, and Dropbox.

开始使用Hugo：http://gohugo.io/overview/quickstart/

## React

### React 的渐进式静态网站生成器 React Static

React Static 是一个 React 的渐进式静态网站生成器。它也是一个服务端渲染 React 应用的简约框架，旨在构建一个满足 SEO，网站性能和用户/开发人员使用体验的标准，帮助每个人无痛地构建下一代、高性能的网站。

**功能特性**

* 100% React。
* 快速运行，高性能构建。
* 数据平台不可知论者（Data Agnostic），可从任何地方提供你的网站数据。
* 为 SEO 而生。
* React 优先的开发体验。
* 无痛的项目设置和迁移。
* 100％ 支持 React 生态系统。 包括 CSS-in-JS 库，自定义 Query 层（如 GraphQL），甚至 Redux。

### ReactJS 静态网站生成器 Gatsby

Gatsby 可以使用 React.js 把纯文本转换到动态博客或者网站上。

特点：

* 无需重载页面转换
* 热重载编辑
* 为构建静态网站创建 React.js 组件模型和生态系统
* 直观的基于目录的 URLs
* 支持 "Starters"

## Vue.js

### Vue.js 后端渲染开源库 Nuxt.js

Nuxt.js 是一个通过 Vue 用于服务端渲染的简单框架，灵感来自 Next.js。 Nuxt 基于 ES2015，这使得代码有着更愉快，更整洁的阅读体验。它不使用任何转换器，并取决于 Core V8 实现的功能。

## 其他

### DocPad

DocPad自带建立好的网站主架，允许你快速的建立功能完整的网站。这个工具支持CoffeeScript、Ruby、PHP、Stylus等等。DocPad removes limitations and closes the gap between experts and beginners. Designers and developers can create websites faster than ever before.

DocPad  可以帮助生成具有布局，元数据，预处理器（markdown，jade，coffeescript 等等），部分，骨架，文件查看器，查询和完美的插件系统的网站前端。这大大减少了有经验开发者和初学者开发网站之间的不同，帮助用户更快速的建立自己的网站。

开始使用DocPad：http://docpad.org/docs/install

### 前端 Web 应用程序构建工具 Brunch

Brunch 是一个轻量级的、优雅和简单的方法构建 HTML5 应用程序的框架，快速的前端 Web 应用程序构建工具，具有简单的声明性配置，用于快速开发的无缝增量编译。

### Expose

### Wintersmith

### 模块化网站编译器 Phenomic

### Lektor

# Pelican 4

G:\SBT-MKDocs>pelican --help

usage: pelican [-h] [-t THEME] [-o OUTPUT] [-s SETTINGS] [-d] [-v] [-q] [-D] [--version] [-r]

[--print-settings [SETTING\_NAME [SETTING\_NAME ...]]] [--relative-urls] [--cache-path CACHE\_PATH]

[--ignore-cache] [-w SELECTED\_PATHS] [--fatal errors|warnings]

[--logs-dedup-min-level {DEBUG,INFO,WARNING,ERROR}] [-l] [-p PORT] [-b BIND]

[path]

A tool to generate a static blog, with restructured text input files.

positional arguments:

path Path where to find the content files. (default: None)

optional arguments:

-h, --help show this help message and exit

-t THEME, --theme-path THEME

Path where to find the theme templates. If not specified, it will use the default one included

with pelican. (default: None)

-o OUTPUT, --output OUTPUT

Where to output the generated files. If not specified, a directory will be created, named

"output" in the current path. (default: None)

-s SETTINGS, --settings SETTINGS

The settings of the application, this is automatically set to pelicanconf.py if a file exists

with this name. (default: None)

-d, --delete-output-directory

Delete the output directory. (default: None)

-v, --verbose Show all messages. (default: None)

-q, --quiet Show only critical errors. (default: None)

-D, --debug Show all messages, including debug messages. (default: None)

--version Print the pelican version and exit.

-r, --autoreload Relaunch pelican each time a modification occurs on the content files. (default: False)

--print-settings [SETTING\_NAME [SETTING\_NAME ...]]

Print current configuration settings and exit. Append one or more setting name arguments to

see the values for specific settings only. (default: None)

--relative-urls Use relative urls in output, useful for site development (default: False)

--cache-path CACHE\_PATH

Directory in which to store cache files. If not specified, defaults to "cache". (default:

None)

--ignore-cache Ignore content cache from previous runs by not loading cache files. (default: False)

-w SELECTED\_PATHS, --write-selected SELECTED\_PATHS

Comma separated list of selected paths to write (default: None)

--fatal errors|warnings

Exit the program with non-zero status if any errors/warnings encountered. (default: )

--logs-dedup-min-level {DEBUG,INFO,WARNING,ERROR}

Only enable log de-duplication for levels equal to or above the specified value (default:

WARNING)

-l, --listen Serve content files via HTTP and port 8000. (default: False)

-p PORT, --port PORT Port to serve HTTP files at. (default: 8000) (default: None)

-b BIND, --bind BIND IP to bind to when serving files via HTTP (default: 127.0.0.1) (default: None)

## Pelican简介

Pelican用Python编写的静态站点生成器。亮点：

* 使用reStructuredText或Markdown格式直接用编辑器编写内容
* 简单的CLI工具来(重新)生成站点
* 易于与分布式版本控制系统和Web对接
* 完全静态输出，可在任何地方托管

Pelican 4目前支持：

* 文章（例如博客文章）和页面（例如，“关于”，“项目”，“联系方式”）
* 评论，通过外部服务（Disqus）。如果您希望更好地控制评论，也可自托管评论是另一种选择。
* 主题支持（使用Jinja2模板创建主题）
* 以多种语言发表文章
* Atom / RSS feeds
* 代码高亮
* 从WordPress，Dotclear或RSS导入
* 与外部工具集成：Twitter，Google Analytics等（可选）
* 借助内容缓存和选择性输出写入，可以快速重建

为什么叫“Pelican”这个名字？

“Pelican”是calepin的字谜，在法语中意为“笔记本”。 ;）

您可以访问以下网址获取源代码：https：[//github.com/getpelican/pelican](https://github.com/getpelican/pelican)。更多相关库参见<https://github.com/china-testing/python-api-tesing>。

搭建博客的工具选用了Pelican，Pelican是一个用Python语言编写的静态网站生成器，支持使用restructuredText和Markdown写文章，配置灵活，扩展性强，有许多优秀的主题和插件可供使用。Pelican 的Github地址是：<https://github.com/getpelican/pelican>;

## pelican 的功能

* 支持 reStructuredText、Markdown、AsciiDoc 编写格式
* 完全静态输出任何地方都很容易承载
* 可以通过 Jinja templates 自定义 Themes
* 多语言发布
* 支持 Atom/Rss feeds
* 代码语法高亮
* 支持从 WordPress、Dotclear 、Rss feeds 等服务中导入信息
* 模块化插件系统强大

## Installing Pelican

先根据xxx搭建python环境   
vscode+pipenv

pip install pelican markdown

### 安装git,pip

|  |  |
| --- | --- |
| 1  2 | emerge --ask dev-cvs/git  emerge --ask dev-python/pip |

### 使用virtualenv工具创建pelican虚拟环境

|  |  |
| --- | --- |
| 1  2  3  4 | emerge --ask virtualenv ## or pip install virtualenv  virtualenv ~/virtualenvs/pelican  cd ~/virtualenvs/pelican  source bin/activate |

### 安装pelican

|  |  |
| --- | --- |
| 1 | pip install pelican |

### 安装markdown,typogrify

|  |  |
| --- | --- |
| 1  2 | pip install Markdown  pip install typogrify |

## Create a project

**环境安装和初始化**

pipenv install pelican markdown

运行初始化pipenv run pelican-quickstart，它会问些关于站点的基本问题。

Welcome to pelican-quickstart v3.7.1.   
This script will help you create a new Pelican-based website.   
Please answer the following questions so this script can generate the files   
needed by Pelican.   
> Where do you want to create your new web site? [.]   
> What will be the title of this web site?   
> Who will be the author of this web site?   
> What will be the default language of this web site?   
> Do you want to specify a URL prefix? e.g., <http://example.com> (Y/n)   
> Do you want to enable article pagination? (Y/n)   
and balabalabala……

blog/

├── content # 存放输入的markdown或RST源文件

│ └── (pages) # 存放手工创建的静态页面，可选

├── output # 存放最终生成的静态博客

├── develop\_server.sh # 测试服务器

├── Makefile # 管理博客的Makefile

├── pelicanconf.py # 配置文件

└── publishconf.py # 发布文件，可删除

> Where do you want to create your new web site? [.] # 默认

> What will be the title of this web site? myblog # 输入博客名称，例如：myblog

> Who will be the author of this web site? obama # 输入博客作者

> What will be the default language of this web site? [Chinese (Simplified)] # 默认中文

> Do you want to specify a URL prefix? e.g., https://example.com (Y/n) # Y ，如果打算放在github上，可以选择 n

> What is your URL prefix? (see above example; no trailing slash) # 输入githup地址，例如：https://myblog.github.io

> Do you want to enable article pagination? (Y/n) # 默认

> How many articles per page do you want? [10] # 默认

> What is your time zone? [Europe/Paris] # Asia/Shanghai

> Do you want to generate a tasks.py/Makefile to automate generation and publishing? (Y/n) # 默认

> Do you want to upload your website using FTP? (y/N) # N

> Do you want to upload your website using SSH? (y/N) # N

> Do you want to upload your website using Dropbox? (y/N) # N

> Do you want to upload your website using S3? (y/N) # N

> Do you want to upload your website using Rackspace Cloud Files? (y/N) # N

> Do you want to upload your website using GitHub Pages? (y/N) # 如果是用 github，默认

Done. Your new project is available at F:\Blog\pelican\_blog

新建一个文件夹，

mkdir -p ~/projects/yoursite

cd ~/projects/yoursite

|  |  |
| --- | --- |
| 1  2  3 | mkdir blog //创建文件夹，名称可根据自己喜欢定  cd blog  pelican-quickstart  mkdir -p ~/projects/yoursite  cd ~/projects/yoursite  pelican-quickstart |

pelican-quickstart执行命令后，会提示输入博客的配置项，除了少数几个必填以外，其它都可以选择默认，而且都可以在pelicanconf.py文件中进行更改，所以你可以随意选择。 命令成功执行后，会出现pelican的框架，如下所示

## pelican站点目录组成

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | blog/  ├── content # 存放输入的markdown或RST源文件  │ └── (pages) # 存放手工创建的静态页面，可选  │ └── (posts) # 存放手工创建的文章，可选  ├── output # 存放最终生成的静态博客  ├── develop\_server.sh # 测试服务器  ├── Makefile # 管理博客的Makefile  ├── pelicanconf.py # 配置文件  └── publishconf.py # 发布文件，可删除 |

## Writing content

### Articles and pages 文章和页面

Pelican 认为 “文章(articles)” 是按时间顺序排列的内容, 如博客上的帖子, 因此与日期相关。

“页面(pages)” 背后的想法是, 它们通常不是时态性质的, 用于不经常更改的内容 (例如, “关于” 或 “联系” 页面)。

你可以在官方库中找到一些例子在https://github.com/getpelican/pelican/tree/master/samples/content

创建文章

~/projects/yoursite/content/keyboard-review.md

Title: My First Review

Date: 2010-12-03 10:20

Category: Review

Following is a review of my favorite mechanical keyboard.

**创建第一篇文章**

在创建文章之前，Pelican是不能运行的。在yoursite/content/目录下添加第一篇markdown。

Title: 我的第一篇文章

Date: 2018-01-25 11:37

Category: 日记

胡乱写点什么试试

源文件顶部Title,Date,Category等内容则是必须的，具体可参见[文档](http://docs.getpelican.com/en/3.3.0/getting_started.html#writing-content-using-pelican)，它们各自意义如下

Title: 文章标题

Date: 创建日期

Modified: 修改日期

Category: 文章分类，标志本文处于该分类下

Tags: 文章标签，标志本文处于该标签下

Slug: URL中该文章的链接地址

Author: 作者

### File metadata 文件中的信息

Pelican 尽可能的自己从文件系统得到信息（如，关于文章的类别，当不指定时pelican会使用文件夹作为分类依据），但是你仍需要在文件中提供一些信息(metadata)。

完成上述博客主体搭建后，使用markdownpad创建一个.md文件，保存于content文件夹中（或自己加入posts文件夹）。 Metadata syntax for Markdown posts should follow this pattern:

|  |  |
| --- | --- |
| 12345678910 | Title: My super titleDate: 2010-12-03 10:20Modified: 2010-12-05 19:30Category: PythonTags: pelican, publishingSlug: my-super-postAuthors: Alexis Metaireau, Conan DoyleSummary: Short version for index and feedsThis is the content of my super blog post. |

如果您以 reStructuredText 格式编写内容, 则可以通过以下语法在文本文件中提供此信息 (以 . rst后缀名的文件):

My super title

:date: 2010-10-03 10:20   
:modified: 2010-10-04 18:40   
:tags: thats, awesome   
:category: yeah   
:slug: my-super-post   
:authors: Alexis Metaireau, Conan Doyle   
:summary: Short version for index and feeds

authors 和 tags 可以用分号（;）或逗号 (,) 分隔。

:tags: pelican, publishing tool; pelican, bird   
:authors: Metaireau, Alexis; Doyle, Conan

Pelican 通过abbr标记实现了对 reStructuredText 的扩展, 以启用对 HTML 的支持。你可以在你的推送(post)中像这样写:

This will be turned into :abbr:HTML (HyperText Markup Language).

你同样可以使用 Markdown 语法 （后缀名为 .md，.markdown，.mkd或.mdown）。Markdown生成器要求你先安装Markdown包，请尝试运行pip install Markdown命令。

Pelican 同时支持扩展的Markdown语法 (Markdown Extensions)，如果扩展语法没有包含在Markdown包中，可能需要另行安装。

Markdown 的文件需要像下面一样添加信息(metadata)：

Title: My super title

Date: 2010-12-03 10:20

Modified: 2010-12-05 19:30

Category: Python

Tags: pelican, publishing

Slug: my-super-post

Authors: Alexis Metaireau, Conan Doyle

Summary: Short version for index and feeds

This is the content of my super blog post.

| **Metadata** | **描述** | **备注** |
| --- | --- | --- |
| title | 文章或页面的标题 |  |
| date | 出版日期(例如，YYYY-MM-DD HH:SS) |  |
| modified | 修改日期(例如，YYYY-MM-DD HH:SS) |  |
| tags | 内容标签，用逗号分隔 |  |
| keywords | 内容关键字，用逗号分隔(仅HTML内容) |  |
| category | 内容类别(只有一个-不是多个) |  |
| slug | URL和翻译中使用的标识符 | 文件名（不含扩展名） |
| author | 内容作者，当只有一个 |  |
| authors | 内容作者，当有多个 |  |
| summary | 索引页内容简介 |  |
| lang | 内容语言ID(en, fr等等) |  |
| translation | 是另一个(true或false) |  |
| status | 内容状况：draft, hidden，或published |  |
| template | 用于生成内容的模板名称(没有扩展) |  |
| save\_as | 将内容保存到此相对文件路径 |  |
| url | 用于本文/页的URL |  |

其他格式（如 AsciiDoc）需要通过插件支持，请参与pelican-plugins的相关说明。

Pelican同样可以处理.html和.htm结尾的HTML文件。它以非常直接的方式读取信息，title从title，body从body：

<html>

<head>

<title>My super title</title>

<meta name="tags" content="thats, awesome" />

<meta name="date" content="2012-07-09 22:28" />

<meta name="modified" content="2012-07-10 20:14" />

<meta name="category" content="yeah" />

<meta name="authors" content="Alexis Métaireau, Conan Doyle" />

<meta name="summary" content="Short version for index and feeds" />

</head>

<body>

This is the content of my super blog post.

</body>

</html>

使用 html, 标准信息有一个简单的做法: tags可以按Pelican标准要求的通过tags标记指定, 也可以按HTML要求的通过keywords提供信息。两者可以互换使用。

请注意, 除了title之外, 本文的其他信息(metadata)都不是必需提供的: 如果未指定日期, 并且 DEFAULT\_DATE 设置为 'fs', 则将依赖于文件的 “mtime” 时间戳, 并且该类别可以由文件所在的目录确定。例如, 位于 python/foobar/myfoobar. rst 的文件将有一个foobar类别。如果您想以其他方式组织文件, 而该文件夹的名称不是一个好的类别名称, 则可以将设置 USE\_FOLDER\_AS\_CATEGORY 设置为 False。在分析页面信息给定的日期时, Pelican支持 W3C’s 建议的[ISO 8601](http://www.w3.org/TR/NOTE-datetime)。

**注意：**   
在尝试使用不同的设置 (特别是信息) 时, 缓存可能会发生干扰，使更改可能不可见。在这种情况下, 禁用缓存 LOAD\_CONTENT\_CACHE = False 或使用--ignore-cache命令行开关忽略缓存。

modified应是上次更新文章的时间, 如果未指定, 则默认为date。除了可以在模板中显示modified外, 当您修改文章后将midified设置为当前日期时, feed readers 中的提要条目将自动更新。

authors是一个逗号分隔的文章作者列表。如果只有一个作者可以使用author。

如果你没有在 post 中指定摘要信息, 则可以使用 SUMMARY\_MAX\_LENGTH 设置来指定从文章开头的多少单词用作摘要。

还可以通过在 FILENAME\_METADATA 设置中设置的正则表达式从文件名中提取信息。所有匹配的命名组将在信息对象中设置。FILENAME\_METADATA 设置的默认值仅从文件名中提取日期。例如, 如果您想提取日期和slug, 可以设置如下内容:'(?P<date>\d{4}-\d{2}-\d{2})\_(?P<slug>.\*)'

请注意, 文件中可用的信息优先于从文件名中提取的信息。

### Pages 页面

如果你在 content 文件夹下建立了一个叫pages的文件夹，所有在这个文件夹内的文件都会用来生成静态页面 ( static pages)，像 About 和 Contact 页面等。（看下面给出的例子。）

你可以使用设置 DISPLAY\_PAGES\_ON\_MENU 来控制是否所有这些页面都显示在主导航菜单中。(默认值为 True。)

如果要关闭某个页面在菜单中链接或列出, 请添加status: hidden属性到其信息中。这对于制作适合网站主题的错误页面(error pages)非常有用。

### Linking to internal content 内部链接

从 Pelican 3.1 开始, 现在可以指定站点内部链接 ( intra-site ) 到**源内容层次 (source content hierarchy)**结构中的文件, 而不是**生成的层次 (generated hierarchy)**结构中的文件。这使得从当前帖子 ( current post)链接到可能与该帖子并排的其他内容更容易 (而不必确定在站点生成之后其他内容的将放置位置)。

要链接到内部内容 (content目录中的文件), 请对链接目标使用以下语法: {filename}path/to/file，注: 正斜线（/）是所有操作系统 (包括 Windows) 的 {filename} 指令中的路径分隔符。

举例如下，加入有一个 Pelican 项目的结构像下面这样：

website/

├── content

│ ├── category/

│ │ └── article1.rst

│ ├── article2.md

│ └── pages

│ └── about.md

└── pelican.conf.py

* 1
* 2
* 3

其中article1.rst可能是这样：

The first article

#################

:date: 2012-12-01 10:02

See below intra-site link examples in reStructuredText format.

`a link relative to the current file <{filename}../article2.md>`\_

`a link relative to the content root <{filename}/article2.md>`\_

article2.md如下：

Title: The second article

Date: 2012-12-01 10:02

See below intra-site link examples in Markdown format.

[a link relative to the current file]({filename}category/article1.rst)

[a link relative to the content root]({filename}/category/article1.rst)

#### Linking to static files 链接到静态文件

像上文提到的使用{filename}标识来连接到既非 aritcle 也非 pages 的内容。必须记住, 这些文件不会被复制到output目录中, 除非包含它们的源目录包含在项目的 pelicanconf.py 文件的 STATIC\_PATHS 设置中。Pelican 的配置默认包括images目录，但必须手动添加其他内容。忘记这样做会导致链接断开。

举例来说，一个项目又如下目录结构：

content

├── images

│ └── han.jpg

├── pdfs

│ └── menu.pdf

└── pages

└── test.md

test.md中的链接：

![Alt Text]({filename}/images/han.jpg)

[Our Menu]({filename}/pdfs/menu.pdf)

* 1
* 2

pelicanconf.py文件设置如下：

STATIC\_PATHS = ['images', 'pdfs']

* 1

网站生成的时候就会复制han.jpg到output/image/han.jpg，menu.pdf到output/pdfs/menu.pdf，并且添加正确的链接到test.md。

#### Mixed content in the same directory 同一文件夹下的混合内容

从 Pelican 3.5 开始, 静态文件可以安全地与页面源文件共享源目录, 而不会在生成的站点中公开页面源。任何此类目录都必须同时添加到 STATIC\_PATHS 和 PAGE\_PATHS (或 STATIC\_PATHS 和 ARTICLE\_PATHS)。Pelican 通常会识别和处理页面源文件, 并将其余的文件复制到为静态文件保留的单独目录中。

Note: 将静态和内容源文件放在同一个源目录中并不能保证它们最终会在生成的站点中的同一位置。为确保如此的最简单方法是使用 {attach} 链接语法 (参见下一节)。或者, 可以将 STATIC\_SAVE\_AS、PAGE\_SAVE\_AS 和 ARTICLE\_SAVE\_AS 属性 (以及相应的 \* \_URL 属性) 设置为将不同类型的文件放在一起, 就像在早期版本的 Pelican 中一样。

#### Attaching static files 附件

从 Pelican 3.5 开始, 静态文件(static files)可以使用{attach}path/to/file将目标 “附加” 到一个网页或文章上，这类似于 {filename} 语法, 但也将静态文件重新定位到链接文档的output目录中。如果静态文件源自链接文档源下的子目录, 则该关系将保留在输出上。否则, 它将成为链接文档的同级。

这只适用于链接到静态文件, 而且仅当它们源自于 STATIC\_PATHS 属性中包含的目录时。

举个栗子，如下的目录结构：

content

├── blog

│ ├── icons

│ │ └── icon.png

│ ├── photo.jpg

│ └── testpost.md

└── downloads

└── archive.zip

pelicanconf.py设置如下：

PATH = 'content'

STATIC\_PATHS = ['blog', 'downloads']

ARTICLE\_PATHS = ['blog']

ARTICLE\_SAVE\_AS = '{date:%Y}/{slug}.html'

ARTICLE\_URL = '{date:%Y}/{slug}.html'

testpost.md内容如下：

Title: Test Post

Category: test

Date: 2014-10-31

![Icon]({attach}icons/icon.png)

![Photo]({attach}photo.jpg)

[Downloadable File]({attach}/downloads/archive.zip)

生成器生成的output目录结构如下：

output

└── 2014

├── archive.zip

├── icons

│ └── icon.png

├── photo.jpg

└── test-post.html

请注意, 所有使用 {attach} 链接的文件最终都在文章的输出目录中或下面。

如果静态文件 ( static file )被多次链接, 则 {attach} 的重新定位功能将只在要处理的第一个链接中起作用。在第一个链接之后, Pelican将像 {filename} 一样处理 {attach}。这样可避免中断已处理的链接。

**从多个文档链接到文件时要小心**： 由于 Pelican 没有定义处理文档的顺序，第一个指向文件的链接确定了其最终位置，因此在由多个文档链接的文件上使用 {attach}时会导致在不同的版本生成时文件位置发生更改（ cause its location to change from one site build to the next）。(这是否在实际中发生受操作系统、文件系统、Pelican 版本以及从项目中添加、修改或删除的文档的影响。）任何链接到文件旧位置的外部站点都可能会发现它们的链接已失效。**因此, 仅当在链接的文档在同一个目录下时，才建议使{attach}，并且要同时全部使用{attach}。It is therefore advisable to use {attach} only if you use it in all links to a file, and only if the linking documents share a single directory.** 在这样的情况下, 文件的输出位置在将来的生成中不会更改。在无法采取这些预防措施的情况下, 请考虑使用 {filename} 链接而不是 {attach} , 并让该文件的位置由项目的 STATIC\_SAVE\_AS 和 STATIC\_URL 设置确定。(每个文件 save\_as 和 url 仍然可以在 EXTRA\_PATH\_METADATA 中设置重写。) 此段落翻译不好，大家将就看下

#### Linking to authors, categories, index and tags

你可以使用{author}name连接到作者；{category}foobar连接到类别；{index}到索引；{tag}tagname到标签。

#### Deprecated internal link syntax 快要过时的内部链接语法

为了与早期版本保持兼容, Pelican 仍然支持垂直条 (| |)的内部链接。例如: | filename | an\_article.rst， | tag | tagname |， |category | foobar。语法从 | |到 {} 的改变是为了避免与扩展 Markdown 或 reST 的指令冲突。可能最终会删除对旧语法的支持。

### Importing an existing site 从已存在的站点导入

可以用一个简单的脚本实现从 WordPress、Tumblr、Dotclear 和Rss feeds的导入，详见[站点导入](http://docs.getpelican.com/en/stable/importer.html#import)

### Translations 翻译

提供多语种的文章是可行的，为了做到这些，你需要在文章或是页面中添加lang信息，并设置DEFAULT\_LANG属性（默认英语[en])。有了这些设置, 将只列出具有默认语言的文章, 并且每篇文章都附有该文章的可用翻译列表。

Note：   
Pelican 的核心功能不创建每个语言的翻译模板的子站点 (例如 example.com/de)。对于这样的高级功能, 可以使用 i18n\_subsites 插件。

Pelican 使用文章的 URL “slug” 来确定两个或更多的文章是否是互相的翻译。可以在文件的信息(metadata)中手动设置slug;如果没有明确设置, Pelican 将通过标题自动生成slug。

下面给个例子，这里有两篇文章，一个英语一个法语。

英语文章：

Foobar is not dead

##################

:slug: foobar-is-not-dead

:lang: en

That's true, foobar is still alive!

* 1

法语版：

Foobar n'est pas mort !

#######################

:slug: foobar-is-not-dead

:lang: fr

Oui oui, foobar est toujours vivant !

发布内容后, 您可以看到这两篇文章中唯一的共同点就是slug, 它在这里是一个标识符。如果您不希望以这种方式显式定义该slug, 则必须确保翻译的文章标题相同, 因为将从文章标题自动生成slug。

如果不希望 DEFAULT\_LANG 检测到某个特定文章的原始版本, 请在信息（metadata）中使用translation指定哪些帖子是翻译的版本：

Foobar is not dead

##################

:slug: foobar-is-not-dead

:lang: en

:translation: true

That's true, foobar is still alive!

### Syntax highlighting 语法高亮

Pelican 可以为您的代码块提供彩色的语法高亮。为此, 您必须在内容文件中使用以下约定。

对于 reStructuredText, 请使用code-block指定要突出显示的代码类型 (在此示例中, 我们将使用 python)：

.. code-block:: python

print("Pelican is a static site generator.")

* 1
* 2
* 3

对于使用 CodeHilite 扩展来提供语法高亮的Markdown, 请在代码块的正上方包含语言标识符, 同时缩进标识符和代码：

There are two ways to specify the identifier:

:::python

print("The triple-colon syntax will \*not\* show line numbers.")

To display line numbers, use a path-less shebang instead of colons:

#!python

print("The path-less shebang syntax \*will\* show line numbers.")

指定的语言标识符 (如 python、ruby) 应该出现在[list of available lexers](http://pygments.org/docs/lexers/)中。

使用 reStructuredText 时, 代码块指令 (code-block directive) 中提供了以下选项:

| **选项** | **有效值** | **描述** |
| --- | --- | --- |
| anchorlinenos | N/A | 如果在<a>标记中显示行号 |
| classprefix | String | 标记类名 |
| hl\_lines | numbers | 要突出显示的行的list, 其中要突出显示的行号由空格分隔。这类似于在Sphinx中的emphasize-lines, 但它不支持用连字符分隔的行号范围, 或逗号分隔的行号。 |
| lineanchors | string | 在lineanchors和-linenumber之间对其中的每一行进行包装。 |
| ………太多了了 | 去看原文吧 | <http://docs.getpelican.com/en/stable/content.html#syntax-highlighting> |

Note：不同的版本中, 您的 Pygments 模块可能没有所有可用的选项。有关每个选项的详细信息, 请参阅 [Pygments 文档](http://pygments.org/docs/formatters/)的 HtmlFormatter 部分。

例如, 下面的代码块启用行号, 从153开始, 并使用 pgcss 前缀 Pygments css 类, 以使名称更加独特, 并避免可能的 css 冲突:

.. code-block:: identifier

:classprefix: pgcss

:linenos: table

:linenostart: 153

<indented code block goes here>

### Publishing drafts 草稿

如果要将文章发布为草稿 (例如, 在发布前供好友审阅)，可以将Status:draft 属性添加到其信息（metadata) 中。然后, 该文章将输出到 draft 文件夹, 而不在索引页上或任何类别或标记页上列出。

如果您想要所有文章自动作为草稿发布 (在文章完成前不会意外地发布), 请在 DEFAULT\_METADATA 中包括以下状态:

DEFAULT\_METADATA = {

'status': 'draft',

}

若要在以上情况，即设置了默认状态为 “草稿” 时发布帖子, 请更新帖子的信息以包括Status: published。

### 学习小结

由于对rst格式并不熟悉，重点还是看了看怎么使用目录结构以及markdown。   
文章中应该包括的信息有：

| **标记** | **内容** |
| --- | --- |
| Title | 标题 |
| Date | yyyy-mm-dd hh:mm |
| Modified | 修改时间，格式同上 |
| Tags | 标签 |
| Slug | Slug用来指定文件名 |
| Authors / Author | 作者，作者 |
| Summary | 概要 |

链接是个重要的东西，会影响build以后的目录结构。让我欣喜的是居然还能够提供附件，这个比我预期要好很多。   
多篇文章指向同一个文件那里没有看懂怎么回事，只能待将来实操多实验几次来总结了。

## Publish Your Site

### Site generation 网站生成

一旦安装了 Pelican , 并且您有一些内容 (例如, 以Markdown 或 reST 格式), 您可以通过 Pelican 命令将内容转换为 HTML, 并指定内容的路径和设置文件的路径 (可选):

pelican /path/to/your/content/ [ -s path/to/your/settings.py]

上面的命令将使用默认的主题生成一个简单的站点，并将其保存在output/文件夹中。默认主题由非常简单的 HTML 组成, 没有样式, 因此人们可以使用它作为创建他们自己的主题的基础。

在处理单个项目或页面时, 使用 --write-selected可以只生成与该内容对应的文件。如下:

pelican --write-selected output/posts/my-post-title.html

NOTE：必须指定生成的输出文件的路径, 而不是源内容。要确定输出文件路径, 请使用 --debug标志来确定正确的文件名和位置。如果需要, --write-selected 可以采取逗号分隔的路径列表, 也可以配置为设置。(请参见: [Writing only selected content](http://docs.getpelican.com/en/stable/settings.html#writing-only-selected-content))

您还可以运行 Pelican时添加使用 -r 或 --autoreload 选项，告诉 Pelican 监视您的修改, 而不是在每次希望看到更改时手动重新运行它。

Pelican 有其他可用的命令行开关。请查看 “帮助” 以查看可以使用的所有选项:

pelican --help

生成网站

pelican content

**生成页面**

pipenv run pelican content

执行make html命令进行博客的生成

make html

(pelican e:/blog/content/ -o e:/blog/output -s e:/blog/pelicanconf.py)

Title等信息一定要在第一行开始，不然会提示找不到。

## Viewing the generated files 预览

Pelican 生成的是静态页面，你并不需要什么特殊的软件来预览。你可以直接用浏览器来打开页面查看：

firefox output/index.html

浏览器不能定位CSS，所以上面的方法会让你困惑。你可以使用 Python 运行一个简单的 web 服务，当使用 Python 3时：

cd output

python -m http.server

一旦服务开始运行，你可以在 http://localhost:8000/预览你生成的静态站。

打开一个新的 terminal ，切换目录到output，然后开启 Pelican 的服务器：

cd /x..xx/yoursite/output

pipenv run python -m pelican.server

make html命令将把刚才写的博文生成html，存放到output目录下，如果你没有make命令，也可执行第二行的pelican命令。接着执行make serve开启测试服务器

make serve

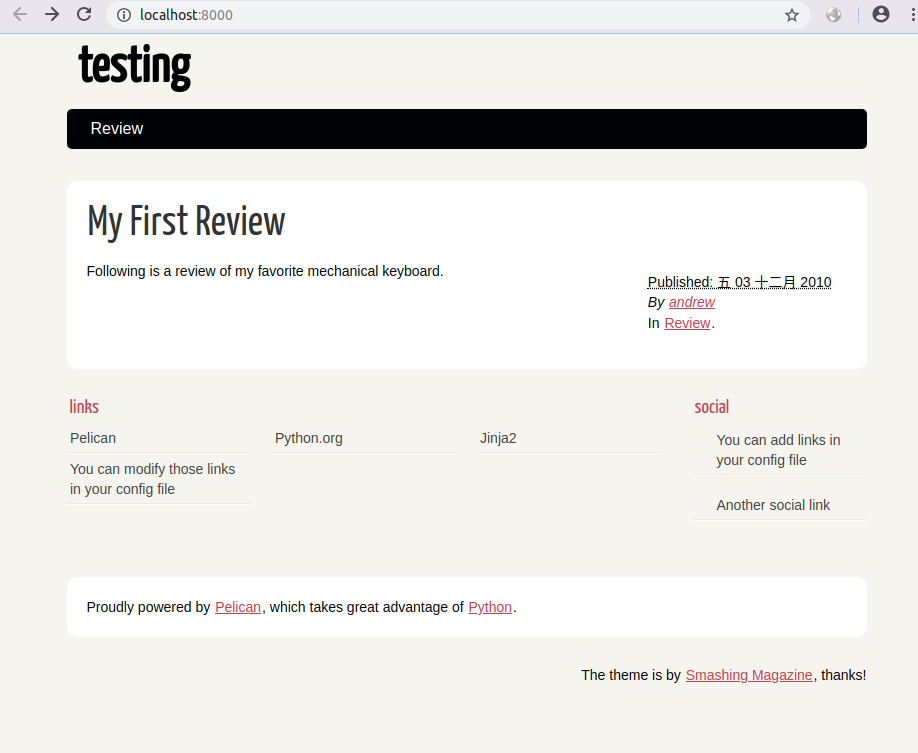
(cd e:/blog/output/ && python -m pelican.server)

make serve命令也可由第二行替代，在浏览器中输入http://localhost:8000即可看到博文效果。

浏览http://localhost:8000/页面预览以下。

pelican --listen

访问：<http://localhost:8000/>。



图片.png

## Deployment 部署

生成站点后, 在本地开发环境中预览它, 并准备将其部署到生产中, 您可能首先使用您定义的任何特定于生产的设置 (例如, 分析提要等) 重新生成站点:

pelican content -s publishconf.py

若要以 pelicanconf.py 作为发布配置的基础, 您可以通过在publishconf.py中导入 pelicanconf 设置:

from pelicanconf import \*

* 1

如果你是通过运行 pelican-quickstart生成的 publishconf.py，以上以默认存在于文件中。

部署站点的步骤将取决于它将部署在何处。如果您对运行 Nginx 或 Apache 的服务器具有 SSH 访问权限, 则可以使用 rsync 工具来传输您的站点文件：

rsync -avc --delete output/ host.example.com:/var/www/your-site/

* 1

还有许多其他部署选项, 其中一些可以在首次通过 pelican-quickstart 命令设置站点时进行配置。有关通过 GitHub 页面发布的详细信息, 请参阅[提示页面](http://docs.getpelican.com/en/stable/tips.html)。

## github部署

比如<https://github.com/china-testing/china-testing.github.io>，这个仓库名要求为[username.github.io](http://username.github.io)。

上传代码，就可以访问你的网站了。

#!/usr/bin/env bash

DATE=$(date +%Y%m%d)

cd /home/andrew/code/yoursite

rm -rf output/\*

pelican content

cd /home/andrew/code/china-testing.github.io

git rm -rf \*

cp -rf /home/andrew/code/yoursite/output/\* .

git add \*

git commit -m $DATE

git push

网站示例：　<https://china-testing.github.io/>

## Automation 自动化

虽然 pelian 命令是生成站点的规范化方法, 但自动化工具可以用来简化生成和发布流程。在 pelican-quickstart 过程中询问的问题之一是您是否希望自动生成和发布站点。如果您对该问题回答 “yes”, 则会在项目的根目录中生成 fabfile.py 和 Makefile 文件。这些文件预先填写了从 pelican-quickstart 过程中提供的其他答案收集的某些信息, 这意味着作为一个起点, 应自定义以适合您的特定需求和使用模式。如果您发现这些自动化工具作用有限, 这些文件可以随时删除, 不会影响 pelican 的使用。

下面是 “wrap” pelican 的自动化工具, 可以简化生成、预览和上传站点的过程。

### Fabric

[Fabric](http://fabfile.org/) 的优点是它是用 Python 编写的, 因此可以在多种环境中使用。缺点是必须单独安装。使用下面的命令来安装 Fabric , 您有可能需要用到 “sudo”：

pip install Fabric

* 1

Note: **Windows 需要安装 PyCrypto**   
  
Fabric 依赖 PyCrypto，如果您的系统没有 C 编译器, 则很难安装。对于 Windows 用户, 在安装Fabric之前, 请使用   
easy\_install http://www.voidspace.org.uk/downloads/pycrypto26/pycrypto-2.6.win32-py2.7.exe   
根据[此 StackOverflow 建议](http://stackoverflow.com/a/11405769/6364)您使用 Python 2.7 的 Win32 版本和PyCrypto比64位版本的Win64—even更容易成功。

请花些时间打开在项目根目录中生成的 fabfile.py 文件。您将看到许多命令, 其中每一个都可以根据您的需要来重命名、删除和/或自定义。使用 out-of-the-box 配置, 您可以通过以下方式生成站点：

fab build

* 1

如果您希望 Pelican 在每次检测到更改时自动重新生成站点 (这在本地测试时很方便), 请改用以下命令:

fab regenerate

* 1

运行本地服务以供 <http://localhost:8000/>预览:

fab serve

* 1

如果在 pelican-quickstart中, 当被问及是否要通过 ssh 上传站点时回答 “是”, 可以使用以下命令通过 ssh 通过 rsync 发布站点:

fab publish

* 1

这些只是一些默认情况下可用的命令, 所以请随时浏览 fabfile.py。更重要的是, 不要犹豫自定义 fabfile.py , 以满足您的特定需求和喜好。

### Make

当您在 pelican-quickstart对相关问题说 “是” 时, 也会自动为您创建 Makefile。这种方法的优点是, 使命令被内置到大多数 POSIX 系统中, 因此不需要安装其他任何东西来使用它。缺点是, 非 POSIX 系统 (例如, Windows) 不包括 make, 并且在这些系统上安装它可能是一个艰巨的任务。

如果你想使用 make根据pelicanconf.py生成你的站：

make html

* 1

利用publishconf.py里的设置将生成站点布置到生产环境：

make publish

* 1

如果你希望当文件更改后不用手动生成，而是 Pelican 自动监控处理，可以：

make regenerate

* 1

提供 <http://localhost:8000> 预览服务：

make serve

* 1

通常情况下, 您需要在两个单独的终端中运行 make regenerate 和 make serve , 但您可以通过以下方式同时运行两个：

make devserver

* 1

上述命令将同时运行 Pelican 的再生模式, 以及输出在 <http://localhost:8000> 的服务。完成对更改的测试后, 应通过以下方式停止开发服务器：

./develop\_server.sh stop

* 1

当您准备发布网站时, 您可以通过在 pelican-quickstart选择的方法将其上传。对于本例, 我们将在 ssh 上使用 rsync:

make rsync\_upload

* 1

（默认的 Makefiel 和 devserver.sh 使用 python 和 pelican 来完成其任务。如果要使用不同的可执行文件 (如 python3), 则可以分别设置 py 和 PELICAN 环境变量来重写默认的可执行文件名称。）

## 更新pelican

pipenv install --upgrade pelican

* 1

## 发布博客站点

[Publish your site](http://docs.getpelican.com/en/3.5.0/publish.html#publish-your-site) 有三种方法可以发布博客： - pelican命令 - make 详细用法查看Makefile文件 - fabric

生成博客站点

[Site generation](http://docs.getpelican.com/en/3.5.0/publish.html#site-generation)

|  |  |
| --- | --- |
| 1 | pelican /path/to/your/content/ [-s path/to/your/settings.py] |

或

|  |  |
| --- | --- |
| 1 | make html |

生成的站点放在output/目录下。

预览生成的站点

For Python 2, run:

|  |  |
| --- | --- |
| 1  2 | cd output  python -m SimpleHTTPServer |

For Python 3, run:

|  |  |
| --- | --- |
| 1  2 | cd output  python -m http.server |

或

|  |  |
| --- | --- |
| 1 | make serve |

浏览<http://localhost:8000/>地址预览效果.

部署博客站点

[Deployment](http://docs.getpelican.com/en/3.5.0/publish.html#deployment) 原理：将output目录下生成的站点部署到自己的github pages.

自动化操作

* Fabric

|  |  |
| --- | --- |
| 1  2  3  4  5 | pip install Fabric  fab build  fab regenerate  fab serve  fab publish |

* Make

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | make html  make regenerate  make serve  make devserver  make stopserver  ./develop\_server.sh stop |

备注：发布的简单流程：pelican content生成页面至output目录，然后git push将output目录推送到github站点的gh-pages分支，即可自动渲生成染博客。

参考资料： <http://www.xycoding.com/articles/2013/11/21/blog-create/>

## Pelican Plugins

Beginning with version 3.0, Pelican supports plugins. Plugins are a way to add features to Pelican without having to directly modify the Pelican core. Starting with 3.2, all plugins (including the ones previously in the core) have been moved to this repository, so this is the central place for all plugins.

### How to use plugins

The easiest way to install and use these plugins is to clone this repo:

git clone --recursive https://github.com/getpelican/pelican-plugins

and activate the ones you want in your settings file:

PLUGIN\_PATHS = ['path/to/pelican-plugins']

PLUGINS = ['assets', 'sitemap', 'gravatar']

PLUGIN\_PATHS can be a path relative to your settings file or an absolute path.

Alternatively, if plugins are in an importable path, you can omit PLUGIN\_PATHS and list them:

PLUGINS = ['assets', 'sitemap', 'gravatar']

or you can import the plugin directly and give that:

import my\_plugin

PLUGINS = [my\_plugin, 'assets']

### Plugin descriptions

添加评论系统

开启个人博客的原因在于分享知识，分享就需要交流，评论模块当然少不了。大多数主题默认支持Disqus。在[Disqus](https://disqus.com/)上申请帐号，按照流程Disqus会分配给你站点的Shortname，记牢Shortname，如果忘了请进入admin/settings中查看。然后同理，在pelicanconf.py添加 国内也可以选择[多说](http://duoshuo.com/)和[友言](http://www.uyan.cc/).

|  |  |
| --- | --- |
| 1 | DISQUS\_SITENAME = Shortname |

## 配置文件设置 重点

# all the following settings are \*optional\*

# HTML metadata

SITEDESCRIPTION = ''

# all defaults to True.

DISPLAY\_HEADER = True

DISPLAY\_FOOTER = True

DISPLAY\_HOME = True

DISPLAY\_MENU = True

# provided as examples, they make 'clean' urls. used by MENU\_INTERNAL\_PAGES.

TAGS\_URL = 'tags'

TAGS\_SAVE\_AS = 'tags/index.html'

AUTHORS\_URL = 'authors'

AUTHORS\_SAVE\_AS = 'authors/index.html'

CATEGORIES\_URL = 'categories'

CATEGORIES\_SAVE\_AS = 'categories/index.html'

ARCHIVES\_URL = 'archives'

ARCHIVES\_SAVE\_AS = 'archives/index.html'

# use those if you want pelican standard pages to appear in your menu

MENU\_INTERNAL\_PAGES = (

('Tags', TAGS\_URL, TAGS\_SAVE\_AS),

('Authors', AUTHORS\_URL, AUTHORS\_SAVE\_AS),

('Categories', CATEGORIES\_URL, CATEGORIES\_SAVE\_AS),

('Archives', ARCHIVES\_URL, ARCHIVES\_SAVE\_AS),

)

# additional menu items

MENUITEMS = (

('GitHub', 'https://github.com/'),

('Linux Kernel', 'https://www.kernel.org/'),

)

## pelican-themes

回到blog目录下，按如下步骤下载pelican官方主题，从里面挑选出自己喜欢的主题吧，大多数主题预览界面你可以打开这个网页[主题预览](http://pelicanthemes.com/)进行查看。不过如今pelican又新出了很多主题，所以你需看看[pelican主题开源库](https://github.com/getpelican/pelican-themes)。

* 克隆主题到本地

|  |  |
| --- | --- |
| 1 | git clone https://github.com/getpelican/pelican-themes.git |

* 使用主题 打开pelicanconf.py配置文件，更改或添加THEME为自己喜欢的主题，例如本博客所挑选的elegant，更多的配置含义请关注[官方文档](http://docs.getpelican.com/en/3.5.0/index.html)。

|  |  |
| --- | --- |
| 1 | THEME = 'pelican-themes/pelican-elegant' |

### pelican-themes 插入主题

### Description 描述

pelican-themes是 Pelican 用来管理主题的命令行工具。

#### Usage 用法

pelican-themes [-h] [-l] [-i theme path [theme path ...]]

