

HELLO

适用于谁？

前端能做什么

- 可视化系统 D3.JS echart.js
- 后台管理系统,运营支撑系统
- 移动端 webapp,native-app
- 桌面客户端网页
- 桌面APP

咱们能做什么？

简历

招聘网站的需求 对比几份简历

没有对比就没有伤害

1.基本信息

- 学历
- 英语水平
- 期望薪资

2.关键信息- 非专业的

- 是否离职
- 非专业工作经验 (ui,java,php,python,c,产品 , 运营 , 项目经理 , 体验顾问)
- 总而言之 : 契合度

3.自我描述- 专业 方面的描述以及工作经历的描述

- 前三条 原则。
- 亮点 原则。
- 契合度 原则。

打动人心的是认真。

做过的案例、工作经历的描述

- 你做过那些比较成功或者比较失败的案例？
- 你做过最漂亮的一个案例、效果、工作是哪个？
- 你解决的那个bug、坑是你映像最深刻的？
- 在工作中，有没有到现在还遗留着，尚未解决的问题？

专业性的问题

- GIT（版本回退，回退后前进，从版本库恢复，本地\远程切换分支，分支状态查看，冲突解决）

- H5 (1px,fixed输入框BUG , 浏览器自动填密码)

The screenshot displays the Taobao mobile app interface. At the top, there is an orange search bar with the text "寻找宝贝店铺" (Find Baby Store) and a magnifying glass icon. Below the search bar, there are three product listings: a light blue dress, a red coat, and a box of Nutrilon Pronutra 4 baby formula. A red arrow points from the red coat to the Nutrilon box. Below the product listings, there are three promotional banners: "天天特价 都给您挑好啦!" (Daily Special Price, We've picked everything for you!), "周末淘宝 买2送1" (Weekend Taobao, Buy 2 Get 1 Free), and "免费试用 手机0元领" (Free Trial, Get mobile phone for 0 yuan). At the bottom, there is a navigation bar with icons for home, search, cart, and other functions. A red circle with the text "包邮价 ¥36.0" (Free shipping price ¥36.0) is visible in the bottom left corner. A smartphone is shown in the bottom right corner, displaying the time 16:22.

- DPI , PIXEL ,

- DOM \ BOM

```
<script type="text/javascript">
  (function() {
    var scale = 1.0;
    var ratio = 1;
    if (window.devicePixelRatio >= 2) {
      scale *= 0.5;
      ratio *= 2;
    }
    var text = '<meta name="viewport" content="initial-scale=' + scale + ', m
    document.write(text);
    document.documentElement.style.fontSize = 50*ratio + "px";
  })();
</script>
```

fixed输入框bug

```
<div class='header'>....</div>  
  <div class='main'>....</div>  
  <div class='footer'>  
    <textarea></textarea>  
</div>
```

解决方法：在键盘弹出来之前按照正常的定位，使用fixed，弹出来的时候将footer这部分position:static;然后这样footer就会跑向页面的最下面了，然后再将页面主动滚动到底部。当blur的时候再把footer设置回去

浏览器自动填密码

```
<input type='text' name='username' autocomplete='off' />  
<input type='text' name='username' autocomplete='off' />
```

class/super关键字

es6 的class:

```
class Super {...};  
class Sub extends Super {...};  
var subObj = new Sub();
```

es5

```
function Super() {...};  
function Sub() {...};  
  
Sub.prototype.__proto__ = Super.prototype;  
//或  
Sub.prototype = Object.create(Super.prototype);  
Sub.prototype.constructor = Sub; //因为上一步造成constructor为Super  
//或  
Sub.prototype = Object.create(Super.prototype, {  
  constructor: subClass,  
});  
  
var subObj = new Sub();
```

es6 的 super:

```
class Super {  
  constructor() {  
    //do something  
  }  
}  
  
class Sub extends Super {  
  constructor() {  
    super();  
    //do something else  
  }  
}
```

