

# 目录

首页	1.1
前端	1.2
CSS	1.2.1
SCSS	1.2.2
HTML	1.2.3
VUE	1.2.4
双向数据绑定	1.2.4.1
MiniProgram	1.2.5
Monitor	1.2.6
RunJS	1.2.7
THINK	1.2.8
TypeScript	1.2.9
图标	1.2.10
其它	1.2.11
面试	1.2.12
考点	1.2.12.1
JS	1.2.12.1.1
SSR	1.2.12.1.2
FontEnd	1.2.12.1.3
招聘	1.2.12.2
考虑的公司	1.2.12.2.1
编辑器	1.3
VSCODE	1.3.1
插件	1.3.1.1
自定义	1.3.1.1.1
浏览器预览（支持任何文件）	1.3.1.1.1.1
自动换行-Markdown-AutoLinefeed	1.3.1.1.2
考试	1.4
MBA	1.4.1
准备	1.4.1.1
考试流程	1.4.1.1.1
大学	1.4.1.2
北京理工大学	1.4.1.2.1
招生简章	1.4.1.2.1.1
武汉大学	1.4.1.2.2

提前面试	1.4.1.2.2.1
2019	1.4.1.2.2.1.1
题目	1.4.1.2.2.1.1.1
2020	1.4.1.2.2.1.2
剖析	1.4.1.2.2.1.2.1
报考	1.4.1.2.2.1.2.2
题目	1.4.1.2.2.1.2.3
联考	1.4.1.2.2.2
2020	1.4.1.2.2.2.1
剖析	1.4.1.2.2.2.1.1
学习	1.5
后台	1.5.1
浏览器	1.5.2
DOCKER	1.5.3
GIT	1.5.4
GITBOOK	1.5.5
插件	1.5.5.1
HTTP	1.5.6
MAC	1.5.7
NPM	1.5.8
科技周刊	1.5.9
WIKI	1.5.10
WINDOW	1.5.11
系统	1.6
IPHONE	1.6.1
生活	1.7
公司	1.7.1
平安	1.7.1.1
腾讯	1.7.1.2
思考	1.7.2
程序员35岁之后，怎么抉择？	1.7.2.1
历史	1.8
书签	1.9
前端	1.9.1
工具	1.10
GITBOOK语法	1.10.1



欢迎大家，来到我的书世界！

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-28  
16:32:28



# 1. video标签的样式

[https://blog.csdn.net/qq\\_20757489/article/details/87879820](https://blog.csdn.net/qq_20757489/article/details/87879820)

object-fit:cover; [https://blog.csdn.net/wuqingyou\\_w/article/details/51671356](https://blog.csdn.net/wuqingyou_w/article/details/51671356)

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-09-09 16:36:32

- 1. video标签的样式

# 1. &的复合选择器

& 必须作为选择器的第一个字符，其后可以跟随后缀生成复合的选择器 参考：

<https://www.jianshu.com/p/f5b41ce5580c>

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-09-06 11:15:11

- [1. &的复合选择器](#)

HTML转义字符对照表: <http://tool.oschina.net/commons?type=2>

字符	转义字符
"	"
&	&
<	<
>	>

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-28 15:22:52







教程：

<https://www.imoooc.com/article/45665> <https://www.jianshu.com/p/6f8d74be3ff8> (入门)

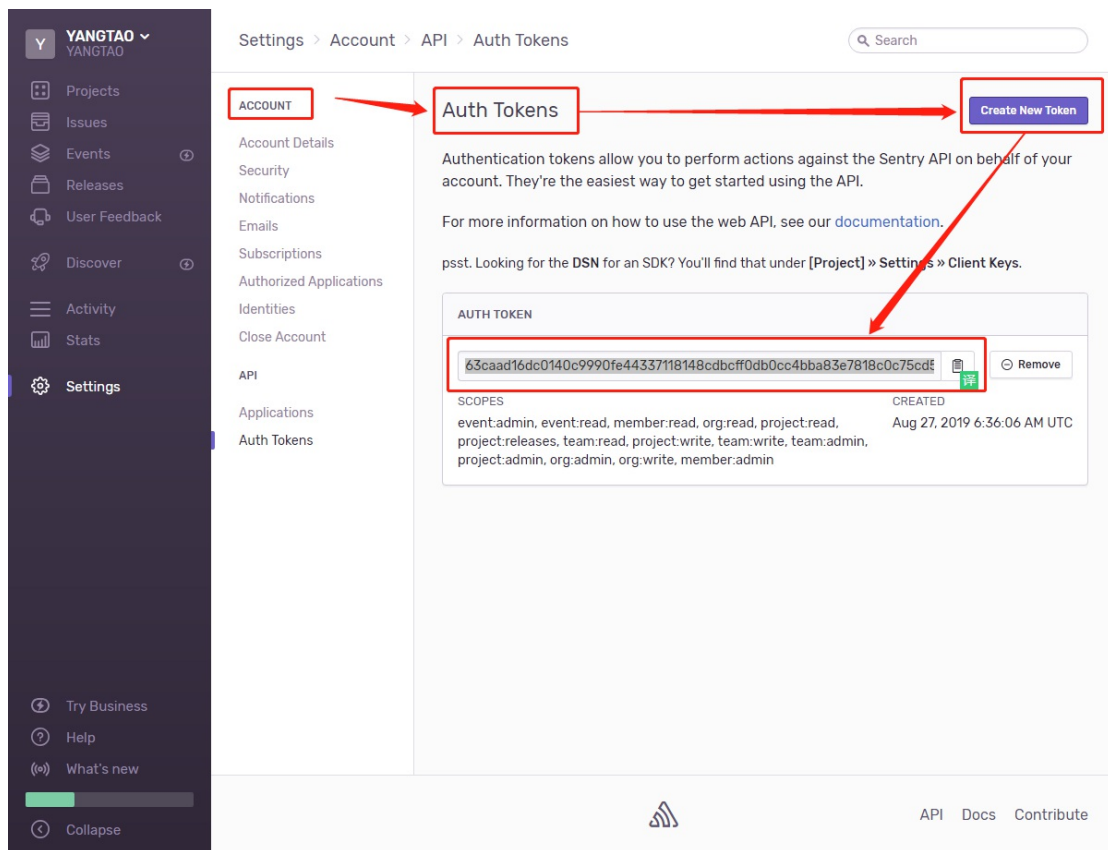
Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-30  
15:59:24