[-r theme name [theme name ...]]

[-s theme path [theme path ...]] [-v] [-version]

* 1
* 2
* 3

#### Optional arguments 参数

| **参数** | **说明** |
| --- | --- |
| -h, –help | 显示帮助 |
| -l, –list | 列出已经安装的主题 |
| -i theme\_path, –install theme\_path | 安装主题（可多个同时安） |
| -r theme\_path, –remove theme\_path | 删除主题 |
| -s theme\_path, –symlink theme\_path | 以符号链接形式安装主题，而非复制，常用于主题开发 |
| -v, –verbose | 详细输出 |
| –version | 版本 |
| git clone https://github.com/getpelican/pelican-themes.git  打开pelicanconf.py配置文件，更改或添加THEME为自己喜欢的主题，例如本博客所挑选的gum，更多的配置含义请关注[官方文档](http://docs.getpelican.com/en/3.3.0/settings.html)。  THEME = 'pelican-themes/gum' |  |

#### Examples 例子

##### Listing the installed themes

$ pelican-themes -l

notmyidea

two-column@

simple

$ pelican-themes --list

notmyidea

two-column@

simple

可以看到主题 two-column后跟着@，表明此主题没有复制安装而是链接安装。

$ pelican-themes -v -l

/usr/local/lib/python2.6/dist-packages/pelican-2.6.0-py2.6.egg/pelican/themes/notmyidea

/usr/local/lib/python2.6/dist-packages/pelican-2.6.0-py2.6.egg/pelican/themes/two-column (symbolic link to '/home/skami/Dev/Python/pelican-themes/two-column')

/usr/local/lib/python2.6/dist-packages/pelican-2.6.0-py2.6.egg/pelican/themes/simple

##### Installing themes 安装主题

你可以安装一个或多个主题：

$ pelican-themes --install ~/Dev/Python/pelican-themes/notmyidea-cms --verbose

# pelican-themes --install ~/Dev/Python/pelican-themes/notmyidea-cms\

~/Dev/Python/pelican-themes/martyalchin \

--verbose

# pelican-themes -vi ~/Dev/Python/pelican-themes/two-column

* 1

##### Removing themes 移除主题

# pelican-themes --remove two-column

* 1

# pelican-themes -r martyachin notmyidea-cmd -v

* 1

##### Creating symbolic links 符号安装

# pelican-themes --symlink ~/Dev/Python/pelican-themes/two-column

* 1

上面的例子，two-column主题作为符号安装进了主题目录，我们可以使用它，同时我们可以开发改进主题而不用reinstall。

这在开发主题的过程中很有用：

$ sudo pelican-themes -s ~/Dev/Python/pelican-themes/two-column

$ pelican ~/Blog/content -o /tmp/out -t two-column

$ firefox /tmp/out/index.html

$ vim ~/Dev/Pelican/pelican-themes/two-column/static/css/main.css

$ pelican ~/Blog/content -o /tmp/out -t two-column

$ cp /tmp/bg.png ~/Dev/Pelican/pelican-themes/two-column/static/img/bg.png

$ pelican ~/Blog/content -o /tmp/out -t two-column

$ vim ~/Dev/Pelican/pelican-themes/two-column/templates/index.html

$ pelican ~/Blog/content -o /tmp/out -t two-column

##### Doing several things an once

--install, --remove和 --symlink可以同时出现在一个命令中：

# pelican-themes --remove notmyidea-cms two-column \

--install ~/Dev/Python/pelican-themes/notmyidea-cms-fr \

--symlink ~/Dev/Python/pelican-themes/two-column \

--verbose

### pelican主题更换

首先在[pelican主题开源库](https://github.com/getpelican/pelican-themes)中找到自己想要的主题，本次替换的主题是[pelican-bootstrap3](https://github.com/getpelican/pelican-themes/tree/master/pelican-bootstrap3)，这个主题的配置好像跟其他的不一样，需要多配置些东西。

在根目录下的pelicanconf.py文件中添加或更改自己喜欢的主题。

THEME = 'pelican-themes/pelican-bootstrap3'

PLUGIN\_PATHS = ['pelican-plugins', ]

PLUGINS = ['i18n\_subsites', ]

JINJA\_ENVIRONMENT = {'extensions': ['jinja2.ext.i18n'],}

1. 然后clone <https://github.com/getpelican/pelican-plugins> 到根目录下(本主题所需)，配置已经在上一步骤完成了。
2. 最后在命令行输入pip install python-gettext安装成功后再输入pelican content 即可输入至output文件夹。

`make html`编译网页

### git提交代码

先输入`git status`可查看修改状态。

git add.

git pull origin master

git commit -m '注释'

git push origin master

最后在网页打开自己的github博客地址即可查看。 例:<https://udbbbn.github.io/> (现已替换成jekyll模板) [参考链接1](https://github.com/getpelican/pelican-themes/issues/507) [参考链接2](https://github.com/getpelican/pelican-themes/tree/master/pelican-bootstrap3)

### 学习笔记

假设我们将主题放在 ~/pelican-themes文件夹下，先将主题拿到本地：

git clone --recursive https://github.com/getpelican/pelican-themes ~/pelican-themes

* 1

现在在本地我们拥有了pelican-themes的分支。安装主题，在blog文件夹下运行：

pipenv run pelican-themes -i ~/pelican-themes/themes-name

pelican-themes -i tuxlite\_tbs

还记得吗？我为我的环境配置了pipenv。所以运行的时候要注意。   
安装完成后用pipenv run pelican-themes -l查看主题   
选用主题时，修改 pelicanconf.py文件，添加：

THEME = "themename"

保存文件并生成站点。

开发主题的时候注意修改~/pelican-themes下的文件，output文件夹下的修改在站点生成的时候会被覆盖.

## Pelican参考资源

### Pelican4.1.0博客系统官方文档中文版

https://pelican-docs-latest-cn.readthedocs.io/zh\_CN/latest/

https://github.com/elvynlee/pelican\_docs\_cn

https://github.com/getpelican/pelican

1. [使用github搭建网站](http://blog.csdn.net/pipisorry/article/details/51707366)   
2. [一步步在GitHub上创建博客主页](http://www.pchou.info/ssgithubPage/2013-01-03-build-github-blog-page-01.html)   
3. [Github Help](https://help.github.com/categories/customizing-github-pages/)   
4. [pelican 的官方页](https://blog.getpelican.com/)   
5. [pelican 中文版](http://pelican-zh.readthedocs.io/en/latest/zh-cn/)，资料太古老，只做参考，不要实操   
6. [Pelican＋Github博客搭建详细教程](https://www.cnblogs.com/cciejh/archive/2016/02/04/blog_building.html)

Github:<https://github.com/>

Github Ppages:<http://pages.github.com/>

git:<http://git-scm.com/>

python:<http://www.python.org/>

pip:<https://pypi.python.org/pypi>

pelican:<http://blog.getpelican.com/>

markdown:<http://daringfireball.net/projects/markdown/syntax>

## Pelican Plugins资源

| **Plugin** | **Description** |
| --- | --- |
| Ace Editor | Replace default **<code>** by an [Ace](https://ace.c9.io) code editor with settings configure on pelicanconf.py. |
| Always modified | Copy created date metadata into modified date for easy "latest updates" indexes |
| AsciiDoc reader | Use AsciiDoc to write your posts. |
| Asset management | Use the Webassets module to manage assets such as CSS and JS files. |
| Auto Pages | Generate custom content for generated Author, Category, and Tag pages (e.g. author biography) |
| Backref Translate | Add a new attribute (is\_translation\_of) to every article/page (which is a translation) pointing back to the original article/page which is being translated |
| Better code line numbers | Allow code blocks with line numbers to wrap |
| Better code samples | Wraps table blocks with div > .hilitewrapper > .codehilitetable class attribute, allowing for scrollable code blocks. |
| Better figures/samples | Adds a style="width: ???px; height: auto;" attribute to any <img> tags in the content |
| bootstrap-rst | Provides most (though not all) of Bootstrap's features as rst directives |
| bootstrapify | Automatically add bootstraps default classes to your content |
| Category Order | Order categories (and tags) by the number of articles in that category (or tag). |
| CJK auto spacing | Inserts spaces between Chinese/Japanese/Korean characters and English words |
| Clean summary | Cleans your summary of excess images |
| Code include | Includes Pygments highlighted code in reStructuredText |
| Collate content | Makes categories of content available to the template as lists through a collations attribute |
| Creole reader | Allows you to write your posts using the wikicreole syntax |
| CSS HTML JS Minify | Minifies all CSS, HTML and JavaScript files in the output path after site generation. |
| Custom article URLs | Adds support for defining different default URLs for different categories |
| CTags generator | Generates a "tags" file following the CTags in the "content/" directory, to provide autocompletion for code editors that support it. |
| Dateish | Treat arbitrary metadata fields as datetime objects |
| Dead Links | Manage dead links (website not available, errors such as 403, 404) |
| Disqus static comments | Adds a disqus\_comments property to all articles. Comments are fetched at generation time using disqus API |
| Encrypt content | Password protect pages and articles |
| Events | Add event start, duration, and location info to post metadata to generate an iCalendar file |
| Extract table of content | Extracts table of contents (ToC) from article.content |
| Figure References | Provides a system to number and references figures |
| Filetime from Git | Uses Git commit to determine page date |
| Filetime from Hg | Uses Mercurial commit to determine page date |
| Footer Insert | Add standardized footer (e.g., author information) at end of every article |
| GA Page View | Display Google Analytics page views on individual articles and pages |
| Gallery | Allows an article to contain an album of pictures |
| Gist directive | This plugin adds a gist reStructuredText directive. |
| GitHub wiki | Converts a flat github wiki into a structured read only wiki on your site |
| GitHub activity | On the template side, you just have to iterate over the github\_activity variable |
| Global license | Allows you to define a LICENSE setting and adds the contents of that license variable to the article's context |
| Glossary | Adds a variable containing definitions extracted from definition lists in articles and pages. This variable is visible to all page templates. |
| Goodreads activity | Lists books from your Goodreads shelves |
| GooglePlus comments | Adds GooglePlus comments to Pelican |
| Gravatar | Assigns the author\_gravatar variable to the Gravatar URL and makes the variable available within the article's context |
| Gzip cache | Enables certain web servers (e.g., Nginx) to use a static cache of gzip-compressed files to prevent the server from compressing files during an HTTP call |
| Headerid | This plugin adds an anchor to each heading so you can deeplink to headers in reStructuredText articles. |
| HTML entities | Allows you to enter HTML entities such as &copy;, &lt;, &#149; inline in a RST document |
| HTML tags for rST | Allows you to use HTML tags from within reST documents |
| I18N Sub-sites | Extends the translations functionality by creating internationalized sub-sites for the default site |
| ical | Looks for and parses an .ics file if it is defined in a given page's calendar metadata. |
| Image Process | Automates the processing of images based on their class attributes |
| Interlinks | Lets you add frequently used URLs to your markup using short keywords |
| Jinja2 Content | Allows the use of Jinja2 template code in articles, including include and import statements. Replacement for pelican-jinja2content. |
| JPEG Reader | Create image gallery pages based on content of JPEG metadata |
| Just table | Allows you to easily create and manage tables. You can embed the tables into posts with a simple way. |
| Libravatar | Allows inclusion of user profile pictures from libravatar.org |
| Link Class | Allows the insertion of class attributes into generated <a> elements (Markdown only) |
| Linker | Allows the definition of custom linker commands in analogy to the builtin {filename}, {attach}, {category}, {tag}, {author}, and {index} syntax |
| Liquid-style tags | Allows liquid-style tags to be inserted into markdown within Pelican documents |
| Load CSV | Adds csv Jinja tag to display the contents of a CSV file as an HTML table |
| Markdown Inline Extend | Enables you to add customize inline patterns to your markdown |
| Markdown-metaYAML | Pelican reader to enable YAML-style metadata in markdown articles |
| Math Render | Gives pelican the ability to render mathematics |
| Mbox Reader | Generate articles automatically via email, given a path to a Unix mbox |
| More Categories | Multiple categories per article; nested categories (foo/bar, foo/baz) |
| Multi Neighbors | Adds a list of newer articles and a list of older articles to every article's context. |
| Multi parts posts | Allows you to write multi-part posts |
| Neighbor articles | Adds next\_article (newer) and prev\_article (older) variables to the article's context |
| Open graph | Generates Open Graph tags for your articles |
| Optimize images | Applies lossless compression on JPEG and PNG images |
| Org Reader | Create posts via Emacs Orgmode files |
| Page View | Pull page view count from Google Analytics. |
| Panorama | Creates charts from posts metadata |
| PDF generator | Automatically exports articles and pages as PDF files |
| PDF Images | If an img tag contains a PDF, EPS or PS file as a source, this plugin generates a PNG preview which will then act as a link to the original file. |
| Pelican Cite | Produces inline citations and a bibliography in articles and pages, using a BibTeX file. |
| Pelican Comment System | Allows you to add static comments to your articles |
| pelican-ert | Allows you to add estimated reading time of an article |
| Pelican-flickr | Brings your Flickr photos & sets into your static website |
| Pelican Genealogy | Add surnames and people so metadata and context can be accessed from within a theme to provide surname and person pages |
| Pelican Gist tag | Easily embed GitHub Gists in your Pelican articles |
| Pelican Github Projects | Embed a list of your public GitHub projects in your pages |
| pelican\_javascript | Allows you to embed Javascript and CSS files into individual articles |
| Pelican Jinja2Content | Allows the use of Jinja2 template code in articles, including include and import statements |
| Pelican Link Class | Set class attribute of <a> elements according to whether the link is external or internal |
| Pelican Meetup Info | Include your Meetup.com group and event information on generated pages and articles |
| Pelican Page Hierarchy | Creates a URL hierarchy for pages that matches the filesystem hierarchy of their sources |
| Pelican Page Order | Adds a page\_order attribute to all pages if one is not defined. |
| Pelican Themes Generator | Generates theme screenshots from the Pelican Themes repository |
| pelican-rdf | Allows the processing of .rdf vocabularies, and the generation of a lightweight documentation. |
| pelican-toc | Generates a Table of Contents and make it available to the theme via article.toc |
| Pelican Vimeo | Enables you to embed Vimeo videos in your pages and articles |
| Pelican YouTube | Enables you to embed YouTube videos in your pages and articles |
| pelicanfly | Lets you type things like i ♥ :fa-coffee: in your Markdown documents and have it come out as little Font Awesome icons in the browser |
| Photos | Add a photo or a gallery of photos to an article, or include photos in the body text. Resize photos as needed. |
| permalink | Enables a kind of permalink using html redirects. |
| Pin to top | Pin Pelican's article(s) to top "Sticky article" |
| PlantUML | Allows you to define UML diagrams directly into rst documents using the great PlantUML tool |
| Post Revision | Extract article and page revision information from Git commit history |
| Post statistics | Calculates various statistics about a post and store them in an article.stats dictionary |
| Random article | Generates a html file which redirect to a random article |
| Read More link | Inserts an inline "read more" or "continue" link into the last html element of the object summary |
| Readtime | Adds article estimated read time calculator to the site, in the form of '<n> minutes'. |
| Related posts | Adds the related\_posts variable to the article's context |
| Render Math | Render mathematics in content via the MathJax Javascript engine |
| Replacer | Replace a text of a generated HTML |
| Representative image | Extracts a representative image (i.e, featured image) from the article's summary or content |
| RMD Reader | Create posts via knitr RMarkdown files |
| Section number | Adds section numbers for article headers, in the form of 2.3.3 |
| Series | Groups related articles into a series |
| Shaarli poster | Upload newly redacted articles onto a specified [Shaarli](https://github.com/shaarli/Shaarli) instance. |
| Share post | Creates share URLs of article |
| Shortcodes | Easy and explicit inline jinja2 macros |
| Show Source | Place a link to the source text of your posts. |
| Similar Posts | Adds a list of similar posts to every article's context. |
| Simple footnotes | Adds footnotes to blog posts |
| Sitemap | Generates plain-text or XML sitemaps |
| Slim | Render theme template files via Plim, a Python port of Slim, instead of Jinja |
| Static comments | Allows you to add static comments to an article |
| Subcategory | Adds support for subcategories |
| Sub parts | Break a very long article in parts, without polluting the timeline with lots of small articles. |
| Summary | Allows easy, variable length summaries directly embedded into the body of your articles |
| tag\_cloud | Provides a tag\_cloud |
| Textile Reader | Adds support for Textile markup |
| Thumbnailer | Creates thumbnails for all of the images found under a specific directory |
| Tipue Search | Serializes generated HTML to JSON that can be used by jQuery plugin - Tipue Search |
| Touch | Does a touch on your generated files using the date metadata from the content |
| Twitter Bootstrap | Defines some rst directive that enable a clean usage of the twitter bootstrap CSS and Javascript components |
| txt2tags\_reader | Reader that renders txt2tags markup in content |
| Unity WebGL | Easily embed Unity3d games into posts and pages |
| Video Privacy Enhancer | Increases user privacy by stopping YouTube, Google, et al from placing cookies via embedded video |
| Webring | Add a webring to your site from a list of web feeds (e.g. RSS/Atom) |
| W3C validate | Submits generated HTML content to the W3C Markup Validation Service |
| Yuicompressor | Minify CSS and JS files on building step |

Please refer to the Readme file in a plugin's folder for detailed information about that plugin.

## Pelican主题资源

### 资源一览

pelican 的主题在 <https://github.com/getpelican/pelican-themes>，页面中有使用说明。选取主题的时候可以看<http://www.pelicanthemes.com/>页面挑选。

|  |  |  |
| --- | --- | --- |
| 主题名称 | GitHub | 维护状态 |
| buruma | https://github.com/ivanhercaz/buruma.git | 2019 |
|  |  |  |
|  |  |  |
| niu-x2 | https://github.com/wilbur-ma/niu-x2 | (中文的，不用调整CSS）最后更新2013年 |
|  |  |  |

[attila](https://github.com/arulrajnet/attila/tree/02dcad911ba1eb2d797a79ec008a810d89a2fde1),

[backdrop](https://github.com/getpelican/pelican-themes/tree/master/backdrop),

[blueidea](https://github.com/blueicefield/pelican-blueidea/tree/8f11c0e3b4b8e9ef45d1243b0175db91b7b42ba8),

[clean-blog](https://github.com/gilsondev/pelican-clean-blog/tree/ea156f8f1741e473bc0ab848b7c8898112d6ffb5),

,

[voidy-bootstrap](https://github.com/robulouski/voidy-bootstrap/tree/83f4d802710bbfa20123a9b7f1921424e292ef53),

### 主题gum修改

[**现状**](https://www.iloveanan.com/pelicanzhu-ti-xiu-gai.html#_1)

现在使用的主题是[gum](https://github.com/getpelican/pelican-themes/tree/master/gum)，不喜欢的地方有：

1. 侧边栏：没有添加pages，但有Pages标题；每篇文章都添加tag，但Tag下没有内容
2. Home导航太单调，希望可以有各个分类的链接

[**修改**](https://www.iloveanan.com/pelicanzhu-ti-xiu-gai.html#_2)

因为这个主题是很早就选定的，所以一开始我是直接去看gum的源码，内容倒也不多，两个文件夹，一个css一个是html模版。但是，这样的方式是错误滴～正确的方式是先过一遍pelican的使用指南，然后再看gum的使用指南。中间遇到的问题就挑重点记录了。

[**菜单显示分类**](https://www.iloveanan.com/pelicanzhu-ti-xiu-gai.html#_3)

pelican内置变量 DISPLAY\_CATEGORIES\_ON\_MENU，决定是否在菜单中显示分类。此外，如果希望同时显示Pages，除了需要在本地添加pages外，还需要在配置文件中添加

MENUITEMS = (

('About', '/pages/About.html'),

)

添加pages的方法是，现在content所在文件夹下新建 pages，然后在其中创建形如article的文件。这里，我使用的是md格式，那么我就创建一个 About.md文件。  
此外，还修改了category.html，将以下部分删除，因为我希望点击分类的时候和首页一样，直接显示文章列表，而不是先出现如下的字样：

{% **block** content\_title %}

**<h2>**Articles in the {{ category }} category**</h2>**

{% **endblock** %}

[**侧边栏**](https://www.iloveanan.com/pelicanzhu-ti-xiu-gai.html#_4)

我希望分类能同时在menu和侧边栏出现，但是查看 sidebar.html 发现，两者分别是 if 和else中，因此将其中的else内容移除。  
同时发现tag下的内容为空，但是html模版文件中有tag内容。此时怀疑是否是变量使用出了问题，排查了一遍将目光定位到

{% **for** tag in tag\_cloud %}

搜索了下，tag\_cloud是一个[插件](http://docs.getpelican.com/en/3.6.3/faq.html#my-tag-cloud-is-missing-broken-since-i-upgraded-pelican)。我的工程文件下已有 plugins 文件夹，赶紧pull了一下，更新到了tag\_cloud。大喜，在配置文件中添加了tag\_cloud的属性，build后发现tag还是依次向下逐个排列，即使我修改模版文件添加了 <ul class="tagcloud">。后来在tag\_cloud库的readme中发现原因：需要将tag\_cloud.py文件直接放置在plugins下，同时要添加相应的css。

现在的形式，差强人意。以后还会陆续修改，借此熟悉pelican和html/css。

### 主题pelican-alchemy修改

https://github.com/nairobilug/pelican-alchemy

### 主题XXXX修改

# Hexo

## 什么是 Hexo？

Hexo 是一个快速、简洁且高效的博客框架。Hexo 使用 [Markdown](http://daringfireball.net/projects/markdown/)（或其他渲染引擎）解析文章，在几秒内，即可利用靓丽的主题生成静态网页。

## 安装

### 安装前提

需要先安装下列应用程序即可：

* [Node.js](http://nodejs.org/) (Node.js 版本需不低于 8.10，建议使用 Node.js 10.0 及以上版本)
* [Git](http://git-scm.com/)

下载node.js并安装（官网下载安装），安装过程和普通软一样，默认会安装npm。 控制台输入 node -v，控制台打印出对应Node版本就说明安装成功了

### 安装 Git

* Windows：下载并安装 [git](https://git-scm.com/download/win).
* Mac：使用 [Homebrew](http://mxcl.github.com/homebrew/), [MacPorts](http://www.macports.org/) 或者下载 [安装程序](http://sourceforge.net/projects/git-osx-installer/)。
* Linux (Ubuntu, Debian)：sudo apt-get install git-core
* Linux (Fedora, Red Hat, CentOS)：sudo yum install git-core

**Mac 用户**

如果在编译时可能会遇到问题，请先到 App Store 安装 Xcode，Xcode 完成后，启动并进入 **Preferences -> Download -> Command Line Tools -> Install** 安装命令行工具。

**Windows 用户**

对于中国大陆地区用户，可以前往 [淘宝 Git for Windows 镜像](https://npm.taobao.org/mirrors/git-for-windows/) 下载 git 安装包。

### 安装 Node.js

Node.js 为大多数平台提供了官方的 [安装程序](https://nodejs.org/en/download/)。对于中国大陆地区用户，可以前往 [淘宝 Node.js 镜像](https://npm.taobao.org/mirrors/node) 下载。

其它的安装方法：

* Windows：通过 [nvs](https://github.com/jasongin/nvs/)（推荐）或者[nvm](https://github.com/nvm-sh/nvm) 安装。
* Mac：使用 [Homebrew](https://brew.sh/) 或 [MacPorts](http://www.macports.org/) 安装。
* Linux（DEB/RPM-based）：从 [NodeSource](https://github.com/nodesource/distributions) 安装。
* 其它：使用相应的软件包管理器进行安装，可以参考由 Node.js 提供的 [指导](https://nodejs.org/en/download/package-manager/)

对于 Mac 和 Linux 同样建议使用 nvs 或者 nvm，以避免可能会出现的权限问题。

**Windows 用户**

使用 Node.js 官方安装程序时，请确保勾选 **Add to PATH** 选项（默认已勾选）

**For Mac / Linux 用户**

如果在尝试安装 Hexo 的过程中出现 EACCES 权限错误，请遵循 [由 npmjs 发布的指导](https://docs.npmjs.com/resolving-eacces-permissions-errors-when-installing-packages-globally) 修复该问题。强烈建议 **不要** 使用 root、sudo 等方法覆盖权限

### 安装 Hexo

所有必备的应用程序安装完成后，即可使用 npm 安装 Hexo。

|  |
| --- |
| $ npm install -g hexo-cli |

### 进阶安装和使用

对于熟悉 npm 的进阶用户，可以仅局部安装 hexo 包。

|  |
| --- |
| $ npm install hexo |

安装以后，可以使用以下两种方式执行 Hexo：

1. npx hexo <command>
2. 将 Hexo 所在的目录下的 node\_modules 添加到环境变量之中即可直接使用 hexo <command>：

|  |
| --- |
| echo 'PATH="$PATH:./node\_modules/.bin"' >> ~/.profile |

## 建站

安装 Hexo 完成后，请执行下列命令，Hexo 将会在指定文件夹中新建所需要的文件。

|  |
| --- |
| $ hexo init <folder> $ cd <folder> $ npm install |

新建完成后，指定文件夹的目录如下：

.  
├── \_config.yml   
├── package.json  
├── scaffolds  
├── source  
| ├── \_drafts  
| └── \_posts  
└── themes

|  |
| --- |
|  |

### Hexo文件夹介绍

node\_modules 依赖包

scaffolds 命令生成文章等的模板

source 用命令创建的各种文章

themes 主题

\_config.yml 整个博客的配置

package.json 项目所需模块项目的配置信息

### config.yml

网站的 [配置](https://hexo.io/zh-cn/docs/configuration) 信息，您可以在此配置大部分的参数。

### package.json

应用程序的信息。[EJS](https://ejs.co/), [Stylus](http://learnboost.github.io/stylus/) 和 [Markdown](http://daringfireball.net/projects/markdown/) renderer 已默认安装，您可以自由移除。

package.json

|  |
| --- |
| {  "name": "hexo-site",  "version": "0.0.0",  "private": true,  "hexo": {  "version": ""  },  "dependencies": {  "hexo": "^3.8.0",  "hexo-generator-archive": "^0.1.5",  "hexo-generator-category": "^0.1.3",  "hexo-generator-index": "^0.2.1",  "hexo-generator-tag": "^0.2.0",  "hexo-renderer-ejs": "^0.3.1",  "hexo-renderer-stylus": "^0.3.3",  "hexo-renderer-marked": "^0.3.2",  "hexo-server": "^0.3.3"  } } |

### scaffolds

[模版](https://hexo.io/zh-cn/docs/writing) 文件夹。当您新建文章时，Hexo 会根据 scaffold 来建立文件。

Hexo的模板是指在新建的文章文件中默认填充的内容。例如，如果您修改scaffold/post.md中的Front-matter内容，那么每次新建一篇文章时都会包含这个修改。

### source

资源文件夹是存放用户资源的地方。除 \_posts 文件夹之外，开头命名为 \_ (下划线)的文件 / 文件夹和隐藏的文件将会被忽略。Markdown 和 HTML 文件会被解析并放到 public 文件夹，而其他文件会被拷贝过去。

### themes

[主题](https://hexo.io/zh-cn/docs/themes) 文件夹。Hexo 会根据主题来生成静态页面。

1. npm install hexo-cli -g
2. hexo init blog
3. cd blog
4. npm install
5. hexo server

打开浏览器，输入[http://localhost:4000](https://link.juejin.im/?target=http%3A%2F%2Flocalhost%3A4000) ，看到以下页面说明Hexo环境安装成功。

### [Hexo常用命令](https://link.juejin.im/?target=https%3A%2F%2Fhexo.io%2Fzh-cn%2Fdocs%2Fcommands.html)

npm install hexo -g *#安装*

npm update hexo -g *#升级*

hexo init *#初始化*

hexo n "我的博客" == hexo new "我的博客" *#新建文章*

hexo p == hexo publish

hexo g == hexo generate*#生成*

hexo s == hexo server *#启动服务预览*

2. hexo d == hexo deploy*#部署*
4. hexo server *#Hexo 会监视文件变动并自动更新，您无须重启服务器。*
6. hexo server -s *#静态模式*
8. hexo server -p 5000 *#更改端口*
10. hexo server -i 192.168.1.1 *#自定义 IP*
12. hexo clean *#清除缓存 网页正常情况下可以忽略此条命令*
14. hexo g *#生成静态网页*
16. hexo d *#开始部署*

## 配置

可以在 \_config.yml 中修改大部分的配置。

### 网站

| **参数** | **描述** |
| --- | --- |
| title | 网站标题 |
| subtitle | 网站副标题 |
| description | 网站描述 |
| keywords | 网站的关键词。使用半角逗号 , 分隔多个关键词。 |
| author | 您的名字 |
| language | 网站使用的语言。对于简体中文用户来说，使用不同的主题可能需要设置成不同的值，请参考你的主题的文档自行设置，常见的有 zh-Hans和 zh-CN。 |
| timezone | 网站时区。Hexo 默认使用您电脑的时区。请参考 [时区列表](https://en.wikipedia.org/wiki/List_of_tz_database_time_zones) 进行设置，如 America/New\_York, Japan, 和 UTC 。一般的，对于中国大陆地区可以使用 Asia/Shanghai。 |

其中，description主要用于SEO，告诉搜索引擎一个关于您站点的简单描述，通常建议在其中包含您网站的关键词。author参数用于主题显示文章的作者。

### 网址

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| url | 网址 |  |
| root | 网站根目录 |  |
| permalink | 文章的 [永久链接](https://hexo.io/zh-cn/docs/permalinks) 格式 | :year/:month/:day/:title/ |
| permalink\_defaults | 永久链接中各部分的默认值 |  |
| pretty\_urls | 改写 [permalink](https://hexo.io/zh-cn/docs/variables) 的值来美化 URL |  |
| pretty\_urls.trailing\_index | 是否在永久链接中保留尾部的 index.html，设置为 false 时去除 | true |
| pretty\_urls.trailing\_html | 是否在永久链接中保留尾部的 .html, 设置为 false 时去除 (对尾部的 *index.html*无效) | true |

**网站存放在子目录**

如果您的网站存放在子目录中，例如 http://yoursite.com/blog，则请将您的 url 设为 http://yoursite.com/blog 并把 root 设为 /blog/。

例如：

|  |
| --- |
| # 比如，一个页面的永久链接是 http://example.com/foo/bar/index.html pretty\_urls:  trailing\_index: false # 此时页面的永久链接会变为 http://example.com/foo/bar/ |

### 目录

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| source\_dir | 资源文件夹，这个文件夹用来存放内容。 | source |
| public\_dir | 公共文件夹，这个文件夹用于存放生成的站点文件。 | public |
| tag\_dir | 标签文件夹 | tags |
| archive\_dir | 归档文件夹 | archives |
| category\_dir | 分类文件夹 | categories |
| code\_dir | Include code 文件夹，source\_dir 下的子目录 | downloads/code |
| i18n\_dir | 国际化（i18n）文件夹 | :lang |
| skip\_render | 跳过指定文件的渲染。匹配到的文件将会被不做改动地复制到 public 目录中。您可使用 [glob 表达式](https://github.com/micromatch/micromatch#extended-globbing)来匹配路径。 |  |

例如：

|  |
| --- |
| skip\_render: "mypage/\*\*/\*" # 将会直接将 `source/mypage/index.html` 和 `source/mypage/code.js` 不做改动地输出到 'public' 目录 # 你也可以用这种方法来跳过对指定文章文件的渲染 skip\_render: "\_posts/test-post.md" # 这将会忽略对 'test-post.md' 的渲染 |

**提示**

如果您刚刚开始接触 Hexo，通常没有必要修改这一部分的值。

### 文章

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| new\_post\_name | 新文章的文件名称 | :title.md |
| default\_layout | 预设布局 | post |
| auto\_spacing | 在中文和英文之间加入空格 | false |
| titlecase | 把标题转换为 title case | false |
| external\_link | 在新标签中打开链接 | true |
| external\_link.enable | 在新标签中打开链接 | true |
| external\_link.field | 对整个网站（site）生效或仅对文章（post）生效 | site |
| external\_link.exclude | 需要排除的域名。主域名和子域名如 www 需分别配置 | [] |
| filename\_case | 把文件名称转换为 (1) 小写或 (2) 大写 | 0 |
| render\_drafts | 显示草稿 | false |
| post\_asset\_folder | 启动 [Asset 文件夹](https://hexo.io/zh-cn/docs/asset-folders) | false |
| relative\_link | 把链接改为与根目录的相对位址 | false |
| future | 显示未来的文章 | true |
| highlight | 代码块的设置 |  |
| highlight.enable | 开启代码块高亮 | true |
| highlight.auto\_detect | 如果未指定语言，则启用自动检测 | false |
| highlight.line\_number | 显示行数 Enabling this option will also enable *wrap* option | true |
| highlight.tab\_replace | 用 n 个空格替换 tabs；如果值为空，则不会替换 tabs | '' |
| highlight.wrap | Wrap the code block in [<table>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/table) | true |
| highlight.hljs | Use the hljs-\* prefix for CSS classes | false |

**相对地址**

默认情况下，Hexo 生成的超链接都是绝对地址。例如，如果您的网站域名为 example.com,您有一篇文章名为 hello，那么绝对链接可能像这样：http://example.com/hello.html，它是**绝对**于域名的。相对链接像这样：/hello.html，也就是说，无论用什么域名访问该站点，都没有关系，这在进行反向代理时可能用到。通常情况下，建议使用绝对地址。

### 分类 & 标签

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| default\_category | 默认分类 | uncategorized |
| category\_map | 分类别名 |  |
| tag\_map | 标签别名 |  |

### 日期 / 时间格式

Hexo 使用 [Moment.js](http://momentjs.com/) 来解析和显示时间。

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| date\_format | 日期格式 | YYYY-MM-DD |
| time\_format | 时间格式 | HH:mm:ss |
| use\_date\_for\_updated | 启用以后，如果 Front Matter 中没有指定 updated， [post.updated](https://hexo.io/zh-cn/docs/variables#%E9%A1%B5%E9%9D%A2%E5%8F%98%E9%87%8F) 将会使用 date 的值而不是文件的创建时间。在 Git 工作流中这个选项会很有用 | true |

### 分页

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| per\_page | 每页显示的文章量 (0 = 关闭分页功能) | 10 |
| pagination\_dir | 分页目录 | page |

### 扩展

| **参数** | **描述** |
| --- | --- |
| theme | 当前主题名称。值为false时禁用主题 |
| theme\_config | 主题的配置文件。在这里放置的配置会覆盖主题目录下的 \_config.yml 中的配置 |
| deploy | 部署部分的设置 |
| meta\_generator | [Meta generator](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/meta#%E5%B1%9E%E6%80%A7) 标签。 值为 false 时 Hexo 不会在头部插入该标签 |

#### 包括或不包括目录和文件

在 Hexo 配置文件中，通过设置 include/exclude 可以让 Hexo 进行处理或忽略某些目录和文件夹。你可以使用 [glob 表达式](https://github.com/isaacs/minimatch) 对目录和文件进行匹配。

include and exclude options only apply to the source/ folder, whereas ignore option applies to all folders.

| **参数** | **描述** |
| --- | --- |
| include | Hexo 默认会忽略隐藏文件和文件夹（包括名称以下划线和 . 开头的文件和文件夹，Hexo 的 \_posts 和 \_data 等目录除外）。通过设置此字段将使 Hexo 处理他们并将它们复制到 source 目录下。 |
| exclude | Hexo 会忽略这些文件和目录 |
| ignore | Ignore files/folders |

举例：

|  |
| --- |
| # Include/Exclude Files/Folders include:  - ".nojekyll"  # 包括 'source/css/\_typing.css'  - "css/\_typing.css"  # 包括 'source/\_css/' 中的任何文件，但不包括子目录及其其中的文件。  - "\_css/\*"  # 包含 'source/\_css/' 中的任何文件和子目录下的任何文件  - "\_css/\*\*/\*"  exclude:  # 不包括 'source/js/test.js'  - "js/test.js"  # 不包括 'source/js/' 中的文件、但包括子目录下的所有目录和文件  - "js/\*"  # 不包括 'source/js/' 中的文件和子目录下的任何文件  - "js/\*\*/\*"  # 不包括 'source/js/' 目录下的所有文件名以 'test' 开头的文件，但包括其它文件和子目录下的单文件  - "js/test\*"  # 不包括 'source/js/' 及其子目录中任何以 'test' 开头的文件  - "js/\*\*/test\*"  # 不要用 exclude 来忽略 'source/\_posts/' 中的文件。你应该使用 'skip\_render'，或者在要忽略的文件的文件名之前加一个下划线 '\_'  # 在这里配置一个 - "\_posts/hello-world.md" 是没有用的。  ignore:  # Ignore any folder named 'foo'.  - "\*\*/foo"  # Ignore 'foo' folder in 'themes/' only.  - "\*\*/themes/\*/foo"  # Same as above, but applies to every subfolders of 'themes/'.  - "\*\*/themes/\*\*/foo" |

列表中的每一项都必须用单引号或双引号包裹起来。

include 和 exclude 并不适用于 themes/ 目录下的文件。如果需要忽略 themes/ 目录下的部分文件或文件夹，可以使用 ignore 或在文件名之前添加下划线 \_。

#### 使用代替配置文件

可以在 hexo-cli 中使用 --config 参数来指定自定义配置文件的路径。你可以使用一个 YAML 或 JSON 文件的路径，也可以使用逗号分隔（无空格）的多个 YAML 或 JSON 文件的路径。例如：

|  |
| --- |
| # use 'custom.yml' in place of '\_config.yml' $ hexo server --config custom.yml  # use 'custom.yml' & 'custom2.json', prioritizing 'custom3.yml', then 'custom2.json' $ hexo generate --config custom.yml,custom2.json,custom3.yml |

当你指定了多个配置文件以后，Hexo 会按顺序将这部分配置文件合并成一个 \_multiconfig.yml。如果遇到重复的配置，排在后面的文件的配置会覆盖排在前面的文件的配置。这个原则适用于任意数量、任意深度的 YAML 和 JSON 文件。

例如，使用 --options 指定了两个自定义配置文件：

|  |
| --- |
| $ hexo generate --config custom.yml,custom2.json |

如果 custom.yml 中指定了 foo: bar，在 custom2.json 中指定了 "foo": "dinosaur"，那么在 \_multiconfig.yml 中你会得到 foo: dinosaur。

#### 覆盖主题配置

通常情况下，Hexo 主题是一个独立的项目，并拥有一个独立的 \_config.yml 配置文件。  
你可以在站点的 \_config.yml 配置文件中配置你的主题，这样你就不需要 fork 一份主题并维护主题独立的配置文件。

以下是一个覆盖主题配置的例子：

|  |
| --- |
| # \_config.yml theme\_config:  bio: "My awesome bio" |

|  |
| --- |
| # themes/my-theme/\_config.yml bio: "Some generic bio" logo: "a-cool-image.png" |

最终主题配置的输出是：

|  |
| --- |
| {  bio: "My awesome bio",  logo: "a-cool-image.png" } |

## Hexo指令

### init

|  |
| --- |
| $ hexo init [folder] |

新建一个网站。如果没有设置 folder ，Hexo 默认在目前的文件夹建立网站。

### new

|  |
| --- |
| $ hexo new [layout] <title> |

新建一篇文章。如果没有设置 layout 的话，默认使用 [\_config.yml](https://hexo.io/zh-cn/docs/configuration) 中的 default\_layout 参数代替。如果标题包含空格的话，请使用引号括起来。

|  |
| --- |
| $ hexo new "post title with whitespace" |

| **参数** | **描述** |
| --- | --- |
| -p, --path | 自定义新文章的路径 |
| -r, --replace | 如果存在同名文章，将其替换 |
| -s, --slug | 文章的 Slug，作为新文章的文件名和发布后的 URL |

默认情况下，Hexo 会使用文章的标题来决定文章文件的路径。对于独立页面来说，Hexo 会创建一个以标题为名字的目录，并在目录中放置一个 index.md 文件。你可以使用 --path 参数来覆盖上述行为、自行决定文件的目录：

|  |
| --- |
| hexo new page --path about/me "About me" |

以上命令会创建一个 source/about/me.md 文件，同时 Front Matter 中的 title 为 "About me"

注意！title 是必须指定的！如果你这么做并不能达到你的目的：

|  |
| --- |
| hexo new page --path about/me |

此时 Hexo 会创建 source/\_posts/about/me.md，同时 me.md 的 Front Matter 中的 title 为 "page"。这是因为在上述命令中，hexo-cli 将 page 视为指定文章的标题、并采用默认的 layout。

### generate

|  |
| --- |
| $ hexo generate |

生成静态文件。

| **选项** | **描述** |
| --- | --- |
| -d, --deploy | 文件生成后立即部署网站 |
| -w, --watch | 监视文件变动 |
| -b, --bail | 生成过程中如果发生任何未处理的异常则抛出异常 |
| -f, --force | 强制重新生成文件 Hexo 引入了差分机制，如果 public 目录存在，那么 hexo g 只会重新生成改动的文件。 使用该参数的效果接近 hexo clean && hexo generate |
| -c, --concurrency | 最大同时生成文件的数量，默认无限制 |

该命令可以简写为

|  |
| --- |
| $ hexo g |

### publish

|  |
| --- |
| $ hexo publish [layout] <filename> |

发表草稿。

### server

|  |
| --- |
| $ hexo server |

启动服务器。默认情况下，访问网址为： http://localhost:4000/。

| **选项** | **描述** |
| --- | --- |
| -p, --port | 重设端口 |
| -s, --static | 只使用静态文件 |
| -l, --log | 启动日记记录，使用覆盖记录格式 |

### deploy

|  |
| --- |
| $ hexo deploy |

部署网站。

| **参数** | **描述** |
| --- | --- |
| -g, --generate | 部署之前预先生成静态文件 |

该命令可以简写为：

|  |
| --- |
| $ hexo d |

### render

|  |
| --- |
| $ hexo render <file1> [file2] ... |

渲染文件。

| **参数** | **描述** |
| --- | --- |
| -o, --output | 设置输出路径 |

### migrate

|  |
| --- |
| $ hexo migrate <type> |

从其他博客系统 [迁移内容](https://hexo.io/zh-cn/docs/migration)。

### clean

|  |
| --- |
| $ hexo clean |

清除缓存文件 (db.json) 和已生成的静态文件 (public)。

在某些情况（尤其是更换主题后），如果发现您对站点的更改无论如何也不生效，您可能需要运行该命令。

### list

|  |
| --- |
| $ hexo list <type> |

列出网站资料。

### version

|  |
| --- |
| $ hexo version |

显示 Hexo 版本。

### 选项

#### 安全模式

|  |
| --- |
| $ hexo --safe |

在安全模式下，不会载入插件和脚本。当您在安装新插件遭遇问题时，可以尝试以安全模式重新执行。

#### 调试模式

|  |
| --- |
| $ hexo --debug |

在终端中显示调试信息并记录到 debug.log。当您碰到问题时，可以尝试用调试模式重新执行一次，并 [提交调试信息到 GitHub](https://github.com/hexojs/hexo/issues/new)。

#### 简洁模式

|  |
| --- |
| $ hexo --silent |

隐藏终端信息。

#### 自定义配置文件的路径

|  |
| --- |
| # 使用 custom.yml 代替默认的 \_config.yml $ hexo server --config custom.yml  # 使用 custom.yml 和 custom2.json，其中 custom2.json 优先级更高 $ hexo generate --config custom.yml,custom2.json,custom3.yml |

自定义配置文件的路径，指定这个参数后将不再使用默认的 \_config.yml。  
你可以使用一个 YAML 或 JSON 文件的路径，也可以使用逗号分隔（无空格）的多个 YAML 或 JSON 文件的路径。例如：

|  |
| --- |
| # 使用 custom.yml 代替默认的 \_config.yml $ hexo server --config custom.yml  # 使用 custom.yml, custom2.json 和 custom3.yml，其中 custom3.yml 优先级最高，其次是 custom2.json $ hexo generate --config custom.yml,custom2.json,custom3.yml |

当你指定了多个配置文件以后，Hexo 会按顺序将这部分配置文件合并成一个 \_multiconfig.yml。如果遇到重复的配置，排在后面的文件的配置会覆盖排在前面的文件的配置。这个原则适用于任意数量、任意深度的 YAML 和 JSON 文件。

#### 显示草稿

|  |
| --- |
| $ hexo --draft |

显示 source/\_drafts 文件夹中的草稿文章。

#### 自定义 CWD

|  |
| --- |
| $ hexo --cwd /path/to/cwd |

自定义当前工作目录（Current working directory）的路径。

## hexo部署到github

1、[安装git](https://links.jianshu.com/go?to=https%3A%2F%2Fgit-scm.com%2Fdownload%2Fwin)

2、[安装nodejs](https://links.jianshu.com/go?to=https%3A%2F%2Fnodejs.org%2Fen%2F)