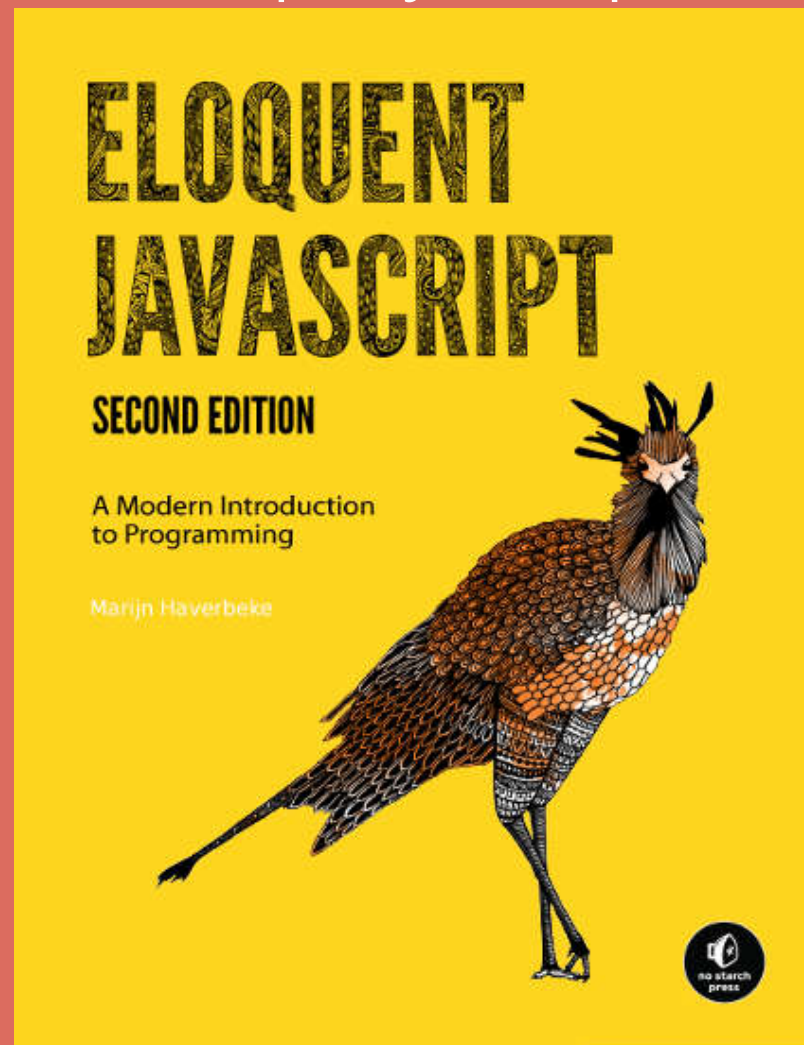
es5 :

```
function Super() {  
  //do something  
}  
  
function Sub() {  
  Sub.prototype.__proto__ = Super.prototype; //extends  
  Super.call(this); //constructor中的super  
  //babel的具体的实现要比这个复杂很多。  
  
  //do something else  
}
```

