

前端开发技术梳理

1. 前言

web前端发展日新月异, 从09年node.js推出后, web 前端已经跳出了旧的前端三剑客的范畴(html,css,js).

现在的web前端, 新技术层出不穷.

本文主要介绍近几年前端的新工具, 新框架.

2. JS新发展

2.1. ES6

ES6 是 JavaScript 语言的下一代标准, 在 2015 年 6 月正式发布了。

它的目标, 是使得 JavaScript 语言可以用来编写复杂的大型应用程序, 成为企业级开发语言。

学习地址:

<http://es6.ruanyifeng.com/#docs/intro>

2.2. 模块化

有了模块, 我们就可以更方便地使用别人的代码, 想要什么功能, 就加载什么模块。

但是, 这样做有一个前提, 那就是大家必须以同样的方式编写模块.

目前, 通行的Javascript模块规范共有: CommonJS、AMD、CMD。

ES6已经加入了原生的模块规范,但是由于浏览器和nodejs对ES6原生模块支持还不全面,所以使用还不多.

学习地址:

知识库文章: 前端大爆炸-JS模块化

<http://113.108.110.30:18383/wcp/webdoc/view/Pubff8080815d534dc8015d5dcea22e013d.html>

3. 前端工具

3.1. gulp

在前端开发过程中, 经常会遇到一些重复性的任务, 比如合并文件、压缩代码、检查语法错误、将Sass代码转成CSS代码、同步刷新浏览器等等。

通常, 我们需要使用不同的工具, 来完成不同的任务, 既重复劳动又非常耗时.

Gulp 是一款基于 JavaScript 的构建工具, 它主要用于自动化运行 Web 开发中的各项任务。

使用Gulp, 可以仅仅只用一行命令就自动化执行上述的所有繁琐任务.

关于gulp, 请看知识库里这篇文章:

前端大爆炸-工具篇-gulp

<http://113.108.110.30:18383/wcp/webdoc/view/Pubff8080815de986ee015dff3d5b850096.html>

3.2. babel

ES6,ES7,ES8...

JS版本不停迭代, 出了很多新语法, 用了之后大幅减少代码量, 但是你敢用吗?

用了之后万一浏览器不支持怎么办?

babel就是为了解决这个问题.

Babel 是一个广泛使用的 ES6 转码器, 可以将 ES6 代码转为 ES5 代码, 从而在现有环境执行。这意味着, 你可以用 ES6 的方式编写程序, 又不用担心现有环境是否支持。

babel学习:

<https://babel.bootcss.com/> <https://babel.books.ironmaxi.com/>

3.3. webpack

自从js开始引入模块化规范后, 网页应用的写法早就变了, 很多网页其实可以看做是功能丰富的应用, 它们拥有着复杂的JavaScript代码和一大堆依赖包。

WebPack可以看做是模块打包机: 它做的事情是, 分析项目结构, 找到JavaScript模块以及其它的一些浏览器不能直接运行的拓展语言

(Scss, TypeScript等), 并将其转换和打包为合适的格式供浏览器使用。

很多情况下,WebPack比gulp更强大,能做的事情更多。

Webpack学习:

<https://webpack.docschina.org/concepts/>

4.前端框架

还在用jQuery? 写的代码跟面条一样长。

后台数据变化了,要先请求一次,拿到新数据后,再逐个的获取DOM元素,再把每个DOM元素的UI进行更新.比如下面的代码:

```
addAddress(address) {
  // state logic
  const id = String(Dat.now())
  this.state = this.state.concat({ address, id })

  // UI logic
  this.updateHelp()

  const li = document.createElement('li')
  const span = document.createElement('span')
  const del = document.createElement('a')
  span.innerText = address
  del.innerText = 'delete'
  del.setAttribute('data-delete-id', id)

  this.ul.appendChild(li)
  li.appendChild(del)
  li.appendChild(span)
  this.items[id] = li
}
```

上面的代码里出现了很多的`document.createElement` 和 `appendChild`,
想保证数据和UI同步,需要很多这样的DOM操作语句,每一句都有可能会出bug.

缺点:

1. 太累了,效率太低,需要频繁去获取每个DOM元素,再去修改它的UI.
2. 累不累还是次要,最主要的是这样手动修改容易出错,导致数据和UI不一致.

现代框架最主要的目的就是解放前端人员,不用再将精力放到操作DOM中.

因为框架最重要的功能就是保持DOM UI 和数据的同步.

4.1. Angular

Angular是这几个里推出最早的,也是曾经使用的人数最多的前端框架.

Angular1.0版使用的是脏值检查功能来实现UI和数据的同步. 性能有短板.

后来Angular2.0出来后,几乎将Angular整个都重新写了,和1.0版一点关系都没有. 一下导致很多人弃坑了.

仅仅过了一年,又推出了Angular4.0版,(直接跳过了3.0版),又是针对Angular2做了很大的改动.

改动这么大这么频繁,自己把自己玩死了.

对了,使用Angular2 或者Angular4,要先学习TypeScript(JavaScript的超集).

学习地址: <https://angular.cn/>

4.2. React

React于2013年发布由Facebook进行开发和维护. 国外用户应该是最多的,当然国内也有很多人在用. 不过去年的时候爆出来专利许可的问题,于是很多人转向了Vue.

学习React,要先学JSX语法,学习JSX语法,要先抛弃HTML,CSS,JavaScript三者分离的概念. 可以说学习曲线还是比较陡峭的.

优点当然有,使用React Native,可以开发跨平台的APP. 一套代码,处处运行.

学习地址: <https://doc.react-china.org/>

4.3. Vue

国内最火的前端框架,简单,易学,对于除了jQuery啥都没用过的开发者,很容易就能学会Vue.

社区发展很快,有啥问题百度一搜就有答案.

和React比, 最大的缺点就是开发跨平台APP这点.

不过Vue已经和阿里的WEEX进行合作, 也能使用Vue语法来开发跨平台APP了.

当然阿里的WEEX和React Native相比还是有很大不足的. (我们公司搞的EasyView解析系统是基于WEEX,但是已经做了很大改动了)

其实我们公司完全可以引入Vue的,不要再使用jQuery了.

但是Vue底层的数据绑定功能实现的核心方法是使用 `Object.defineProperty`, 就这个方法来说, IPTV肯定不能使用Vue的.

不过要说起来IPTV, 连ES6语法都不能用, 还奢求什么Vue.

所以还是等公司什么时候OTT发展壮大, 才有可能搞这些新技术.开发人员才能不这么累.

学习地址: <https://cn.vuejs.org/>