3、用 npm 安装话经常出现卡住而导致无法正常安装，解决办法就是修改 npm 的安装源，这里选择淘宝 NPM 镜像，这是一个完整 npmjs.org 镜像，你可以用此代替官方版本，同步频率目前为 10分钟 一次以保证尽量与官方服务同步。

npm config set registry https://registry.npm.taobao.org

4、安装hexo

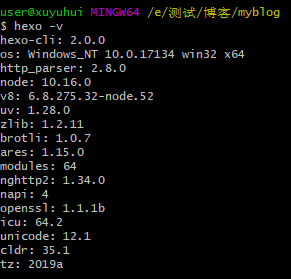
npm install -g hexo-cli



安装hexo

5、查看版本信息

hexo -v



查看版本信息

6、初始化hexo程序

hexo init myblog

（更新npm/并更换为国内的npm镜像命令：npm install npm -g/  
npm install -g cnpm --registry=[https://registry.npm.taobao.org](https://links.jianshu.com/go?to=https%3A%2F%2Fregistry.npm.taobao.org)）

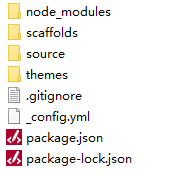
7、进入myblog文件夹

cd myblog

（8、npm install）

9、查看文件夹目录

* node\_modules: 依赖包
* scaffolds：生成文章的一些模板
* source：用来存放你的文章
* themes：主题
* \*\* \_config.yml: 博客的配置文件\*\*



文件夹目录

10、打开hexo

hexo g

hexo server

11、创建以用于提交

git config --global user.name "yourname"

git config --global user.email "youremail"

12、连接公钥

ssh-keygen -t rsa -C "email"

13、查看是否成功

ssh -T git@github.com

（You've successfully authenticated, but GitHub does not provide shell access.）

14、将hexo和GitHub关联起来，修改配置文件\_config.yml

deploy:

type: （中间有个空格，以下相同）git

repo: https://github.com/YourgithubName/YourgithubName.github.io.git

branch: master

15、安装deploy-git部署命令

npm install hexo-deployer-git --save

16、清除之前生成的东西

hexo clean

17、生成静态文章

hexo generate/hexo g

18、部署文章

hexo deploy/hexo d

19、github中要新建一个仓库用户名+github.io，访问用户名.github.io即可

##### 设置hexo主题

1、[选择hexo主题](https://links.jianshu.com/go?to=https%3A%2F%2Fhexo.io%2Fthemes%2F)，到github上下载主题放到themes文件夹下  
2、修改\_config.yml里的theme为该文件夹名称

##### 新增菜单栏选项

1、添加新页面：hexo new page "xx"  
2、在主题配置文件的menu中加上该页面  
3、在zh-CN.yml文件中加上中文意思

##### 新增草稿

1、添加草稿：hexo new draft "xx"  
2、预览草稿：hexo server --draft  
3、发布草稿：hexo publish draft "xx"

##### 图片处理

1、配置文件中的post\_asset\_folder改成true  
2、npm install hexo-asset-image --save  
3、放到该文件夹下，按照md格式输入(文件名称/图片名.jpg)

###### 搜索

1、npm install hexo-generator-searchdb --save  
2、站点配置文件的扩展下添加

search:

path: search.xml

field: post

format: html

limit: 10000

3、主题配置文件下，local\_search改成true即可

local\_search:

enable: true

###### 字数

1、npm install hexo-symbols-count-time --save  
2、站点配置

symbols\_count\_time:

symbols: true

time: true

total\_symbols: true

total\_time: true

###### 宠物

1、npm install --save hexo-helper-live2d  
2、[live2d插件](https://links.jianshu.com/go?to=https%3A%2F%2Fhuaji8.top%2Fpost%2Flive2d-plugin-2.0%2F)  
3、安装合适的宠物npm install live2d-widget-model-wanko  
4、在配置文件中添加即可

live2d:

enable: true

scriptFrom: local

pluginRootPath: live2dw/

pluginJsPath: lib/

pluginModelPath: assets/

model:

# use: live2d-widget-model-tororo

# use: live2d-widget-model-hijiki

use: live2d-widget-model-wanko

display:

position: right

width: 150

height: 300

# 水平位置

# hOffset: 0

# 垂直位置

# vOffset: -20

mobile:

show: true

###### 点击弹爱心

1、在source->js目录下新建一个js文件为click-love.js

! function (e, t, a) {

function n() {

c(".heart{width: 10px;height: 10px;position: fixed;background: #f00;transform: rotate(45deg);-webkit-transform: rotate(45deg);-moz-transform: rotate(45deg);}.heart:after,.heart:before{content: '';width: inherit;height: inherit;background: inherit;border-radius: 50%;-webkit-border-radius: 50%;-moz-border-radius: 50%;position: fixed;}.heart:after{top: -5px;}.heart:before{left: -5px;}"), o(), r()

}

function r() {

for (var e = 0; e < d.length; e++) d[e].alpha <= 0 ? (t.body.removeChild(d[e].el), d.splice(e, 1)) : (d[e].y--, d[e].scale += .004, d[e].alpha -= .013, d[e].el.style.cssText = "left:" + d[e].x + "px;top:" + d[e].y + "px;opacity:" + d[e].alpha + ";transform:scale(" + d[e].scale + "," + d[e].scale + ") rotate(45deg);background:" + d[e].color + ";z-index:99999");

requestAnimationFrame(r)

}

function o() {

var t = "function" == typeof e.onclick && e.onclick;

e.onclick = function (e) {

t && t(), i(e)

}

}

function i(e) {

var a = t.createElement("div");

a.className = "heart", d.push({

el: a,

x: e.clientX - 5,

y: e.clientY - 5,

scale: 1,

alpha: 1,

color: s()

}), t.body.appendChild(a)

}

function c(e) {

var a = t.createElement("style");

a.type = "text/css";

try {

a.appendChild(t.createTextNode(e))

} catch (t) {

a.styleSheet.cssText = e

}

t.getElementsByTagName("head")[0].appendChild(a)

}

function s() {

return "rgb(" + ~~(255 \* Math.random()) + "," + ~~(255 \* Math.random()) + "," + ~~(255 \* Math.random()) + ")"

}

var d = [];

e.requestAnimationFrame = function () {

return e.requestAnimationFrame || e.webkitRequestAnimationFrame || e.mozRequestAnimationFrame || e.oRequestAnimationFrame || e.msRequestAnimationFrame || function (e) {

setTimeout(e, 1e3 / 60)

}

}(), n()

}(window, document);

2、在主题theme->layout->\_partials->footer文件中导入该js文件即可

<script type="text/javascript" src="/js/click-love.js"></script>

## Hexo迁移

### RSS

首先，安装 hexo-migrator-rss 插件。

|  |
| --- |
| $ npm install hexo-migrator-rss --save |

插件安装完成后，执行下列命令，从 RSS 迁移所有文章。source 可以是文件路径或网址。

|  |
| --- |
| $ hexo migrate rss <source> |

### Jekyll

把 \_posts 文件夹内的所有文件复制到 source/\_posts 文件夹，并在 \_config.yml 中修改 new\_post\_name 参数。

|  |
| --- |
| new\_post\_name: :year-:month-:day-:title.md |

### Octopress

把 Octopress source/\_posts 文件夹内的所有文件转移到 Hexo 的 source/\_posts 文件夹，并修改 \_config.yml 中的 new\_post\_name 参数。

|  |
| --- |
| new\_post\_name: :year-:month-:day-:title.md |

### WordPress

首先，安装 hexo-migrator-wordpress 插件。

|  |
| --- |
| $ npm install hexo-migrator-wordpress --save |

在 WordPress 仪表盘中导出数据(“Tools” → “Export” → “WordPress”)（详情参考[WP支持页面](http://en.support.wordpress.com/export/)）。

插件安装完成后，执行下列命令来迁移所有文章。source 可以是 WordPress 导出的文件路径或网址。

|  |
| --- |
| $ hexo migrate wordpress <source> |

**注意**

这个插件并不能完美地实现WordPress->Hexo的数据转换，尤其是在处理WordPress的分类方面存在问题（见[Front-matter中的分类与标签](https://hexo.io/zh-cn/docs/front-matter)）。因此，建议您在迁移完成后，手工审阅所有生成的markdown文件，检查其中是否有错误。对于文章数量较大的WordPress站点，这项工作可能要花很长的时间。

### Joomla

首先，安装 hexo-migrator-joomla 插件。

|  |
| --- |
| $ npm install hexo-migrator-joomla --save |

使用 [J2XML](http://extensions.joomla.org/extensions/migration-a-conversion/data-import-a-export/12816?qh=YToxOntpOjA7czo1OiJqMnhtbCI7fQ%253D%253D) 组件导出 Joomla 文章。  
插件安装完成后，执行下列命令来迁移所有文章。source 可以是 Joomla 导出的文件路径或网址。

|  |
| --- |
| $ hexo migrate joomla <source> |

## 内容

你可以执行下列命令来创建一篇新文章或者新的页面。

|  |
| --- |
| $ hexo new [layout] <title> |

您可以在命令中指定文章的布局（layout），默认为 post，可以通过修改 \_config.yml 中的 default\_layout 参数来指定默认布局。

#### 布局（Layout）

Hexo 有三种默认布局：post、page 和 draft。在创建者三种不同类型的文件时，它们将会被保存到不同的路径；而您自定义的其他布局和 post 相同，都将储存到 source/\_posts 文件夹。

| **布局** | **路径** |
| --- | --- |
| post | source/\_posts |
| page | source |
| draft | source/\_drafts |

**不要处理我的文章**

如果你不想你的文章被处理，你可以将 Front-Matter 中的layout: 设为 false 。

#### 文件名称

Hexo 默认以标题做为文件名称，但您可编辑 new\_post\_name 参数来改变默认的文件名称，举例来说，设为 :year-:month-:day-:title.md 可让您更方便的通过日期来管理文章。

| **变量** | **描述** |
| --- | --- |
| :title | 标题（小写，空格将会被替换为短杠） |
| :year | 建立的年份，比如， 2015 |
| :month | 建立的月份（有前导零），比如， 04 |
| :i\_month | 建立的月份（无前导零），比如， 4 |
| :day | 建立的日期（有前导零），比如， 07 |
| :i\_day | 建立的日期（无前导零），比如， 7 |

#### 草稿

刚刚提到了 Hexo 的一种特殊布局：draft，这种布局在建立时会被保存到 source/\_drafts 文件夹，您可通过 publish 命令将草稿移动到 source/\_posts 文件夹，该命令的使用方式与 new 十分类似，您也可在命令中指定 layout 来指定布局。

|  |
| --- |
| $ hexo publish [layout] <title> |

草稿默认不会显示在页面中，您可在执行时加上 --draft 参数，或是把 render\_drafts 参数设为 true 来预览草稿。

### 模版（Scaffold）

在新建文章时，Hexo 会根据 scaffolds 文件夹内相对应的文件来建立文件，例如：

|  |
| --- |
| $ hexo new photo "My Gallery" |

在执行这行指令时，Hexo 会尝试在 scaffolds 文件夹中寻找 photo.md，并根据其内容建立文章，以下是您可以在模版中使用的变量：

| **变量** | **描述** |
| --- | --- |
| layout | 布局 |
| title | 标题 |
| date | 文件建立日期 |

#### 支持的格式

Hexo 支持以任何格式书写文章，只要安装了相应的渲染插件。

例如，Hexo 默认安装了 hexo-renderer-marked 和 hexo-renderer-ejs，因此你不仅可以用 Markdown 写作，你还可以用 EJS 写作。如果你安装了 hexo-renderer-pug，你甚至可以用 Pug 模板语言书写文章。

只需要将文章的扩展名从 md 改成 ejs，Hexo 就会使用 hexo-renderer-ejs 渲染这个文件，其他格式同理。

## Front-matter

Front-matter 是文件最上方以 --- 分隔的区域，用于指定个别文件的变量，举例来说：

|  |
| --- |
| --- title: Hello World date: 2013/7/13 20:46:25 --- |

以下是预先定义的参数，您可在模板中使用这些参数值并加以利用。

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| layout | 布局 |  |
| title | 标题 | 文章的文件名 |
| date | 建立日期 | 文件建立日期 |
| updated | 更新日期 | 文件更新日期 |
| comments | 开启文章的评论功能 | true |
| tags | 标签（不适用于分页） |  |
| categories | 分类（不适用于分页） |  |
| permalink | 覆盖文章网址 |  |
| keywords | 仅用于 meta 标签和 Open Graph 的关键词（不推荐使用） |  |

### 分类和标签

只有文章支持分类和标签，您可以在 Front-matter 中设置。在其他系统中，分类和标签听起来很接近，但是在 Hexo 中两者有着明显的差别：分类具有顺序性和层次性，也就是说 Foo, Bar 不等于 Bar, Foo；而标签没有顺序和层次。

|  |
| --- |
| categories: - Diary tags: - PS3 - Games |

**分类方法的分歧**

如果您有过使用 WordPress 的经验，就很容易误解 Hexo 的分类方式。WordPress 支持对一篇文章设置多个分类，而且这些分类可以是同级的，也可以是父子分类。但是 Hexo 不支持指定多个同级分类。下面的指定方法：

|  |
| --- |
| categories:  - Diary  - Life |

会使分类Life成为Diary的子分类，而不是并列分类。因此，有必要为您的文章选择尽可能准确的分类。

如果你需要为文章添加多个分类，可以尝试以下 list 中的方法。

|  |
| --- |
| categories: - [Diary, PlayStation] - [Diary, Games] - [Life] |

此时这篇文章同时包括三个分类： PlayStation 和 Games 分别都是父分类 Diary 的子分类，同时 Life 是一个没有子分类的分类。

### JSON Front-matter

除了 YAML 外，你也可以使用 JSON 来编写 Front-matter，只要将 --- 代换成 ;;; 即可。

|  |
| --- |
| "title": "Hello World", "date": "2013/7/13 20:46:25" ;;; |

## 标签插件（Tag Plugins）

标签插件和 Front-matter 中的标签不同，它们是用于在文章中快速插入特定内容的插件。

虽然你可以使用任何格式书写你的文章，但是标签插件永远可用，且语法也都是一致的。

### 引用块

在文章中插入引言，可包含作者、来源和标题。

**别号：** quote

|  |
| --- |
| {% blockquote [author[, source]] [link] [source\_link\_title] %} content {% endblockquote %} |

#### 样例

**没有提供参数，则只输出普通的 blockquote**

|  |
| --- |
| {% blockquote %} Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque hendrerit lacus ut purus iaculis feugiat. Sed nec tempor elit, quis aliquam neque. Curabitur sed diam eget dolor fermentum semper at eu lorem. {% endblockquote %} |

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque hendrerit lacus ut purus iaculis feugiat. Sed nec tempor elit, quis aliquam neque. Curabitur sed diam eget dolor fermentum semper at eu lorem.

**引用书上的句子**

|  |
| --- |
| {% blockquote David Levithan, Wide Awake %} Do not just seek happiness for yourself. Seek happiness for all. Through kindness. Through mercy. {% endblockquote %} |

Do not just seek happiness for yourself. Seek happiness for all. Through kindness. Through mercy.

**David Levithan**Wide Awake

**引用 Twitter**

|  |
| --- |
| {% blockquote @DevDocs https://twitter.com/devdocs/status/356095192085962752 %} NEW: DevDocs now comes with syntax highlighting. http://devdocs.io {% endblockquote %} |

NEW: DevDocs now comes with syntax highlighting. [http://devdocs.io](http://devdocs.io/)

**@DevDocs**[*twitter.com/devdocs/status/356095192085962752*](https://twitter.com/devdocs/status/356095192085962752)

**引用网络上的文章**

|  |
| --- |
| {% blockquote Seth Godin http://sethgodin.typepad.com/seths\_blog/2009/07/welcome-to-island-marketing.html Welcome to Island Marketing %} Every interaction is both precious and an opportunity to delight. {% endblockquote %} |

Every interaction is both precious and an opportunity to delight.

**Seth Godin**[*Welcome to Island Marketing*](http://sethgodin.typepad.com/seths_blog/2009/07/welcome-to-island-marketing.html)

### 代码块

在文章中插入代码。

**别名：** code

|  |
| --- |
| {% codeblock [title] [lang:language] [url] [link text] [additional options] %} code snippet {% endcodeblock %} |

Specify additional options in option:value format, e.g. line\_number:false first\_line:5.

| **Extra Options** | **Description** | **Default** |
| --- | --- | --- |
| line\_number | Show line number | true |
| highlight | Enable code highlighting | true |
| first\_line | Specify the first line number | 1 |
| mark | Line highlight specific line(s), each value separated by a comma. Specify number range using a dash Example: mark:1,4-7,10 will mark line 1, 4 to 7 and 10. |  |
| wrap | Wrap the code block in [<table>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/table) | true |

#### 样例

**普通的代码块**

|  |
| --- |
| {% codeblock %} alert('Hello World!'); {% endcodeblock %} |

|  |
| --- |
| alert('Hello World!'); |

**指定语言**

|  |
| --- |
| {% codeblock lang:objc %} [rectangle setX: 10 y: 10 width: 20 height: 20]; {% endcodeblock %} |

|  |
| --- |
| [rectangle setX: 10 y: 10 width: 20 height: 20]; |

**附加说明**

|  |
| --- |
| {% codeblock Array.map %} array.map(callback[, thisArg]) {% endcodeblock %} |

Array.map

|  |
| --- |
| array.map(callback[, thisArg]) |

**附加说明和网址**

|  |
| --- |
| {% codeblock \_.compact http://underscorejs.org/#compact Underscore.js %} \_.compact([0, 1, false, 2, '', 3]); => [1, 2, 3] {% endcodeblock %} |

\_.compact[Underscore.js](http://underscorejs.org/#compact)