ES5兼容模拟ES6模板字符串；

学习技术相关的问题

eloquentjavascript



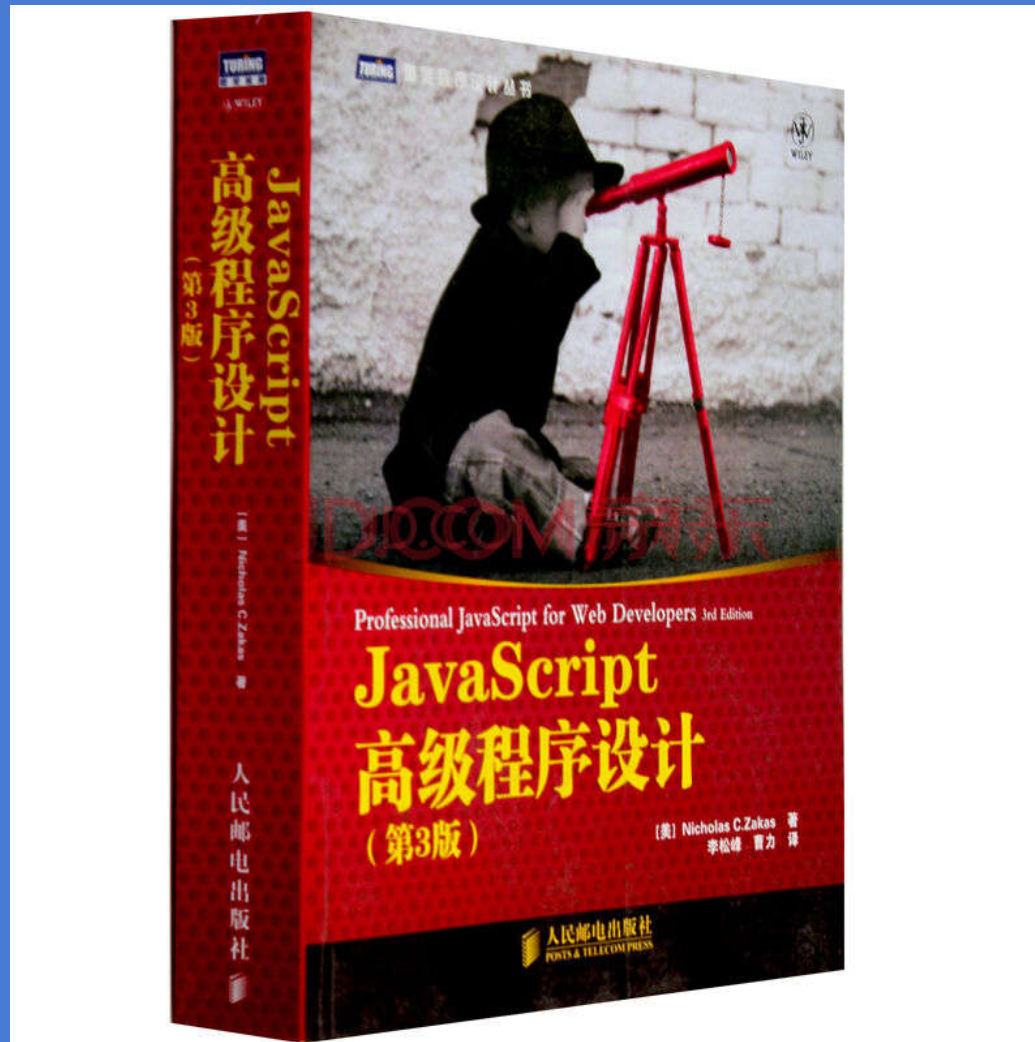
eloquentjavascript

freecodecamp



freecodecamp

javascript高级程序设计



javascript高级程序设计

项目驱动



HTTP (400 , 404 , 304缓存)

400 - bad required

404 - java映射目录错误、React路由跳转错误、<a>链接配置错误

304 - cache缓存

浏览器缓存是存在浏览器本地的文件，304是HTTP状态码，服务器用来标识这个文件没修改，不返回内容，浏览器在接收到个状态码后，会使用浏览器已缓存的文件

过程如下:

1. 客户端请求一个页面 (A) 。
2. 服务器返回页面A , 并在给A加上一个Last-Modified/ETag。
3. 客户端展现该页面 , 并将页面连同Last-Modified/ETag一起缓存。
4. 客户再次请求页面A , 并将上次请求时服务器返回的Last-Modified/ETag一起传递给服务器。
5. 服务器检查该Last-Modified或ETag , 并判断出该页面自上次客户端请求之后还未被修改 , 直接返回响应304和一个空的响应体。

读取过的文件在http header设置了expire(http 1.0) / max-age(http 1.1)，在正常浏览时，如未超时，并且浏览器也有缓存时，会直接从浏览器缓存取出，但如果你在当前页面按刷新按钮（F5）时，有的浏览器会再次向服务器发出请求，有些浏览器不会。

