三分天下如何选择vue,angular，react

随着近几年的发展，前端已经不单单是html+css编写样式静态页了，而是往着更深层次发展，越来越多的前端js框架：angular、vue、react纷纷研发出来了。而作为前端的你。该如何选择？

如果你现在在学前端或者工作是前端，欢迎加入我们大前端交流学习群：575546903【点击右边群名即可进群→】：点击链接加入群【WEB前端开发】：WEB前端开发

Vue.js ： 网址：网页链接

Vue.js 专注于 MVVM 模型的 ViewModel 层。它通过双向数据绑定把 View 层和 Model 层连接了起来。实际的 DOM 封装和输出格式都被抽象为了 Directives 和 Filters。Vue.js和其他库相比是一个小而美的库，作者的主要目的是通过一个尽量简单的 API 产生可反映的数据绑定和可组合的视图组件，感觉作者的思路非常清晰。

优点：

简单：官方文档很清晰，比 Angular 简单易学。
快速：异步批处理方式更新 DOM。

组合：用解耦的、可复用的组件组合你的应用程序。
紧凑：~18kb min+gzip，且无依赖。
强大：表达式 & 无需声明依赖的可推导属性 (computed properties)。
对模块友好：可以通过 NPM、Bower 或 Duo 安装，不强迫你所有的代码都遵循 Angular 的各种规定，使用场景更加灵活。

缺点：

新生儿：Vue.js是一个新的项目，没有angular那么成熟。
影响度不是很大：google了一下，有关于Vue.js多样性或者说丰富性少于其他一些有名的库。
不支持IE8：不过AngularJS 1.3也抛弃了对IE8的支持。这一点对于那些需要支持IE8的项目就不好了，不过这也是web前端开发的一个趋势，像IE低版本就应该退出历史舞台了，通过改变我们的前端思维，而不是顺应那些使用老版本而不去升级的人。

Angularjs 网址：网页链接

AngularJS最近很火，追随者也很多。完全使用JavaScript编写的客户端技术。同其他历史悠久的Web技术（HTML、CSS和JavaScript）配合使用，使Web应用开发比以往更简单、更快捷“。当你学习它的时候，我相信你会被它的很多新特效所吸引。

优点：

动态视图：以前从来没有想过js可以如此扩展HTML的属性，但是AngularJs做到了，它替我们静态的HTML加了很多扩展性功能，有一种让HTML由死变活的感觉。
完善：是一个比较完善的前端MVW框架，包含模板，数据双向绑定，路由，模块化，服务，依赖注入等所有功能，模板功能强大丰富，并且是声明式的，自带了丰富的 Angular 指令。
Google维护：AngularJS有Google来维护，无疑有了一个强大的后台，对于推广和维护明显比Vue.js和avalon有优势，社区也非常活泼，能够很好促进它的发展。
AngularJS & Ionic：Ionic: Advanced HTML5 Hybrid Mobile App Framework，这俩就是一个好基友，Ionic通过用AngularJS为了创建一个框架，最适合开发的丰富和强大的应用程序。
如果你现在在学前端或者工作是前端，欢迎加入我们大前端交流学习群：575546903【点击右边群名即可进群→】：点击链接加入群【WEB前端开发】：WEB前端开发

缺点：

大而全：学习起来有难度，比较难理解一些。
推翻重写：AngularJS2.0和1.0相比，会把之前的推翻重写，两个框架的改变很大，基本是两个框架了，等于是说等到2.0出来后又需要从头开始