|  |
| --- |
| \_.compact([0, 1, false, 2, '', 3]); => [1, 2, 3] |

### 反引号代码块

另一种形式的代码块，不同的是它使用三个反引号来包裹。

``` [language] [title] [url] [link text] code snippet ```

### Pull Quote

在文章中插入 Pull quote。

|  |
| --- |
| {% pullquote [class] %} content {% endpullquote %} |

### jsFiddle

在文章中嵌入 jsFiddle。

|  |
| --- |
| {% jsfiddle shorttag [tabs] [skin] [width] [height] %} |

### Gist

在文章中嵌入 Gist。

|  |
| --- |
| {% gist gist\_id [filename] %} |

### iframe

在文章中插入 iframe。

|  |
| --- |
| {% iframe url [width] [height] %} |

### Image

在文章中插入指定大小的图片。

|  |
| --- |
| {% img [class names] /path/to/image [width] [height] '"title text" "alt text"' %} |

### Link

在文章中插入链接，并自动给外部链接添加 target="\_blank" 属性。

|  |
| --- |
| {% link text url [external] [title] %} |

### Include Code

插入 source/downloads/code 文件夹内的代码文件。source/downloads/code 不是固定的，取决于你在配置文件中 code\_dir 的配置。

|  |
| --- |
| {% include\_code [title] [lang:language] [from:line] [to:line] path/to/file %} |

#### 样例

**嵌入 test.js 文件全文**

|  |
| --- |
| {% include\_code lang:javascript test.js %} |

**只嵌入第 3 行**

|  |
| --- |
| {% include\_code lang:javascript from:3 to:3 test.js %} |

**嵌入第 5 行至第 8 行**

|  |
| --- |
| {% include\_code lang:javascript from:5 to:8 test.js %} |

**嵌入第 5 行至文件结束**

|  |
| --- |
| {% include\_code lang:javascript from:5 test.js %} |

**嵌入第 1 行至第 8 行**

|  |
| --- |
| {% include\_code lang:javascript to:8 test.js %} |

### Youtube

在文章中插入 Youtube 视频。

|  |
| --- |
| {% youtube video\_id %} |

### Vimeo

在文章中插入 Vimeo 视频。

|  |
| --- |
| {% vimeo video\_id %} |

### 引用文章

引用其他文章的链接。

|  |
| --- |
| {% post\_path filename %} {% post\_link filename [title] [escape] %} |

在使用此标签时可以忽略文章文件所在的路径或者文章的永久链接信息、如语言、日期。

例如，在文章中使用 {% post\_link how-to-bake-a-cake %} 时，只需有一个名为 how-to-bake-a-cake.md 的文章文件即可。即使这个文件位于站点文件夹的 source/posts/2015-02-my-family-holiday 目录下、或者文章的永久链接是 2018/en/how-to-bake-a-cake，都没有影响。

默认链接文字是文章的标题，你也可以自定义要显示的文本。此时不应该使用 Markdown 语法 []()。

默认对文章的标题和自定义标题里的特殊字符进行转义。可以使用escape选项，禁止对特殊字符进行转义。

**链接使用文章的标题**

{% post\_link hexo-3-8-released %}

[Hexo 3.8.0 Released](https://hexo.io/news/2018/10/19/hexo-3-8-released/)

**链接使用自定义文字**

{% post\_link hexo-3-8-released '通往文章的链接' %}

[通往文章的链接](https://hexo.io/news/2018/10/19/hexo-3-8-released/)

**对标题的特殊字符进行转义**

|  |
| --- |
| {% post\_link hexo-4-released 'How to use <b> tag in title' %} |

[How to use <b> tag in title](https://hexo.io/news/2019/10/14/hexo-4-released/)

**禁止对标题的特殊字符进行转义**

|  |
| --- |
| {% post\_link hexo-4-released '<b>bold</b> custom title' false %} |

[bold custom title](https://hexo.io/news/2019/10/14/hexo-4-released/)

### 引用资源

引用文章的资源。

|  |
| --- |
| {% asset\_path filename %} {% asset\_img filename [title] %} {% asset\_link filename [title] [escape] %} |

### Raw

如果您想在文章中插入 Swig 标签，可以尝试使用 Raw 标签，以免发生解析异常。

|  |
| --- |
| {% raw %} content {% endraw %} |

### 文章摘要和截断

在文章中使用 <!-- more -->，那么 <!-- more --> 之前的文字将会被视为摘要。首页中将只出现这部分文字，同时这部分文字也会出现在正文之中。

例如：

|  |
| --- |
| Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. <!-- more --> Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. |

首页中将只会出现

|  |
| --- |
| Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. |

正文中则会出现

|  |
| --- |
| Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.  Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. |

注意，摘要可能会被 Front Matter 中的 excerpt 覆盖。

## 资源文件夹

资源（Asset）代表 source 文件夹中除了文章以外的所有文件，例如图片、CSS、JS 文件等。比方说，如果你的Hexo项目中只有少量图片，那最简单的方法就是将它们放在 source/images 文件夹中。然后通过类似于 ![](/images/image.jpg) 的方法访问它们。

### 文章资源文件夹

对于那些想要更有规律地提供图片和其他资源以及想要将他们的资源分布在各个文章上的人来说，Hexo也提供了更组织化的方式来管理资源。这个稍微有些复杂但是管理资源非常方便的功能可以通过将 config.yml文件中的 post\_asset\_folder 选项设为 true 来打开。

\_config.yml

|  |
| --- |
| post\_asset\_folder: true |

当资源文件管理功能打开后，Hexo将会在你每一次通过 hexo new [layout] <title> 命令创建新文章时自动创建一个文件夹。这个资源文件夹将会有与这个文章文件一样的名字。将所有与你的文章有关的资源放在这个关联文件夹中之后，你可以通过相对路径来引用它们，这样你就得到了一个更简单而且方便得多的工作流。

### 相对路径引用的标签插件

通过常规的 markdown 语法和相对路径来引用图片和其它资源可能会导致它们在存档页或者主页上显示不正确。在Hexo 2时代，社区创建了很多插件来解决这个问题。但是，随着Hexo 3 的发布，许多新的标签插件被加入到了核心代码中。这使得你可以更简单地在文章中引用你的资源。

|  |
| --- |
| {% asset\_path slug %} {% asset\_img slug [title] %} {% asset\_link slug [title] %} |

比如说：当你打开文章资源文件夹功能后，你把一个 example.jpg 图片放在了你的资源文件夹中，如果通过使用相对路径的常规 markdown 语法 ![](/example.jpg) ，它将 不会 出现在首页上。（但是它会在文章中按你期待的方式工作）

正确的引用图片方式是使用下列的标签插件而不是 markdown ：

|  |
| --- |
| {% asset\_img example.jpg This is an example image %} |

通过这种方式，图片将会同时出现在文章和主页以及归档页中。

## 数据文件

有时您可能需要在主题中使用某些资料，而这些资料并不在文章内，并且是需要重复使用的，那么您可以考虑使用 Hexo 3.0 新增的「数据文件」功能。此功能会载入 source/\_data 内的 YAML 或 JSON 文件，如此一来您便能在网站中复用这些文件了。

举例来说，在 source/\_data 文件夹中新建 menu.yml 文件：

|  |
| --- |
| Home: / Gallery: /gallery/ Archives: /archives/ |

您就能在模板中使用这些资料：

|  |
| --- |
| <% for (var link in site.data.menu) { %>  <a href="<%= site.data.menu[link] %>"> <%= link %> </a> <% } %> |

渲染结果如下 :

|  |
| --- |
| <a href="/"> Home </a> <a href="/gallery/"> Gallery </a> <a href="/archives/"> Archives </a> |

## 服务器

### [hexo-server](https://github.com/hexojs/hexo-server)

Hexo 3.0 把服务器独立成了个别模块，您必须先安装 [hexo-server](https://github.com/hexojs/hexo-server) 才能使用。

|  |
| --- |
| $ npm install hexo-server --save |

安装完成后，输入以下命令以启动服务器，您的网站会在 http://localhost:4000 下启动。在服务器启动期间，Hexo 会监视文件变动并自动更新，您无须重启服务器。

|  |
| --- |
| $ hexo server |

如果您想要更改端口，或是在执行时遇到了 EADDRINUSE 错误，可以在执行时使用 -p 选项指定其他端口，如下：

|  |
| --- |
| $ hexo server -p 5000 |

#### 静态模式

在静态模式下，服务器只处理 public 文件夹内的文件，而不会处理文件变动，在执行时，您应该先自行执行 hexo generate，此模式通常用于生产环境（production mode）下。

|  |
| --- |
| $ hexo server -s |

#### 自定义 IP

服务器默认运行在 0.0.0.0，您可以覆盖默认的 IP 设置，如下：

|  |
| --- |
| $ hexo server -i 192.168.1.1 |

指定这个参数后，您就只能通过该IP才能访问站点。例如，对于一台使用无线网络的笔记本电脑，除了指向本机的127.0.0.1外，通常还有一个192.168.\*.\*的局域网IP，如果像上面那样使用-i参数，就不能用127.0.0.1来访问站点了。对于有公网IP的主机，如果您指定一个局域网IP作为-i参数的值，那么就无法通过公网来访问站点。

### Pow

[Pow](http://pow.cx/) 是一个 Mac 系统上的零配置 Rack 服务器，它也可以作为一个简单易用的静态文件服务器来使用。

#### 安装

|  |
| --- |
| $ curl get.pow.cx | sh |

#### 设置

在 ~/.pow 文件夹建立链接（symlink）。

|  |
| --- |
| $ cd ~/.pow $ ln -s /path/to/myapp |

您的网站将会在 http://myapp.dev 下运行，网址根据链接名称而定。

## 生成文件

使用 Hexo 生成静态文件快速而且简单。

|  |
| --- |
| $ hexo generate |

#### 监视文件变动

Hexo 能够监视文件变动并立即重新生成静态文件，在生成时会比对文件的 SHA1 checksum，只有变动的文件才会写入。

|  |
| --- |
| $ hexo generate --watch |

#### 完成后部署

您可执行下列的其中一个命令，让 Hexo 在生成完毕后自动部署网站，两个命令的作用是相同的。

|  |
| --- |
| $ hexo generate --deploy $ hexo deploy --generate |

**简写**

上面两个命令可以简写为  
$ hexo g -d  
$ hexo d -g

## 将 Hexo 部署到 GitHub Pages

在本教程中，我们将会使用 [Travis CI](https://travis-ci.com/) 将 Hexo 博客部署到 GitHub Pages 上。Travis CI 对于开源 repository 是免费的，但是这意味着你的站点文件将会是公开的。如果你希望你的站点文件不被公开，请直接前往本文 [私有 Repository](#私有 Repository) 部分。

1. 新建一个 repository。如果你希望你的站点能通过 <你的 GitHub 用户名>.github.io 域名访问，你的 repository 应该直接命名为 <你的 GitHub 用户名>.github.io。
2. 将你的 Hexo 站点文件夹推送到 repository 中。默认情况下不应该 public 目录将不会被推送到 repository 中，你应该检查 .gitignore 文件中是否包含 public 一行，如果没有请加上。
3. 将 [Travis CI](https://github.com/marketplace/travis-ci) 添加到你的 GitHub 账户中。
4. 前往 GitHub 的 [Applications settings](https://github.com/settings/installations)，配置 Travis CI 权限，使其能够访问你的 repository。
5. 你应该会被重定向到 Travis CI 的页面。如果没有，请 [手动前往](https://travis-ci.com/)。
6. 在浏览器新建一个标签页，前往 GitHub [新建 Personal Access Token](https://github.com/settings/tokens)，只勾选 repo 的权限并生成一个新的 Token。Token 生成后请复制并保存好。
7. 回到 Travis CI，前往你的 repository 的设置页面，在 **Environment Variables** 下新建一个环境变量，**Name** 为 GH\_TOKEN，**Value** 为刚才你在 GitHub 生成的 Token。确保 **DISPLAY VALUE IN BUILD LOG**保持 **不被勾选** 避免你的 Token 泄漏。点击 **Add** 保存。
8. 在你的 Hexo 站点文件夹中新建一个 .travis.yml 文件：

|  |
| --- |
| sudo: false language: node\_js node\_js:  - 10 # use nodejs v10 LTS cache: npm branches:  only:  - master # build master branch only script:  - hexo generate # generate static files deploy:  provider: pages  skip-cleanup: true  github-token: $GH\_TOKEN  keep-history: true  on:  branch: master  local-dir: public |

1. 将 .travis.yml 推送到 repository 中。Travis CI 应该会自动开始运行，并将生成的文件推送到同一 repository 下的 gh-pages 分支下
2. 在 GitHub 中前往你的 repository 的设置页面，修改 GitHub Pages 的部署分支为 gh-pages。
3. 前往 https://<你的 GitHub 用户名>.github.io 查看你的站点是否可以访问。这可能需要一些时间。

### Project page

如果你更希望你的站点部署在 <你的 GitHub 用户名>.github.io 的子目录中，你的 repository 需要直接命名为子目录的名字，这样你的站点可以通过 https://<你的 GitHub 用户名>.github.io/<repository 的名字> 访问。你需要检查你的 Hexo 配置文件，将 url 修改为 https://<你的 GitHub 用户名>.github.io/<repository 的名字>、将 root 的值修改为 /<repository 的名字>/

### 私有 Repository

下面的指示基于 [一键部署](https://hexo.io/docs/one-command-deployment) 编写。

1. 安装 [hexo-deployer-git](https://github.com/hexojs/hexo-deployer-git).
2. 在 **\_config.yml**（如果有已存在的请删除）添加如下配置：

|  |
| --- |
| deploy:  type: git  repo: https://github.com/<username>/<project>  # example, https://github.com/hexojs/hexojs.github.io  branch: gh-pages |

1. 运行 hexo clean && hexo deploy 。
2. 查看 username.github.io 上的网页是否部署成功。

### 有用的参考链接

* [GitHub Pages 使用文档](https://help.github.com/categories/github-pages-basics/)
* [Travis CI 使用文档](https://docs.travis-ci.com/user/tutorial/)
* [Awesome Hexo](https://github.com/hexojs/awesome-hexo)
* [在百度上搜索 “Hexo GitHub”](https://www.baidu.com/s?wd=Hexo%20GitHub)

## 将 Hexo 部署到 GitLab Pages

在本教程中，我们将会使用 GitLab CI 将 Hexo 博客部署到 GitLab Pages 上。

1. 新建一个 repository。如果你希望你的站点能通过 <你的 GitLab 用户名>.gitlab.io 域名访问，你的 repository 应该直接命名为 <你的 GitLab 用户名>.gitlab.io。
2. 将你的 Hexo 站点文件夹推送到 repository 中。默认情况下不应该 public 目录将不会被推送到 repository 中，你应该检查 .gitignore 文件中是否包含 public 一行，如果没有请加上。
3. 在你的站点文件夹中新建 .gitlab-ci.yml 文件：

|  |
| --- |
| image: node:10-alpine # use nodejs v10 LTS cache:  paths:  - node\_modules/  before\_script:  - npm install hexo-cli -g  - npm install  pages:  script:  - hexo generate  artifacts:  paths:  - public  only:  - master |

1. GitLab CI 应该会自动开始运行，构建成功以后你应该可以在 https://<你的 GitLab 用户名>.gitlab.io 查看你的网站。
2. (可选) 如果你需要查看生成的文件，可以在 [job artifact](https://docs.gitlab.com/ee/user/project/pipelines/job_artifacts.html) 中找到。

在 GitLab.com 上，GitLab CI 是默认启用的。如果你使用的是自托管的 GitLab，你可能需要在 Settings -> CI / CD -> Shared Runners 启用 GitLab CI。

### Project page

如果你更希望你的站点部署在 <你的 GitLab 用户名>.gitlab.io 的子目录中，你的 repository 需要直接命名为子目录的名字，这样你的站点可以通过 https://<你的 GitLab 用户名>.gitlab.io/<repository 的名字> 访问。你需要检查你的 Hexo 配置文件，将 url 的值修改为 https://<你的 GitLab 用户名>.gitlab.io/<repository 的名字>、将 root 的值修改为 /<repository 的名字>/

### Useful links

* [GitLab Pages 相关文档](https://docs.gitlab.com/ee/user/project/pages/index.html)
* [GitLab CI 相关文档](https://docs.gitlab.com/ee/ci/README.html)
* [在百度上搜索 “Hexo GitLab”](https://www.baidu.com/s?wd=Hexo%20GitLab)

## 部署

Hexo 提供了快速方便的一键部署功能，让您只需一条命令就能将网站部署到服务器上。

|  |
| --- |
| $ hexo deploy |

在开始之前，您必须先在 \_config.yml 中修改参数，一个正确的部署配置中至少要有 type 参数，例如：

|  |
| --- |
| deploy:  type: git |

您可同时使用多个 deployer，Hexo 会依照顺序执行每个 deployer。

|  |
| --- |
| deploy: - type: git  repo: - type: heroku  repo: |

Refer to the [Plugins](https://hexo.io/plugins/) list for more deployment plugins.

**缩进**

YAML依靠缩进来确定元素间的从属关系。因此，请确保每个deployer的缩进长度相同，并且使用空格缩进。

### Git

1. 安装 [hexo-deployer-git](https://github.com/hexojs/hexo-deployer-git)。

|  |
| --- |
| $ npm install hexo-deployer-git --save |

1. 修改配置。

|  |
| --- |
| deploy:  type: git  repo: <repository url> #https://bitbucket.org/JohnSmith/johnsmith.bitbucket.io  branch: [branch]  message: [message] |

| **参数** | **描述** | **默认** |
| --- | --- | --- |
| repo | 库（Repository）地址 |  |
| branch | 分支名称 | gh-pages (GitHub) coding-pages (Coding.net) master (others) |
| message | 自定义提交信息 | Site updated: {{ now('YYYY-MM-DD HH:mm:ss') }}) |
| token | Optional token value to authenticate with the repo. Prefix with $ to read token from environment variable |  |

1. 生成站点文件并推送至远程库。执行 hexo clean && hexo deploy。
   * You will be prompted with username and password of the target repository, unless you authenticate with a token or ssh key.
   * hexo-deployer-git does not store your username and password. Use [git-credential-cache](https://git-scm.com/docs/git-credential-cache) to store them temporarily.
2. 登入 Github/BitBucket/Gitlab，请在库设置（Repository Settings）中将默认分支设置为\_config.yml配置中的分支名称。稍等片刻，您的站点就会显示在您的Github Pages中。

#### 这一切是如何发生的？

当执行 hexo deploy 时，Hexo 会将 public 目录中的文件和目录推送至 \_config.yml 中指定的远端仓库和分支中，并且**完全覆盖**该分支下的已有内容。

**For 使用 Git 管理站点目录的用户**

由于 Hexo 的部署默认使用分支 master，所以如果你同时正在使用 Git 管理你的站点目录，你应当注意你的部署分支应当不同于写作分支。  
一个好的实践是将站点目录和 Pages 分别存放在两个不同的 Git 仓库中，可以有效避免相互覆盖。  
Hexo 在部署你的站点生成的文件时并不会更新你的站点目录。因此你应该手动提交并推送你的写作分支。

此外，如果您的 Github Pages 需要使用 CNAME 文件**自定义域名**，请将 CNAME 文件置于 source 目录下，只有这样 hexo deploy 才能将 CNAME 文件一并推送至部署分支。

### Heroku

安装 [hexo-deployer-heroku](https://github.com/hexojs/hexo-deployer-heroku)。

|  |
| --- |
| $ npm install hexo-deployer-heroku --save |

修改配置。

|  |
| --- |
| deploy:  type: heroku  repo: <repository url>  message: [message] |

| **参数** | **描述** |
| --- | --- |
| repo | Heroku 库（Repository）地址 |
| message | 自定提交信息 (默认为 Site updated: {{ now('YYYY-MM-DD HH:mm:ss') }}) |

### Netlify

[Netlify](https://www.netlify.com/) 是一个提供网络托管的综合平台。它集持续集成（CI）CDN 自定义域名 HTTPS 持续部署（CD）等诸多功能于一身。您可以通过以下两种方式将Hexo站点部署到Netlify。

首先，也是最通用的方式，就是使用Netlify提供的网页端用户界面。前往[新建一个网站页面](https://app.netlify.com/start)，选择需要关联的 Github/BitBucket/Gitlab 库，然后遵循网站提示。

另一种方式是使用Netlify提供的命令行客户端工具 [Node based CLI](https://www.netlify.com/docs/cli/) 管理和部署您的站点。

此外，您还可以在项目的README中增加一个 [部署至Netlify按钮](https://www.netlify.com/docs/deploy-button/)，这样其他用户在fork或clone了您的项目之后可以方便快捷地一键部署。

### Rsync

安装 [hexo-deployer-rsync](https://github.com/hexojs/hexo-deployer-rsync)。

|  |
| --- |
| $ npm install hexo-deployer-rsync --save |

修改配置。

|  |
| --- |
| deploy:  type: rsync  host: <host>  user: <user>  root: <root>  port: [port]  delete: [true|false]  verbose: [true|false]  ignore\_errors: [true|false] |

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| host | 远程主机的地址 |  |
| user | 使用者名称 |  |
| root | 远程主机的根目录 |  |
| port | 端口 | 22 |
| delete | 删除远程主机上的旧文件 | true |
| verbose | 显示调试信息 | true |
| ignore\_errors | 忽略错误 | false |

**rsync部署模块的工作方式**

需要注意的是，要求您提供的实际上是一个能通过SSH登陆远程主机的Linux用户。Hexo会自动处理关于rsync使用的一切操作。因此，您需要在远程主机上为您的Hexo站点建立一个用户，并允许其通过SSH登陆。不过，这里的port，的确是指rsync监听的端口，请确保防火墙打开了该端口。

### OpenShift

安装 [hexo-deployer-openshift](https://github.com/hexojs/hexo-deployer-openshift)。

|  |
| --- |
| $ npm install hexo-deployer-openshift --save |

修改配置。

|  |
| --- |
| deploy:  type: openshift  repo: <repository url>  message: [message] |

| **参数** | **描述** |
| --- | --- |
| repo | OpenShift 库（Repository）地址 |
| message | 自定提交信息 (默认为 Site updated: {{ now('YYYY-MM-DD HH:mm:ss') }}) |

### FTPSync

安装 [hexo-deployer-ftpsync](https://github.com/hexojs/hexo-deployer-ftpsync)。

|  |
| --- |
| $ npm install hexo-deployer-ftpsync --save |

修改配置。

|  |
| --- |
| deploy:  type: ftpsync  host: <host>  user: <user>  pass: <password>  remote: [remote]  port: [port]  ignore: [ignore]  connections: [connections]  verbose: [true|false] |

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| host | 远程主机的地址 |  |
| user | 使用者名称 |  |
| pass | 密码 |  |
| remote | 远程主机的根目录 | / |
| port | 端口 | 21 |
| ignore | 忽略的文件或目录 |  |
| connections | 使用的连接数 | 1 |
| verbose | 显示调试信息 | false |

**FTP部署可能出现的问题**

您可能需要预先通过其他方式将所有文件上传到远程主机中。否则初次使用ftpsync插件就可能出现报错。另外，由于FTP协议的特征，它每传送一个文件就需要一次握手，相对速度较慢。

### SFTP

安装 [hexo-deployer-sftp](https://github.com/lucascaro/hexo-deployer-sftp)。

|  |
| --- |
| $ npm install hexo-deployer-sftp --save |

修改配置。

|  |
| --- |
| deploy:  type: sftp  host: <host>  user: <user>  pass: <password>  remotePath: [remote path]  port: [port]  privateKey: [path/to/privateKey]  passphrase: [passphrase]  agent: [path/to/agent/socket] |

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| host | 远程主机的地址 |  |
| user | 使用者名称 |  |
| pass | 密码 |  |
| remotePath | 远程主机的根目录 | / |
| port | 端口 | 22 |
| privateKey | ssh私钥的目录地址 |  |
| passphrase | （可省略）ssh私钥的密码短语 |  |
| agent | ssh套接字的目录地址 | $SSH\_AUTH\_SOCK |

### ZEIT Now

[ZEIT Now](https://zeit.co/) 是一个托管静态网站和 Serverless APIs 的云平台。你可以一键将你的 Hexo 网站部署到 ZEIT Now、并使用自己的域名或者由 ZEIT Now 提供的免费子域名 now.sh。

1. 安装 [Now CLI](https://zeit.co/download):

|  |
| --- |
| $ npm i -g now |

1. 在 package.json 中添加 npm script:

|  |
| --- |
| {  "scripts": {  "build": "hexo generate"  } } |

1. 在 Hexo 所在目录执行下述指令：

|  |
| --- |
| now |

你也可以使用下面的按钮在 ZEIT Now 创建一个 Hexo 项目。

### 21云盒子

1. 在 [21云盒子](https://www.21yunbox.com/) 中， 创建一个新的 静态网页，然后使用以下设置：

* **构建命令:** yarn && hexo deploy
* **发布目录:** public

1. 点击 “部署” 按钮！

样例已经部署在 <https://hexo.21yunbox.com/>.

请查看 [用 21云盒子 部署一个 Hexo 样例](https://www.21yunbox.com/docs/#/deploy-hexo).

### 其他方法

Hexo 生成的所有文件都放在 public 文件夹中，您可以将它们复制到您喜欢的地方。

## 永久链接（Permalinks）

可以在 \_config.yml 配置中调整网站的永久链接或者在每篇文章的 Front-matter 中指定。

#### 变量

除了下列变量外，您还可使用 Front-matter 中的所有属性。

| **变量** | **描述** |
| --- | --- |
| :year | 文章的发表年份（4 位数） |
| :month | 文章的发表月份（2 位数） |
| :i\_month | 文章的发表月份（去掉开头的零） |
| :day | 文章的发表日期 (2 位数) |
| :i\_day | 文章的发表日期（去掉开头的零） |
| :hour | 文章发表时的小时 (2 位数) |
| :minute | 文章发表时的分钟 (2 位数) |
| :title | 文件名称 |
| :post\_title | 文章标题 |
| :id | 文章 ID (not persistent across [*cache reset*](https://hexo.io/zh-cn/docs/commands#clean)) |
| :category | 分类。如果文章没有分类，则是 default\_category 配置信息。 |

您可在 permalink\_defaults 参数下调整永久链接中各变量的默认值：

|  |
| --- |
| permalink\_defaults:  lang: en |

#### 示例

假设 source/\_posts 文件夹中有个 hello-world.md，包含以下内容：

|  |
| --- |
| title: Hello World date: 2013-07-14 17:01:34 categories: - foo - bar |

| **参数** | **结果** |
| --- | --- |
| :year/:month/:day/:title/ | 2013/07/14/hello-world/ |
| :year-:month-:day-:title.html | 2013-07-14-hello-world.html |
| :category/:title/ | foo/bar/hello-world/ |

#### 多语种支持

若要建立一个多语种的网站，您可修改 new\_post\_name 和 permalink 参数，如下：

|  |
| --- |
| new\_post\_name: :lang/:title.md permalink: :lang/:title/ |

当您建立新文章时，文章会被储存到：

|  |
| --- |
| $ hexo new "Hello World" --lang tw  # => source/\_posts/tw/Hello-World.md |

而网址会是：

|  |
| --- |
| http://localhost:4000/tw/hello-world/ |

## 主题

创建 Hexo 主题非常容易，您只要在 themes 文件夹内，新增一个任意名称的文件夹，并修改 \_config.yml 内的 theme 设定，即可切换主题。一个主题可能会有以下的结构：

|  |
| --- |
| . ├── \_config.yml ├── languages ├── layout ├── scripts └── source |

#### \_config.yml

主题的配置文件。和 Hexo 配置文件不同，主题配置文件修改时会自动更新，无需重启 Hexo Server。

#### languages

语言文件夹。请参见 [国际化 (i18n)](https://hexo.io/zh-cn/docs/internationalization)。

#### layout

布局文件夹。用于存放主题的模板文件，决定了网站内容的呈现方式，Hexo 内建 [Swig](https://github.com/node-swig/swig-templates) 模板引擎，您可以另外安装插件来获得 [EJS](https://github.com/hexojs/hexo-renderer-ejs)、[Haml](https://github.com/hexojs/hexo-renderer-haml)、[Jade](https://github.com/hexojs/hexo-renderer-jade) 或 [Pug](https://github.com/maxknee/hexo-render-pug) 支持，Hexo 根据模板文件的扩展名来决定所使用的模板引擎，例如：

|  |
| --- |
| layout.ejs - 使用 EJS layout.swig - 使用 Swig |

您可参考 [模板](https://hexo.io/zh-cn/docs/templates) 以获得更多信息。

#### scripts

脚本文件夹。在启动时，Hexo 会载入此文件夹内的 JavaScript 文件，请参见 [插件](https://hexo.io/zh-cn/docs/plugins) 以获得更多信息。

#### source

资源文件夹，除了模板以外的 Asset，例如 CSS、JavaScript 文件等，都应该放在这个文件夹中。文件或文件夹开头名称为 \_（下划线线）或隐藏的文件会被忽略。

如果文件可以被渲染的话，会经过解析然后储存到 public 文件夹，否则会直接拷贝到 public 文件夹。

#### 发布

当您完成主题后，可以考虑将它发布到 [主题列表](https://hexo.io/themes)，让更多人能够使用您的主题。在发布前建议先进行 [主题单元测试](https://github.com/hexojs/hexo-theme-unit-test)，确保每一项功能都能正常使用。发布主题的步骤和 [更新文档](https://hexo.io/zh-cn/docs/contributing#%E6%9B%B4%E6%96%B0%E6%96%87%E6%A1%A3) 非常类似。

1. Fork [hexojs/site](https://github.com/hexojs/site)
2. 把库（repository）复制到电脑上，并安装所依赖的插件。

|  |
| --- |
| $ git clone https://github.com/<username>/site.git $ cd site $ npm install |

1. 编辑 source/\_data/themes.yml，在文件中新增您的主题，例如：

|  |
| --- |
| - name: landscape  description: A brand new default theme for Hexo.  link: https://github.com/hexojs/hexo-theme-landscape  preview: http://hexo.io/hexo-theme-landscape  tags:  - official  - responsive  - widget  - two\_column  - one\_column |

1. 在 source/themes/screenshots 新增同名的截图档案，图片必须为 800x500 的 PNG 文件。
2. 推送（push）分支。
3. 建立一个新的合并申请（pull request）并描述改动。

## 模版

模板决定了网站内容的呈现方式，每个主题至少都应包含一个 index 模板，以下是各页面相对应的模板名称：

| **模板** | **用途** | **回退** |
| --- | --- | --- |
| index | 首页 |  |
| post | 文章 | index |
| page | 分页 | index |
| archive | 归档 | index |
| category | 分类归档 | archive |
| tag | 标签归档 | archive |

### 布局（Layout）

如果页面结构类似，例如两个模板都有页首（Header）和页脚（Footer），您可考虑通过「布局」让两个模板共享相同的结构。一个布局文件必须要能显示 body 变量的内容，如此一来模板的内容才会被显示，举例来说：

index.ejs

|  |
| --- |
| index |

layout.ejs

|  |
| --- |
| <!DOCTYPE html> <html>  <body><%- body %></body> </html> |

生成：

|  |
| --- |
| <!DOCTYPE html> <html>  <body>index</body> </html> |

每个模板都默认使用 layout 布局，您可在 front-matter 指定其他布局，或是设为 false 来关闭布局功能，您甚至可在布局中再使用其他布局来建立嵌套布局。

### 局部模版（Partial）

局部模板让您在不同模板之间共享相同的组件，例如页首（Header）、页脚（Footer）或侧边栏（Sidebar）等，可利用局部模板功能分割为个别文件，让维护更加便利。举例来说：

partial/header.ejs

|  |
| --- |
| <h1 id="logo"><%= config.title %></h1> |

index.ejs

|  |
| --- |
| <%- partial('partial/header') %> <div id="content">Home page</div> |

生成：

|  |
| --- |
| <h1 id="logo">My Site</h1> <div id="content">Home page</div> |

#### 局部变量

您可以在局部模板中指定局部变量并使用。

partial/header.ejs

|  |
| --- |
| <h1 id="logo"><%= title %></h1> |

index.ejs

|  |
| --- |
| <%- partial('partial/header', {title: 'Hello World'}) %> <div id="content">Home page</div> |

生成：

|  |
| --- |
| <h1 id="logo">Hello World</h1> <div id="content">Home page</div> |

### 优化

如果您的主题太过于复杂，或是需要生成的文件量太过于庞大，可能会大幅降低性能，除了简化主题外，您可以考虑 Hexo 2.7 新增的局部缓存（Fragment Caching） 功能。

本功能借鉴于 [Ruby on Rails](http://guides.rubyonrails.org/caching_with_rails.html#fragment-caching)，它储存局部内容，下次便能直接使用缓存内容，可以减少文件夹查询并使生成速度更快。

它可用于页首、页脚、侧边栏等文件不常变动的位置，举例来说：

|  |
| --- |
| <%- fragment\_cache('header', function(){  return '<header></header>'; }); |

如果您使用局部模板的话，可以更简单：

|  |
| --- |
| <%- partial('header', {}, {cache: true}); |

fragment\_cache() 将会缓存第一次的渲染结果，并在之后直接输出缓存的结果。因此只有在不同页面的渲染结果都相同时才应使用局部缓存。  
比如，在配置中启用了 relative\_link 后不应该使用局部缓存，因为相对链接在每个页面可能不同。

## Hexo变量

#### 全局变量

| **变量** | **描述** | **类型** |
| --- | --- | --- |
| site | [网站变量](https://hexo.io/zh-cn/docs/variables#%E7%BD%91%E7%AB%99%E5%8F%98%E9%87%8F) | object; 见 [网站变量](https://hexo.io/zh-cn/docs/variables#%E7%BD%91%E7%AB%99%E5%8F%98%E9%87%8F) |
| page | 针对该页面的内容以及 front-matter 中自定义的变量。 | object; 见 [页面变量](https://hexo.io/zh-cn/docs/variables#%E9%A1%B5%E9%9D%A2%E5%8F%98%E9%87%8F) |
| config | 网站配置 | object (站点的配置文件) |
| theme | 主题配置。继承自网站配置。 | object (主题配置文件) |
| \_ (单下划线) | [Lodash](https://lodash.com/) 函数库 | [Lodash](https://lodash.com/) 文档 |
| path | 当前页面的路径（不含根路径） | string |
| url | 当前页面的完整网址 | string |
| env | 环境变量 | ??? |

#### 网站变量

| **变量** | **描述** | **类型** |
| --- | --- | --- |
| site.posts | 所有文章 | array of post objects |
| site.pages | 所有分页 | array of page objects |
| site.categories | 所有分类 | object，包含了站点全部的分类 |
| site.tags | 所有标签 | array，包含了站点全部的标签 |

#### 页面变量

**页面（page）**

| **变量** | **描述** | **类型** |
| --- | --- | --- |
| page.title | 页面标题 | string |
| page.date | 页面建立日期 | [Moment.js](http://momentjs.com/) 对象 |
| page.updated | 页面更新日期 | [Moment.js](http://momentjs.com/) 对象 |
| page.comments | 留言是否开启 | boolean |
| page.layout | 布局名称 | string |
| page.content | 页面的完整内容 | string |
| page.excerpt | 页面摘要 | string |
| page.more | 除了页面摘要的其余内容 | string |
| page.source | 页面原始路径 | string |
| page.full\_source | 页面的完整原始路径 | string |
| page.path | 页面网址（不含根路径）。我们通常在主题中使用 url\_for(page.path)。 | string |
| page.permalink | 页面的完整网址 | string |
| page.prev | 上一个页面。如果此为第一个页面则为 null。 | string or null |
| page.next | 下一个页面。如果此为最后一个页面则为 null。 | string or null |
| page.raw | 文章的原始内容 | ??? |
| page.photos | 文章的照片（用于相簿） | array |
| page.link | 文章的外部链接（用于链接文章） | string |

**文章 (post):** 与 page 布局相同，但新增以下变量。

| **变量** | **描述** | **类型** |
| --- | --- | --- |
| page.published | 如果该文章已发布则为 true | boolean |
| page.categories | 该文章的所有分类 | array of ??? |
| page.tags | 该文章的所有标签 | array of ??? |

**首页（index）**

| **变量** | **描述** | **类型** |
| --- | --- | --- |
| page.per\_page | 每页显示的文章数量 | number |
| page.total | 总页数 | number |
| page.current | 目前页数 | number |
| page.current\_url | 目前分页的网址 | string |
| page.posts | 本页文章 ([Data Model](https://hexojs.github.io/warehouse/)) | object |
| page.prev | 上一页的页数。如果此页是第一页的话则为 0。 | number |
| page.prev\_link | 上一页的网址。如果此页是第一页的话则为 ''。 | string |
| page.next | 下一页的页数。如果此页是最后一页的话则为 0。 | number |
| page.next\_link | 下一页的网址。如果此页是最后一页的话则为 ''。 | string |
| page.path | 当前页面的路径（不含根目录）。我们通常在主题中使用 url\_for(page.path)。 | string |

**归档 (archive)**：与 index 布局相同，但新增以下变量。

| **变量** | **描述** | **类型** |
| --- | --- | --- |
| page.archive | 等于 true | boolean |
| page.year | 年份归档 (4位) | number |
| page.month | 月份归档 (没有前导零的2位数) | number |

**分类 (category)**：与 index 布局相同，但新增以下变量。

| **变量** | **描述** | **类型** |
| --- | --- | --- |
| page.category | 分类名称 | string |

**标签 (tag)**：与 index 布局相同，但新增以下变量。

| **变量** | **描述** | **类型** |
| --- | --- | --- |
| page.tag | 标签名称 | string |

## 辅助函数（Helpers）

辅助函数帮助您在模版中快速插入内容。辅助函数不能在源文件中使用。

### 网址

#### url\_for

在路径前加上根路径，从 Hexo 2.7 开始您应该使用此函数而不是 config.root + path。

|  |
| --- |
| <%- url\_for(path, [option]) %> |

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| relative | 是否输出相对链接 | config.relative\_link 的值 |

**示例：**

|  |
| --- |
| \_config.yml root: /blog/ |

|  |
| --- |
| <%- url\_for('/a/path') %> // /blog/a/path |

是否输出相对链接，默认遵循配置文件中 relative\_link 的值  
例如， post/page 的相对路径值可能是 /foo/bar/index.html

|  |
| --- |
| \_config.yml relative\_link: true |

|  |
| --- |
| <%- url\_for('/css/style.css') %> // ../../css/style.css /\* 覆盖配置  \* 即使配置文件中启用了 relative\_link，你也可以使用 relative 参数禁用相对链接输出，反之亦然  \*/ <%- url\_for('/css/style.css', {relative: false}) %> // /css/style.css |

#### relative\_url

取得与 from 相对的 to 路径。

|  |
| --- |
| <%- relative\_url(from, to) %> |

**示例：**

|  |
| --- |
| <%- relative\_url('foo/bar/', 'css/style.css') %> // ../../css/style.css |

#### gravatar

根据邮箱地址返回 Gravatar 头像 URL。

如果你不指定 options 参数，将会应用默认参数。否则，你可以将其设置为一个数字，这个数字将会作为 Gravatar 的大小参数。最后，如果你设置它一个对象，它将会被转换为 Gravatar 的一个查询字符串参数。

|  |
| --- |
| <%- gravatar(email, [options]) %> |

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| s | 图片大小 | 80 |
| d | 默认头像 |  |
| f | 强制使用默认图象 |  |
| r | 头像等级限制 |  |

访问 [Gravatar](https://en.gravatar.com/site/implement/images/) 了解更多。

**示例：**

|  |
| --- |
| <%- gravatar('a@abc.com') %> // https://www.gravatar.com/avatar/b9b00e66c6b8a70f88c73cb6bdb06787  <%- gravatar('a@abc.com', 40) %> // https://www.gravatar.com/avatar/b9b00e66c6b8a70f88c73cb6bdb06787?s=40  <%- gravatar('a@abc.com' {s: 40, d: 'https://via.placeholder.com/150'}) %> // https://www.gravatar.com/avatar/b9b00e66c6b8a70f88c73cb6bdb06787?s=40&d=https%3A%2F%2Fvia.placeholder.com%2F150 |

#### full\_url\_for

在路径前加上根路径和域名。输出会被自动转码。

|  |
| --- |
| <%- full\_url\_for(path) %> |

**示例：**

|  |
| --- |
| \_config.yml url: https://example.com/blog # example |

|  |
| --- |
| <%- full\_url\_for('/a/path') %> // https://example.com/blog/a/path |

### HTML 标签

#### css

载入 CSS 文件。path 可以是数组或字符串，如果 path 开头不是 / 或任何协议，则会自动加上根路径；如果后面没有加上 .css 扩展名的话，也会自动加上。Use object type for custom attributes.

|  |
| --- |
| <%- css(path, ...) %> |

**示例：**

|  |
| --- |
| <%- css('style.css') %> // <link rel="stylesheet" href="/style.css">  <%- css(['style.css', 'screen.css']) %> // <link rel="stylesheet" href="/style.css"> // <link rel="stylesheet" href="/screen.css">  <%- css({ href: 'style.css', integrity: 'foo' }) %> // <link rel="stylesheet" href="/style.css" integrity="foo">  <%- css([{ href: 'style.css', integrity: 'foo' }, { href: 'screen.css', integrity: 'bar' }]) %> // <link rel="stylesheet" href="/style.css" integrity="foo"> // <link rel="stylesheet" href="/screen.css" integrity="bar"> |

