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；

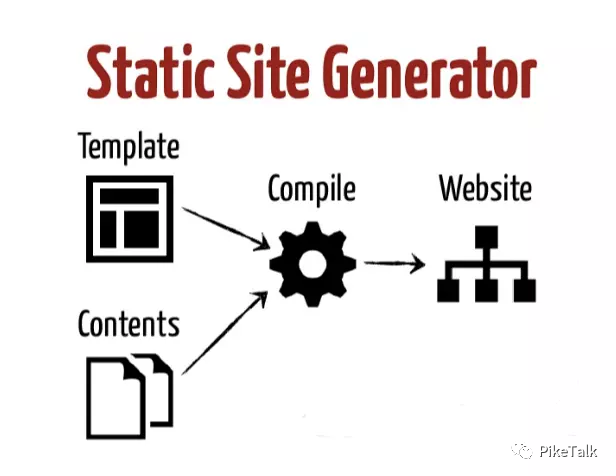
2.没有服务端处理或者数据库交互；

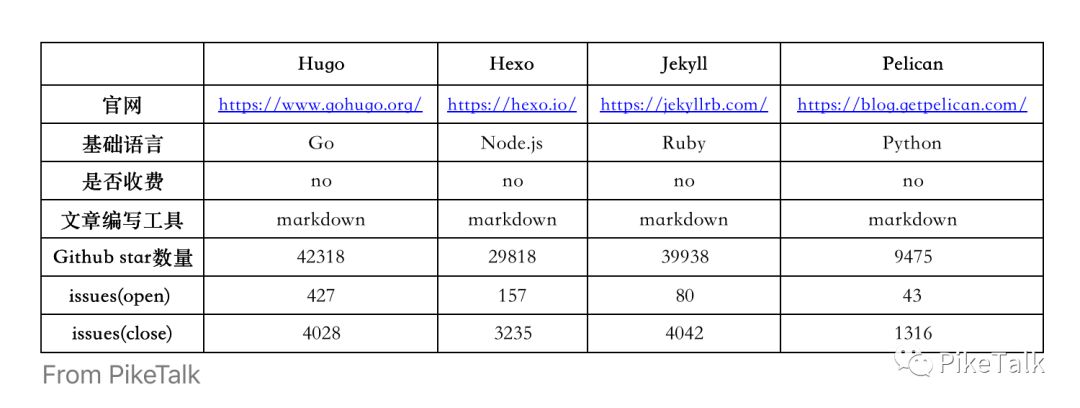
3.比动态网站更安全；

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。





**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)

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。

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**

        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

[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。

**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

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/

# 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/

**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）。

# 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