React

网址：网页链接

中文社区：网页链接

优点：

1、React速度很快：它并不直接对DOM进行操作，引入了一个叫做虚拟DOM的概念，安插在javascript逻辑和实际的DOM之间，性能好。

2、跨浏览器兼容：虚拟DOM帮助我们解决了跨浏览器问题，它为我们提供了标准化的API，甚至在IE8中都是没问题的。

3、一切都是component：代码更加模块化，重用代码更容易，可维护性高。

4、单向数据流：Flux是一个用于在JavaScript应用中创建单向数据层的架构，它随着React视图库的开发而被Facebook概念化。

5、同构、纯粹的javascript：因为搜索引擎的爬虫程序依赖的是服务端响应而不是JavaScript的执行，预渲染你的应用有助于搜索引擎优化。

6、兼容性好：比如使用RequireJS来加载和打包，而Browserify和Webpack适用于构建大型应用。它们使得那些艰难的任务不再让人望而生畏。

缺点：

1、React本身只是一个V而已，并不是一个完整的框架，所以如果是大型项目想要一套完整的框架的话，基本都需要加上ReactRouter和Flux才能写大型应用。

2、大多数坑没踩出来。。。。。

大概就是现在还太新了很难说将来有没有大的API变化，目前在大的稳定的项目上采用React的，我也就只知道有Yahoo的Email。

所以现在很少有批评React的声音，也许不是他真的就没有坑，而是那些坑还没有被踩出来而已。

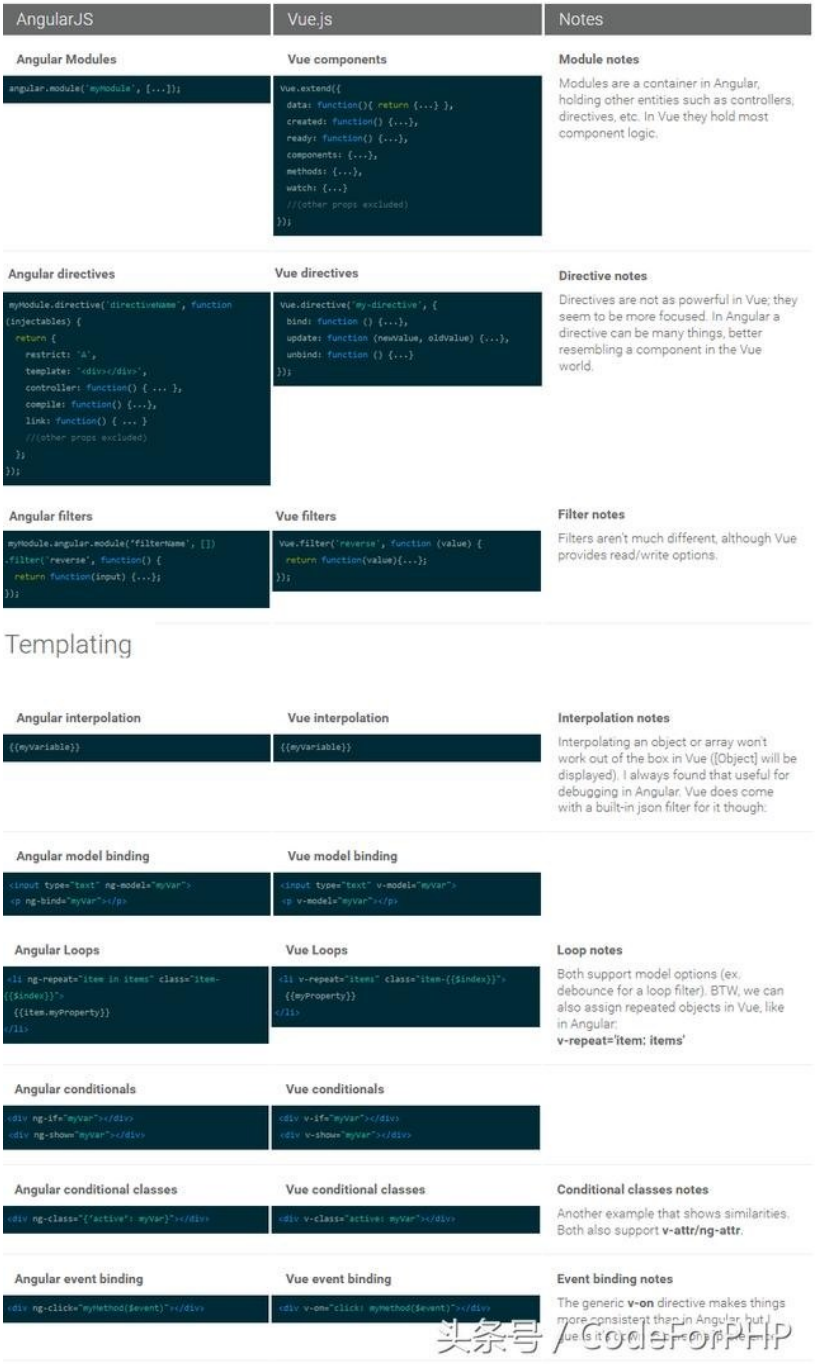
Angularjs vs Vue.js

在比较这两者时，我们首先定性一下， 如果把 Angular（主要是Angular 2 发布之后的版本）比作一头猛犸象， 而 Vue.js 则是一头已经很饿，很快就能变强大的老虎。然而，许多原因都会导致开发人员偏向 Vue。Evan You 做为 Vue 的拥有者对原因进行了正确的描述：