#### js

载入 JavaScript 文件。path 可以是数组或字符串，如果 path 开头不是 / 或任何协议，则会自动加上根路径；如果后面没有加上 .js 扩展名的话，也会自动加上。Use object type for custom attributes.

|  |
| --- |
| <%- js(path, ...) %> |

**示例：**

|  |
| --- |
| <%- js('script.js') %> // <script src="/script.js"></script>  <%- js(['script.js', 'gallery.js']) %> // <script src="/script.js"></script> // <script src="/gallery.js"></script>  <%- js({ src: 'script.js', integrity: 'foo', async: true }) %> // <script src="/script.js" integrity="foo" async></script>  <%- js([{ src: 'script.js', integrity: 'foo' }, { src: 'gallery.js', integrity: 'bar' }]) %> // <script src="/script.js" integrity="foo"></script> // <script src="/gallery.js" integrity="bar"></script> |

#### link\_to

插入链接。

|  |
| --- |
| <%- link\_to(path, [text], [options]) %> |

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| external | 在新视窗打开链接 | false |
| class | Class 名称 |  |
| id | ID |  |

**示例：**

|  |
| --- |
| <%- link\_to('http://www.google.com') %> // <a href="http://www.google.com" title="http://www.google.com">http://www.google.com</a>  <%- link\_to('http://www.google.com', 'Google') %> // <a href="http://www.google.com" title="Google">Google</a>  <%- link\_to('http://www.google.com', 'Google', {external: true}) %> // <a href="http://www.google.com" title="Google" target="\_blank" rel="noopener">Google</a> |

#### mail\_to

插入电子邮箱链接。

|  |
| --- |
| <%- mail\_to(path, [text], [options]) %> |

| **参数** | **描述** |
| --- | --- |
| class | Class 名称 |
| id | ID |
| subject | 邮件主题 |
| cc | 抄送（CC） |
| bcc | 密送（BCC） |
| body | 邮件内容 |

**示例：**

|  |
| --- |
| <%- mail\_to('a@abc.com') %> // <a href="mailto:a@abc.com" title="a@abc.com">a@abc.com</a>  <%- mail\_to('a@abc.com', 'Email') %> // <a href="mailto:a@abc.com" title="Email">Email</a> |

#### image\_tag

插入图片。

|  |
| --- |
| <%- image\_tag(path, [options]) %> |

| **参数** | **描述** |
| --- | --- |
| alt | 图片的替代文字 |
| class | Class 名称 |
| id | ID |
| width | 图片宽度 |
| height | 图片高度 |

#### favicon\_tag

插入 favicon。

|  |
| --- |
| <%- favicon\_tag(path) %> |

#### feed\_tag

插入 feed 链接。

|  |
| --- |
| <%- feed\_tag(path, [options]) %> |

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| title | Feed 标题 |  |
| type | Feed 类型 | atom |

### 条件函数

#### is\_current

检查 path 是否符合目前页面的网址。开启 strict 选项启用严格比对。

|  |
| --- |
| <%- is\_current(path, [strict]) %> |

#### is\_home

检查当前页面是否为首页。

|  |
| --- |
| <%- is\_home() %> |

#### is\_post

检查当前页面是否为文章。

|  |
| --- |
| <%- is\_post() %> |

#### is\_page

检查当前页面是否为独立页面。

|  |
| --- |
| <%- is\_page() %> |

#### is\_archive

检查当前页面是否为存档页面。

|  |
| --- |
| <%- is\_archive() %> |

#### is\_year

检查当前页面是否为年度归档页面。

|  |
| --- |
| <%- is\_year() %> |

#### is\_month

检查当前页面是否为月度归档页面。

|  |
| --- |
| <%- is\_month() %> |

#### is\_category

检查当前页面是否为分类归档页面。  
如果给定一个字符串作为参数，将会检查目前是否为指定分类。

|  |
| --- |
| <%- is\_category() %> <%- is\_category('hobby') %> |

#### is\_tag

检查当前页面是否为标签归档页面。  
如果给定一个字符串作为参数，将会检查目前是否为指定标签。

|  |
| --- |
| <%- is\_tag() %> <%- is\_tag('hobby') %> |

### 字符串处理

#### trim

清除字符串开头和结尾的空格。

|  |
| --- |
| <%- trim(string) %> |

#### strip\_html

清除字符串中的 HTML 标签。

|  |
| --- |
| <%- strip\_html(string) %> |

**示例：**

|  |
| --- |
| <%- strip\_html('It\'s not <b>important</b> anymore!') %> // It's not important anymore! |

#### titlecase

把字符串转换为正确的 Title case。

|  |
| --- |
| <%- titlecase(string) %> |

**示例：**

|  |
| --- |
| <%- titlecase('this is an apple') %> # This is an Apple |

#### markdown

使用 Markdown 解析字符串。

|  |
| --- |
| <%- markdown(str) %> |

**示例：**

|  |
| --- |
| <%- markdown('make me \*\*strong\*\*') %> // make me <strong>strong</strong> |

#### render

解析字符串。

|  |
| --- |
| <%- render(str, engine, [options]) %> |

**Examples:**

|  |
| --- |
| <%- render('p(class="example") Test', 'pug'); %> // <p class="example">Test</p> |

See [Rendering](https://hexo.io/zh-cn/api/rendering) for more details.

#### word\_wrap

使每行的字符串长度不超过 length。length 预设为 80。

|  |
| --- |
| <%- word\_wrap(str, [length]) %> |

**示例：**

|  |
| --- |
| <%- word\_wrap('Once upon a time', 8) %> // Once upon\n a time |

#### truncate

移除超过 length 长度的字符串。length 的默认值是 30。

|  |
| --- |
| <%- truncate(text, length) %> |

**示例：**

|  |
| --- |
| <%- truncate('Once upon a time in a world far far away', {length: 17}) %> // Once upon a ti...  <%- truncate('Once upon a time in a world far far away', {length: 17, separator: ' '}) %> // Once upon a...  <%- truncate('And they found that many people were sleeping better.', {length: 25, omission: '... (continued)'}) %> // And they f... (continued) |

### 模板

#### partial

载入其他模板文件，您可在 locals 设定区域变量。

|  |
| --- |
| <%- partial(layout, [locals], [options]) %> |

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| cache | 缓存（使用 Fragment cache） | false |
| only | 限制局部变量。在模板中只能使用 locals 中设定的变量。 | false |

#### fragment\_cache

局部缓存。它储存局部内容，下次使用时就能直接使用缓存。

|  |
| --- |
| <%- fragment\_cache(id, fn); |

**示例：**

|  |
| --- |
| <%- fragment\_cache('header', function(){  return '<header></header>'; }) %> |

### 日期与时间

#### date

插入格式化的日期。date 可以是 UNIX 时间、ISO 字符串、Date 对象或 [Moment.js](http://momentjs.com/) 对象。format 默认为 date\_format 配置信息。

|  |
| --- |
| <%- date(date, [format]) %> |

**示例：**

|  |
| --- |
| <%- date(Date.now()) %> // 2013-01-01  <%- date(Date.now(), 'YYYY/M/D') %> // Jan 1 2013 |

#### date\_xml

插入 XML 格式的日期。date 可以是 UNIX 时间、ISO 字符串、Date 对象或 [Moment.js](http://momentjs.com/) 对象。

|  |
| --- |
| <%- date\_xml(date) %> |

**示例：**

|  |
| --- |
| <%- date\_xml(Date.now()) %> // 2013-01-01T00:00:00.000Z |

#### time

插入格式化的时间。date 可以是 UNIX 时间、ISO 字符串、Date 对象或 [Moment.js](http://momentjs.com/) 对象。format 默认为 time\_format 配置信息。

|  |
| --- |
| <%- time(date, [format]) %> |

**示例：**

|  |
| --- |
| <%- time(Date.now()) %> // 13:05:12  <%- time(Date.now(), 'h:mm:ss a') %> // 1:05:12 pm |

#### full\_date

插入格式化的日期和时间。date 可以是 UNIX 时间、ISO 字符串、Date 对象或 [Moment.js](http://momentjs.com/) 对象。format默认为 date\_format + time\_format。

|  |
| --- |
| <%- full\_date(date, [format]) %> |

**示例：**

|  |
| --- |
| <%- full\_date(new Date()) %> // Jan 1, 2013 0:00:00  <%- full\_date(new Date(), 'dddd, MMMM Do YYYY, h:mm:ss a') %> // Tuesday, January 1st 2013, 12:00:00 am |

#### moment

[Moment.js](http://momentjs.com/) 函数库。

### 列表

#### list\_categories

插入分类列表。

|  |
| --- |
| <%- list\_categories([options]) %> |

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| orderby | 分类排列方式 | name |
| order | 分类排列顺序。1, asc 升序；-1, desc 降序。 | 1 |
| show\_count | 显示每个分类的文章总数 | true |
| style | 分类列表的显示方式。使用 list 以无序列表（unordered list）方式显示。 | list |
| separator | 分类间的分隔符号。只有在 style 不是 list 时有用。 | , |
| depth | 要显示的分类层级。0 显示所有层级的分类；-1 和 0 很类似，但是显示不分层级；1 只显示第一层的分类。 | 0 |
| class | 分类列表的 class 名称。 | category |
| transform | 改变分类名称显示方法的函数 |  |
| suffix | 为链接添加前缀 | None |

**用例:**

|  |
| --- |
| <%- list\_categories(post.categories, {  class: 'post-category',  transform(str) {  return titlecase(str);  } }) %>  <%- list\_categories(post.categories, {  class: 'post-category',  transform(str) {  return str.toUpperCase();  } }) %> |

#### list\_tags

插入标签列表。

|  |
| --- |
| <%- list\_tags([options]) %> |

| **选项** | **描述** | **预设值** |
| --- | --- | --- |
| orderby | 标签排列方式 | name |
| order | 标签排列顺序。1, asc 升序；-1, desc 降序。 | 1 |
| show\_count | 显示每个标签的文章总数 | true |
| style | 标签列表的显示方式。使用 list 以无序列表（unordered list）方式显示。 | list |
| separator | 标签间的分隔符号。只有在 style 不是 list 时有用。 | , |
| class | 标签列表的 class 名称。 | tag |
| transform | 改变标签名称显示方法的函数。请查看 [list\_categories](https://hexo.io/zh-cn/docs/helpers#list-categories) 中给出的例子 |  |
| amount | 要显示的标签数量（0 = 无限制） | 0 |
| suffix | 为链接添加前缀 | None |

#### list\_archives

插入归档列表。

|  |
| --- |
| <%- list\_archives([options]) %> |

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| type | 类型。此设定可为 yearly 或 monthly。 | monthly |
| order | 排列顺序。1, asc 升序；-1, desc 降序。 | 1 |
| show\_count | 显示每个归档的文章总数 | true |
| format | 日期格式 | MMMM YYYY |
| style | 归档列表的显示方式。使用 list 以无序列表（unordered list）方式显示。 | list |
| separator | 归档间的分隔符号。只有在 style 不是 list 时有用。 | , |
| class | 归档列表的 class 名称。 | archive |
| transform | 改变归档名称显示方法的函数。请查看 [list\_categories](https://hexo.io/zh-cn/docs/helpers#list-categories) 中给出的例子 |  |

#### list\_posts

插入文章列表。

|  |
| --- |
| <%- list\_posts([options]) %> |

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| orderby | 文章排列方式 | date |
| order | 文章排列顺序。1, asc 升序；-1, desc 降序。 | -1 |
| style | 文章列表的显示方式。使用 list 以无序列表（unordered list）方式显示。 | list |
| separator | 文章间的分隔符号。只有在 style 不是 list 时有用。 | , |
| class | 文章列表的 class 名称。 | post |
| amount | 要显示的文章数量（0 = 无限制） | 6 |
| transform | 改变文章名称显示方法的函数。请查看 [list\_categories](https://hexo.io/zh-cn/docs/helpers#list-categories) 中给出的例子 |  |

#### tagcloud

插入标签云。

|  |
| --- |
| <%- tagcloud([tags], [options]) %> |

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| min\_font | 最小字体尺寸 | 10 |
| max\_font | 最大字体尺寸 | 20 |
| unit | 字体尺寸的单位 | px |
| amount | 标签总量 | 40 |
| orderby | 标签排列方式 | name |
| order | 标签排列顺序。1, sac 升序；-1, desc 降序 | 1 |
| color | 使用颜色 | false |
| start\_color | 开始的颜色。您可使用十六进位值（#b700ff），rgba（rgba(183, 0, 255, 1)），hsla（hsla(283, 100%, 50%, 1)）或 [颜色关键字](http://www.w3.org/TR/css3-color/#svg-color)。此变量仅在 color 参数开启时才有用。 |  |
| end\_color | 结束的颜色。您可使用十六进位值（#b700ff），rgba（rgba(183, 0, 255, 1)），hsla（hsla(283, 100%, 50%, 1)）或 [颜色关键字](http://www.w3.org/TR/css3-color/#svg-color)。此变量仅在 color 参数开启时才有用。 |  |

### 其他

#### paginator

插入分页链接。

|  |
| --- |
| <%- paginator(options) %> |

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| base | 基础网址 | / |
| format | 网址格式 | page/%d/ |
| total | 分页总数 | 1 |
| current | 目前页数 | 0 |
| prev\_text | 上一页链接的文字。仅在 prev\_next 设定开启时才有用。 | Prev |
| next\_text | 下一页链接的文字。仅在 prev\_next 设定开启时才有用。 | Next |
| space | 空白文字 | … |
| prev\_next | 显示上一页和下一页的链接 | true |
| end\_size | 显示于两侧的页数 | 1 |
| mid\_size | 显示于中间的页数 | 2 |
| show\_all | 显示所有页数。如果开启此参数的话，end\_size 和 mid\_size 就没用了。 | false |
| escape | Escape HTML tags | true |

**Examples:**

|  |
| --- |
| <%- paginator({  prev\_text: '<',  next\_text: '>' }) %> |

|  |
| --- |
| <!-- Rendered as --> <a href="/1/">&lt;</a> <a href="/1/">1</a> 2 <a href="/3/">3</a> <a href="/3/">&gt;</a> |

|  |
| --- |
| <%- paginator({  prev\_text: '<i class="fa fa-angle-left"></i>',  next\_text: '<i class="fa fa-angle-right"></i>',  escape: false }) %> |

|  |
| --- |
| <!-- Rendered as --> <a href="/1/"><i class="fa fa-angle-left"></i></a> <a href="/1/">1</a> 2 <a href="/3/">3</a> <a href="/3/"><i class="fa fa-angle-right"></i></a> |

#### search\_form

插入 Google 搜索框。

|  |
| --- |
| <%- search\_form(options) %> |

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| class | 表单的 class name | search-form |
| text | 搜索提示文字 | Search |
| button | 显示搜索按钮。此参数可为布尔值（boolean）或字符串，当设定是字符串的时候，即为搜索按钮的文字。 | false |

#### number\_format

格式化数字。

|  |
| --- |
| <%- number\_format(number, [options]) %> |

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| precision | 数字精度。此选项可为 false 或非负整数。 | false |
| delimiter | 千位数分隔符号 | , |
| separator | 整数和小数之间的分隔符号 | . |

**示例：**

|  |
| --- |
| <%- number\_format(12345.67, {precision: 1}) %> // 12,345.68  <%- number\_format(12345.67, {precision: 4}) %> // 12,345.6700  <%- number\_format(12345.67, {precision: 0}) %> // 12,345  <%- number\_format(12345.67, {delimiter: ''}) %> // 12345.67  <%- number\_format(12345.67, {separator: '/'}) %> // 12,345/67 |

#### meta\_generator

Inserts [generator tag](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/meta).

|  |
| --- |
| <%- meta\_generator() %> |

**Examples:**

|  |
| --- |
| <%- meta\_generator() %> // <meta name="generator" content="Hexo 4.0.0"> |

#### open\_graph

插入 open graph 资源。

|  |
| --- |
| <%- open\_graph([options]) %> |

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| title | 页面标题 (og:title) | page.title |
| type | 页面类型 (og:type) | blog |
| url | 页面网址 (og:url) | url |
| image | 页面图片 (og:image) | 内容中的图片 |
| site\_name | 网站名称 (og:site\_name) | config.title |
| description | 页面描述 (og:description) | 内容摘要或前 200 字 |
| twitter\_card | Twitter 卡片类型 (twitter:card) | summary |
| twitter\_id | Twitter ID (twitter:creator) |  |
| twitter\_site | Twitter 网站 (twitter:site) |  |
| google\_plus | Google+ 个人资料链接 |  |
| fb\_admins | Facebook 管理者 ID |  |
| fb\_app\_id | Facebook 应用程序 ID |  |

#### toc

解析内容中的标题标签 (h1~h6) 并插入目录。

|  |
| --- |
| <%- toc(str, [options]) %> |

| **参数** | **描述** | **默认值** |
| --- | --- | --- |
| class | Class 名称 | toc |
| list\_number | 显示编号 | true |
| max\_depth | 生成 TOC 的最大深度 | 6 |
| min\_depth | 生成 TOC 的最小深度 | 1 |

**示例：**

|  |
| --- |
| <%- toc(page.content) %> |

## 插件

Hexo 有强大的插件系统，使您能轻松扩展功能而不用修改核心模块的源码。在 Hexo 中有两种形式的插件：

#### 脚本（Scripts）

如果您的代码很简单，建议您编写脚本，您只需要把 JavaScript 文件放到 scripts 文件夹，在启动时就会自动载入。

#### 插件（Packages）

如果您的代码较复杂，或是您想要发布到 NPM 上，建议您编写插件。首先，在 node\_modules 文件夹中建立文件夹，文件夹名称开头必须为 hexo-，如此一来 Hexo 才会在启动时载入否则 Hexo 将会忽略它。

文件夹内至少要包含 2 个文件：一个是主程序，另一个是 package.json，描述插件的用途和所依赖的插件。

|  |
| --- |
| . ├── index.js └── package.json |

package.json 中至少要包含 name, version, main 属性，例如：

package.json

|  |
| --- |
| {  "name": "hexo-my-plugin",  "version": "0.0.1",  "main": "index" } |

#### 工具

您可以使用 Hexo 提供的官方工具插件来加速开发：

* [hexo-fs](https://github.com/hexojs/hexo-fs)：文件 IO
* [hexo-util](https://github.com/hexojs/hexo-util)：工具程式
* [hexo-i18n](https://github.com/hexojs/hexo-i18n)：本地化（i18n）
* [hexo-pagination](https://github.com/hexojs/hexo-pagination)：生成分页资料

#### 发布

当您完成插件后，可以考虑将它发布到 [插件列表](https://hexo.io/plugins)，让更多人能够使用您的插件。发布插件的步骤和 [更新文件](https://hexo.io/zh-cn/docs/contributing#%E6%9B%B4%E6%96%B0%E6%96%87%E4%BB%B6)非常类似。

1. Fork [hexojs/site](https://github.com/hexojs/site)
2. 把库（repository）复制到电脑上，并安装所依赖的插件。

|  |
| --- |
| $ git clone https://github.com/<username>/site.git $ cd site $ npm install |

1. 编辑 source/\_data/plugins.yml，在档案中新增您的插件，例如：

|  |
| --- |
| - name: hexo-server  description: Server module for Hexo.  link: https://github.com/hexojs/hexo-server  tags:  - official  - server  - console |

1. 推送（push）分支。
2. 建立一个新的合并申请（pull request）并描述改动。

## 国际化（i18n）

若要让您的网站以不同语言呈现，您可使用国际化（internationalization）功能。请先在 \_config.yml 中调整 language 设定，这代表的是预设语言，您也可设定多个语言来调整预设语言的顺位。

|  |
| --- |
| language: zh-tw  language:  - zh-tw - en |

#### 语言文件

语言文件可以使用 YAML 或 JSON 编写，并放在主题文件夹中的 languages 文件夹。您可以在语言文件中使用 [printf 格式](https://github.com/alexei/sprintf.js)。

#### 模板

在模板中，透过 \_\_ 或 \_p 辅助函数，即可取得翻译后的字符串，前者用于一般使用；而后者用于复数字符串。例如：

en.yml

|  |
| --- |
| index:  title: Home  add: Add  video:  zero: No videos  one: One video  other: %d videos |

|  |
| --- |
| <%= \_\_('index.title') %> // Home  <%= \_p('index.video', 3) %> // 3 videos |

#### 路径

您可在 front-matter 中指定该页面的语言，也可在 \_config.yml 中修改 i18n\_dir 设定，让 Hexo 自动侦测。

|  |
| --- |
| i18n\_dir: :lang |

i18n\_dir 的预设值是 :lang，也就是说 Hexo 会捕获网址中的第一段以检测语言，举例来说：

|  |
| --- |
| /index.html => en /archives/index.html => en /zh-tw/index.html => zh-tw |

捕获到的字符串唯有在语言文件存在的情况下，才会被当作是语言，因此例二 /archives/index.html 中的 archives 就不被当成是语言。

## Hexo多语言化

### 1.前言

大家都知道Hexo是一款很出色的生成静态网站的组件，对于初学者或量比较小的用户来说，静态网站托管可满足大部分需要。目前的托管平台有Github Page, Gitlab Page, 都大同小异，看自己的博客定位如果是要方便国外友人访问则最好是Github, 如果主要是面向国内用户则Gitlab也可以满足需求。

Hexo有很多Theme, 可在官网详细了解，适合不同风格的用户所需。对于本人来说，选型的标准主要是：简洁，大方，快速。基于此标准主要尝试的主题有:

* **maupassant**: 界面清爽，无多余内容，不过可惜对多语系博客支持不太好，只有简单的i18n国际化，不满足需求 ;
* **nova**: 尝试的另一款，作者很用心，把类似于博客园的风格借鉴到了hexo，号称支持双语系，不过尝试过程中发现切换语系无法切换中英博客，且两种语系博客无论怎么设置，都只能显示在一个列表中，所以也不满足需求;
* **minos**: 风格类似于maupassant, 比它强的是多语系支持这块比较完善，在尝试过后，正符合多语系的场景，遂采纳。

### 2.前置条件

#### 2.1 环境准备

* **OS**: Darwin MacBook-Pro-2.local 17.7.0
* **NodeJS**: v10.10.0
* **Hexo**: 3.3.9
* **Github page**: 开通github账号

#### 2.2 软件安装

此步略，参考网上教程。安装好后，执行Hexo命令初始化一个静态网站目录:

hexo init hexo\_blog

#### 2.3 hexo依赖

**minos**: 从官方Github下下载

cd hexo\_blog

git clone https://github.com/ppoffice/hexo-theme-minos.git themes/minos

#### 2.4 npm依赖

* **hexo-renderer-sass**

cd hexo\_blog

npm install hexo-renderer-sass --save

* **cheerio**

cd hexo\_blog

npm install cheerio --save

### 3.折腾过程

整个过程分为几个部分:

* **全站配置**: 配置站点
* **主题配置**: 配置minos
* **语系配置**: 配置切换语系

#### 3.1 全站配置

这里主要考虑和语系有关的配置，其它配置不在本文讨论范畴，大家可参照网上配置自由设定。全站配置文件为hexo\_blog/\_config.yml。配置过程如下：

##### 3.1.1 语系配置

#配置默认以English为主，中文为辅

language: [en,zh-cn]

##### 3.1.2 主题配置

#配置主题为minos

theme: minos

##### 3.1.3 url配置

#配置自己的静态网站部署地址

url: https://zendwind.github.io

##### 3.1.4 deploy配置

deploy:

#配置git部署模式

type: git

repo: https://github.com/zendwind/zendwind.github.io.git

branch: master

#### 3.2 主题配置

主题配置主要涉及语系方面，步骤如下:

##### 3.2.1 主题配置生成

cd test/themes/minos

#默认主题配置名称为\_config.yml.example

rsync -av \_config.yml.example \_config.yml

#生成英语系配置

rsync -av \_config.yml.example \_config.en.yml

#生成中文语系配置

rsync -av \_config.yml.example \_config.zh-cn.yml

##### 3.2.2 多语系配置

* \_英文语系配置: config.en.yml

menu:

Archives: /archives

Categories: /categories

Tags: /tags

About: /about

* \_中文语系配置: config.zh-cn.yml

menu:

归档: /zh-cn/zh-cn/archives #这里是个bug, 如果不加双层zh-cn后面无法定位到archives目录

分类: /zh-cn/categories

关于: /zh-cn/about

#### 3.3 站点文档配置

上面已经把hexo配置完成了，接下来为了要实现双语博客的访问，需要配置文档目录。首先，我们来看下没配置之前的站点目录结构是怎么样的：

.

source

└── \_posts

└── hello-world.md

在站点根目录下，有一个source目录，同时里面又有一个\_post子目录，初始化时会生成一个hello-world.md的文档。由于我们站点的默认语系是English, 所以在\_post目录下存放的就是英文文档。

##### 3.3.1 source/\_post目录文档配置

在source/\_posts目录下，新增一个zh-cn的目录，专门用于存放中文文档。在zh-cn目录下新增一个中文文档命名为hello-wolrd-cn.md。此时目录结构如下:

.

source

└── \_posts

├── hello-world.md

└── zh-cn

└── hello-world-cn.md

##### 3.3.2 source目录文档配置

在\_post增加完中文文档目录后，在source目录直接再新增一个zh-cn目录，用于存放其它一些中文文档,　如about之类的。同时在source目录下新增一个about目录用于存放英文的about文档，目录结构如下:

source

├── \_posts

│   ├── hello-world.md

│   └── zh-cn

│   └── hello-world-cn.md

├── about

│   └── index.md

└── zh-cn

└── about

└── index.md

关于about中的index.md, 其内容如下:

* **英文语系**

---

layout: "about"

title: 'About Me'

date: 2020-04-24 10:00:00

---

* **中文语系**

---

layout: "about"

title: '关于我'

date: 2020-04-26 10:00:00

--

测试About页面

Test About Page  
这样初步完成站点文档的目录设置。

##### 3.3.3 增加多语系切换配置

此步，需要修改下脚本，把语系相关代码调整到合适位置，如在themes/minos/layout/common/footer.ejs脚本中，找到:

<%- partial('common/languages') %>

把上述代码剪切到themes/minos/layout/common/navbar.ejs的合适位置（75行）,　如下：

<% } %>

<%- partial('common/languages') %>

</div>

</div>

</nav>

同时找到themes/minos/layout/common/languages.ejs, 增加一个配置项style="top:100%" 到如下代码中:

<div class="dropdown-menu has-text-left" role="menu" style="top:100%">

#### 3.4 查看效果

##### 3.4.1 编译静态文件

cd hexo\_blog

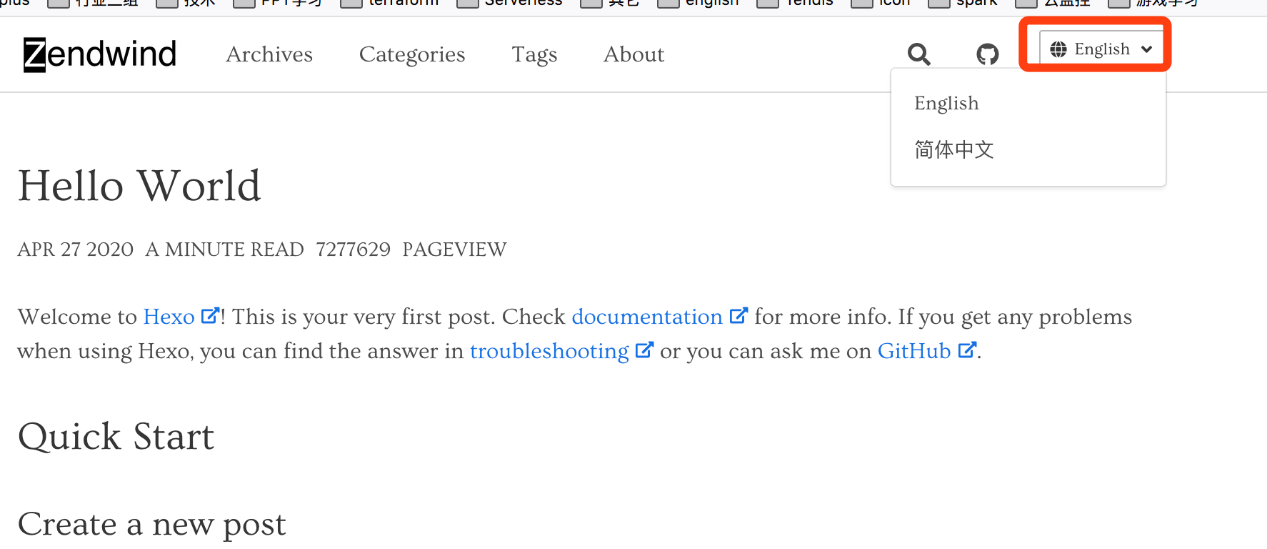
hexo g

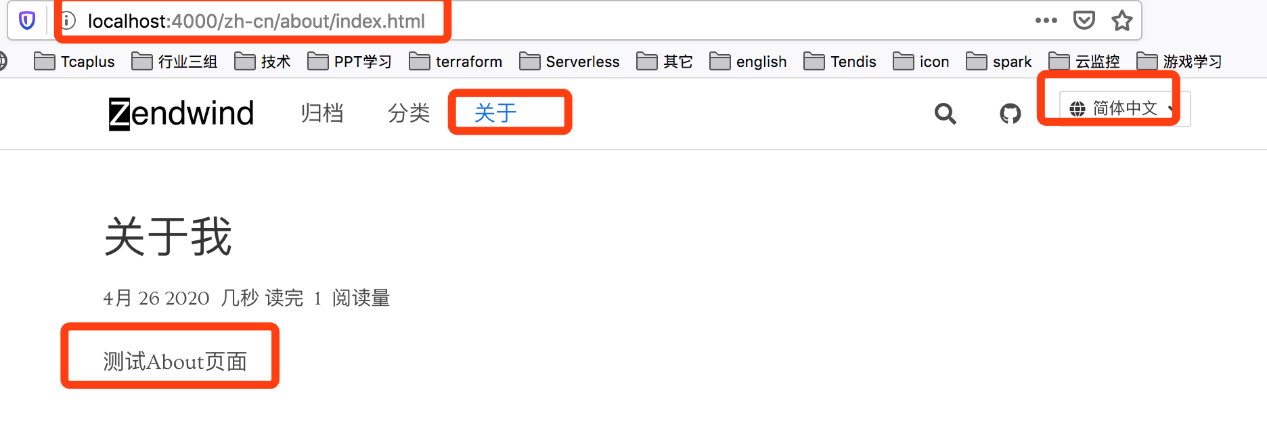
##### 3.4.2 本地测试

cd hexo\_blog

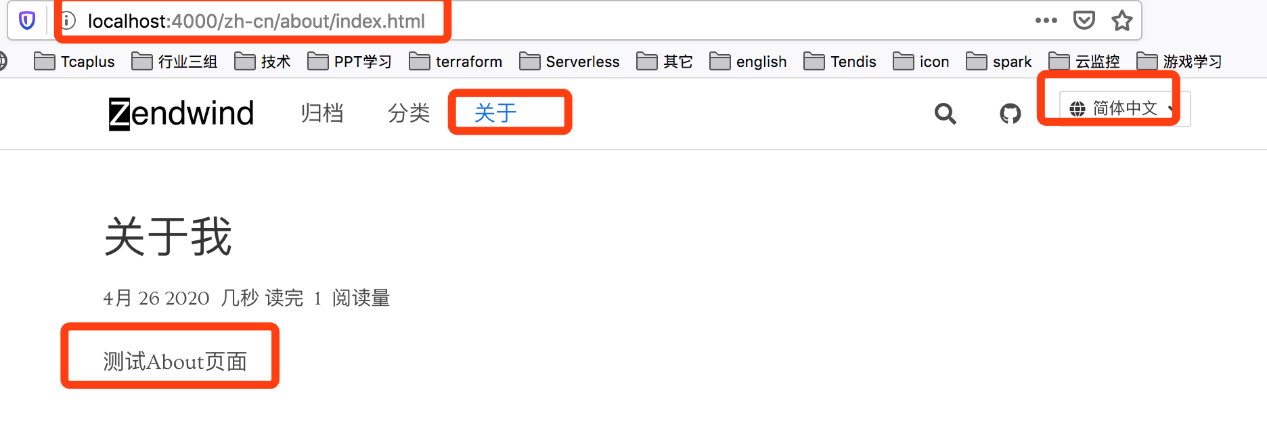
hexo s

打开浏览器，查看localhost:4000网页是否正常。

打开默认的英文语系站点测试效果如下:  




切换到中文语系站点测试效果如下:  

#### 3.5 部署到github

如果查看效果ok, 可以把站点部署到github上，执行命令如下:

hexo d -g

备注：在部署之前，需要检查是否有创建好站点仓库（在配置中指定的repo），　如https://github.com/zendwind/zendwind.github.io.git。

然后通过zendwind.github.io直接访问站点，看是否ok。

### 4. 其它功能

如果要为你博客添加评论系统、添加计数及分享功能，在hexo中也比较方便。

### 4.1 添加评论功能

评论系统有很多，主流的有两种：

* **国外**: Disqus, 这个在国内访问很慢
* **国内**: Valine, 这个在国内访问比较快  
  本文采用Valine模式来设置评论功能，Valine已和LeanCloud打通，即评论数据可放在LeanCloud存储，且是免费的，方便广大初学者学习使用。设置步骤如下：

##### 4.1.1 创建应用

登录leancloud.com官网，注册账号,　登录后创建一个开发版应用，如下图所示:  


##### 4.1.2 查看应用app\_id和app\_key



##### 4.1.3 配置应用key

在主题配置中配置comment：\_config.en.yml 和\_config.zh-cn.yml

comments:

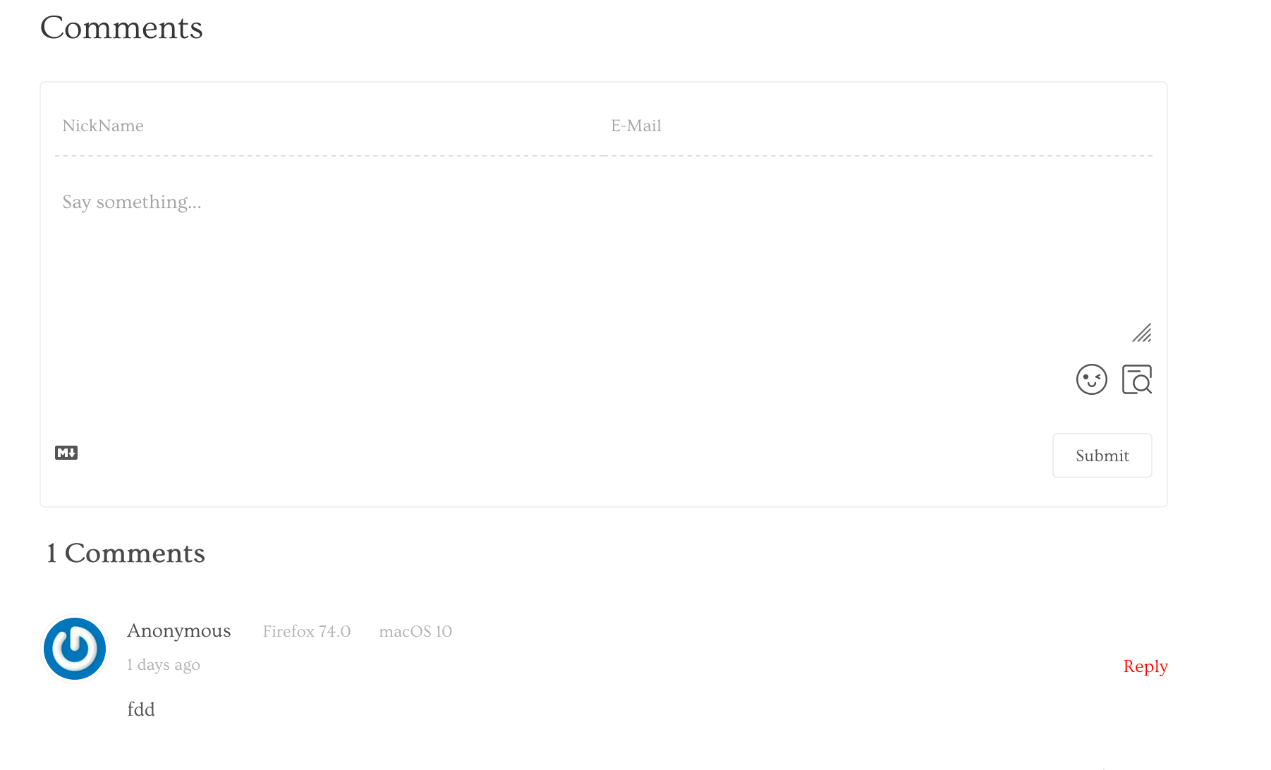
type: valine

app\_id: xxx #replace your app id

app\_key: xxx # replace your app key

##### 4.1.4 文章开启comment或关闭comment

在文档开头可以通过设置comments: false来关闭文章评论

评论效果类似如下:  


#### 4.2 添加文章访问量及站点访问统计功能

一般有两种模式：

* **Valine模式**: 集成Vistor统计功能，与leancloud打通，把统计数据存储到leancloud
* **不蒜子模式**: 简单，直接集成脚本即可。

本文主要介绍用不蒜子模式来统计访问量，步骤如下：

##### 4.2.1 修改主题配置

修改主题配置 ：\_config.en.yml和\_config.zh-cn.yml，增加如下内容：

article:

#增加一个pageview变量，用来控制是否显示文章访问量

pageview: true

##### 4.2.2 增加busuanzi脚本

在主题目录下，找到themes/minos/layout/common/head.ejs, 在最后面添加如下代码：

<% if (!has\_config('article.pageview') || get\_config('article.pageview') === true) { %>

<script async="" src="//busuanzi.ibruce.info/busuanzi/2.3/busuanzi.pure.mini.js"></script>

<% } %>

##### 4.2.3 增加访问量统计

在主题目录下，找到themes/minos/layout/common/article.ejs, 在合适位置(第28行后），添加如下代码：

<% if (!has\_config('article.pageview') || get\_config('article.pageview') === true) { %>

<span class="column is-narrow" id="busuanzi\_value\_page\_pv">0</span>

<span class="column is-narror"> <%=`${ \_\_('article.vistor') }` %></span>

<% } %>

##### 4.2.4 修改多语系配置

在themes/minos/languages目录下，分别修改en.yml和zh-cn.yml,增加两个对应项，如下所示:

* **en.yml**

article:

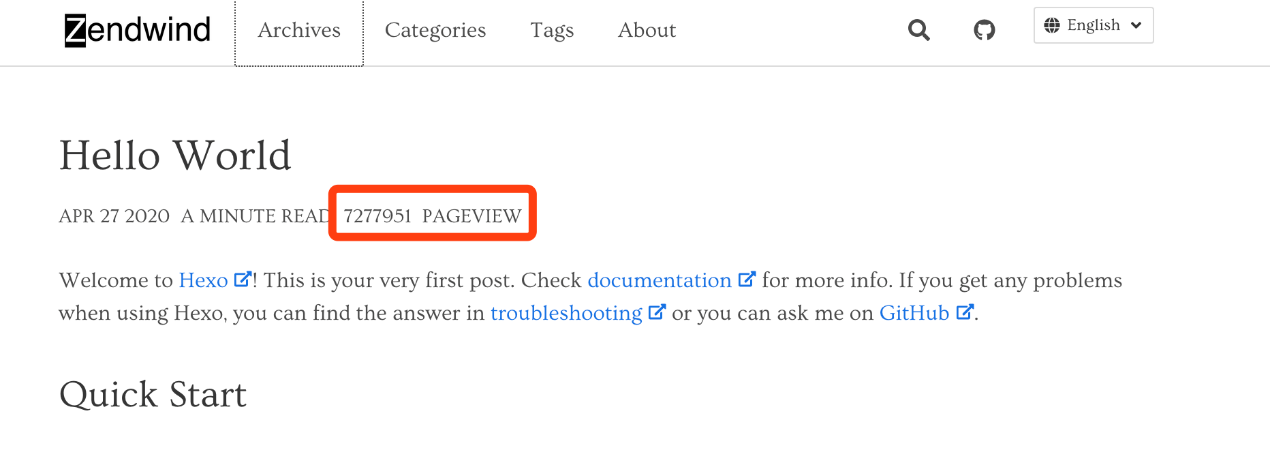
vistor: PageView

* **zh-cn.yml**

article:

vistor: 阅读量

##### 4.2.5 查看效果

添加完后，类似如下截图显示，文章访问量：  


### 5.总结

本文主要描述了关于Hexo如何搭建多语系方面的内容，同时结合很常见的一些功能展开了介绍，希望大家能从本文中获得一些帮助。

## Hexo的多语言配置

关于**Hexo**的中英文语言配置，我的做法是，在菜单栏添加一个按键，通过按键切换到中/英文界面，让中文文章只显示在中文页面，英文显示在英文页面。

### 插件安装

首先，安装**hexo**-generator-i18n插件

|  |  |
| --- | --- |
| 1 | npm install **hexo**-generator-i18n --save |

然后修改\_config.yml

|  |  |
| --- | --- |
| 1 2 3 4 5 6 | language: [zh-CN,en] # **hexo**-generator-i18n 选项（可选，默认使用如下设置） i18n:  type: [page, post]  generator: [index, archive, category, tag] |

这里并没有明白tyte和generator的作用，但是似乎不影响

之后是**在主题languages目录下添加对应的语言.yml文件(如zh.yml, en.yml)**

但是由于我使用的主题并不支持**多**语言，我也懒得弄，所有暂时没有管这一项

插件安装完成后，执行**hexo** clean & **hexo** g

会生成public/en目录，对于\_post内的所有博文，都会在public里存放一份，en下存放一份。

那么就需要根据页面的语言，显示不同的博文。

### 博文配置

对于所有的markdown文章，都需要在文章内添加属性lang:zh-CN or lang:en，来区分中英文

### 网页配置

我通过url来区分中英文页面，对于中文，url是www.yoursite.com，英文，url是www.yoursite.com/en/,

这样，在js中检索url是否包含en，来判断显示中文还是英文界面。

**修改主题**, 我的主题路径是themes/vexo/layout/index.ejs

获取当前html的路径，并判断路径内是否存在en

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 | <main class="app-body">  <% var currentURL = page.path.split('/') %>  <% if (currentURL.length === 1) { %>  <% var lang = page.lang %>  <% } else { %>  <% var lang = currentURL.shift() %>  <% } %>  <% page.posts.each(function(post) { %>  <% if (lang == post.lang) { %>  <article class="article-card">  <h2 class="article-head">  <a href="<%- url\_for(post.path) %>"><%- post.title %></a>  </h2>  <p class="article-date"><%- date(post.date, "LL") %></p>  <% if (post.tags && post.tags.length) { %>  <%- partial('\_partial/tag', { tags: post.tags }) %>  <% } %>  <div class="article-summary">  <% if (post.excerpt) { %>  <%- post.excerpt %>  <% } else { %>  <%- truncate(strip\_html(post.content), { length: 150, omission: ' ...' }) %>  <% } %>  </div>  <a class="more" href="<%- url\_for(post.path) %>"><%- theme.excerpt\_link %></a>  </article>  <% } %>  <% }) %>   <% if (page.total > 1){ %>  <%- partial('\_partial/pager') %>  <% } %> </main> |

上述都完成，就可以通过www.yoursite.com/en进入英文网页，且该网页只有英文文章

### 添加切换语言按钮

修改theme下的\_config.yml文件, 添加一个EN的按钮，跳转到英文界面

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 | menu:  Home: /  Tags: /tags/  Archives: /archives/  Projects: /project/  About: /about/  EN: /en/ |

这里为了简单，没有动态修改EN这个按钮，提供从英文网页，跳转回中文网页的方式(实际上，点击Home按键，就跳转回中文了)

### 部署测试

|  |  |
| --- | --- |
| 1 2 | **hexo** clean & **hexo** g **hexo** d |

#### 

#### 

#### 

## 更新 Hexo 官方插件

我们也欢迎给 [Hexo 官方插件](https://github.com/hexojs) 提交 PR 和 Issue 🤗

## 问题解答

在使用 Hexo 时，您可能会遇到一些问题，下列的常见问题解答可能会对您有所帮助。如果您在这里找不到解答，可以在 [GitHub](https://github.com/hexojs/hexo/issues) 或 [Google Group](https://groups.google.com/group/hexo) 上提问。

### YAML 解析错误

|  |
| --- |
| JS-YAML: incomplete explicit mapping pair; a key node is missed at line 18, column 29:  last\_updated: Last updated: %s |

如果 YAML 字符串中包含冒号（:）的话，请加上引号。

|  |
| --- |
| JS-YAML: bad indentation of a mapping entry at line 18, column 31:  last\_updated:"Last updated: %s" |

请确认您使用空格进行缩进（Soft tab），并确认冒号后有加上一个空格。

您可参阅 [YAML 规范](http://www.yaml.org/spec/1.2/spec.html) 以取得更多信息。

### EMFILE 错误

|  |
| --- |
| Error: EMFILE, too many open files |

虽然 Node.js 有非阻塞 I/O，同步 I/O 的数量仍被系统所限制，在生成大量静态文件的时候，您可能会碰到 EMFILE 错误，您可以尝试提高同步 I/O 的限制数量来解决此问题。

|  |
| --- |
| $ ulimit -n 10000 |

（这一命令只对Linux系统有效）

### Git 部署问题

#### RPC failed

|  |
| --- |
| error: RPC failed; result=22, HTTP code = 403  fatal: 'username.github.io' does not appear to be a git repository |

请确认您已经在电脑上 [配置 git](https://help.github.com/articles/set-up-git)，或改用 HTTPS 库（repository）地址。

#### Error: ENOENT: no such file or directory

如果你遇到了这个错误，有可能是你的文件名、分类或者标签的名字混淆了大写和小写，你可以尝试检查每一个标签和分类的名称，是否大小写一致来修复这一问题。

### 服务器问题

|  |
| --- |
| Error: listen EADDRINUSE |

您可能同时开启两个 Hexo 服务器，或者有其他应用程序正在占用相同的端口，请尝试修改 port 参数，或是在启动 Hexo 服务器时加上 -p 选项。

|  |
| --- |
| $ hexo server -p 5000 |

### 插件安装问题

|  |
| --- |
| npm ERR! node-waf configure build |

当您尝试安装以 C/C++ 或其他非 JavaScript 语言所编写的插件时，可能会遇到此类问题，请确认您已经在电脑上安装相对应的编译器。

### DTrace 错误 （Mac OS X）

|  |
| --- |
| { [Error: Cannot find module './build/Release/DTraceProviderBindings'] code: 'MODULE\_NOT\_FOUND' } { [Error: Cannot find module './build/default/DTraceProviderBindings'] code: 'MODULE\_NOT\_FOUND' } { [Error: Cannot find module './build/Debug/DTraceProviderBindings'] code: 'MODULE\_NOT\_FOUND' } |

DTrace 安装可能有错误 , 使用下列命令:

|  |
| --- |
| $ npm install hexo --no-optional |

参考 [#1326](https://github.com/hexojs/hexo/issues/1326#issuecomment-113871796)

### 在 Jade 或 Swig 遍历资料

Hexo 使用 [Warehouse](https://github.com/hexojs/warehouse) 存储资料，它不是一般数组所以必须先进行类型转型才能遍历。

|  |
| --- |
| {% for post in site.posts.toArray() %} {% endfor %} |

### 资料没有更新

有时资料可能没有被更新，或是生成的文件与修改前的相同，您可以尝试清除缓存并再执行一次。

|  |
| --- |
| $ hexo clean |

### 命令没有执行

如果你除了 hexo help、hexo init 和 hexo version 以外不能执行任何命令、并且你的任何命令都只返回了 hexo help 的内容，这可能是由于 package.json 中缺乏 hexo 字段导致的。

|  |
| --- |
| {  "hexo": {  "version": "3.9.0"  } } |

### 泄露（Escape）内容

Hexo 使用 [Nunjucks](http://mozilla.github.io/nunjucks/) 来解析文章（旧版本使用 [Swig](http://paularmstrong.github.io/swig/)，两者语法类似），内容若包含 {{ }} 或 {% %} 可能导致解析错误，您可以用 raw 标签包裹来避免潜在问题发生。

|  |
| --- |
| {% raw %} Hello {{ sensitive }} {% endraw %} |

### ENOSPC 错误 （Linux）

运行 $ hexo server 命令有时会返回这样的错误：

|  |
| --- |
| Error: watch ENOSPC ... |

它可以用过运行 $ npm dedupe 来解决，如果不起作用的话，可以尝试在 Linux 终端中运行下列命令：

|  |
| --- |
| $ echo fs.inotify.max\_user\_watches=524288 | sudo tee -a /etc/sysctl.conf && sudo sysctl -p |

这将会提高你能监视的文件数量。

### EMPERM Error (Windows Subsystem for Linux)

在执行 hexo server 后返回如下错误：

|  |
| --- |
| Error: watch /path/to/hexo/theme/ EMPERM |

这是由于你使用的 WSL 版本不支持监听文件系统改动。 最新版的 WSL 已经解决了这一问题。

您也仍然可以通过先使用 hexo g 生成文件然后将其作为静态服务器运行来从 WSL 环境运行服务器：

|  |
| --- |
| $ hexo generate $ hexo server -s |

关于 WSL 的这一 Issue 请前往 <https://github.com/Microsoft/BashOnWindows/issues/216> 查看。目前这一问题已经得到了解决。

### 模板渲染错误

有的时候你在执行 hexo generate 时会返回以下错误信息：

|  |
| --- |
| FATAL Something's wrong. Maybe you can find the solution here: http://hexo.io/docs/troubleshooting.html Template render error: (unknown path) |

这表明你的文件中存在一些不可被识别的字符，比如不可见的零宽度字符。有可能你的新文章存在这个问题，或者你在修改配置文件时导致了这个错误。

检查你的 \_config.yml 文件中是否漏掉了列表前的空格。你可以查阅 Wikipedia 中 [YAML](https://zh.wikipedia.org/wiki/YAML) 相关页面来学习 YAML 语法。

这个是错误的：

|  |
| --- |
| plugins: - hexo-generator-feed - hexo-generator-sitemap |

正确的应该是这样：

|  |
| --- |
| plugins:  - hexo-generator-feed  - hexo-generator-sitemap |

### 开始之前

请遵守以下准则：

* 遵守 [Google JavaScript 代码风格](https://google.github.io/styleguide/jsguide.html)。
* 使用 2 个空格缩排。
* 不要把逗号放在最前面。

### 工作流程

1. Fork [hexojs/hexo](https://github.com/hexojs/hexo)
2. 把库（repository）复制到电脑上，并安装所依赖的插件。

|  |
| --- |