CSS是一次渲染全部后再接着做其它工作，重复的图片URL，在没缓存的情况下，只会发出一次请求，如果相同的图片URL已在页面或内存中，不会再次请求。

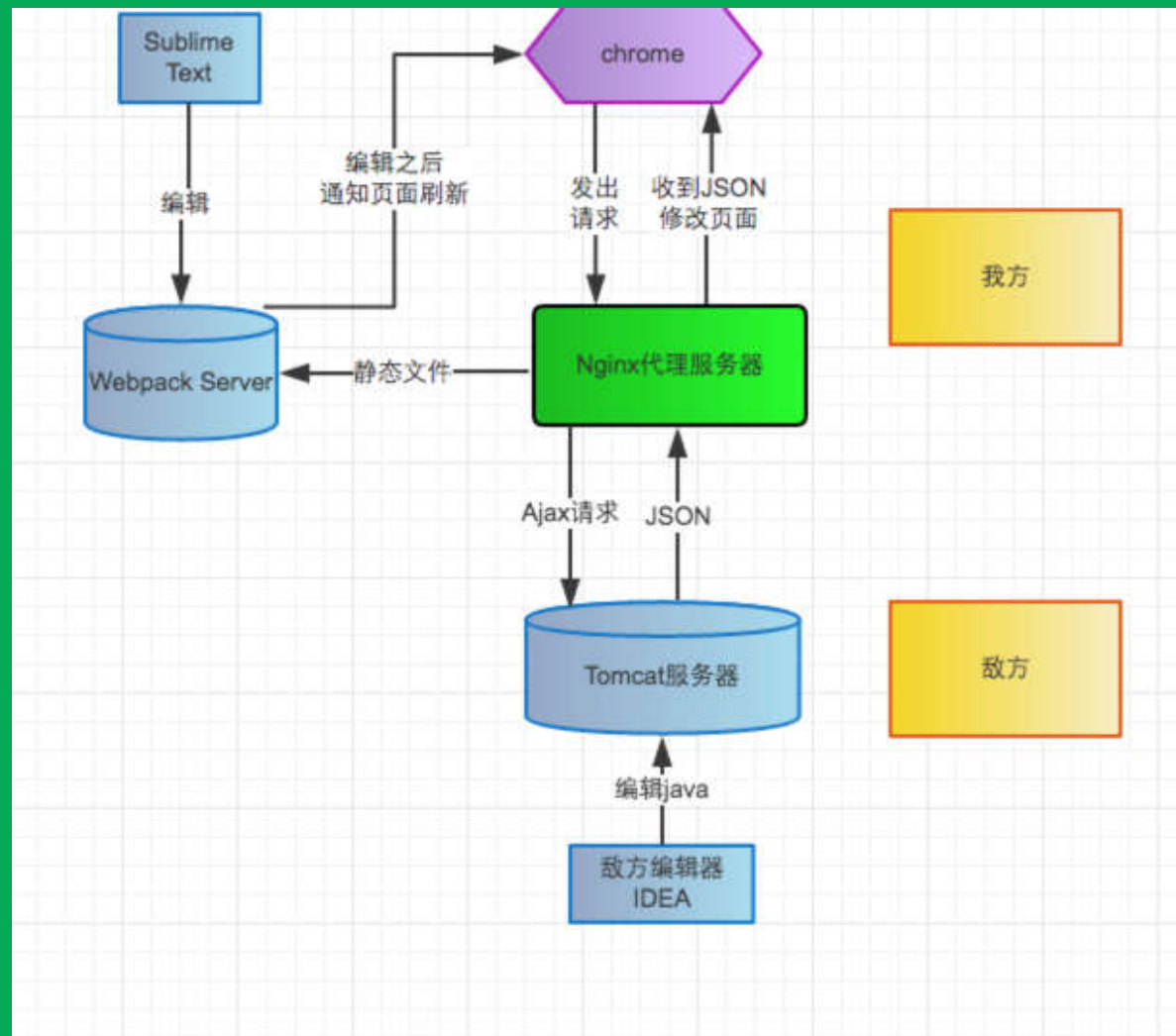
Http2

- 减少http请求，时间占用，http2可以多路复用
 - 浏览器并发限制，http2 影响
 - http头压缩

联调的问题（如何定的接口，如何讨论接口是否齐全便利）

前后端是否分离？如何分离的？为什么这样分离？

方案一：我的基于Nginx的前后端分离



```
location ~ .*\. (gif|jpg|jpeg|bmp|png|ico|txt|js|css)$ {  
    proxy_pass http://192.168.2.72:8888;  
}  
location ~ /admin/api/ {  
    proxy_pass http://192.168.2.147:8080;  
}
```

方案二：基于Node(express,koa2等)

```
// 配置服务器
devServer: {
  historyApiFallback: true,
  hot: true,
  inline: true,
  progress: true,
  contentBase: "./app", //最好写上，否则报错，难道这里是一个坑？
  port: 8888,
  proxy: { // 联调中跨域问题解决
    '/promxingzuo/index': {
      changeOrigin: true,
      target: 'http://192.168.2.74:8080',
      secure: false,
    }
  }
},
```

接口定义相关

- 统一前缀 - 例如 (/jiuye/api/)
- 请求类型 - get\post
- 参数 - 名称 (驼峰、下划线) , 类型 (integer/number,对象 , 数组 , 字符串等) , 是否必须 ,
- 错误处理 - 正常的错误码-出错的错误码

接口地址

/jiuye/api/getCityPage

提交方式

GET

请求参数

1. cityName 城市名称 [必须输入] string
2. pageNo 当前页 [非必输入] Integer
3. pageSize 每页的行数 [非必输入] Integer

响应结果（正确）

```
{
  "code": "0",
  "message": "分页获取城市信息成功！",
  "data": {
    "total_num": 2
    "list": [正确的值]
  }
}
```

响应结果（错误）

```
{
  "code": "-1",
  "message": "分页获取城市信息失败！",
  "data": null
}
```

- 优化以及优化相关（首屏优化，性能优化，合理缓存）

减少HTTP请求

- 原因：浏览器和同一个域的服务器，只能同时存在6个链接-分布式。
- 解决方案：
 - webpack等构建工具，
 - requireJS 实现页面JS的异步加载，定时间延迟加载，
 - 自己写类库功能减少类库依赖。

页面的优化

- 解决方案：
- 延迟加载、事件委托、
- 缓存（代码缓存：ajax请求，较深的变量，、浏览器缓存（cdn，图片-雪碧图，css），浏览器本地存储）