> Vue.js 更加灵活，（比起 Angular）更少专制，它能然你按照自己想要的方式构建应用，而非凡事非得 Angular 如此如

此。它只不过是一层界面而已，因此你可以拿它作为页面中一个轻量的功能来使用，而不是一个完整的 SPA。

现在看看下面几行代码可以让你对此有一些基础的认识（来源：fadeit.dk）。



Angularjs vs Vue.js 概要：Angularjs 拥有许多工具，而如此多的复杂语法有时也会让你感到迷惑。另外一方面，Vue.js 比起 Angular 要简单的多，甚至于要更好。如果你是在担心这个框架的未来流行趋势，我认为你不必想那么多。它是需要长时间坚持下去的，而且在未来两年，无论如何都是不会过时。

Vue.js vs Reactjs

React 和 Vue.js 拥有一些类似的功能特性，如：

1）使用了一个虚拟 DOM

2）提供了响应式的，并且可组合式的视图组件。

3）保持对核心库的专注，而像路由和全局状态管理这样的关注点则交给附带的库来处理。

这个说明了 React 和 Vue.js 在功能上是相当类似的。因此我们想从开发人员的角度，用几个简单的操作来试试每个框架，通过这样做来对这两个框架进行一下对比，看看会发生什么！

Hello World:

Vue 这样做：

```
// React.js (jsx)

var Message = React.createClass({
  getInitialState() {
    return {
      message: ''
    }
  },

  handleMessageChange(e) {
    this.setState({message: e.target.value});
  },

  render() {
    return (
      <div>
        <input type="text" onChange={this.handleMessageChange} />
        <span>{this.state.message}</span>
      </div>
    )
  }
});
```


React 这样做：

```
<!DOCTYPE html>
<html>
  <head>
    <title>React Hello World</title>
    <script src="https://fb.me/react-15.0.0.js"></script>
    <script src="https://fb.me/react-dom-15.0.0.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-core/5.8.34/browser.min.js"></script>
  </head>
  <body>
    <div id="greeting"></div>
    <script type="text/babel">
      var Greeting = React.createClass({
        render: function() {
          return (
            <p>Hello from React</p>
          )
        }
      });
      ReactDOM.render(
        <Greeting/>,
        document.getElementById('greeting')
      );
    </script>
  </body>
</html>
```


这个很简单。使用一堆 script 标签就可以让代码跑起来。从这里可以看出，利用 Vue 的功能特性的好处就是无需学习任何新的技术。

双向数据绑定

Vue 这样做：

```
// Vue.js (js)

var Message = new Vue({
  el: '#message',
  data: {
    message: ''
  }
});
```

```
<div id="message">
  <input type="text" v-model="message" />
  <span>{{message}}</span>
</div>
```


React 这样做：

```
// React.js (jsx)

var Message = React.createClass({
  getInitialState() {
    return {
      message: ''
    }
  },

  handleMessageChange(e) {
    this.setState({message: e.target.value});
  },

  render() {
    return (
      <div>
        <input type="text" onChange={this.handleMessageChange} />
        <span>{this.state.message}</span>
      </div>
    )
  }
});
```


Vue.js 中的双向数据绑定在你使用了 v-model 时就会相当的简单。而在 React 中，过程就比较漫长了。

迭代

Vue 这样做：

```
// Vue.js (js)

var Iteration = new Vue({
  el: '#array',
  data: {
    array: [1,2,3,4]
  }
});
```

```
<!-- Vue.js (html) -->

<div id="array">
  <span v-for="value in array">{{value}}</span>
</div>
```


React 这样做：

```
// React.js  (.jsx)

var Iteration = React.createClass({
  getInitialState() {
    return {
      array: [1,2,3,4]
    }
  },

  render() {
    this.state.array.map( function(value) {
      return (
        <span>{value}</span>
      )
    });
  }
});

ReactDOM.render(<Iteration />, document.getElementById('array'));
```

```
<!-- React.js (html) -->

<div id="array"></div>
```


在这里 ，Vue 的优势也是代码更少更简单。

Reactjs vs Vue.js 概要：这些示例所要表明的意思就是 Vue.js 更容易学习，而且可以快速形成生产力。它还提供了一条途径，使用新的工具和模式来简化大型代码库的管理工作。Vue.js 会随着你知识的日渐丰富而不断扩展，因此你可以利用它来学习最新的工具以及进行最佳的实践。

现在， Vue 还没有 React（由 Facebook 维护）或者 Angular 2（受到

Google 的支持）流行。不过，许多开发者都已经转向 Vue 了。Laravel (PHP框架)社区也在考虑将它作为可选用的前端框架之一。

> 总之，Vue 给 React & Angular 的弊病提供了一道良方，为你提供了一种更加简单和轻松的方法来编写代码。

声明：以上内容有小编整合而成，仅供学习参考，如有侵权，联系删除！

如果你现在在学前端或者工作是前端，欢迎加入我们大前端交流学习群：575546903【点击右边群名即可进群→】：点击链接加入群【WEB前端开发】：WEB前端开发