| $ git clone https://github.com/<username>/hexo.git $ cd hexo $ npm install $ git submodule update --init |

1. 新增一个功能分支。

|  |
| --- |
| $ git checkout -b new\_feature |

1. 开始开发。
2. 推送（push）分支。

|  |
| --- |
| $ git push origin new\_feature |

1. 建立一个新的合并申请（pull request）并描述变动。

### 注意事项

* 不要修改 package.json 的版本号。
* 只有在测试通过的情况下您的合并申请才会被批准，在提交前别忘了进行测试。

|  |
| --- |
| $ npm test |

## 更新文档

Hexo 文档开放源代码，您可以在 [hexojs/site](https://github.com/hexojs/site) 找到源代码。

### 工作流程

1. Fork [hexojs/site](https://github.com/hexojs/site)
2. 把库（repository）复制到电脑上，并安装所依赖的插件。

|  |
| --- |
| $ git clone https://github.com/<username>/site.git $ cd site $ npm install |

1. 开始编辑文件，您可以通过服务器预览变动。

|  |
| --- |
| $ hexo server |

1. 推送（push）分支。
2. 建立一个新的合并申请（pull request）并描述变动。

### 翻译

1. 在 source 资料夹中建立一个新的语言资料夹（全小写）。
2. 把 source 资料夹中相关的文件（Markdown 和模板文件）复制到新的语言资料夹中。
3. 在 source/\_data/language.yml 中新增语言。
4. 将 en.yml 复制到 themes/navy/languages中并命名为语言名称（全小写）。

## 反馈问题

当您在使用 Hexo 时遇到问题，您可以尝试在 [问题解答](https://hexo.io/zh-cn/docs/troubleshooting) 中寻找解答，或是在 [GitHub](https://github.com/hexojs/hexo/issues) 或 [Google Group](https://groups.google.com/group/hexo) 上提问。如果你没有找答案，请在 Github 报告它。

1. 在 [调试模式](https://hexo.io/zh-cn/docs/commands#%E8%B0%83%E8%AF%95%E6%A8%A1%E5%BC%8F) 中重现问题。
2. 运行 hexo version 并检查版本信息。
3. 把调试信息和版本信息都贴到 GitHub。

## Hexo 巧用 abbrlink 插件实现文章多语言版本

　　目前，Hexo 似乎没有完美的实现文章多语言版本的方式，而有时针对某些特定的文章，有多语言版本的需求，而呉真的博客又使用 [hexo-abbrlink](https://github.com/rozbo/hexo-abbrlink) 插件实现持久连接的生成。基于这些条件，这里提出一种利用 hexo-abbrlink 插件生成多语言文章的方法。

### 使用方法

　　为每一种语言生成一个 md 文件，可以命名为 xxx.md，xxx\_en.md，xxx\_jp.md 格式，或者将额外语言的文件放入资源文件夹内，例如 xxx/en.md，xxx/jp.md。在 hexo-abbrlink 插件生成永久链接后，手动将链接修改为 89abcdef\_en 之类的格式，如此你即可生成多个文章页面，每个页面对应一种语言版本，你可以在每个语言版本中自行添加超链接至别的语言页面。

### 在首页中隐藏不需要的语言

　　利用这个方法可以针对有多重语言需求的文章灵活的撰写多语言版本，但是同样也有缺点，在首页中以及分类菜单下，会同时出现同篇文章的多个语言版本，再利用 [hexo-hide-posts](https://github.com/printempw/hexo-hide-posts) 插件，将不需要显示在首页上的语言版本隐藏。

提示：你可以在配置文件内设置在 sitemap 中不隐藏文章，利于搜索引擎收录多语言版本。  
演示文章：[修改群晖存储池及存储空间顺序](https://blog.kuretru.com/posts/2752cb63/)

## hexo-abbrlink

### How to install

Add plugin to Hexo:

npm install hexo-abbrlink --save

Modify permalink in config.yml file:

permalink: posts/:abbrlink/

There are two settings:

alg -- Algorithm (currently support crc16 and crc32, which crc16 is default)

rep -- Represent (the generated link could be presented in hex or dec value)

# abbrlink config

abbrlink:

alg: crc32 #support crc16(default) and crc32

rep: hex #support dec(default) and hex

drafts: false #(true)Process draft,(false)Do not process draft. false(default)

# Generate categories from directory-tree

# depth: the max\_depth of directory-tree you want to generate, should > 0

auto\_category:

enable: true #true(default)

depth: #3(default)

auto\_title: false #enable auto title, it can auto fill the title by path

auto\_date: false #enable auto date, it can auto fill the date by time today

force: false #enable force mode,in this mode, the plugin will ignore the cache, and calc the abbrlink for every post even it already had abbrlink.

### Sample

The generated link will look like the following:

crc16 & hex

https://post.zz173.com/posts/66c8.html

crc16 & dec

https://post.zz173.com/posts/65535.html

crc32 & hex

https://post.zz173.com/posts/8ddf18fb.html

crc32 & dec

https://post.zz173.com/posts/1690090958.html

### Limitation

[fixed] Maximum number of posts is 65535 for crc16. (now, if a abbrlink already exist, it will change another one and try again and again...)

### More info

see [this](https://post.zz173.com/detail/hexo-abbrlink.html)(Chinese)

## hexo-hide-posts

[**English Documentation**](https://github.com/printempw/hexo-hide-posts/blob/master/README.md)

本 Hexo 插件可以在博客中隐藏指定的文章，并使它们仅可通过链接访问。

当一篇文章被设置为「隐藏」时，它不会出现在任何列表中（包括首页、存档、分类页面、标签页面、Feed、站点地图等），也不会被搜索引擎索引（前提是搜索引擎遵守 noindex 标签）。

只有知道文章链接的人才可以访问被隐藏的文章。

### 安装

$ npm install hexo-hide-posts --save

### 使用

在文章的 [front-matter](https://hexo.io/docs/front-matter) 中添加 hidden: true 即可隐藏文章。

比如我们隐藏了 source/\_posts/lorem-ipsum.md 这篇文章：

---

title: 'Lorem Ipsum'

date: '2019/8/10 11:45:14'

hidden: true

---

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

虽然首页上被隐藏了，但你仍然可以通过 https://hexo.test/lorem-ipsum/ 链接访问它。

你可以在命令行运行 hexo hidden:list 来获取当前所有的已隐藏文章列表。

插件也在 [Local Variables](https://hexo.io/api/locals) 中添加了 all\_posts 和 hidden\_posts 变量，供自定义主题使用。

### 配置

在你的站点 \_config.yml 中添加如下配置：

# hexo-hide-posts

hide\_posts:

# 可以改成其他你喜欢的名字

filter: hidden

# 指定你想要传递隐藏文章的 generator，比如让所有隐藏文章在存档页面可见

# 常见的 generators 有：index, tag, category, archive, sitemap, feed, etc.

public\_generators: []

# 为隐藏的文章添加 noindex meta 标签，阻止搜索引擎收录

noindex: true

举个栗子：设置 filter: secret 之后，你就可以在 front-matter 中使用 secret: true 来隐藏文章了

## 设置 RSS

NexT 中 RSS 有三个设置选项，满足特定的使用场景。 更改 主题配置文件，设定 rss 字段的值：

* false：禁用 RSS，不在页面上显示 RSS 连接。
* 留空：使用 Hexo 生成的 Feed 链接。 你可以需要先安装 [hexo-generator-feed](https://github.com/hexojs/hexo-generator-feed) 插件。
* 具体的链接地址：适用于已经烧制过 Feed 的情形。

#### 添加「标签」页面

新建「标签」页面，并在菜单中显示「标签」链接。「标签」页面将展示站点的所有标签，若你的所有文章都未包含标签，此页面将是空的。 底下代码是一篇包含标签的文章的例子：

title: 标签测试文章

tags:

- Testing

- Another Tag

---

请参阅 [Hexo 的分类与标签文档](https://hexo.io/zh-cn/docs/front-matter.html#分类和标签)，了解如何为文章添加标签或者分类。

* [新建页面](http://theme-next.iissnan.com/theme-settings.html#new-page-tags)
* [设置页面类型](http://theme-next.iissnan.com/theme-settings.html#set-page-tags)
* [修改菜单](http://theme-next.iissnan.com/theme-settings.html#update-menu-for-tags-page)

在菜单中添加链接。编辑 主题配置文件 ， 添加 tags 到 menu 中，如下:

menu:

home: /

archives: /archives

tags: /tags

**注意：**如果有集成评论服务，页面也会带有评论。 若需要关闭的话，请添加字段 comments 并将值设置为 false，如：

禁用评论示例

title: 标签

date: 2014-12-22 12:39:04

type: "tags"

comments: false

---

#### 添加「分类」页面

新建「分类」页面，并在菜单中显示「分类」链接。「分类」页面将展示站点的所有分类，若你的所有文章都未包含分类，此页面将是空的。 底下代码是一篇包含分类的文章的例子：

title: 分类测试文章

categories: Testing

---

请参阅 [Hexo 的分类与标签文档](https://hexo.io/zh-cn/docs/front-matter.html#分类和标签)，了解如何为文章添加标签或者分类。

* [新建页面](http://theme-next.iissnan.com/theme-settings.html#new-page-categories)
* [设置页面类型](http://theme-next.iissnan.com/theme-settings.html#set-page-categories)
* [修改菜单](http://theme-next.iissnan.com/theme-settings.html#update-menu-for-categories-page)

在菜单中添加链接。编辑 主题配置文件 ， 添加 categories 到 menu 中，如下:

menu:

home: /

archives: /archives

categories: /categories

**注意：**如果有集成评论服务，页面也会带有评论。 若需要关闭的话，请添加字段 comments 并将值设置为 false，如：

禁用评论示例

title: 分类

date: 2014-12-22 12:39:04

type: "categories"

comments: false

---

#### 设置字体

**注意：** 此特性在版本 5.0.1 中引入，要使用此功能请确保所使用的 NexT 版本在此之后

为了解决 [Google Fonts API](https://www.google.com/fonts) 不稳定的问题，NexT 在 5.0.1 中引入此特性。 通过此特性，你可以指定所使用的字体库外链地址；与此同时，NexT 开放了 5 个特定范围的字体设定，他们是：

* 全局字体：定义的字体将在全站范围使用
* 标题字体：文章内标题的字体（h1, h2, h3, h4, h5, h6）
* 文章字体：文章所使用的字体
* Logo字体：Logo 所使用的字体
* 代码字体： 代码块所使用的字体

各项所指定的字体将作为首选字体，当他们不可用时会自动 Fallback 到 NexT 设定的基础字体组：

* 非代码类字体：Fallback 到 "PingFang SC", "Microsoft YaHei", sans-serif
* 代码类字体： Fallback 到 consolas, Menlo, "PingFang SC", "Microsoft YaHei", monospace

另外，每一项都有一个额外的 external 属性，此属性用来控制是否使用外链字体库。 开放此属性方便你设定那些已经安装在系统中的字体，减少不必要的请求（请求大小）。

配置示例

font:

enable: true

# 外链字体库地址，例如 //fonts.googleapis.com (默认值)

host:

# 全局字体，应用在 body 元素上

global:

external: true

family: Monda

# 标题字体 (h1, h2, h3, h4, h5, h6)

headings:

external: true

family: Roboto Slab

# 文章字体

posts:

external: true

family:

# Logo 字体

logo:

external: true

family: Lobster Two

size: 24

# 代码字体，应用于 code 以及代码块

codes:

external: true

family: PT Mono

#### 设置代码高亮主题

NexT 使用 [Tomorrow Theme](https://github.com/chriskempson/tomorrow-theme) 作为代码高亮，共有5款主题供你选择。 NexT 默认使用的是 白色的 normal 主题，可选的值有 normal，night， night blue， night bright， night eighties：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

更改 highlight\_theme 字段，将其值设定成你所喜爱的高亮主题，例如：

高亮主题设置示例

# Code Highlight theme

# Available value: normal | night | night eighties | night blue | night bright

# https://github.com/chriskempson/tomorrow-theme

highlight\_theme: normal

#### 侧边栏社交链接

侧栏社交链接的修改包含两个部分，第一是链接，第二是链接图标。 两者配置均在 主题配置文件 中。

1. 链接放置在 social 字段下，一行一个链接。其键值格式是 显示文本: 链接地址。

配置示例

# Social links

social:

GitHub: https://github.com/your-user-name

Twitter: https://twitter.com/your-user-name

微博: http://weibo.com/your-user-name

豆瓣: http://douban.com/people/your-user-name

知乎: http://www.zhihu.com/people/your-user-name

# 等等

1. 设定链接的图标，对应的字段是 social\_icons。其键值格式是 匹配键: Font Awesome 图标名称， 匹配键 与上一步所配置的链接的 显示文本 相同（大小写严格匹配），图标名称 是 Font Awesome 图标的名字（不必带 fa- 前缀）。 enable 选项用于控制是否显示图标，你可以设置成 false 来去掉图标。

配置示例

# Social Icons

social\_icons:

enable: true

# Icon Mappings

GitHub: github

Twitter: twitter

微博: weibo

#### 开启打赏功能 由 [habren](https://github.com/iissnan/hexo-theme-next/pull/687) 贡献

越来越多的平台（微信公众平台，新浪微博，简书，百度打赏等）支持打赏功能，付费阅读时代越来越近，特此增加了打赏功能，支持微信打赏和支付宝打赏。 只需要 主题配置文件 中填入 微信 和 支付宝 收款二维码图片地址 即可开启该功能。

打赏功能配置示例

reward\_comment: 坚持原创技术分享，您的支持将鼓励我继续创作！

wechatpay: /path/to/wechat-reward-image

alipay: /path/to/alipay-reward-image

#### 友情链接 由 [iamwent](https://github.com/iissnan/hexo-theme-next/pull/250) 贡献

编辑 主题配置文件 添加：

友情链接配置示例

# title

links\_title: Links

links:

MacTalk: http://macshuo.com/

Title: http://example.com/

#### 腾讯公益404页面 由 [xirong](https://github.com/iissnan/hexo-theme-next/pull/126) 贡献

腾讯公益404页面，寻找丢失儿童，让大家一起关注此项公益事业！效果如下 <http://www.ixirong.com/404.html>

使用方法，新建 404.html 页面，放到主题的 source 目录下，内容如下：

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="content-type" content="text/html;charset=utf-8;"/>

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />

<meta name="robots" content="all" />

<meta name="robots" content="index,follow"/>

<link rel="stylesheet" type="text/css" href="https://qzone.qq.com/gy/404/style/404style.css">

</head>

<body>

<script type="text/plain" src="http://www.qq.com/404/search\_children.js"

charset="utf-8" homePageUrl="/"

homePageName="回到我的主页">

</script>

<script src="https://qzone.qq.com/gy/404/data.js" charset="utf-8"></script>

<script src="https://qzone.qq.com/gy/404/page.js" charset="utf-8"></script>

</body>

</html>

#### 站点建立时间

这个时间将在站点的底部显示，例如 © 2013 - 2015。 编辑 主题配置文件，新增字段 since。

配置示例

since: 2013

#### 订阅微信公众号 由 [huiwang](https://github.com/iissnan/hexo-theme-next/pull/788) 贡献

**注意：** 此特性在版本 5.0.1 中引入，要使用此功能请确保所使用的 NexT 版本在此之后

在每篇文章的末尾显示微信公众号二维码，扫一扫，轻松订阅博客。

在微信公众号平台下载您的二维码，并将它存放于博客source/uploads/目录下。

然后编辑 主题配置文件，如下：

配置示例

# Wechat Subscriber

wechat\_subscriber:

enabled: true

qcode: /uploads/wechat-qcode.jpg

description: 欢迎您扫一扫上面的微信公众号，订阅我的博客！

#### 设置「动画效果」

NexT 默认开启动画效果，效果使用 JavaScript 编写，因此需要等待 JavaScript 脚本完全加载完毕后才会显示内容。 如果您比较在乎速度，可以将设置此字段的值为 false 来关闭动画。

编辑 主题配置文件， 搜索 use\_motion，根据您的需求设置值为 true 或者 false 即可：

use\_motion: true # 开启动画效果

use\_motion: false # 关闭动画效果

#### 设置「背景动画」

NexT 自带两种背景动画效果

编辑 主题配置文件， 搜索 canvas\_nest 或 three\_waves，根据您的需求设置值为 true 或者 false 即可：

**注意：** three\_waves 在版本 5.1.1 中引入。只能同时开启一种背景动画效果。

canvas\_nest 配置示例

# canvas\_nest

canvas\_nest: true //开启动画

canvas\_nest: false //关闭动画

three\_waves 配置示例

# three\_waves

three\_waves: true //开启动画

three\_waves: false //关闭动画

## 第三方服务来

### 评论系统

NexT 支持多款评论系统。

如需取消某个 页面/文章 的评论，在 md 文件的 [front-matter](https://hexo.io/docs/front-matter.html) 中增加 comments: false

#### DISQUS

* [>= 5.1.1](http://theme-next.iissnan.com/third-party-services.html#disqus-new-settings)
* [< 5.1.1](http://theme-next.iissnan.com/third-party-services.html#disqus-old-settings)

编辑 主题配置文件， 将 disqus 下的 enable 设定为 true，同时提供您的 shortname。count 用于指定是否显示评论数量。

disqus:

enable: false

shortname:

count: true

#### Facebook Comments 由 [hydai](https://github.com/iissnan/hexo-theme-next/pull/925) 贡献

需要先集成 [Facebook SDK](http://theme-next.iissnan.com/third-party-services.html#facebook-sdk)。修改 主题配置文件 以下配置：

# Facebook comments plugin

# This plugin depends on Facebook SDK.

# If facebook\_sdk.enable is false, Facebook comments plugin is unavailable.

facebook\_comments\_plugin:

enable: false

num\_of\_posts: 10 # min posts num is 1

width: 100% # default width is 550px

scheme: light # default scheme is light (light or dark)

#### HyperComments 由 [ivan-nginx](https://github.com/iissnan/hexo-theme-next/pull/1155) 贡献

Add support for [Hypercomments](https://www.hypercomments.com/) system instead of Disqus. 修改 主题配置文件 以下配置：

# Hypercomments

#hypercomments\_id:

#### 网易云跟帖 由 [geekrainy](https://github.com/iissnan/hexo-theme-next/pull/1301) 贡献

登陆 [网易云跟帖](https://gentie.163.com/) 获取你的 Product Key。 编辑 主题配置文件， 编辑 gentie\_productKey 字段，设置如下：

gentie\_productKey: #your-gentie-product-key

请注意，您在云跟帖管理后台设置的域名必须跟您站点的域名一致。在本地测试时，需要做两步骤前置设定：

1. 修改 hosts 文件，将您域名的请求指向本地。例如：127.0.0.1 notes.example.com
2. 修改 Hexo 监听的端口为 80：hexo s --debug -p 80

测试完成后请将 hosts 文件中的域名映射删除。

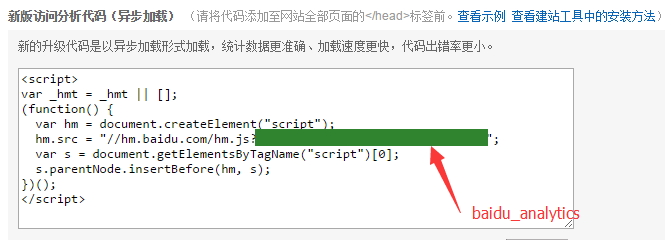
#### 来必力 由 [asmoker](https://github.com/iissnan/hexo-theme-next/pull/1415) 贡献

登陆 [来必力](https://livere.com/) 获取你的 LiveRe UID。 编辑 主题配置文件， 编辑 livere\_uid 字段，设置如下：

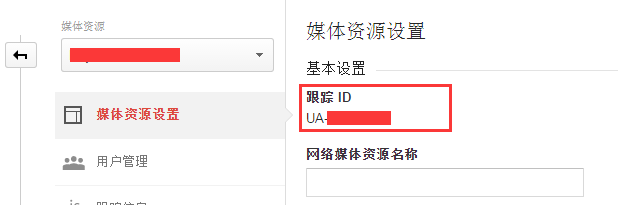
livere\_uid: #your livere\_uid

### 数据统计与分析

#### 百度统计

1. 登录 [百度统计](http://tongji.baidu.com/)，定位到站点的代码获取页面
2. 复制 hm.js? 后面那串统计脚本 id，如下图所示：  
   
3. 编辑 主题配置文件， 修改字段 baidu\_analytics，值设置成你的百度统计脚本 id。

#### Google 分析

编辑 主题配置文件， 修改字段 google\_analytics， 值设置成你的 Google 跟踪 ID。跟踪 ID 通常是以 UA- 开头。 

#### 腾讯分析 由 [Cissoid](https://github.com/iissnan/hexo-theme-next/pull/603) 贡献

请登录 [腾讯分析](http://ta.qq.com/)，登录并获取分析的 ID。 然后在 主题配置文件 里将 ID 放置 tencent\_analytics 字段。

#### CNZZ 统计 由 [jerrybendy](https://github.com/iissnan/hexo-theme-next/pull/712) 贡献

在 主题配置文件 中增加了一项 cnzz\_siteid的配置项，值为 CNZZ 里面添加统计的站点ID。 这个ID可以在地址栏里，或者自动生成的脚本里面找到。

默认脚本生成的内容会产生“站长统计”几个字，感觉很丑而且和主题不太搭，我用了一个 display: none;把这几个字给隐藏了。

#### 不蒜子统计 由 [panzhitian](https://github.com/iissnan/hexo-theme-next/pull/809) 贡献

**注意：** 此特性在版本 5.0.1 中引入，要使用此功能请确保所使用的 NexT 版本在此之后

* [全局配置](http://theme-next.iissnan.com/third-party-services.html#analytics-busuanzi-enable)
* [站点UV配置](http://theme-next.iissnan.com/third-party-services.html#analytics-busuanzi-site-uv)
* [站点PV配置](http://theme-next.iissnan.com/third-party-services.html#analytics-busuanzi-site-pv)
* [单页面PV配置](http://theme-next.iissnan.com/third-party-services.html#analytics-busuanzi-page-pv)

编辑 主题配置文件 中的busuanzi\_count的配置项。

当enable: true时，代表开启全局开关。若site\_uv、site\_pv、page\_pv的值均为false时，不蒜子仅作记录而不会在页面上显示。

#### 腾讯移动分析 由 [aleonchen](https://github.com/iissnan/hexo-theme-next/pull/1408) 贡献

编辑 主题配置文件，填写 tencent\_mta 的值。

# Tencent MTA ID

tencent\_mta: your-tencent-mta-id

#### 阅读次数统计（LeanCloud) 由 [Doublemine](https://github.com/iissnan/hexo-theme-next/pull/439) 贡献

请查看 [为NexT主题添加文章阅读量统计功能](https://notes.wanghao.work/2015-10-21-%E4%B8%BANexT%E4%B8%BB%E9%A2%98%E6%B7%BB%E5%8A%A0%E6%96%87%E7%AB%A0%E9%98%85%E8%AF%BB%E9%87%8F%E7%BB%9F%E8%AE%A1%E5%8A%9F%E8%83%BD.html#%E9%85%8D%E7%BD%AELeanCloud)

### 内容分享服务

#### JiaThis

编辑 主题配置文件， 添加/修改字段 jiathis，值为 true。

JiaThis 内容分享服务配置示例

# JiaThis 分享服务

jiathis: true

#### [百度分享](http://share.baidu.com/)

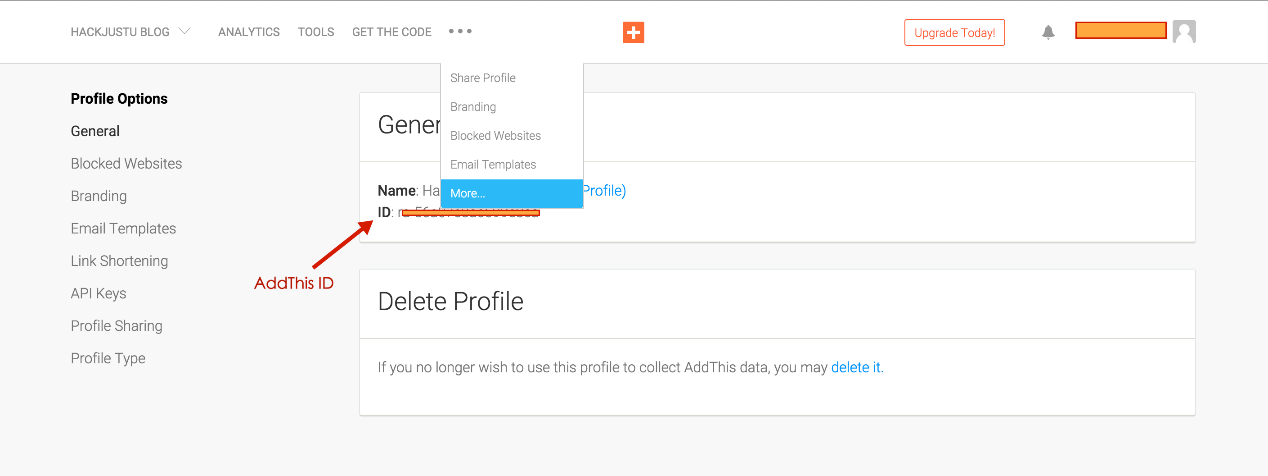
编辑 主题配置文件，添加/修改字段 baidushare，值为 true。

百度内容分享服务配置示例

# 百度分享服务

baidushare: true

#### AddThis 由 [hackjustu](https://github.com/iissnan/hexo-theme-next/pull/660) 贡献

1. 在网站 [AddThis](https://www.addthis.com/) 上注册账号。 可以使用 Google/Facebook/Twitter 账号进行第三方登陆
2. 从下面菜单获得 AddThis id：More.. --> General --> ID。 具体 ID 获得方式参考以下截图   
   
3. 在 主题配置文件 中，把#Share下的 #add\_this\_id取消注释， 改为add\_this\_id: put\_your\_add\_this\_id\_here

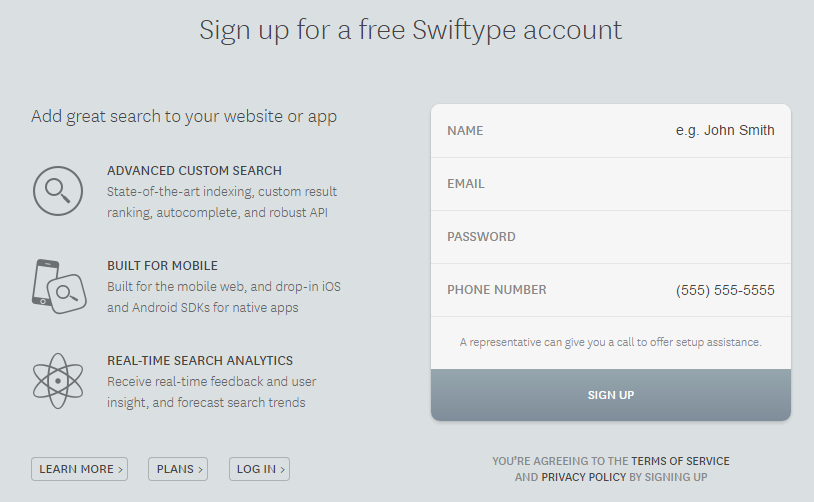
### 搜索服务

NexT 支持集成 [Swiftype](https://swiftype.com/)、 [微搜索](http://tinysou.com/)、Local Search 和 Algolia。

#### Swiftype 站内搜索 由 [lzlun129](https://github.com/iissnan/hexo-theme-next/pull/84) 贡献

* [注册账户](http://theme-next.iissnan.com/third-party-services.html#swiftype-step-1)
* [创建搜索引擎](http://theme-next.iissnan.com/third-party-services.html#swiftype-step-2)
* [定制与激活搜索](http://theme-next.iissnan.com/third-party-services.html#swiftype-step-3)
* [获取 Key](http://theme-next.iissnan.com/third-party-services.html#swiftype-step-4)
* [更新主题配置](http://theme-next.iissnan.com/third-party-services.html#swiftype-step-5)

前往 [Swiftype 注册页面](https://swiftype.com/users/sign_up)，注册一个新账户。



#### 微搜索 由 [lzlun129](https://github.com/iissnan/hexo-theme-next/pull/84) 贡献

TBD

#### Local Search 由 [flashlab](https://github.com/iissnan/hexo-theme-next/pull/694) 贡献

添加百度/谷歌/本地 自定义站点内容搜索

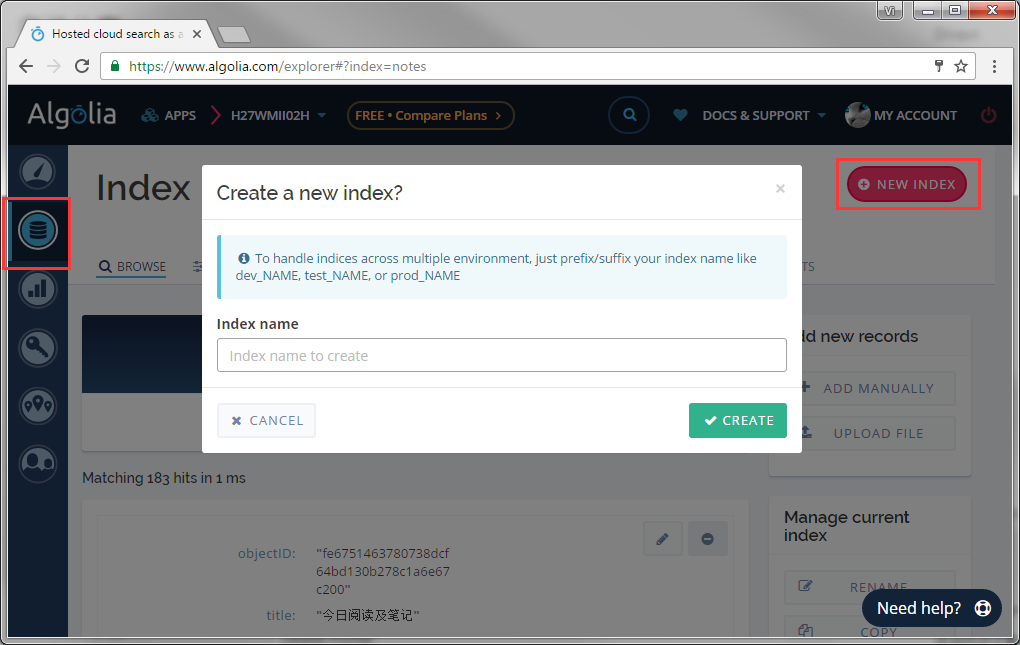
1. 安装 hexo-generator-searchdb，在站点的根目录下执行以下命令：
2. $ npm install hexo-generator-searchdb --save
3. 编辑 站点配置文件，新增以下内容到任意位置：
4. search:
5. path: search.xml
6. field: post
7. format: html
8. limit: 10000
9. 编辑 主题配置文件，启用本地搜索功能：
10. # Local search
11. local\_search:
12. enable: true

#### Algolia

**注意：** 此特性在版本 5.1.0 中引入，要使用此功能请确保所使用的 NexT 版本在此之后

* [注册 Algolia，创建 Index](http://theme-next.iissnan.com/third-party-services.html#algolia-step-1)
* [安装 Hexo Algolia](http://theme-next.iissnan.com/third-party-services.html#algolia-step-2)
* [获取 Key，更新站点配置](http://theme-next.iissnan.com/third-party-services.html#algolia-step-3)
* [更新 Index](http://theme-next.iissnan.com/third-party-services.html#algolia-step-4)
* [主题集成](http://theme-next.iissnan.com/third-party-services.html#algolia-step-5)

前往 [Algolia 注册页面](https://www.algolia.com)，注册一个新账户。 可以使用 GitHub 或者 Google 账户直接登录，注册后的 14 天内拥有所有功能（包括收费类别的）。之后若未续费会自动降级为免费账户，免费账户 总共有 10,000 条记录，每月有 100,000 的可以操作数。注册完成后，创建一个新的 Index，这个 Index 将在后面使用。



### 其他服务

#### MathJax

NexT 借助于 MathJax 来显示数学公式，此选项默认关闭。

* [5.0.0 及之前的版本](http://theme-next.iissnan.com/third-party-services.html#mathjax-old-settings)
* [5.0.1 及以上的版本](http://theme-next.iissnan.com/third-party-services.html#mathjax-new-settings)

编辑 主题配置文件，将 mathjax 设定为 true 即可。

# MathJax Support

mathjax:

ProTip: [使用七牛 CDN 来加速 MathJax 脚本的加载](https://github.com/iissnan/hexo-theme-next/pull/363)

#### Facebook SDK 由 [changyuheng](https://github.com/iissnan/hexo-theme-next/pull/410) 贡献

修改 主题配置文件 以下配置：

# Facebook SDK Support.

facebook\_sdk:

enable: false

app\_id:

fb\_admin:

like\_button: #true

webmaster: #true

#### Google Webmaster tools

设置 [Google站点管理工具](https://www.google.com/webmasters/tools/) 的验证字符串，用于提交 sitemap

1. 获取 google site verification code  
   登录 Google Webmaster Tools，导航到验证方法，并选择 HTML Tag。将会获取到一段代码：

<meta name="google-site-verification" content="XXXXXXXXXXXXXXXXXXXXXXX" />

将 content 里面的 XXXXXXXXXXXXXXXXXXXXXXX 复制出来。

1. 设置主题。编辑 主题配置文件，新增/修改字段 google\_site\_verification。
2. google\_site\_verification: XXXXXXXXXXXXXXXXXXXXXXX

## 设置 「JavaScript 第三方库」

**注意：** 此特性在版本 5.0.1 中引入，要使用此功能请确保所使用的 NexT 版本在此之后

设定成合适的 CDN 地址，此特性可以加速静态资源（JavaScript 第三方库）的加载，目前 NexT 使用的第三方库包括：

| **名称** | **官方地址** | **NexT 内部使用版本** |
| --- | --- | --- |
| jQuery | <http://jquery.com/> | 2.1.3 |
| Fancybox | <http://fancyapps.com/fancybox/> | 2.1.5 |
| FastClick | <https://github.com/ftlabs/fastclick> | 1.0.6 |
| jQuery LazyLoad | <https://github.com/tuupola/jquery_lazyload> | 2.1.5 |
| VelocityJS | <http://julian.com/research/velocity/> | 1.2.1 |
| VelocityJS UI Pack | <http://julian.com/research/velocity/#uiPack> | 1.2.1 |
| UAParser.js | <https://faisalman.github.io/ua-parser-js/> | 0.7.9 |
| FontAwesome | <http://fontawesome.io/> | 4.4.0 |

此特性配置位于 主题配置文件，键值 vendors。 其中 \_internal: vendors 用于指定 NexT 内部使用的路径前缀，通常情况下不需要更改。接着便是各个库的设定，若设定为空，NexT 将使用内部集成的文件。每一个设定都是 库： CDN 地址 的键值形式。例如：

jquery: //cdn.jsdelivr.net/jquery/2.1.3/jquery.min.js

「Internal version」说明 NexT 内部使用的版本，经过测试的版本。因此若你需要指定其他版本，需要进行额外的自行测试。

设置示例

vendors:

# Internal path prefix. Please do not edit it.

\_internal: vendors

# Internal version: 2.1.3

jquery: //cdn.jsdelivr.net/jquery/2.1.3/jquery.min.js

# Internal version: 2.1.5

# Fancybox: http://fancyapps.com/fancybox/

fancybox: //cdn.jsdelivr.net/fancybox/2.1.5/jquery.fancybox.pack.js

fancybox\_css: //cdn.jsdelivr.net/fancybox/2.1.5/jquery.fancybox.min.css

# Internal version: 1.0.6

fastclick: //cdn.jsdelivr.net/fastclick/1.0.6/fastclick.min.js

# Internal version: 1.9.7

lazyload: //cdn.jsdelivr.net/jquery.lazyload/1.9.3/jquery.lazyload.min.js

# Internal version: 1.2.1

velocity: //cdn.jsdelivr.net/velocity/1.2.3/velocity.min.js

# Internal version: 1.2.1

velocity\_ui: //cdn.jsdelivr.net/velocity/1.2.3/velocity.ui.min.js

# Internal version: 0.7.9

ua\_parser: //cdn.jsdelivr.net/ua-parser.js/0.7.10/ua-parser.min.js

# Internal version: 4.4.0

# http://fontawesome.io/

fontawesome: //maxcdn.bootstrapcdn.com/font-awesome/4.6.2/css/font-awesome.min.css

# Internal version: 1

# https://www.algolia.com

algolia\_instant\_js:

algolia\_instant\_css:

# Internal version: 1.0.0

# https://github.com/hustcc/canvas-nest.js

canvas\_nest:

# Internal version: r84

# https://github.com/mrdoob/three.js/releases

three:

# Internal version: 1.0.0

# https://github.com/jjandxa/three\_waves

three\_waves:

### 如何设置「阅读全文」？

在首页显示一篇文章的部分内容，并提供一个链接跳转到全文页面是一个常见的需求。 NexT 提供三种方式来控制文章在首页的显示方式。 也就是说，在首页显示文章的摘录并显示 **阅读全文** 按钮，可以通过以下方法：

1. 在文章中使用 <!-- more --> 手动进行截断，Hexo 提供的方式 推荐
2. 在文章的 [front-matter](https://hexo.io/docs/front-matter.html) 中添加 description，并提供文章摘录
3. 自动形成摘要，在 主题配置文件 中添加：
4. auto\_excerpt:
5. enable: true
6. length: 150

默认截取的长度为 150 字符，可以根据需要自行设定

建议使用 <!-- more -->（即第一种方式），除了可以精确控制需要显示的摘录内容以外， 这种方式也可以让 Hexo 中的插件更好的识别。

### Favicon 设置后没有生效？

将你的 favicon 放置到 **站点** 的 source 目录下，与站点的配置文件同级。 若你发现设置的 Favicon 并未生效，请先清除浏览器的缓存后直接访问 Favicon 的地址，这个地址通常是 http://your-domain.com/favicon.ico。

如果你的站点是放在子目录下，请设置为：favicon: favicon.ico

### 如何更改字体？

NexT 从 **5.0.1** 版本开始提供一个 [字体定制特性](http://theme-next.iissnan.com/theme-settings.html#fonts-customization)， 请先查看此特性是否能满足你的需求。以下的修改将覆盖 字体定制 的特性。 编辑主题下的 source/css/\_variables/custom.styl 文件，新增两个变量：

// 标题，修改成你期望的字体族

$font-family-headings = Georgia, sans

// 修改成你期望的字体族

$font-family-base = "Microsoft YaHei", Verdana, sans-serif

// 代码字体

$code-font-family = "Input Mono", "PT Mono", Consolas, Monaco, Menlo, monospace

// 正文字体的大小

$font-size-base = 16px

// 代码字体的大小

$code-font-size = 13px

### 如何更改内容区域的宽度？

NexT 对于内容的宽度的设定如下：

* 700px，当屏幕宽度 < 1600px
* 900px，当屏幕宽度 >= 1600px
* 移动设备下，宽度自适应

如果你需要修改内容的宽度，同样需要编辑样式文件。 编辑主题的 source/css/\_variables/custom.styl 文件，新增变量：

// 修改成你期望的宽度

$content-desktop = 700px

// 当视窗超过 1600px 后的宽度

$content-desktop-large = 900px

此方法不适用于 Pisces Scheme，关于如何修改 Pisces Scheme 的宽度请参看 [这个 Issue](https://github.com/iissnan/hexo-theme-next/issues/759#issuecomment-202242848)

### 为何新增的菜单项显示 Menu.xxx

NexT 使用了 Hexo 提供的国际化功能，因此在 Menu 部分配置的值并非直接用于界面的展示，而是作为一个名称。 当渲染的时候，NexT 去寻找对应的语言文件，并替换这个键所定义的文本后展示。

以 home 为例，在 主题配置文件 中配置为 home: /， 此处的键即是 home，在不同的语言中，这个键所对应的文本都不同：

# 简体中文

# 语言文件 languages/zh-Hans.yml

menu:

home: 首页

# 英文

# 语言文件 languages/en.yml

menu:

home: Home

因此，当新增一个项时，必须在 **所使用的语言** 中添加对应的翻译文本。

关于菜单的配置，请参考 [菜单的配置](http://theme-next.iissnan.com/getting-started.html#menu-settings)

### 标签/分类数量统计不准确？

因为 Hexo 有缓存的功能，因此有时候你会发现在 标签 和 分类 页面中的数量统计并不准确。 出现这个问题时，可以按照以下步骤重新生成站点的内容：

1. 删除站点目录下的 db.json 文件
2. 在站点目录下执行命令 hexo clean
3. 在站点目录下执行命令，重新生成 hexo generate

当执行完以上步骤后，可以在本地启动服务器来验证下是否已经解决问题。

### 为何语言显示不正确？

若没有设定 language 的值或者 language: default，会出现语言选择不正确的情况。 因此为了确保语言正确显示，请明确指定需要使用的语言。 编辑 站点配置文件， 将 language 字段设置成明确的语言代码。 例如，设置语言为简体中文：

language: zh-Hans

| **语言** | **对应值** |
| --- | --- |
| English | en |
| 简体中文 | zh-Hans |
| 法语 | fr-FR |
| 繁体中文 | zh-hk/zh-tw |
| 俄语 | ru |
| 德语 | de |
| 葡萄牙语 | pt |
| 日语 | ja |

### 如何关闭新建页面的评论功能？

当集成了评论系统，如 Disqus，所有新建的页面都将自动开启评论。若你不需要评论，请在页面的 Front-matter 里添加 comments 字段，并将值设置为 false。如下所示：

title: All tags

date: 2015-12-16 17:05:24

type: "tags"

comments: false

---

### 如何设置页面文章的篇数？

在 Hexo 里可以为首页和归档页面设置不同的文章篇数，但可能需要安装 Hexo 插件。详细步骤如下。

1. 使用 npm install --save 命令来安装需要的 Hexo 插件。
2. npm install --save hexo-generator-index
3. npm install --save hexo-generator-archive
4. npm install --save hexo-generator-tag
5. 等待扩展全部安装完成后，在 **站点配置文章** 中，设定如下选项：
6. index\_generator:
7. per\_page: 5
8. archive\_generator:
9. per\_page: 20
10. yearly: true
11. monthly: true
12. tag\_generator:
13. per\_page: 10

per\_page 即文章的数量。

需要注意的是，Hexo 3.2 版本不允许配置文件中存在重复的选项设置。 因此，最好检查下 站点配置文件 中是否有存在上述同名的配置。 如果存在，请将两者配置在一起。

### 如何优化 NexT 主题？

NexT 并未对 HTML、JavaScript 以及 CSS 做压缩处理。如果你比较在意这点，可以在 NexT 的 Issues 列表搜索 [他人共享的优化方案](https://github.com/iissnan/hexo-theme-next/issues?utf8=%E2%9C%93&q=+label%3AOptimization) 。

### Cannot find module XXX

NexT 提供的标签插件引用了 Hexo 模块中的一些子模块，例如 hexo-util。 NPM 在版本 3 中引入了新的模块安装机制， NexT 将使用这种机制，意味着如果是旧的 NPM 将会碰到这个错误。 因此，如果您碰到 "Cannot find module XXX" 的错误时，请首先使用 npm -v 确认您的 NPM 版本。

* 版本 > 3 依然报错： 请先删除 node\_module 目录，然后使用 npm install 重新安装一下模块。
* 版本 < 3： 您可以选择升级您的 NPM； 或者在站点目录下明确指定模块依赖 npm install --save hexo-util。 其中 `hexo-util` 仅是示例，请替换成错误中提示的模块名称。

## Hexo插件

Hexo 有强大的插件系统，让我们能够轻松扩展功能而不用修改核心模块的源码。在 Hexo 中有两种形式的插件：

* 脚本（Scripts）
* 插件（Packages）

如果我们的代码很简单，我们可以编写脚本，只需要把 JavaScript 文件放到 scripts 文件夹，在启动时就会自动载入。简单来说，脚本文件可以相当于一些这样的的工具函数，当我们发现Hexo官方提供的函数不能满足我们的需求时，我们可以通过添加一个脚本来实现。

比如，我们现在有这样一个简单的需求，我们想给首页文章列表中的文章块添加一个背景颜色，背景颜色我们可以在文章md文件中定义，如果未定义，则随机选用一种颜色。

首先，我们先在文章md文件中顶部Front-matter添加一个color字段：

\_posts/hello-world-1.md:

title: Hello World 1  
date: 2019-02-12 17:49:32  
categories: 分类1  
tags:   
    - 标签1  
color: blue  
---

定义完成后，我们就可以在文章信息字段post或者page中获取到color。

然后，我们需要添加一个脚本函数来根据color字段来获取文章块的背景颜色，新增scripts/getPostBgColor.js:

const arr = [ 'blue', 'purple', 'green', 'yellow', 'red', 'orange' ];  
  
var getPostBgColor = function(color) {  
  if (arr.indexOf(color) >= 0) {  
    return `bg-${color}`;  
  }  
  return 'bg-' + randBgColor();  
};  
  
function randBgColor() {  
  return arr[randomInt(0, 5)];  
}  
  
function randomInt(min, max) {  
  return Math.round(Math.random() \* (max - min)) + min;  
}  
  
hexo.extend.helper.register('getPostBgColor', getPostBgColor);

最后一行我们通过hexo.extend.helper.register全局注册一个脚本函数，注册完成后，我们就可以在模板文件中和Hexo官方提供的全局函数一样使用。

修改layout/index.ejs:

...  
<article class="post <%= getPostBgColor(post.color) %>">  
...

添加背景颜色样式，编辑css/index.styl:

...  
.bg-blue {  
    background-color: #6fa3ef;  
}  
.bg-purple {  
    background-color: #bc99c4;  
}  
.bg-green {  
    background-color: #46c47c;  
}  
.bg-yellow {  
    background-color: #f9bb3c;  
}  
.bg-red {  
    background-color: #e8583d;  
}  
.bg-orange {  
    background-color: #f68e5f;  
}

看下效果，Hello World 1这篇文章我们定义了color：blue，因此是蓝色，其他文章，我们未定义color，因此是随机颜色。

这样，我们就实现了随机背景色这个小需求，当然这只是一个非常简单的例子，如果有其他复杂的需求，我们可以通过编写更加复杂的脚本来实现。

## Hexo的数据DB扩展查询

我们已经知道，Hexo已经为我们预先定义了很多常用的变量供我们使用，具体可以在 Hexo | 变量 查询。但是如果系统提供的变量数据不能满足我们的要求，那我们该怎么办呢？其实我们可以通过扩展查询来获取到我们期望的数据。

其实Hexo所有的文章分类标签等等变量信息，在编译成本地静态文件之前，都是本地存储在一个db.json中的，相当于小型的本地数据库，Hexo在运行阶段，所有的数据相关操作其实都是在这个小型数据库上进行操作，其底层使用的查询引擎就是Warehouse。因此我们可以通过Warehouse的语法进行自定义扩展查询。

比如我们需要在页面的底部展示全站的最近6篇文章列表，由于Hexo首页只提供了第一页的数据，因此我们可以基于site变量进行扩展查询：

site.posts.sort({date: -1}).limit(6)

site.posts表示所有的文章，sort（{date: -1})表示按创建时间倒序排列，limit(6)表示只取前6条数据，这样我们就可以拿到了全站的最近6文章信息，后续进行相应展示操作即可。

其他更多复杂的扩展查询都可以根据Warehouse语法文档进行按需扩展。

## 总结

其实说白了，Hexo就是把那些 Markdown 文件，按照我们编写的对应布局模板，填上对应的数据生成 HTML 页面，然后在编译的过程中将JS/CSS等文件引入HTML，然后生成每个页面的对应HMTL静态文件。

而Hexo主题的作用就是决定每个布局模板长什么样。

## 新建仓库

都说了要给他人访问，如果你连个仓库都没有，是不是说不过去？出门右转上 github 新建一个仓库，仓库名必须为 <user-name>.github.io 格式，其中 <user-name> 是你 github 的昵称。不要问我为什么，问我我也不想回答你。

#### 2. 全局安装hexo

如果你连 npm 是啥都不知道，对不起，我们的缘分尽了，你走吧。...别，别，别，开个玩笑，有啥问题可以留言。打开命令行，输入下面的命令，全局安装 hexo。

npm install -g hexo

#### 3. 初始化项目

Just one 命令，hexo init，就会自动构建一个 hexo 项目，紧接着执行 hexo s，浏览器访问 localhost:4000 就可以看到效果啦！过程一定要快，凉了就不好吃了！

hexo init

// 本地运行

hexo s

#### 4. 部署到github

虽然只有一份 Hello World，但是我也想让全世界的人看到，成为最闪耀的那个崽，可咋整？不要慌，快速在项目根目录下找到 \_congif.yml，找到 deploy 字段并填写完整，下面请开始你上学时最擅长的事情：抄写！

# Deployment

## Docs: https://hexo.io/docs/deployment.html

deploy:

type: git

repo: <你的仓库地址> # https://github.com/TJ-XiaJiaHao/TJ-XiaJiaHao.github.io

branch: master

抄写完毕！但是！我们需要额外的一个工具来帮助我们推到仓库上，那就是！那就是！那就是 hexo-deployer-git。搞它！

npm install hexo-deployer-git --save

万事俱备，只欠东南风！执行下面两个命令，就可以把项目自动部署到 github 上啦啦啦啦啦。

hexo clean

hexo deploy

#### 5. 查看效果

凡是连环杀手都喜欢在杀人后留在现场，因为他喜欢欣赏自己的杰作！

浏览器访问：https://tj-xiajiahao.github.io/ 即可看到效果。

至此，已经完成了个人博客的搭建，也可以在浏览器中访问，下面将介绍一些常用的野外生存技能。

#### 如何创建新文章

阳光明媚，一杯温暖的咖啡，一束倾斜的阳光，惬意地抿一口咖啡，我要开始写博客了！

// [layout] 为布局，可选项为 `post`、`page`、`draft`，这将决定文章所在文件路径。

// <title> 为文章标题

// 如 hexo new post 除了帅气，我还有啥！

hexo new [layout] <title>

#### 更换主题皮肤

风格不喜欢？换之。

更换主题流程：下载主题 -> 配置主题，以 xoxo 为例（不要想歪了！）

# 下载到themes文件夹下

git clone https://github.com/KevinOfNeu/hexo-theme-xoxo xoxo

# 修改 \_config.yml 配置

theme: xoxo

#### 部署优化

每次都要执行 hexo clean 和 hexo deploy，不如写个新的脚本

// package.json

"dev": "hexo s",

"build": "hexo clean & hexo deploy"

部署命令

npm run build

#### 接入评论系统 - valine

<https://valine.js.org/quickstart.html>

## Hexo主题开发实践

typecho 主题写了一个非常漂亮的 hexo 主题，地址是：https://github.com/cccyb/hexo-theme-pinghsu，欢迎大家使用。

### 前言

本文将会从零开始编写一个简单的Hexo博客主题，目的是了解一个Hexo博客主题的构成以及如何编写，因此，本示例中的博客页面样式不做过多描绘，样式主要参考 Hexo theme 中的 Noise 主题。

在开始前，你需要对以下的一些知识点有必要的了解：

* 模板引擎语法
* CSS预处理器
* YML语法
* Hexo文档

本文使用的模板引擎为 ejs，使用的 CSS 预处理器为 stylus。这也是 hexo 项目预装了的 render 插件，如果想使用其他模板引擎或者其他 CSS 预处理器，可以安装相对应的 render 插件。

本文的代码：https://github.com/cccyb/theme-example

### 目录结构

根据下图，在themes目录下新建一个theme-example文件夹作为我们的主题目录。

如图所示，一个hexo主题的目录主要包括以下五部分：

* languages：用于国际化的语言文件
* layout：主题布局模板文件
* scripts：hexo脚本插件目录，可以编写一些辅助函数脚本
* source：资源文件目录，包括页面样式，js脚本等
* \_config.yml：主题配置文件

### 局部模板

我们通过分析常见的博客网站可以知道，大部分的博客网站都由三部分组成：顶部导航栏，中间内容区域，以及底部信息展示区域。每次点击导航栏选项跳转页面时，顶部导航栏以及底部信息展示区域是不变的，只是中间的内容区域重新渲染，因此，我们可以将通用的代码抽离成局部模板以复用。

我们在layout目录下新建\_partial目录，在该目录下添加head.ejs,header.ejs以及footer.ejs文件。

layout/\_partial/head.ejs:

<head>  
    <meta http-equiv="content-type" content="text/html; charset=utf-8">  
    <meta content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0" name="viewport">  
    <title>标题</title>  
</head>

layout/\_partial/header.ejs:

<header>我是导航栏</header>

layout/\_partial/footer.ejs:

<footer>我是底部信息</footer>

我们在layout中创建layout.ejs，并引入head.ejs,header/ejs和footer.ejs文件，layout.ejs文件是通用的布局文件模板，我们在后面新增的ejs文件都会继承layout.ejs，并将其内容填充到body中。

layout/layout.ejs:

<!DOCTYPE html>  
<html>  
<%- partial('\_partial/head') %>  
<body>  
    <div class="container">  
    <%- partial('\_partial/header') %>  
    <%- body %>  
    <%- partial('\_partial/footer') %>  
    </div>  
</body>  
</html>

### 首页

首页是我们的网站加载完毕后的第一个页面。

我们在 layout 中创建 index.ejs 文件，index.ejs首页将会继承layout.ejs布局模板生成 HTML 文件。

partial()函数的作用是可以引入其他模板文件，详情参考hexo文档

layout/index.ejs:

<h1>Hello World</h1>

修改**站点配置文件**中的主题配置，使用我们刚刚创建的 theme-example 主题：

# Extensions  
## Plugins: https://hexo.io/plugins/  
## Themes: https://hexo.io/themes/  
theme: theme-example

运行 hexo server 开启 Hexo 本地服务器预览，访问 http://localhost:4000/。

### 编写导航栏和底部信息

前面，我们只是搭了个页面框架，接下来我们就将开始正式开始一步步的完善我们的主题，以下两个文档我们将频繁的使用，最好可以提前阅读一遍有个了解：

* Hexo | 变量
* Hexo | 辅助函数

layout/\_partial/head.ejs:

<head>  
    <meta http-equiv="content-type" content="text/html; charset=utf-8">  
    <meta content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0" name="viewport">  
    <title><%= config.title %></title>  
</head>

这里我们使用了config全局变量，该变量包含的是站点配置（即站点根目录下 \_config.yml 中的配置）。除此之外，我还有将经常使用的是theme变量，该变量是主题配置（即主题根目录下 \_config.yml 中的配置），其他变量参见hexo文档。

编辑导航栏部分，layout/\_partial/header.ejs：

<header class="header">  
    <div class="title">  
        <a href="<%= url\_for() %>" class="logo"><%= config.title %></a>  
    </div>  
    <nav class="navbar">  
        <ul class="menu">  
            <li class="menu-item">  
                <a href="/" class="menu-item-link">Home</a>  
            </li>  
            <li class="menu-item">  
                <a href="/categories" class="menu-item-link">Categories</a>  
            </li>  
            <li class="menu-item">  
                <a href="/tags" class="menu-item-link">Tags</a>  
            </li>  
            <li class="menu-item">  
                <a href="/archives" class="menu-item-link">Archives</a>  
            </li>  
        </ul>  
    </nav>  
</header>

编辑底部信息部分，layout/\_partial/footer.ejs：

<footer>  
    <p>Theme is <a href="/" target="\_blank">Theme-example</a> by <a href="<%= config.url %>" target="\_blank"><%= config.author %></a></p>  
    <p>Powered by <a href="https://hexo.io/" target="\_blank" rel="nofollow">hexo</a> &copy; <%- date(Date.now(), 'YYYY') %></p>  
</footer>

这样，我们就得到了一个包含导航栏和底部信息的简单页面。

### 添加主题配置

实际上我们需要让导航菜单根据我们的需要显示不同的项，上面这种写法不方便修改。所以我们会在主题的配置文件中添加导航菜单的配置。在 theme-example 下的配置文件 \_config.yml，在其中添加需要配置的字段。然后可以通过 theme 这个变量来拿到该配置文件中的配置。

theme-example/\_config.yml:

menu:  
  home: /  
  categories: /categories  
  tags: /tags  
  archives: /archives

这样我们就可以在 header.ejs 中使用 theme.menu 获取到导航菜单的设置。将 header.ejs 修改为：

<header class="header">  
    <div class="title">  
        <a href="<%= url\_for() %>" class="logo"><%= config.title %></a>  
    </div>  
    <nav class="navbar">  
        <ul class="menu">  
            <% for (name in theme.menu) { %>  
            <li class="menu-item">  
                <a href="<%- url\_for(theme.menu[name]) %>" class="menu-item-link"><%= name %></a>  
            </li>  
            <% } %>  
        </ul>  
    </nav>  
</header>

这样，我们就可以动态的配置导航的信息了，我们还可以在主题配置文件中添加其他配置项供我们使用。

### 添加文章列表

接着我们完善首页的模板，使其能够显示文章列表。前面已经说过 Hexo 提供了各种有用的变量，在这里将会使用到 page 这个变量。page 会根据不同的页面拥有不同的属性。具体有什么属性，可以获取到哪些数据可以查看这里。

那么这里我们会使用 page 变量的 posts 属性拿到文章数据的集合。编辑 index.ejs 文件：

<section class="posts">  
    <% page.posts.each(function (post) { %>  
    <article class="post">  
        <div class="post-title">  
            <a class="post-title-link" href="<%- url\_for(post.path) %>"><%= post.title %></a>  
        </div>  
        <div class="post-content">  
            <%- post.content %>  
        </div>  
        <div class="post-meta">  
            <span class="post-time"><%- date(post.date, "YYYY-MM-DD") %></span>  
        </div>  
    </article>  
    <% }) %>  
</section>

从 page.posts 中获取单篇文章的数据，并获取文章的标题，内容等数据填充到模板中。处理文章创建时间的时候使用了 date() 函数，这是 Hexo 提供的时间处理的辅助函数。

由于首页显示文章内容时使用的是 post.content，即文章的全部内容。所以首页会显示每一篇文章的内容，实际上我们并不想在首页显示那么多内容，只想显示文章的摘录。

Hexo 提供了 excerpt 属性来获取文章的摘录部分，不过这里需要在文章中添加一个 <!--more--> 标记。添加了这个标记之后，post.excerpt 将会获取到标记之前的内容。如果没有这个标记，那么 post.excerpt 会是空的。所以我们可以把首页文章内容部分的 post.content 替换成 post.excerpt。

<div class="post-content">  
    <%- post.excerpt %>  
</div>

### 添加样式

到目前为止，我们完成了首页的整体页面结构，不过由于没添加css样式，因此整体页面非常难看，所以我们需要给页面加上一些样式来美化一下我们的页面。

由于 Hexo 在新建项目的时候会安装 hexo-renderer-stylus 这个插件，所以我们无需其他步骤，只需要将样式文件放到 css 文件夹中。Hexo 在生成页面的时候会将 source 中的所有文件复制到生成的 public 文件中，并且在此之前会编译 styl 为 css 文件。

在 css 文件夹中创建 style.styl，编写一些基础的样式，并把所有样式 import 到这个文件。所以最终编译之后只会有 style.css 一个文件。创建 \_partial/header.styl 与 \_partial/post.styl 存放页面导航以及文章的样式，并且在 style.styl 中 import 这两个文件。

\_partial/header.styl：

.header {  
    margin-top: 2em  
    display: flex  
    align-items: baseline  
    justify-content: space-between  
  
    .blog-title .logo {  
        color: #AAA;  
        font-size: 2em;  
        font-family: "Comic Sans MS",cursive,LiSu,sans-serif;  
        text-decoration: none;  
    }  
  
    .menu {  
        margin: 0;  
        padding: 0;  
  
        .menu-item {  
            display: inline-block;  
            margin-right: 10px;  
        }  
  
        .menu-item-link {  
            color: #AAA;  
            text-decoration: none;  
  
            &:hover {  
                color: #368CCB;  
            }  
        }  
    }  
}

index.styl:

.post {  
    margin: 1em auto;  
    padding: 30px 50px;  
    background-color: #fff;  
    border: 1px solid #ddd;  
    box-shadow: 0 0 2px #ddd;  
}  
  
.posts  {  
    .post:first-child {  
        margin-top: 0;  
    }  
  
    .post-title {  
        font-size: 1.5em;  
  
        .post-title-link {  
            color: #368CCB;  
            text-decoration: none;  
        }  
    }  
  
    .post-content {  
        a {  
            color: #368CCB;  
            text-decoration: none;  
        }  
    }  
  
    .post-meta {  
        color: #BABABA;  
    }  
}

style.styl:

body {  
    background-color: #F2F2F2;  
    font-size: 1.25rem;  
    line-height: 1.5;  
}  
  
.container {  
    max-width: 960px;  
    margin: 0 auto;  
}  
  
@import "\_partial/header";  
@import "\_partial/index";

最后，我们需要把样式添加到页面中，这里使用了另外一个辅助函数 css():

layout/\_partial/head.ejs:

<head>  
    <meta http-equiv="content-type" content="text/html; charset=utf-8">  
    <meta content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0" name="viewport">  
    <title><%= config.title %></title>  
    <%- css('css/style.styl') %>  
</head>

至此，我们会看到站点的首页是这个样子的：

### 添加分页

在站点的 source/\_post/ 目录下存放的是我们的文章，现在我们把原本的 hello-world.md 复制黏贴 10+ 次，再查看站点首页。会发现，首页只显示了 10 篇文章。

首页显示的文章数量我们可以通过站点配置文件中的 per\_page 字段来修改，但是我们不可能把所有文章都放在一页，所以我们现在来添加文章列表的分页。

新建 \_partial/paginator.ejs:

<% if (page.total > 1){ %>  
    <nav class="page-nav">  
    <%- paginator({  
        prev\_text: "&laquo; Prev",  
        next\_text: "Next &raquo;"  
    }) %>  
    </nav>  
<% } %>

在 index.ejs 中添加这个文件的内容：

...  
</section>  
<%- partial('\_partial/paginator') %>

这里我们使用到了另外的一个辅助函数 paginator，它能够帮助我们插入分页链接，这里我们只是实现了一个最基本的分页，具体的样式可以自行添加，或者根据文档使用其他配置自定义分页。

### 添加文章详情页

文章详情页对应的布局文件是 post.ejs，新建 post.ejs:

<article class="post">  
    <div class="post-title">  
        <h2 class="title"><%= page.title %></h2>  
    </div>  
    <div class="post-meta">  
        <span class="post-time"><%- date(page.date, "YYYY-MM-DD") %></span>  
    </div>  
    <div class="post-content">  
        <%- page.content %>  
    </div>  
</article>

由于这里是文章的模板，所以变量 page 表示的是文章的数据，而不是首页的文章数据集合。

### 添加归档页

创建归档页使用的模板文件 archive.ejs:

<section class="archive">  
    <ul class="post-archive">  
    <% page.posts.each(function (post) { %>  
        <li class="post-item">  
            <span class="post-date"><%= date(post.date, "YYYY-MM-DD") %></span>  
            <a class="post-title" href="<%- url\_for(post.path) %>"><%= post.title %></a>  
        </li>  
    <% }) %>  
    </ul>  
</section>  
<%- partial('\_partial/paginator') %>

其实结构跟首页差不多，只是不显示文章内容而已。添加归档页的样式：

css/\_partial/archive.styl:

.archive {  
    margin: 1em auto;  
    padding: 30px 50px;  
    background-color: #fff;  
    border: 1px solid #ddd;  
    box-shadow: 0 0 2px #ddd;  
  
    .post-archive {  
        list-style: none;  
        padding: 0;  
  
        .post-item {  
            margin: 5px 0;  
  
            .post-date {  
                display: inline-block;  
                margin-right: 10px;  
                color: #BABABA;  
            }  
  
            .post-title {  
                color: #368CCB;  
                text-decoration: none;  
            }  
        }  
    }  
}

### 添加分类页

分类页跟归档页也类似，相当于根据不同的分类进行归档。

分类页和标签页的模板编写比较特殊，本质上，分类页和标签页属于自定义页面，我们需要新建自定义页面模板page.ejs:

<% if (is\_current(theme.menu.categories)) { %>  
<%- partial('\_partial/category') %>  
<% } else if (is\_current(theme.menu.tags)) { %>  
<%- partial('\_partial/tag') %>  
<% } else { %>  
<%- partial('\_partial/custom') %>  
<% } %>

这里，我们需要根据当前自定义页面的类型来决定渲染何种自定义页面模板。

新建\_partial/category.ejs:

<section class="archive">  
    <ul class="post-archive">  
    <% site.categories.each(function (category) { %>  
        <span><%= category.name %></span>  
        <% category.posts.forEach(function(post) { %>  
        <li class="post-item">  
            <span class="post-date"><%= date(post.date, "YYYY-MM-DD") %></span>  
            <a class="post-title" href="<%- url\_for(post.path) %>"><%= post.title %></a>  
        </li>  
        <% }) %>  
    <% }) %>  
    </ul>  
</section>

site.categories包括了站点所有的分类信息，可以遍历获取分类信息，其中category.posts又包含了该分类的所有文章信息。

需要注意的是，要想在页面中展示分类页，需要先执行hexo new page categories生成分类页面，并添加type为categories：

title: categories  
date: 2019-02-25 18:19:55  
type: "categories"  
---

后续的标签页类似。

### 添加标签页

标签页与分类页及其类似，只是根据标签进行归档。

新建\_partial/tag.ejs:

<section class="archive">  
    <ul class="post-archive">  
    <% site.tags.each(function (tag) { %>  
        <span><%= tag.name %></span>  
        <% tag.posts.forEach(function(post) { %>  
        <li class="post-item">  
            <span class="post-date"><%= date(post.date, "YYYY-MM-DD") %></span>  
            <a class="post-title" href="<%- url\_for(post.path) %>"><%= post.title %></a>  
        </li>  
        <% }) %>  
    <% }) %>  
    </ul>  
</section>

### 添加自定义页面

自定义页面与文章详情页类似。

新建\_partial/custom.ejs:

<article class="post">  
    <div class="post-title">  
        <h2 class="title"><%= page.title %></h2>  
    </div>  
    <div class="post-meta">  
        <span class="post-time"><%- date(page.date, "YYYY-MM-DD") %></span>  
    </div>  
    <div class="post-content">  
        <%- page.content %>  
    </div>  
</article>

我们在主题配置文件中添加自定义页面的菜单：

menu:  
  home: /  
  categories: /categories  
  tags: /tags  
  archives: /archivesz  
  about: /about

执行hexo new page about进行手动生成页面，编辑文件内容即可。

## 将Hexo部署到个人网站上

首先安装 [hexo-deployer-sftp](https://link.juejin.im/?target=https%3A%2F%2Fgithub.com%2Flucascaro%2Fhexo-deployer-sftp)

1. npm install hexo-deployer-sftp *--save*
2. 复制代码

### 修改配置

(**记得冒号后必须有空格，不然配置不生效，这是Hexo的一个坑！**）

1. deploy:
2. type: sftp
3. host: <host>
4. user: <user>
5. pass: <password>
6. remotePath: [remote path]
7. port: [port]
8. privateKey: [path/to/privateKey]
9. passphrase: [passphrase]
10. agent: [path/to/agent/socket]
11. 复制代码

host 远程主机的地址

user 使用者名称

pass 密码

remotePath 远程主机的根目录

port 端口

privateKey ssh私钥的目录地址

passphrase （可省略）ssh私钥的密码短语

agent ssh套接字的目录地址

### 执行部署命令

输入hexo g -d就可以生成并部署到自己的个人服务器上了

## Hexo资源

https://hexo.io/zh-cn/docs/configuration.html

## Hexo主题资源

按评价排序

### 主题NexT

http://theme-next.iissnan.com/getting-started.html

[Hexo](https://hexo.io/zh-cn/) 是高效的静态站点生成框架，她基于 [Node.js](https://nodejs.org)。 通过 Hexo 你可以轻松地使用 Markdown 编写文章，除了 Markdown 本身的语法之外，还可以使用 Hexo 提供的 [标签插件](https://hexo.io/zh-cn/docs/tag-plugins.html) 来快速的插入特定形式的内容。在这篇文章中，假定你已经成功安装了 Hexo，并使用 Hexo 提供的命令创建了一个站点。

你可以访问 [Hexo 的文档](https://hexo.io/zh-cn/docs/) 了解如何安装 Hexo

在 Hexo 中有两份主要的配置文件，其名称都是 \_config.yml。 其中，一份位于站点根目录下，主要包含 Hexo 本身的配置；另一份位于主题目录下，这份配置由主题作者提供，主要用于配置主题相关的选项。

为了描述方便，在以下说明中，将前者称为 站点配置文件， 后者称为 主题配置文件。

#### 安装 NexT

Hexo 安装主题的方式非常简单，只需要将主题文件拷贝至站点目录的 themes 目录下， 然后修改下配置文件即可。具体到 NexT 来说，安装步骤如下。

##### 下载主题

如果你熟悉 [Git](http://git-scm.com/)， 建议你使用 克隆最新版本 的方式，之后的更新可以通过 git pull 来快速更新， 而不用再次下载压缩包替换。

* [克隆最新版本](http://theme-next.iissnan.com/getting-started.html#clone)
* [下载稳定版本](http://theme-next.iissnan.com/getting-started.html#stable)

在终端窗口下，定位到 Hexo 站点目录下。使用 Git checkout 代码：

$ cd your-hexo-site

$ git clone https://github.com/theme-next/hexo-theme-next themes/next

如果你对 Git 感兴趣，可以通过《Pro Git》这本书来学习。你可以访问我制作的一个 [在线版本（第一版）](http://iissnan.com/progit)。

##### 启用主题

与所有 Hexo 主题启用的模式一样。 当 克隆/下载 完成后，打开 站点配置文件， 找到 theme 字段，并将其值更改为 next。

启用 NexT 主题

theme: next

到此，NexT 主题安装完成。下一步我们将验证主题是否正确启用。在切换主题之后、验证之前， 我们最好使用 hexo clean 来清除 Hexo 的缓存。

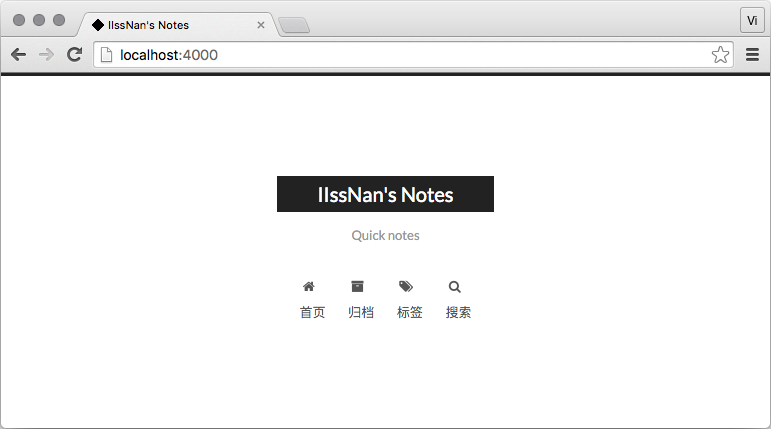
##### 验证主题

首先启动 Hexo 本地站点，并开启调试模式（即加上 --debug），整个命令是 hexo s --debug。 在服务启动的过程，注意观察命令行输出是否有任何异常信息，如果你碰到问题，这些信息将帮助他人更好的定位错误。 当命令行输出中提示出：

INFO Hexo is running at http://0.0.0.0:4000/. Press Ctrl+C to stop.

此时即可使用浏览器访问 http://localhost:4000，检查站点是否正确运行。

当你看到站点的外观与下图所示类似时即说明你已成功安装 NexT 主题。这是 NexT 默认的 Scheme —— Muse



现在，你已经成功安装并启用了 NexT 主题。下一步我们将要更改一些主题的设定，包括个性化以及集成第三方服务。

#### 主题设定

##### 选择 Scheme

Scheme 是 NexT 提供的一种特性，借助于 Scheme，NexT 为你提供多种不同的外观。同时，几乎所有的配置都可以 在 Scheme 之间共用。目前 NexT 支持三种 Scheme，他们是：

* Muse - 默认 Scheme，这是 NexT 最初的版本，黑白主调，大量留白
* Mist - Muse 的紧凑版本，整洁有序的单栏外观
* Pisces - 双栏 Scheme，小家碧玉似的清新

Scheme 的切换通过更改 主题配置文件，搜索 scheme 关键字。 你会看到有三行 scheme 的配置，将你需用启用的 scheme 前面注释 # 去除即可。

选择 Pisces Scheme

#scheme: Muse

#scheme: Mist

scheme: Pisces

##### 设置 语言

编辑 站点配置文件， 将 language 设置成你所需要的语言。建议明确设置你所需要的语言，例如选用简体中文，配置如下：

language: zh-Hans

目前 NexT 支持的语言如以下表格所示：

| **语言** | **代码** | **设定示例** |
| --- | --- | --- |
| English | en | language: en |
| 简体中文 | zh-Hans | language: zh-Hans |
| Français | fr-FR | language: fr-FR |
| Português | pt | language: pt or language: pt-BR |
| 繁體中文 | zh-hk 或者 zh-tw | language: zh-hk |
| Русский язык | ru | language: ru |
| Deutsch | de | language: de |
| 日本語 | ja | language: ja |
| Indonesian | id | language: id |
| Korean | ko | language: ko |

##### 设置 菜单

菜单配置包括三个部分，第一是菜单项（名称和链接），第二是菜单项的显示文本，第三是菜单项对应的图标。 NexT 使用的是 [Font Awesome](http://fontawesome.io/) 提供的图标， Font Awesome 提供了 600+ 的图标，可以满足绝大的多数的场景，同时无须担心在 Retina 屏幕下 图标模糊的问题。

编辑 主题配置文件，修改以下内容：

1. 设定菜单内容，对应的字段是 menu。 菜单内容的设置格式是：item name: link。其中 item name 是一个名称，这个名称并不直接显示在页面上，她将用于匹配图标以及翻译。

菜单示例配置

menu:

home: /

archives: /archives

#about: /about

#categories: /categories

tags: /tags

#commonweal: /404.html

若你的站点运行在子目录中，请将链接前缀的 / 去掉

NexT 默认的菜单项有（标注 的项表示需要手动创建这个页面）：

| **键值** | **设定值** | **显示文本（简体中文）** |
| --- | --- | --- |
| home | home: / | 主页 |
| archives | archives: /archives | 归档页 |
| categories | categories: /categories | 分类页 |
| tags | tags: /tags | 标签页 |
| about | about: /about | 关于页面 |
| commonweal | commonweal: /404.html | 公益 404 |

1. 设置菜单项的显示文本。在第一步中设置的菜单的名称并不直接用于界面上的展示。Hexo 在生成的时候将使用 这个名称查找对应的语言翻译，并提取显示文本。这些翻译文本放置在 NexT 主题目录下的 languages/{language}.yml （{language} 为你所使用的语言）。

以简体中文为例，若你需要添加一个菜单项，比如 something。那么就需要修改简体中文对应的翻译文件 languages/zh-Hans.yml，在 menu 字段下添加一项：

menu:

home: 首页

archives: 归档

categories: 分类

tags: 标签

about: 关于

search: 搜索

commonweal: 公益404

something: 有料

1. 设定菜单项的图标，对应的字段是 menu\_icons。 此设定格式是 item name: icon name，其中 item name 与上一步所配置的菜单名字对应，icon name 是 Font Awesome 图标的 名字。而 enable 可用于控制是否显示图标，你可以设置成 false 来去掉图标。

菜单图标配置示例

menu\_icons:

enable: true

# Icon Mapping.

home: home

about: user

categories: th

tags: tags

archives: archive

commonweal: heartbeat

在菜单图标开启的情况下，如果菜单项与菜单未匹配（没有设置或者无效的 Font Awesome 图标名字） 的情况下，NexT 将会使用 作为图标。

请注意键值（如 home）的大小写要严格匹配

##### 设置 侧栏

默认情况下，侧栏仅在文章页面（拥有目录列表）时才显示，并放置于右侧位置。 可以通过修改 主题配置文件 中的 sidebar 字段来控制侧栏的行为。侧栏的设置包括两个部分，其一是侧栏的位置， 其二是侧栏显示的时机。

1. 设置侧栏的位置，修改 sidebar.position 的值，支持的选项有：
   * left - 靠左放置
   * right - 靠右放置

目前仅 Pisces Scheme 支持 position 配置。影响版本5.0.0及更低版本。

sidebar:

position: left

1. 设置侧栏显示的时机，修改 sidebar.display 的值，支持的选项有：
   * post - 默认行为，在文章页面（拥有目录列表）时显示
   * always - 在所有页面中都显示
   * hide - 在所有页面中都隐藏（可以手动展开）
   * remove - 完全移除
2. sidebar:
3. display: post

已知侧栏在 use motion: false 的情况下不会展示。 影响版本5.0.0及更低版本。

##### 设置 头像

编辑 主题配置文件， 修改字段 avatar， 值设置成头像的链接地址。其中，头像的链接地址可以是：

| **地址** | **值** |
| --- | --- |
| 完整的互联网 URI | http://example.com/avatar.png |
| 站点内的地址 | 将头像放置主题目录下的 source/uploads/ （新建 uploads 目录若不存在）  配置为：avatar: /uploads/avatar.png  或者 放置在 source/images/ 目录下  配置为：avatar: /images/avatar.png |

头像设置示例

avatar: http://example.com/avatar.png

##### 设置 作者昵称

编辑 站点配置文件， 设置 author 为你的昵称。

##### 站点描述

编辑 站点配置文件， 设置 description 字段为你的站点描述。站点描述可以是你喜欢的一句签名:)

#### NexT标签

「标签」(Tag Plugin) 是 Hexo 提供的一种快速生成特定内容的方式。 例如，在标准 Markdown 语法中，我们无法指定图片的大小。这种情景，我们即可使用标签来解决。 Hexo 内置来许多标签来帮助写作者可以更快的书写， [完整的标签列表](https://hexo.io/docs/tag-plugins.html) 可以参考 Hexo 官网。 另外，Hexo 也开放来接口给主题，使主题有可能提供给写作者更简便的写作方法。 以下标签便是 NexT 主题当前提供的标签。

##### 文本居中的引用

此标签将生成一个带上下分割线的引用，同时引用内文本将自动居中。 文本居中时，多行文本若长度不等，视觉上会显得不对称，因此建议在引用单行文本的场景下使用。 例如作为文章开篇引用 或者 结束语之前的总结引用。

使用方式

* HTML方式：使用这种方式时，给 img 添加属性 class="blockquote-center" 即可。
* 标签方式：使用 centerquote 或者 简写 cq。

此标签要求 NexT 的版本在 0.4.5 或以上。 若你正在使用的版本比较低，可以选择使用 HTML 方式。

标签调用方法

<!-- HTML方式: 直接在 Markdown 文件中编写 HTML 来调用 -->

<!-- 其中 class="blockquote-center" 是必须的 -->

<blockquote class="blockquote-center">blah blah blah</blockquote>

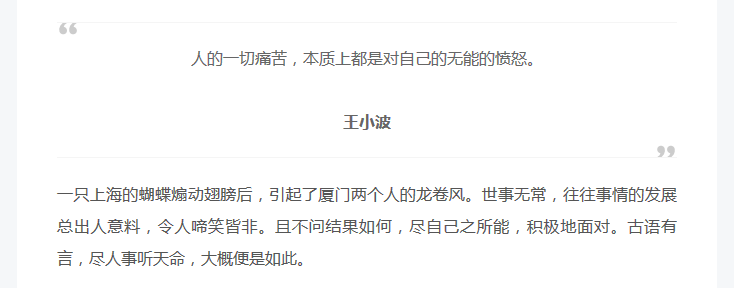
<!-- 标签 方式，要求版本在0.4.5或以上 -->

{% centerquote %}blah blah blah{% endcenterquote %}

<!-- 标签别名 -->

{% cq %} blah blah blah {% endcq %}

效果示例:



##### 突破容器宽度限制的图片

当使用此标签引用图片时，图片将自动扩大 26%，并突破文章容器的宽度。 此标签使用于需要突出显示的图片, 图片的扩大与容器的偏差从视觉上提升图片的吸引力。 此标签有两种调用方式（详细参看底下示例）：

使用方式

* HTML方式：使用这种方式时，为 img 添加属性 class="full-image"即可。
* 标签方式：使用 fullimage 或者 简写 fi， 并传递图片地址、 alt 和 title 属性即可。 属性之间以逗号分隔。

此标签要求 NexT 的版本在 0.4.5 或以上。 若你正在使用的版本比较低，可以选择使用 HTML 方式。  
如果要在图片下显示图片的标题，请使用 标签方式 并给定 title 属性。

标签调用方法

<!-- HTML方式: 直接在 Markdown 文件中编写 HTML 来调用 -->

<!-- 其中 class="full-image" 是必须的 -->

<img src="/image-url" class="full-image" />

<!-- 标签 方式，要求版本在0.4.5或以上 -->

{% fullimage /image-url, alt, title %}

<!-- 别名 -->

{% fi /image-url, alt, title %}

效果示例:



##### Bootstrap Callout 由 [ivan-nginx](https://github.com/iissnan/hexo-theme-next/pull/1160) 贡献

这些样式出现在 [Bootstrap 的官方文档](http://getbootstrap.com/)中。

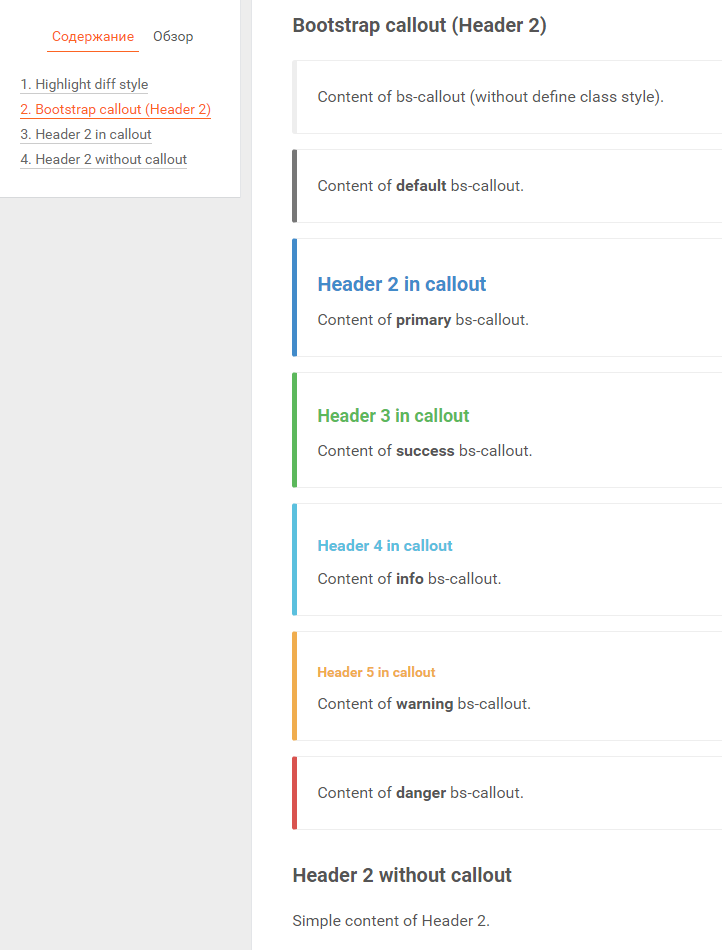
使用方式

{% note class\_name %} Content (md partial supported) {% endnote %}

其中，class\_name 可以是以下列表中的一个值：

* default
* primary
* success
* info
* warning
* danger

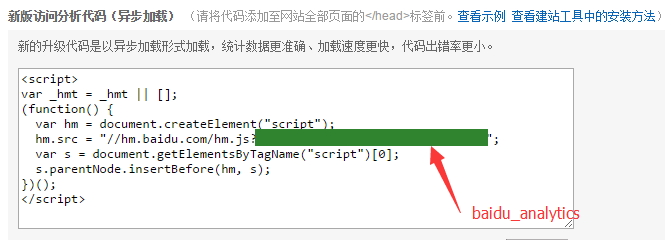
效果示例:



###### 集成常用的第三方服务

百度统计

注意： baidu\_analytics 不是你的百度 id 或者 百度统计 id

1. 登录 [百度统计](http://tongji.baidu.com/)， 定位到站点的代码获取页面
2. 复制 hm.js? 后面那串统计脚本 id，如： 
3. 编辑 主题配置文件， 修改字段 baidu\_analytics 字段，值设置成你的百度统计脚本 id。

阅读次数统计（LeanCloud） 由 [Doublemine](https://github.com/iissnan/hexo-theme-next/pull/439) 贡献

请查看 [为NexT主题添加文章阅读量统计功能](https://notes.wanghao.work/2015-10-21-%E4%B8%BANexT%E4%B8%BB%E9%A2%98%E6%B7%BB%E5%8A%A0%E6%96%87%E7%AB%A0%E9%98%85%E8%AF%BB%E9%87%8F%E7%BB%9F%E8%AE%A1%E5%8A%9F%E8%83%BD.html#%E9%85%8D%E7%BD%AELeanCloud)

Algolia 搜索

详细的配置请参考： [第三方服务 - Algolia](http://theme-next.iissnan.com/third-party-services.html#algolia-search)

### 主题hexo icarus

#### 安装icarus

* 直接下载主题模块放到blog项目 ,blog项目根目录执行

git clone https://github.com/ppoffice/hexo-theme-icarus.git themes/icarus

此时已经下载到项目中。

* 顶级\_config.yml中选择icarus主题
* # Extensions
* ## Plugins: https://hexo.io/plugins/
* ## Themes: https://hexo.io/themes/
* theme: icarus
* 此时主题已经安装好，清除、编译、部署可以看到效果了

在博客目录中输入

git clone https://github.com/ppoffice/hexo-theme-icarus.git themes/icarus

修改 \_config.yml 内的 theme:icarus 设定，即可切换主题

#### 配置icarus

* 完全参照[官网](https://blog.zhangruipeng.me/hexo-theme-icarus/categories/Configuration/Posts/)配置，进行翻译解说

##### 配置 icarus 主题

下载 [icarus](https://github.com/ppoffice/hexo-theme-icarus/releases) ，在 hexo 的 themes 文件夹，新加入 [icarus](https://github.com/ppoffice/hexo-theme-icarus) 目录，配置 hexo 的根目录的 \_config.yml

\_config.yml

|  |  |
| --- | --- |
| 1 2 | -theme: yilia +theme: icarus |

然后更新部署就好了

|  |  |
| --- | --- |
| 1 | hexo d -g |

##### 配置文章部分

##### 顶部图片添加

icarus 主题中的配置\_config.yml中开启图片开关

article:

thumbnail: true

文章.md文件头中添加图片绝对/相对地址

title: Getting Started with Icarus

thumbnail: /gallery/thumbnails/desert.jpg

// thumbnail:https://raw.githubusercontent.com/removeif/blog\_image/master/20190620152744.png

---

Post content...

配置完成后部署显示效果 (最新文章列表显示缩略图、文章开头显示一张设置图片)

##### 左边文章导航栏开启

icarus 主题中的配置\_config.yml中开关

widgets:

-

type: toc

position: left

##### 提取文章目录

title: Table of Contents Example

toc: true

---

Post content...

##### 评论系统开启

icarus 主题中的配置\_config.yml中开启（部分评论系统需要翻墙才能使用，valine不用翻墙个人推荐，[valine安装参考](https://valine.js.org/quickstart.html)）

comment:

type: valine

app\_id: xxxxxxxx # (required) LeanCloud application id

app\_key: xxxxxxxx # (required) LeanCloud application key

notify: false # (optional) receive email notification

verify: false # (optional) show verification code

placeholder: xxxxxxxx # (optional) comment box placeholder text

##### 捐赠收款开启

icarus 主题中的配置\_config.yml中开启 注意如果默认不配置，编译时有报错，# 把它注释掉后会报错，不启用功能方法是注释掉代码

donate:

-

# Donation entry name

type: alipay

# Qrcode image URL

qrcode: 'https://wx2.sinaimg.cn/large/b5d1b710gy1g0lvxdcwm0j20p011i4bg.jpg'

-

# Donation entry name

type: wechat

# Qrcode image URL

qrcode: 'https://wx2.sinaimg.cn/large/b5d1b710gy1g0lvwdcpb5j20u014qgy2.jpg'

##### 全局搜索开启

icarus 主题中的配置\_config.yml中开启,不同的搜索类型需要[安装插件参考官网](https://blog.zhangruipeng.me/hexo-theme-icarus/categories/Plugins/Search/),type: insight此类型不需要安装，已经内置

search:

type: insight

##### 更多配置请参考官网配置

目前配置基本已经够使用，还需要更多配置请参考[连接](https://blog.zhangruipeng.me/hexo-theme-icarus/categories/Configuration/Posts/)

https://blog.zhangruipeng.me/hexo-theme-icarus/categories/Configuration/Posts/

#### 自定义 icarus 主题

#### 代码高亮

[代码高亮款式](https://github.com/highlightjs/highlight.js/tree/master/src/styles) , [预览参见这里](https://highlightjs.org/)，Android 程序员选择 androidstudio 效果不错

themes/icarus/\_config.yml

|  |  |
| --- | --- |
| 1 | highlight: androidstudio |

#### 更新logo

icarus 有默认的 logo，这里想改变一下，logo 是在 themes\icarus\source\images 里面logo.svg 文件，我们只需要替换成自己的 logo 文件即可，可以从[这里](https://icomoon.io/app/#/select) 去获取。

#### 页面类型判定

注: 可以通过变量 page.layout 来判断当前页面类型，'post' 表示当前是文章页面，具体可参考 [hexo](https://hexo.io/zh-cn/docs/variables) 文档。

#### 布局

##### 文章页面两栏布局

主题默认是三栏布局，并且显示了很多的 widget ，但在阅读文章时显得有些拥挤。因此在文章页面，修改为两栏布局，并显示特定的 widget。

diff:includes/helpers/layout.js

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 | const widgets = hexo.extend.helper.get('get\_config').bind(this)('widgets'); - return widgets.filter(widget => widget.hasOwnProperty('position') && widget.position === position); + if (this.page.layout !== 'post') { + return widgets.filter(widget => widget.hasOwnProperty('position') && widget.position === position); + } + if (position === 'left') { + return widgets.filter(widget => widget.hasOwnProperty('position') && (widget.type === 'toc' || widget.type === 'profile')); + } else { + return [] + } |

可以参考上述代码，即可实现不同页面不同 widget。

但两栏整体宽度跟三栏不同。因此强制指定为三栏布局，并且修改相应的宽度，这样所有的页面侧边栏宽度保持一致。

diff:layout/common/widget.ejs

|  |  |
| --- | --- |
| 1 2 3 4 5 6 | <% function side\_column\_class() {  switch (column\_count()) {  case 2: - return 'is-4-tablet is-4-desktop is-4-widescreen'; + return 'is-4-tablet is-4-desktop is-3-widescreen';  case 3: |

diff:layout/layout.ejs

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 | -<body class="is-<%= column\_count() %>-column"> +<body class="is-3-column">  <%- partial('common/navbar', { page }) %>  <% function main\_column\_class() {  switch (column\_count()) {  case 1:  return 'is-12';  case 2: - return 'is-8-tablet is-8-desktop is-8-widescreen'; + return 'is-8-tablet is-8-desktop is-9-widescreen'; |

并修改在不同屏幕小大下的宽度

diff:source/css/style.styl

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 | .is-2-column .container  max-width: screen-desktop - 2 \* gap  width: screen-desktop - 2 \* gap + .is-3-column .container + max-width: screen-widescreen - gap + width: screen-widescreen - gap  @media screen and (min-width: screen-fullhd) + .is-3-column .container + max-width: screen-fullhd - 2 \* gap + width: screen-fullhd - 2 \* gap  .is-2-column .container  max-width: screen-widescreen - 2 \* gap  width: screen-widescreen - 2 \* gap |

##### 调整阅读模式双栏

判断 post 页面，显示目录 toc ，修改宽度

themes/icarus/includes/helpers/layout.js

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 | - return widgets.filter(widget => widget.hasOwnProperty('position') && widget.position === position); + if (this.page.layout !== 'post') { + return widgets.filter(widget => widget.hasOwnProperty('position') && widget.position === position); + } + if (position === 'right') { + return widgets.filter(widget => widget.hasOwnProperty('position') && (widget.type === 'toc')); + } else { + return [] + } |

themes/icarus/layout/common/widget.ejs

|  |  |
| --- | --- |
| 1 2 3 | case 2: - return 'is-4-tablet is-4-desktop is-4-widescreen'; + return 'is-4-tablet is-4-desktop is-3-widescreen'; |

themes/icarus/layout/layout.ejs

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 | <head>  <%- partial('common/head') %>  </head> -<body class="is-<%= column\_count() %>-column"> +<body class="is-3-column">  <%- partial('common/navbar', { page }) %>  case 2: - return 'is-8-tablet is-8-desktop is-8-widescreen'; + return 'is-8-tablet is-8-desktop is-9-widescreen';  case 3:  return 'is-8-tablet is-8-desktop is-6-widescreen'  } |

themes/icarus/source/css/style.styl

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 | .is-2-column .container  max-width: screen-desktop - 2 \* gap  width: screen-desktop - 2 \* gap + .is-3-column .container + max-width: screen-widescreen - gap + width: screen-widescreen - gap  @media screen and (min-width: screen-fullhd) + .is-3-column .container + max-width: screen-fullhd - 2 \* gap + width: screen-fullhd - 2 \* gap |

##### 优化文章标题布局

标题移动到文章信息上方，增加更新时间，并增加了icon。

diff:layout/common/article.ejs

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 | <div class="card-content article <%= post.hasOwnProperty('direction') ? post.direction : '' %>"> + <h1 class="title is-size-3 is-size-4-mobile has-text-weight-normal"> + <% if (index) { %> + <a class="has-link-black-ter" href="<%- url\_for(post.link ? post.link : post.path) %>"><i class="fas fa-angle-double-right"></i><%= post.title %></a> + <% } else { %> + <i class="fas fa-angle-double-right"></i><%= post.title %> + <% } %> + </h1>  <% if (post.layout != 'page') { %>  <div class="level article-meta is-size-7 is-uppercase is-mobile is-overflow-x-auto">  <div class="level-left"> - <time class="level-item has-text-grey" datetime="<%= date\_xml(post.date) %>"><%= date(post.date) %></time> + <time class="level-item has-text-grey" datetime="<%= date\_xml(post.date) %>"><i class="far fa-calendar-alt">&nbsp;</i><%= date(post.date) %></time> + <% if (post.updated && post.updated > post.date) { %> + <time class="level-item has-text-grey is-hidden-mobile" datetime="<%= date\_xml(post.updated) %>"><i class="far fa-calendar-check">&nbsp;</i><%= date(post.updated) %></time> + <% } %>  <% if (post.categories && post.categories.length) { %> |

其中创建时间使用日期。

diff:source/js/main.js

|  |  |
| --- | --- |
| 1 2 3 4 5 | - if (typeof(moment) === 'function') { - $('.article-meta time').each(function () { - $(this).text(moment($(this).attr('datetime')).fromNow()); - }); - } |

##### 优化文章结尾布局

在文章结尾增加一个 hr，并修改 tags 展示。在预览时也显示 tags，并且将 Read More 按钮放置在右边。

diff:layout/common/article.ejs

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 | <% if (!index && post.tags && post.tags.length) { %> + <hr style="height:1px;margin:1rem 0"/>  <div class="level is-size-7 is-uppercase">  <div class="level-start">  <div class="level-item"> - <span class="is-size-6 has-text-grey has-mr-7">#</span> + <i class="fas fa-tags has-text-grey"></i>&nbsp;  <%- list\_tags(post.tags, {  class: 'has-link-grey ',  show\_count: false, - style: 'link' + style: 'link', + separator: ',&nbsp;'  }) %>  </div>  </div>  </div>  <% } %>  <% if (index && post.excerpt) { %> - <div class="level is-mobile"> + <hr style="height:1px;margin:1rem 0"/> + <div class="level is-mobile is-flex"> + <div class="level-start"> + <% if (post.tags && post.tags.length) { %> + <div class="level-item is-size-7 is-uppercase"> + <i class="fas fa-tags has-text-grey"></i>&nbsp; + <%- list\_tags(post.tags, { + class: 'has-link-grey ', + show\_count: false, + style: 'link', + separator: ',&nbsp;' + }) %> + </div> + <% } %> + </div>  <div class="level-start"> |

##### 优化个人信息布局

减少头像大小，头像下方计数的地方增加链接，follow前增加icon。

diff:layout/widget/profile.ejs

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 | - <nav class="level is-mobile"> + <nav class="level menu-list is-mobile" style="margin-bottom:1rem">  <div class="level-item has-text-centered is-marginless"> - <div> + <a href="<%- url\_for('/archives/') %>">  <p class="heading"> ...... |

##### 优化移动端显示

在移动端，隐藏 archive 和 tagcloud。

diff:layout/widget/archive.ejs

|  |  |
| --- | --- |
| 1 2 | -<div class="card widget"> +<div class="card widget is-hidden-mobile"> |

diff:layout/widget/tagcloud.ejs

|  |  |
| --- | --- |
| 1 2 | -<div class="card widget"> +<div class="card widget is-hidden-mobile"> |

##### 目录粘性布局

增加 column-left is-sticky 类。

diff:layout/widget/toc.ejs

|  |  |
| --- | --- |
| 1 2 | -<div class="card widget" id="toc"> +<div class="card widget column-left is-sticky" id="toc"> |

#### 功能

##### 增加版权说明

diff:layout/common/article.ejs

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 | <div class="content">  <%- index && post.excerpt ? post.excerpt : post.content %>  </div> + <% if (!index && post.layout === 'post' && post.copyright !== false) { %> + <ul class="post-copyright"> + <li><strong>本文标题：</strong><a href="<%= post.permalink %>"><%= page.title %></a></li> + <li><strong>本文作者：</strong><a href="<%= theme.url %>"><%= theme.author %></a></li> + <li><strong>本文链接：</strong><a href="<%= post.permalink %>"><%= post.permalink %></a></li> + <li><strong>发布时间：</strong><%= post.date.format("YYYY-MM-DD") %></li> + <li><strong>版权声明：</strong>本博客所有文章除特别声明外，均采用 <a href="https://creativecommons.org/licenses/by-nc-sa/4.0/deed.zh" rel="external nofollow" target="\_blank">CC BY-NC-SA 4.0</a> 许可协议。转载请注明出处！ + </li> + </ul> + <% } %>  <% if (!index && post.tags && post.tags.length) { %> |

并增加样式

diff:source/css/style.styl

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 | +.post-copyright + font-size: 1rem + letter-spacing: 0.02rem + word-break: break-all + margin: 2.5rem 0 0 + padding: 1rem 1rem + border-left: 3px solid #FF1700 + background-color: #F9F9F9 |

##### 增加文章版权

themes/icarus/layout/common/article.ejs

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 | <div class="content">  <%- index && post.excerpt ? post.excerpt : post.content %>  </div> + <% if (!index && post.layout === 'post' && post.copyright !== false) { %> + <ul class="post-copyright"> + <li><strong>本文标题：</strong><a href="<%= post.permalink %>"><%= page.title %></a></li> + <li><strong>本文作者：</strong><a href="<%= theme.url %>"><%= theme.author %></a></li> + <li><strong>本文链接：</strong><a href="<%= post.permalink %>"><%= post.permalink %></a></li> + <li><strong>发布时间：</strong><%= post.date.format("YYYY-MM-DD") %></li> + <li><strong>版权声明：</strong>本博客所有文章除特别声明外，均采用 <a href="https://creativecommons.org/licenses/by-nc-sa/4.0/deed.zh" rel="external nofollow" target="\_blank">CC BY-NC-SA 4.0</a> 许可协议。转载请注明出处！ + </li> + </ul> + <% } %>  <% if (!index && post.tags && post.tags.length) { %> |

下面文件加入格式

themes/icarus/source/css/style.styl

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 | border: none  .file  all: initial + +.post-copyright + font-size: 1rem + letter-spacing: 0.02rem + word-break: break-all + margin: 2.5rem 0 0 + padding: 1rem 1rem + border-left: 3px solid #FF1700 + background-color: #F9F9F9 |

这里大部分是参照 [自定义icarus](https://www.alphalxy.com/2019/03/customize-icarus/) 做的修改(点赞)

##### 配置 gitalk 评论

icarus 默认是支持 gitalk 的，所以只需要在 themes/icarus/\_config.yml 中设置一下即可

themes/icarus/\_config.yml

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 | comment:  # Name of the comment plugin  type: gitalk  owner: cloudy-liu # (required) GitHub user name  repo: cloudy-liu.github.io # (required) GitHub repository name  client\_id: xxxx # (required) OAuth application client id  client\_secret: xxx # (required) OAuth application client secret  admin: cloudy-liu |

还没配置的可以[参考这篇](https://www.liuyun.fun/2018/07/14/Hexo%E6%9B%B4%E6%8D%A2%E4%B8%BAGitTalk%E8%AF%84%E8%AE%BA%E7%B3%BB%E7%BB%9F/)做配置

##### 增加标题自动计数

diff:source/css/style.styl

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 | +.article {counter-reset:section} +.article h2{counter-reset:sub-section} +.article h3{counter-reset:composite} +.article h4{counter-reset:detail} +.article h2:before{content:counter(section) " ";counter-increment:section} +.article h3:before{content:counter(section) "." counter(sub-section) " ";counter-increment:sub-section} +.article h4:before{content:counter(section) "." counter(sub-section) "." counter(composite) " ";counter-increment:composite} |

##### 默认显示目录

正常开启目录需要在 meta 信息中加入 toc: true 开启，但绝大部分文章都是有目录的，因此修改为默认开启。因为有多个地方引用了 get\_config('toc') 并且需要判断当前页面，改起来稍微麻烦一些，因此直接修改了 get\_config 方法。

diff:includes/helpers/config.js

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 | return defaultValue;  } else {  const property = readProperty(specs, configName); - return property === null ? null : property[descriptors.defaultValue]; + const result = property === null ? null : property[descriptors.defaultValue]; + return (configName === 'toc' && this.page.layout === 'post' && result === null) ? true : result;  } |

##### 页面footer显示一组icon

默认情况下一个icon对应一个链接，但例如 CC BY-NC-SA 4.0 需要四个图标一组。因此修改代码，使得配置 link.icon可以是一个数组，效果可以参考页面底部。

diff:layout/common/footer.ejs

|  |  |
| --- | --- |
| 1 2 3 4 | <% } else { %> - <i class="<%= link.icon %>"></i> + <% for (let icon of (Array.isArray(link.icon) ? link.icon : [link.icon])) { %><i class="<%= icon %>"></i>&nbsp;<% } %>  <% } %> |

diff:includes/specs/icon\_link.spec.js

|  |  |
| --- | --- |
| 1 2 3 4 5 | icon: {  [required]: true, - [type]: 'string', + [type]: ['string', 'array'],  [doc]: 'Link icon class names' |

\_config.yml 中配置如下

\_config.yml

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 | footer:  links:  CC BY-NC-SA 4.0:  icon:  - fab fa-creative-commons  - fab fa-creative-commons-by  - fab fa-creative-commons-nc  - fab fa-creative-commons-sa  url: 'https://creativecommons.org/licenses/by-nc-sa/4.0/' |

##### 可以配置文章开头是否显示图片

Icarus 支持文章设置一个图片，在文章开头、最近的文章、时间线等地方显示。但有些图放大之后会显得很不协调，因此修改以支持自定义是否显示。修改 has\_thumbnail 方法，增加参数 isArticle 参数，默认 false，并在文章页面修改调用参数。

diff:includes/helpers/page.js

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 | - hexo.extend.helper.register('has\_thumbnail', function (post) { + hexo.extend.helper.register('has\_thumbnail', function (post, isArticle = false) {  const getConfig = hexo.extend.helper.get('get\_config').bind(this);  const allowThumbnail = getConfig('article.thumbnail', true);  if (!allowThumbnail) {  return false;  } + if (isArticle && post['article-thumbnail'] === false){ + return false; + }  return post.hasOwnProperty('thumbnail') && post.thumbnail; |

diff:layout/common/article.ejs

|  |  |
| --- | --- |
| 1 2 3 4 | <div class="card"> - <% if (has\_thumbnail(post)) { %> + <% if (has\_thumbnail(post, true)) { %>  <div class="card-image"> |

这样修改之后，如果文章 meta 信息中包含 article-thumbnail: false，就可以取消图片的显示。

##### ShareThis

**安装指南** [在线预览](https://blog.zhangruipeng.me/hexo-theme-icarus/uncategorized/sharethis-share-buttons/)

1. 访问[ShareThis](https://sharethis.com/)并点击页面上的“从分享按钮开始”(Start with Share Buttons)按钮。
2. 在“选择分享按钮类型”(Choose type of sharing button)页面选择你需要的按钮类型。 不要启用”GDPR规范工具“，否则可能会导致一些问题。 你也可以点击“自定义你的分享按钮”(Customize your share buttons)链接来进行按钮的进一步配置。 完成后点击“下一步”(Next)。

[选择按钮类型 - ShareThis](https://blog.zhangruipeng.me/hexo-theme-icarus/gallery/screenshots/sharethis-choose-button-type.png)

1. 在”选择你的CMS平台“(Choose your CMS platform)页面选择”HTML“并点击”下一步“(Next)。

[选择平台 - ShareThis](https://blog.zhangruipeng.me/hexo-theme-icarus/gallery/screenshots/sharethis-choose-platform.png)

1. 在“注册并获取代码”(Register and get the code!)页面输入邮箱和密码完成ShareThis的注册。
2. 最后，从HTML代码段中复制src中的URL地址到分享按钮配置中。

[获取代码 - ShareThis](https://blog.zhangruipeng.me/hexo-theme-icarus/gallery/screenshots/sharethis-get-code.png)

例如，下面的ShareThis代码：

AddThis代码

|  |  |
| --- | --- |
| 1 | <script type="text/javascript" src="https://platform-api.sharethis.com/js/sharethis.js#property=xxxxxxxxxxxxx&product=inline-share-buttons" async="async"></script> |

对应下面的主题配置：

themes/icarus/\_config.yml

|  |  |
| --- | --- |
| 1 2 3 | share:  type: sharethis  install\_url: https://platform-api.sharethis.com/js/sharethis.js#property=xxxxxxxxxxxxx&product=inline-share-buttons |

文章内容有误？请点击[此处](https://github.com/ppoffice/hexo-theme-icarus/edit/site/source/_posts/zh-CN/Share-Buttons.md)提交修改。

##### 总访问量统计

icarus 是用卜算子统计，默认只统计了访问人数，并没有访问量统计，需要在页脚修改下

themes/icarus/layout/common/footer.ejs

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 | @@ -15,9 +15,8 @@  href="https://github.com/ppoffice/hexo-theme-icarus" target="\_blank">Icarus</a>  <% if (has\_config('plugins.busuanzi') ? get\_config('plugins.busuanzi') : false) { %>  <br> - <span id="busuanzi\_container\_site\_uv"> - <%- \_p('plugin.visitor', '<span id="busuanzi\_value\_site\_uv">0</span>') %> - </span> + <span id="busuanzi\_container\_site\_uv"> 来访 <span id="busuanzi\_value\_site\_uv"></span>人</span> + <span id="busuanzi\_container\_site\_pv">, 总访问 <span id="busuanzi\_value\_site\_pv"></span>次</span>  <% } %>  </p> |

修改主题后，建议先进行清理，不然可能由于缓存问题，导致本地预览与部署不一致，KO !

|  |  |
| --- | --- |
| 1 2 | hexo clean hexo d -g |

#### 样式

##### 修改 logo 和 favicon

[用 Python 设计 Logo](https://www.alphalxy.com/2019/03/python-logo/)，并微调样式。

##### 按钮背景颜色增加渐变

diff:source/css/style.styl

|  |  |
| --- | --- |
| 1 2 3 4 5 6 | .menu-list li ul  margin-right: 0 + .menu-list a + transition: background-color 0.3s ease-in-out  .menu-list a.level  display: flex |

##### card 增加浮动效果

:hover 时增大阴影，并增加动画属性 ease-in-out。

diff:source/css/style.styl

|  |  |
| --- | --- |
| 1 2 3 4 5 | .card  border-radius: 4px  box-shadow: 0 4px 10px rgba(0,0,0,0.05), 0 0 1px rgba(0,0,0,0.1) + &:hover + box-shadow: 0 6px 15px rgba(0,0,0,0.15), 0 0 1px rgba(0,0,0,0.1) |

diff:source/js/animation.js

|  |  |
| --- | --- |
| 1 2 3 | element.style.transform = ''; - element.style.transition = 'opacity 0.3s ease-out, transform 0.3s ease-out'; + element.style.transition = 'opacity 0.3s ease-out, transform 0.3s ease-out, box-shadow 0.3s ease-in-out'; |

##### 更新

2020-02-16 合并了 [2.7.0](https://github.com/ppoffice/hexo-theme-icarus/releases/tag/2.7.0) 版本，冲突不算太多，有两个点需要注意一下。

icarus 的配置里面，plugins.clipboard 需要移除，相关的配置移动到 article.highlight.clipboard。

\_config.yml

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 | article:  # Code highlight settings  highlight:  # Code highlight themes  # https://github.com/highlightjs/highlight.js/tree/master/src/styles  theme: github-gist  # Show code copying button  clipboard: true  # Default folding status of the code blocks. Can be "", "folded", "unfolded"  fold: unfolded |

如果之前 links widget 没有配置链接，可能会遇到编译错误，需要增加一下判断。

diff:layout/widget/links.locals.js

|  |  |
| --- | --- |
| 1 2 3 4 5 | const links = get\_config\_from\_obj(locals.widget, 'links'); - if (Object.keys(links).length == 0) { + if (!links || Object.keys(links).length == 0) {  return null;  } |

##### 总结

这里只列举了部分改动，详细的差异可以查看 [diff](https://github.com/ppoffice/hexo-theme-icarus/compare/2.7.0...AlphaLxy:master)。

#### 参考资料

[icarus文档](https://blog.zhangruipeng.me/hexo-theme-icarus/)

https://blog.zhangruipeng.me/hexo-theme-icarus/

[自定义 icarus](https://www.alphalxy.com/2019/03/customize-icarus/)

**https://www.alphalxy.com/2019/03/customize-icarus/**

#### 修改要点

主题配置文件

Share button

89行注释掉

{!index ? <Share config={config} page={page} helper={helper} /> : null}

将捐赠注释掉后会报错 恢复

93行

\themes\icarus\layout\common\article.jsx"(93,17): {/\* Donate button \*/}

注释掉 {!index ? <Donates config={config} helper={helper} /> : null}

{/\* Comment \*/}

注释掉

{!index ? <Comment config={config} page={page} helper={helper} /> : null}

# Jekyll

## 安装

一、官网链接

官网安装方式介绍：

http://www.ruby-lang.org/en/downloads/

二、安装

安装方式一：源码安装

安装方式二：RubyInstaller（windows下），使用这种方式安装时会自动安装好Ruby以及RubyGems。

Ruby 安装 - Windows

https://rubyinstaller.org/downloads/

Ruby Gems 下载

https://rubygems.org/pages/download

三、测试

Ruby：在cmd中输入“ruby -v”如可以正确显示版本信息则安装成功。

RubyGems：在cmd中输入“gem -v”如可以正确显示版本信息则安装成功。

## 增加新的pem证书，解决问题

下载新证书： <http://curl.haxx.se/ca/cacert.pem>

配置环境变量：SSL\_CERT\_FILE 值为刚下载证书的存放位置： C:\Ruby21-x64\lib\ruby\site\_ruby\2.1.0\rubygems\ssl\_certs\cacert.pem

解决方法：

把地址换成：https://gems.ruby-china.com

gem source -a https://gems.ruby-china.com

安装完以后，输入以下命令查看是否替换成功

gem sources -l

//更换gem源

gem sources --remove https://rubygems.org/

//查看gem源

gem sources -l

//更新gem

gem update --system

//安装jekyll

gem install jekyll

//创建jekyll工程目录

jekyll new myblog

//切换到工程目录，并开启服务

cd myblog

jekyll serve

https://rubyinstaller.org/downloads/

https://github.com/oneclick/rubyinstaller2

使用Jekyll

jekyll的下载安装

安装ruby：因为jekyll依赖于ruby，所以必须先安装ruby，前往https://www.ruby-lang.org/en/downloads/根据自己的系统下载安装ruby。

安装gem：安装了ruby之后，默认会安装gem，可以在命令行里测试是否已经安装成功，如下图。如果没有安装gem，可以前往此处https://rubygems.org/pages/download下载安装

在这里插入图片描述

安装jekyll：在命令行中输入gem install jekyll，jekyll安装完成后，还不能使用，还要安装一些组件。此时在命令行中输入gem install bundler，安装完成后，再在命令行中输入bundle install。这些步骤做完之后，jekyll就可以使用了。

jekyll的使用

关于Jekyll的使用，可以前往Jekyll官网查看，这里给出简单的使用方法。

利用命令行进入一个空文件夹，然后执行jekyll new my-first-website，然后jekyll就生成一个my-first-website文件夹，里面的内容是这样的：

在这里插入图片描述

在命令行中进入该文件夹，执行jekyll server，然后再浏览器中打开地址http://127.0.0.1:4000/，就可以看到网页的内容在这里插入图片描述

更换jekyll主题

寻找jekyll主题

前面可以看到，Jekyll默认生成的网页主题单调，我们可以换一个更好的主题。前往jekyll theme网站寻找自己喜欢的主题，我是用的主题是Less-Or-More。点击下图中的dowload按钮即可下载。

在这里插入图片描述

使用jekyll主题

将刚刚下载好的主题压缩包解压，然后在命令行中进入LessOrMore-master文件夹，执行jekyll server命令，然后再浏览器中打开地址http://127.0.0.1:4000/LessOrMore/，即可看到网站内容。那么，使用了这个主题之后，之前利用jekyll new my-first-website生成的文件夹及里面的内容就可以删掉不要了。

在这里插入图片描述

将本地网站内容部署到github上

复制文件：之前的步骤我们利用git工具将github上一个名为username.github.io的空仓库克隆到了本地文件夹，那么我们将LessOrMore主题文件夹里面的全部文件复制到那个本地文件夹里。（注意，是复制LessOrMore主题文件夹里面的全部文件，而不是LessOrMore主题文件夹），即以下文件

在这里插入图片描述

修改\_config.yml文件：注意，该主题的README.md文件里面有些一些注意事项，即我们要做一个修改，把\_config.yml文件里的baseurl属性值修改为空，如下图所示：

在这里插入图片描述

提交到github：利用git工具执行git add -A和git commit -m "xxx"命令，然后再执行git push命令，就能将这些文件提交到自己的github账号上，如下图

在这里插入图片描述

浏览器查看网站：在浏览器中输入https://username.github.io/，把username修改为自己的github用户名，即可查看网站内容。

如何更新博客

这一点在LessOrMore的README.md文件里面有提到，我把那段文字复制下来。

在LessOrMore/\_posts目录下新建一个文件，可以创建文件夹并在文件夹中添加文件，方便维护。在新建文件中粘贴如下信息，并修改以下的titile,date,categories,tag的相关信息，添加\* content {:toc}为目录相关信息，在进行正文书写前需要在目录和正文之间输入至少2行空行。然后按照正常的Markdown语法书写正文。

---

layout: post

#标题配置

title: 标题

#时间配置

date: 2016-08-27 01:08:00 +0800

#大类配置

categories: document

#小类配置

tag: 教程

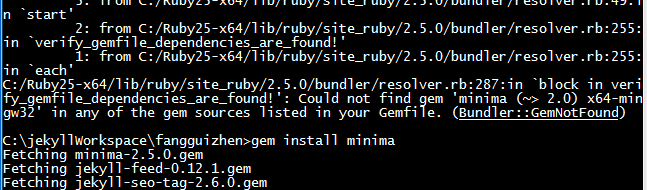
---

\* content

{:toc}

我是正文。我是正文。我是正文。我是正文。我是正文。我是正文。

问题如下：



 解决方法如下

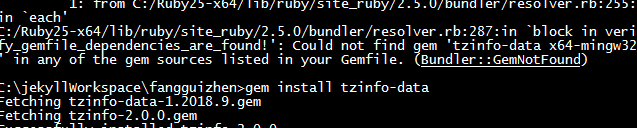
执行：

gem install minima

即可解决。

同样，对于类似的问题

Could not find gem 'tzinfo-data x64-mingw32' in any...



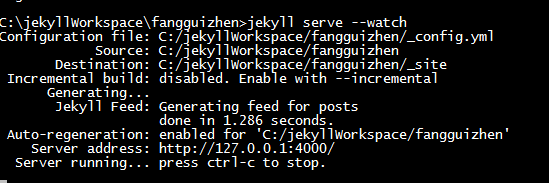
 执行：

gem install tzinfo-data

即可解决。

后面要是遇到类似的问题执行相似的命令即可。

最后，服务器启动成功



## 主题

 安装主题

rake theme:install git="https://github.com/mmistakes/minimal-mistakes.git"

$ rake theme:install git="[https://github.com/jekyllbootstrap/theme-the-program.git](https://link.jianshu.com?t=https:/github.com/jekyllbootstrap/theme-the-program.git)" // 在线安装

$ rake theme:install name="THEME-NAME" // 本地安装，将下载的主题放在 ./\_theme\_packages 中

 切换主题  
$ rake theme:switch name="THEME-NAME"

## 前言

最近在利用GitHub Pages搭建自己的博客，决定用GitHub推荐的Jekyll作为网站生成器。博客搭建过程中至关重要的一点就是选择一个自己喜欢的主题模板。目前我也没有能力自己修改，所以还是花点时间找个自己心仪的主题。

## 对主题的要求

清楚自己的需求才能做出正确的选择。我对博客主题的要求主要有：

* **简洁**。看着舒服，不用太多花哨的动画、背景，加载慢，也容易分心。
* **注重阅读体验**。供人阅读，这点重重之重。阅读界面两边不要有多余边框，但是最好有目录，可以辅助阅读。
* **美观**。兼具一定的美观效果，不要看起来根本没有样式一样。
* **响应式设计**。有时候确实不方便开电脑。

## 推荐主题

正题，推荐的主题是我觉得不错的，个人喜好问题，我希望你能喜欢，但更希望你能找到你喜欢的。图片不贴了，自己点连接实际看一下才知道。

### 2.1 NexT

地址：[NexT官方地址](http://jekyllthemes.org/themes/jekyll-theme-next/)  
优点：简约美观、阅读界面有路径。  
缺点：只有标签，没有主题。我还是希望我的博客能按主题归类。

### 2.2 Hux Blog

地址：[huxpro.github.io](https://github.com/Huxpro/huxpro.github.io)  
评价同上，没有主题分类。

### 2.3 码志

地址：[码志](https://github.com/mzlogin/mzlogin.github.io)  
评价：简约，有分类有目录，相应式设计。很贴合我的要求，爱上了，直接用。

## 附上一些主题站点

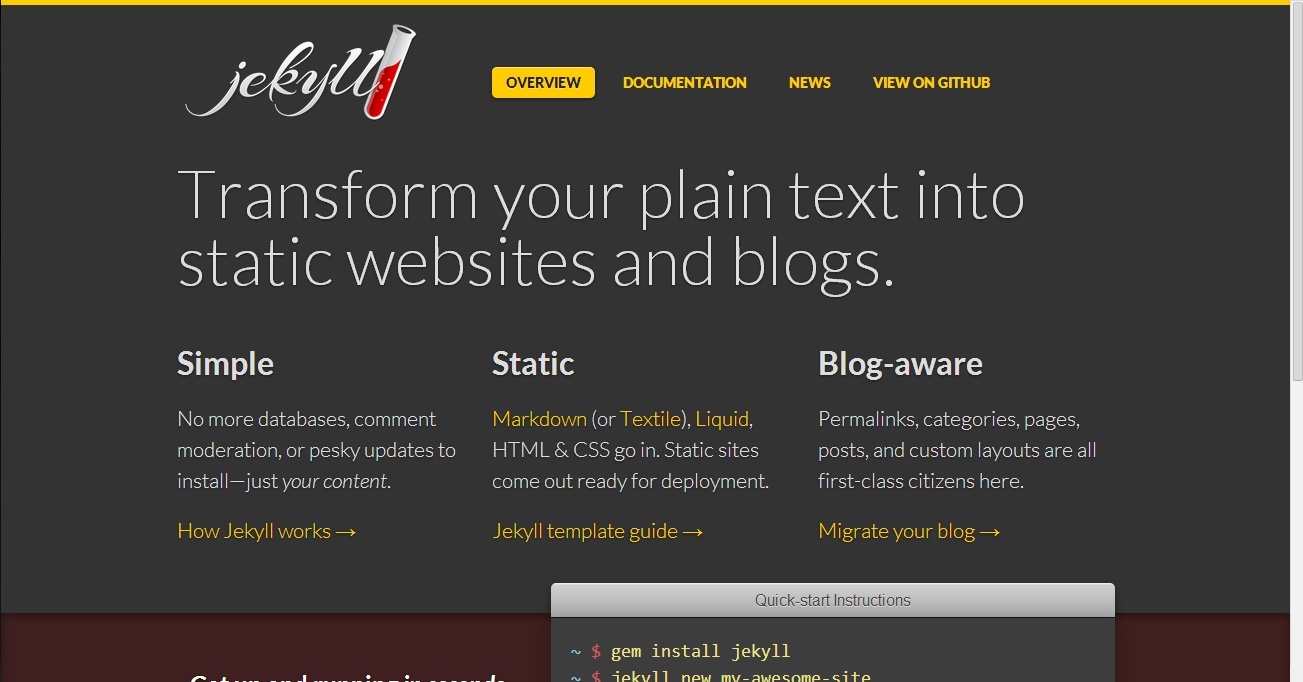
如果你觉得不喜欢，可以自己去主题站点找找看，不过那真的是海淘，累死人，你还不是道那些主题对中文支持怎么样。

1. [Jekyll官方主题](http://jekyllthemes.org/)
2. [GitHub上整理的Jekyll主题](https://github.com/jekyll/jekyll/wiki/Sites)
3. [知乎-有哪些简洁明快的 Jekyll 模板？](https://www.zhihu.com/question/20223939)

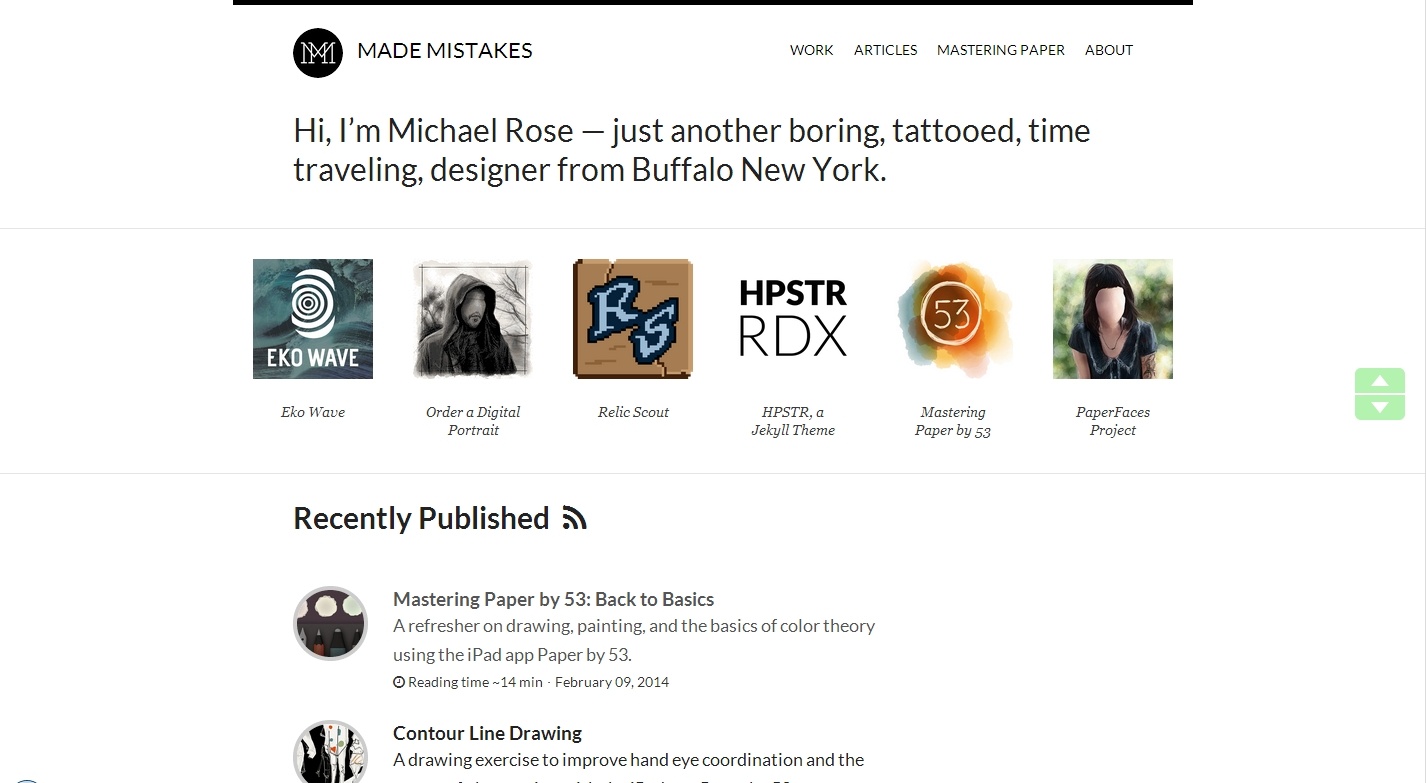
以下主题来源

https://blog.csdn.net/garfielder007/article/details/82875394

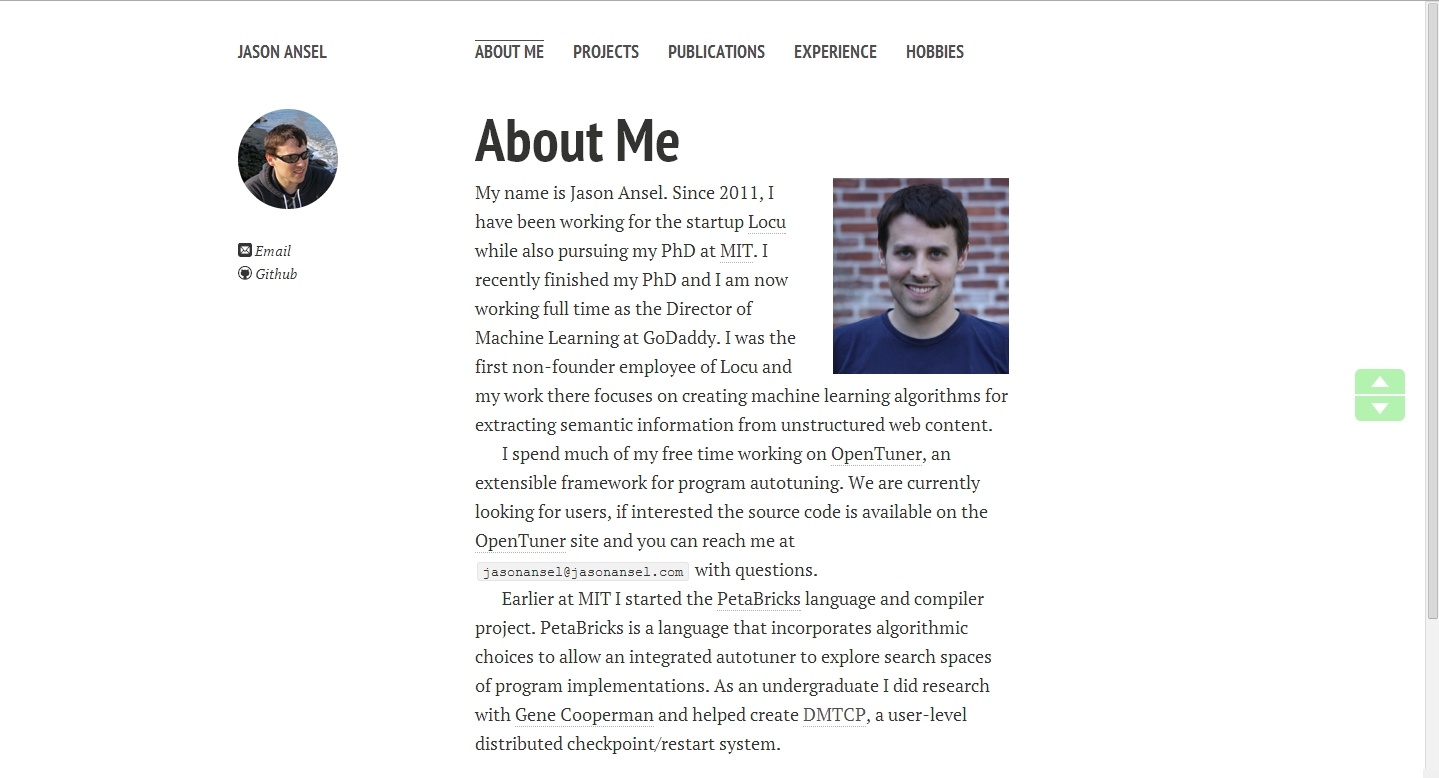
**项目模板**

[Jeklly](https://jekyllrb.com/)官网本身采用的就是Jekyll,设计大方美观，不过我不怎么喜欢它页面的配置。灰黑配色虽然很适合在编程时可以护眼，但作为一个网页配色，灰黑我会很慎用它们作为主配色。GitHub上fork该repository的人很多啊。对应中文项目模板为[jekyllcn](http://jekyllcn.com/)

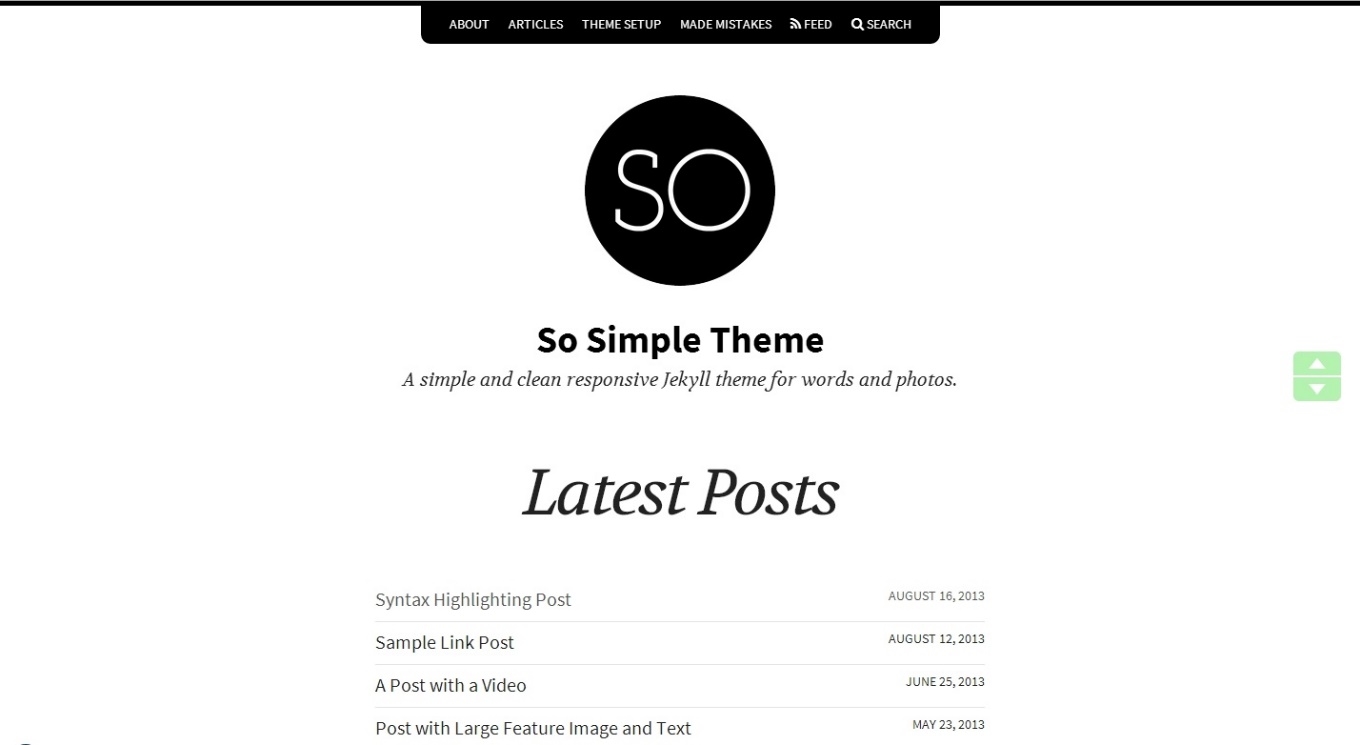
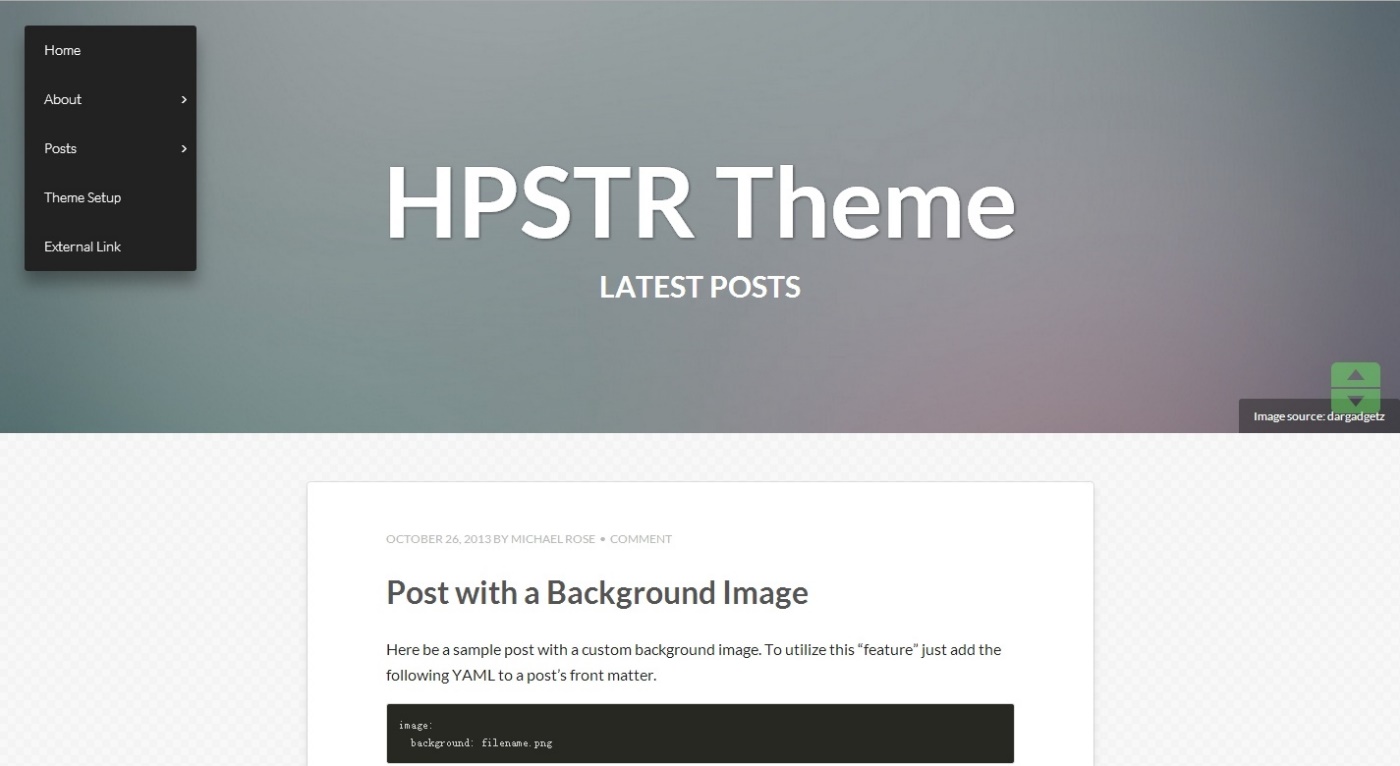
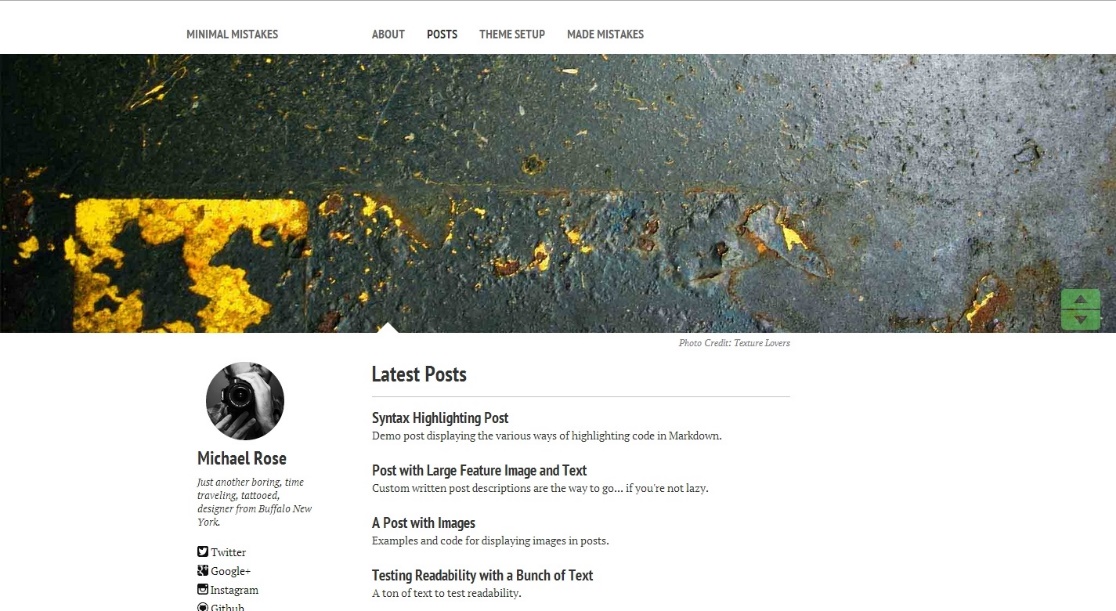
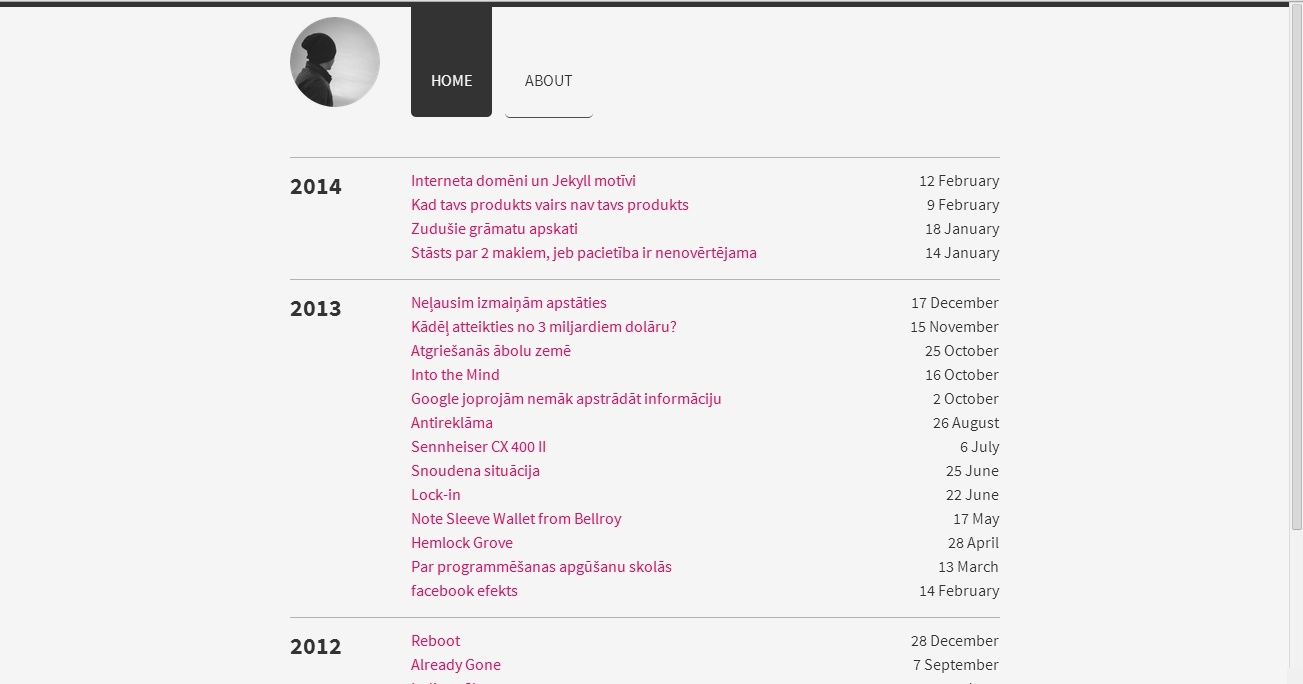
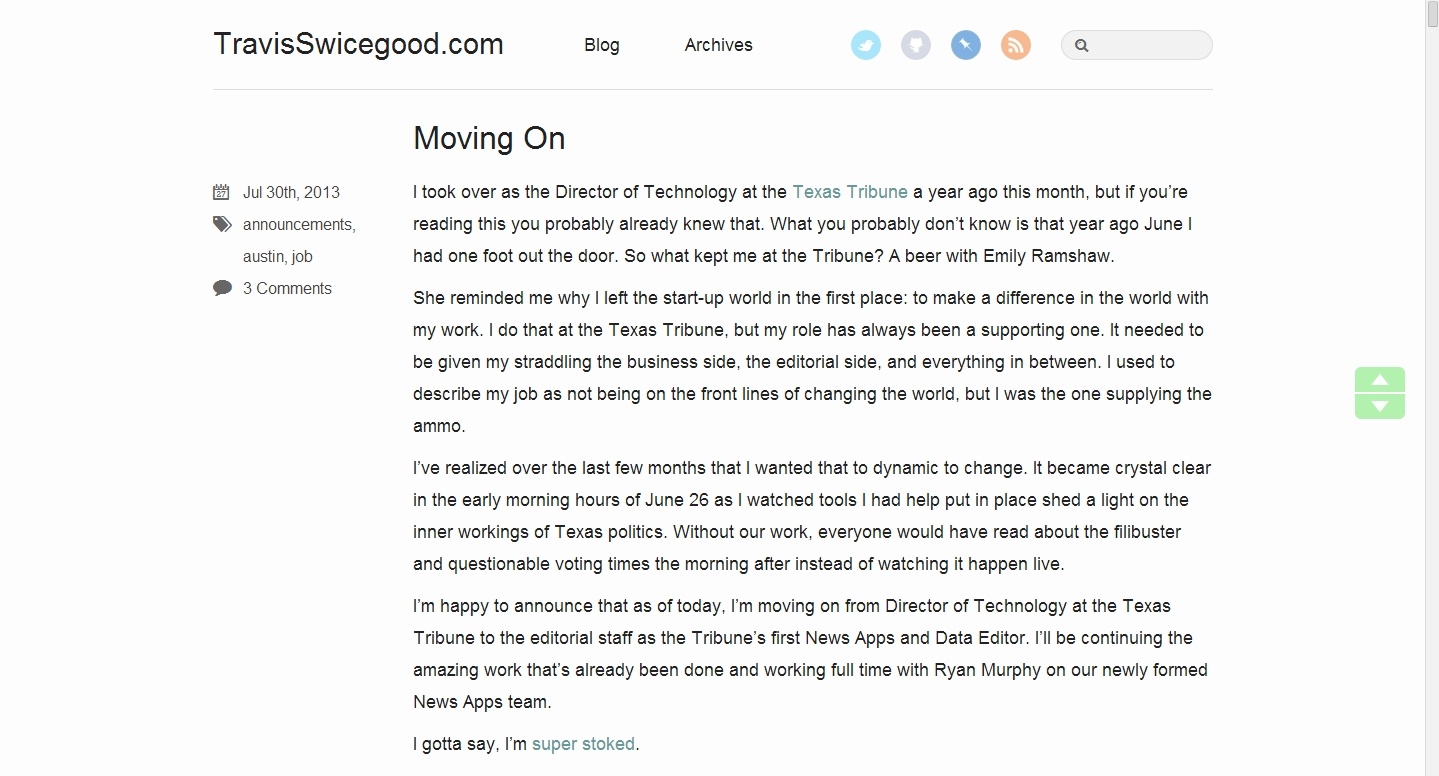
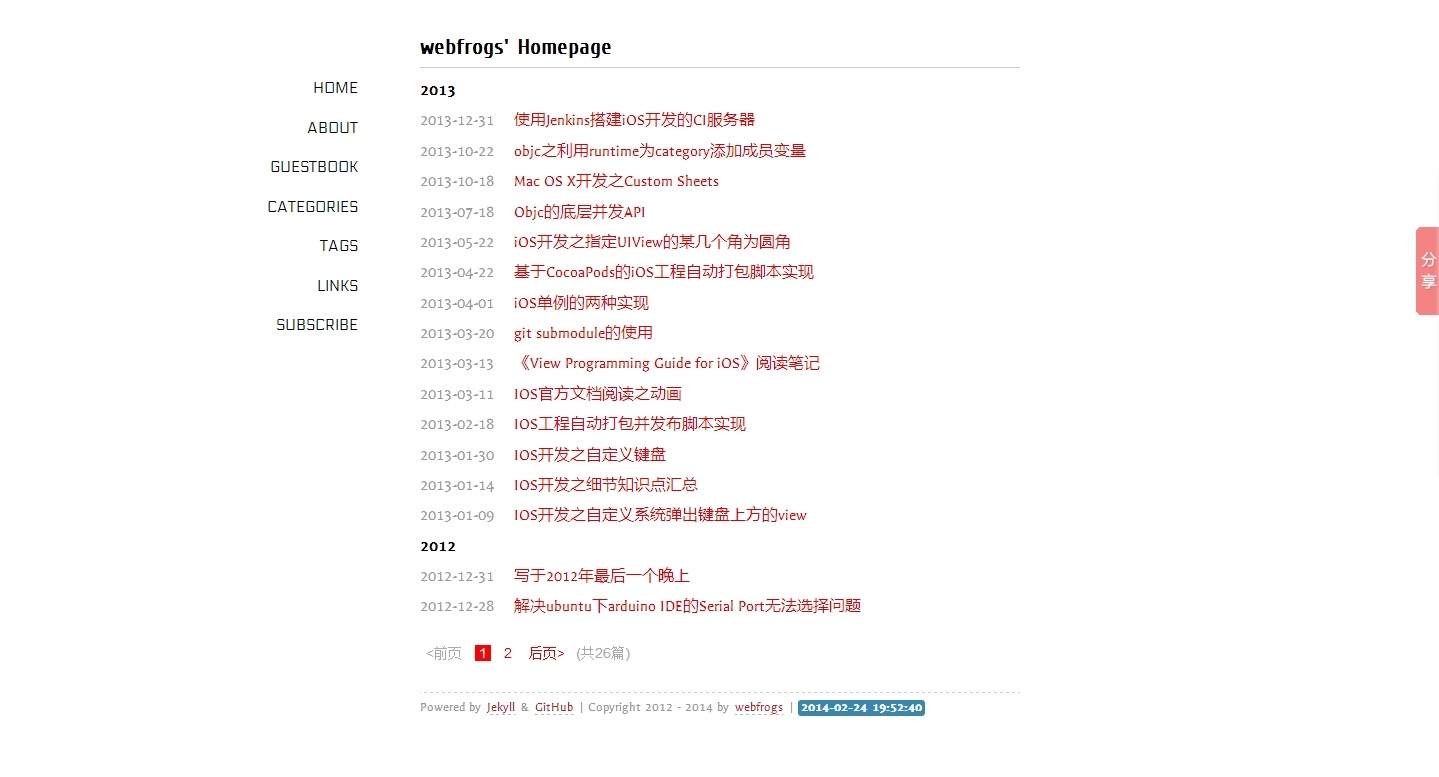
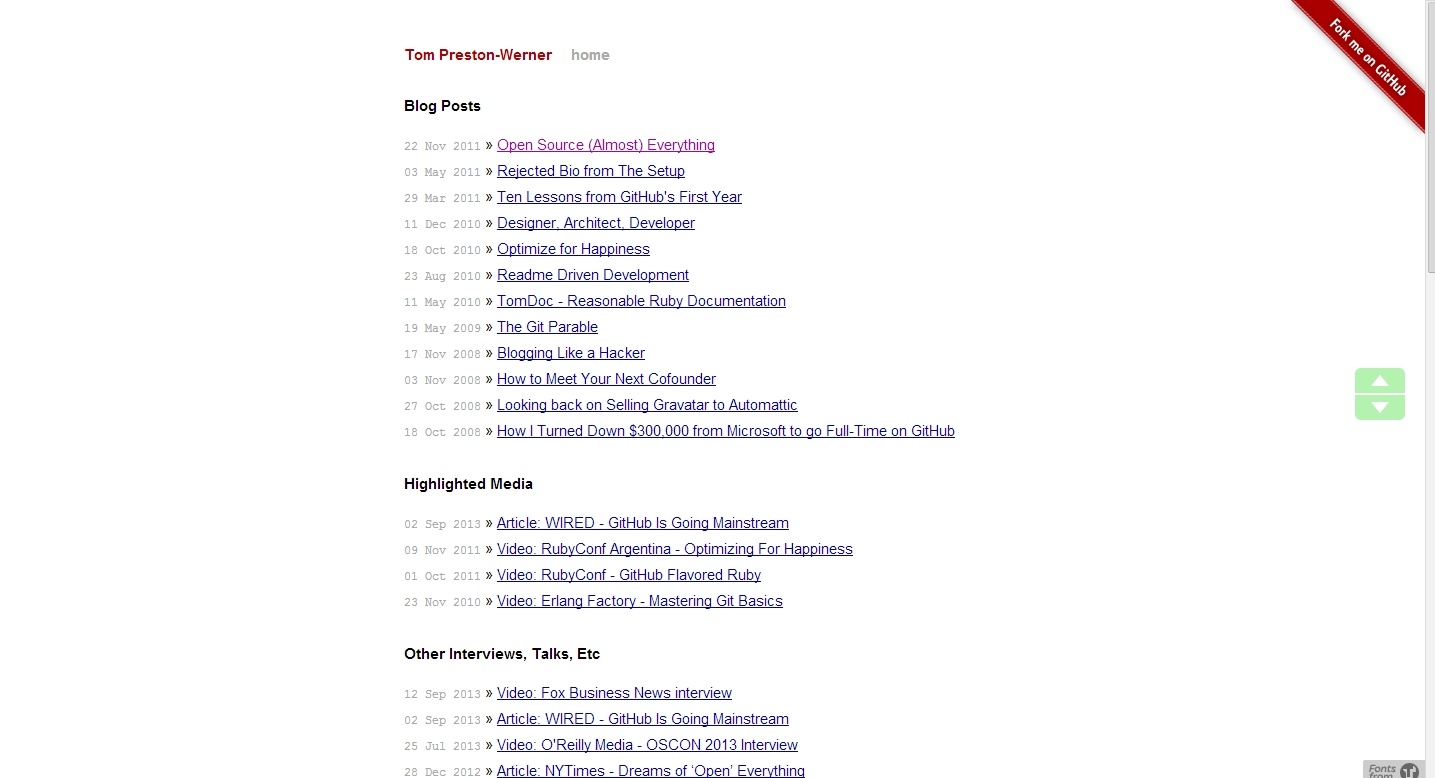
**公司模板**

1. mmistakes/made-mistakes，演示[主页](http://mademistakes.com/)，稍微改改应该可以直接作为公司用的模板。
2. [libgit2](https://libgit2.github.com/)

**学术主页模板**

1. jansel.github.io，演示[主页](http://jasonansel.com/) 页面元素简约、可以作为学术主页模板

**博客模板**

1. mmistakes/so-simple-theme，演示[主页](https://mmistakes.github.io/so-simple-theme/) 很简约大方的一个博客模板。
2. mmistakes/hpstr-jekyll-theme，演示[主页](https://mmistakes.github.io/hpstr-jekyll-theme/) 也是[mmistakes](https://github.com/mmistakes)设计的，很喜欢他的作品，堪称艺术品。
3. mmistakes/minimal-mistakes，演示[主页](https://mmistakes.github.io/minimal-mistakes/) 不解释，直接看网页截图。
4. aigarsdz/brume，演示[主页](http://dzerviniks.com/),很简洁美观的一个博客主题，也是我现在博客用的模板。前段时间想用该模板在我的master下简历博客分支，结果发现了这个模板还有点问题，然后在brume向aigarsdz提交了个问题，没想到aigarsdz很快就回了，很nice。
5. Github [likebeta](https://github.com/likebeta)，演示[主页](http://blog.ixxoo.me/),页面大方美观，还用了豆瓣API把豆瓣上的个人书单列出了，页面配色也很不错。
6. tswicegood/travisswicegood.com，演示[主页](http://travisswicegood.com/),第5个主题跟这个很相似。
7. Github [luolinjia](https://github.com/luolinjia),设计得很不错的一个博客，作者写作很勤啊。[主页](http://ideex.name/)
8. webfrogs.github.com，演示[主页](http://webfrogs.me/),文章排版很好看，之前用过一段时间，后来觉得导航没涉及好，就换成别的了。
9. mojombo/mojombo.github.io，演示[主页](http://tom.preston-werner.com/)，简约风。
10. liberize.github.com，演示[主页](http://liberize.me/) 看着还行。
11. [corincole.com](https://github.com/corincole/corincole.com),演示[主页](http://www.corincole.com/)
12. [lazywei.github.io]，演示[主页](https://lazywei.github.io/)，简约风。
13. hexo-theme-raytaylorism [Demo](http://raytaylorlin.com/)。
14. [xguox.github.com](https://xguox.github.io/)。
15. [webfrogs](http://webfrogs.me/),跟[yihui.name](http://yihui.name/cn/)的一样。
16. [macduff.github.com](https://macduff.github.io/),原来为自己配置MathJax是找到的一个博客，容器居中应该更好看。
17. [Poised-flw-blog](http://www.poised-flw.com/),[Github](https://github.com/luofei2011/Poised-flw-blog)。

补充三个寻找模板的地方：

1. [Jekyll Theme](http://jekyllthemes.org/)官方提供的主题模板，上面有很多好看的模板供选择。
2. [Jekyll wiki site](https://github.com/jekyll/jekyll/wiki/Sites),表单上列举了很多采用Jekyll模板的网站，可以在这里面找别人的模板，fork过来就可以用了。
3. [掌心](http://www.zhanxin.info/themes.html),在该页，作者提供了几个自己制作的Jekyll主题，其中有几个本小子很喜欢。

https://github.com/mzlogin/mzlogin.github.io

## Jekyll资源

## Jekyll主题资源

### NexT

http://theme-next.simpleyyt.com/getting-started.html

### minimal-mistakes

https://github.com/mmistakes/minimal-mistakes

### 码志

https://github.com/mzlogin/mzlogin.github.io

# MKDocs

<https://github.com/mkdocs>

https://www.mkdocs.org/user-guide/configuration/

pip install -i https://pypi.tuna.tsinghua.edu.cn/simple

pip install -i https://pypi.tuna.tsinghua.edu.cn/simple mkdocs mkdocs-material

Pip

python -m pip install --upgrade pip

python -m pip install --upgrade pip

## mkdocs-material介绍

符合google material ui规范的静态文档网站生成器，使用markdown进行文档书写

**mkdocs**

* python编写的markdown解释器、编译器，带有本地cli工具
* 自带基于Tornado的小型http服务，用于本地调试
* 内置一键式发布至GitHub Pages
* 内置mkdocs风格、readthedocs风格的主题，并支持自定义主题
* 支持调用python模块实现语法及渲染的扩展

**mkdocs-material**

* python模块，符合google material ui规范的mkdocs自定义主题
* 针对特定语法、功能做了渲染优化
* 根据客户端浏览器页面尺寸自动缩放，对PC、移动设备都友好
* 丰富的页面配色，多达19种主体配色和16种悬停链接文字配色
* 支持中文搜索
* 支持统计功能，如百度统计，谷歌统计

## 安装

pip install mkdocs mkdocs-material

**初始化项目**

mkdocs new my-project

会生成my-project目录，进入该目录里，可以看到默认放置了一些文件，包括mkdocs.yml，这是主配置文件

**修改主题**

mkdocs.yml里添加:

theme:

name: material

## 运行

# 在my-project目录里执行

mkdocs serve

**生成站点文件**

在git目录下执行命令

mkdocs build

## mkdocs.yml注意事项

由于是yaml格式，因此首先要符合yaml的语法要求。

docs下需要一个index.md，作为站点首页

文档层次结构虽然可以很多层，但最佳实践是控制在2层内，最多不要超过3层，否则展示会不够友好

## 添加页面

在my-project/docs/里放置.md文件，可以自行组织目录结构

然后在mkdocs.yml里添加，比如这样:

nav:

- 介绍: index.md

- 安装:

- 本地环境搭建: install/local.md

- 发布至GitHub Pages: install/github-pages.md

- 发布至自己的HTTP Server: install/http-server.md

- 语法:

- 语法总览: syntax/main.md

- 标题: syntax/headline.md

- 段落: syntax/paragraph.md

* 上面的index.md就是放置在my-project/docs/index.md
* 上面的local.md就是放置在my-project/docs/install/local.md

## 添加扩展

只有添加了扩展，才能完美使用mkdocs-material官方支持的所有markdown语法

mkdocs.yml里添加:

markdown\_extensions:

- admonition

- codehilite:

guess\_lang: false

linenums: false

- toc:

permalink: true

- footnotes

- meta

- def\_list

- pymdownx.arithmatex

- pymdownx.betterem:

smart\_enable: all

- pymdownx.caret

- pymdownx.critic

- pymdownx.details

- pymdownx.emoji:

emoji\_generator: !!python/name:pymdownx.emoji.to\_png

- pymdownx.inlinehilite

- pymdownx.magiclink

- pymdownx.mark

- pymdownx.smartsymbols

- pymdownx.superfences

- pymdownx.tasklist

- pymdownx.tilde

## 主题

**# Welcome to MkDocs**

For full documentation visit [mkdocs.org](https://www.mkdocs.org).

**## Commands**

\* `mkdocs new [dir-name]` - Create a new project.

\* `mkdocs serve` - Start the live-reloading docs server.

\* `mkdocs build` - Build the documentation site.

\* `mkdocs -h` - Print help message and exit.

**## Project layout**

    mkdocs.yml    # The configuration file.

    docs/

        index.md  # The documentation homepage.

        ...       # Other markdown pages, images and other files.

## 使用MkDocs 管理博客

### 需求

* **MarkDown 文本**
* **文档结构化，方便组织**
* **搜索整理**
* **shell 编辑**
* **随时随地编辑**

### MkDocs 优点

* **轻量简单** :想用HEXO，但是NodeJS安装不上，费事费力
* **方面客户化**: Python包，修改简单

### 方案

* 内部服务器 稳定，但是访问困难
* Home Raspberry
* 发布到**Github Page**

### 试验环境

* 安装环境Centos6

[官方网站](https://link.jianshu.com?t=http:/www.mkdocs.org/#installation)

pip install mkdocs

# 创建笔记项目

mkdocs new mdnote

cd mdnote

* 模板

pip install mkdocs-bootswatch

可用的样式：

* amelia
* cerulean
* cosmo
* cyborg
* flatly
* journal
* readable
* simplex
* slate
* spacelab
* united
* yeti

A clean, responsive PowerShell flavored MkDocs theme - forked from Cinder

 源代码名称:**cinder**

 源代码网址:<http://www.github.com/chrissimpkins/cinder>

 [cinder源代码文档](http://www.github.com/chrissimpkins/cinder/wiki)

 [cinder源代码下载](http://www.github.com/chrissimpkins/cinder/releases) Git URL:

git://www.github.com/chrissimpkins/cinder.git

Git Clone代码到本地:

git clone http://www.github.com/chrissimpkins/cinder

Subversion代码到本地:

$ svn co --depth empty http://www.github.com/chrissimpkins/cinder

Checked out revision 1.

$ cd repo

$ svn up trunk

## Cinder

一个干净的，响应式的MkDocs static 文档站点生成器主题

[演示和文档管理](http://sourcefoundry.org/cinder/)

## MKDocs资源

## MKDocs主题资源

# Vue

# React

# Angular

# Django

# 附录：资源

Python： http://www.python.org/

pip： https://pypi.python.org/pypi

[pelican](http://blog.getpelican.com/)：http://blog.getpelican.com/

[markdown](http://daringfireball.net/projects/markdown/syntax)：http://daringfireball.net/projects/markdown/syntax

[markdownpad](http://markdownpad.com/)：http://markdownpad.com/

[git：](http://msysgit.github.io/) http://msysgit.github.io/

[windows下make](http://www.equation.com/servlet/equation.cmd?fa=make)：http://www.equation.com/servlet/equation.cmd?fa=make