## 前端错误监控

- 原理
  - 写入前端代码 `window.onerror=function(a,b,c,d,e){(new Image).src= /m?p=${location.href}& }`
- 第三方组件——badjs

## 错误日志监控

- Doc
  - [官方git](#)
  - [官方git-前端js](#)
  - [官方api-doc](#)
- 开始
  - <https://sentry.io/onboarding/yangtao/get-started/>
- vue+sentry
  - 接入
    - [https://docs.sentry.io/platforms/javascript/vue/?\\_ga=2.131033679.1746061538.1566874846-1957175966.1566874846](https://docs.sentry.io/platforms/javascript/vue/?_ga=2.131033679.1746061538.1566874846-1957175966.1566874846)
  - 开始
    - <https://segmentfault.com/a/1190000016309667>
- sentry-后台管理系统
  - <https://sentry.io/yangtao/yangtao/getting-started/javascript/>
- source map上报sentry
  - auth token创建



<https://sentry.io/settings/account/api/auth-tokens/>

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-29 10:16:49

## 在线执行html/js/css

<http://jsrun.pro/new>

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-28  
15:22:52

## 【感想】开源协同

只有去中心化、开源化、协同化，项目才能做大。航空母舰需要全世界的人贡献每个人的想法，让航空母舰更优、最优。举例子：

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-28 15:22:52

## 【问题】 Cannot find module '../package.json'. Consider using '--resolveJsonModule' to import module with '.json' extensions(2732)

- STEP 1:

```
{
  "compilerOptions": {
    "resolveJsonModule": true,
  }
}
```

- STEP 2: restart vscode <https://github.com/microsoft/TypeScript/issues/25400>

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-28 15:22:52



阿里的免费图标: <https://www.iconfont.cn/search/index?q=%E4%B9%A6%E7%B1%8D&page=1>

Copyright (c) 2019 Fit Web Front End Tobayang. all right reserved, powered by Gitbooktime: 2019-08-29  
10:30:15

# 伪代码

不正式、不标准、接近自然语言的语言，把程序员的意思表达出来。

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-28  
16:34:22

# gitbook-最全的面试题

<https://www.yuque.com/gi4u62/gitbook-hi1tpd/183367e569e23fab6a071f9a8ce7ec41>

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-09-09 15:40:23



# escape

对字符串进行编码

- 区别  
字符串转义，两者结果不一样

## 1. Unicode编码与解码

- 编码：  
`stringObject.charCodeAt(index)`
  - 注：  
`index`，必需。表示字符串中某个位置的数字，即字符在字符串中的下标。
- 解码：

参考：

[https://www.w3school.com.cn/js/jsref\\_charCodeAt.asp](https://www.w3school.com.cn/js/jsref_charCodeAt.asp)

## 2. proxy

用法：

<https://www.jianshu.com/p/77eaaf34e732>

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-09-09 15:14:28

[escape](#)

- [1. Unicode编码与解码](#)
- [2. proxy](#)

## 伪服务端渲染


- 概述

### (伪)服务端渲染

那么既然在 HTML 加载到 js 执行期间会有时间等待,那么为什么不直接服务端渲染呢?直接返回的 HTML 就是带完整 DOM 结构的,省得还得调用 js 执行各种创建 dom 的工作,不仅如此还对 SEO 友好。

正是有这种需求 vue 和 react 都支持服务端渲染,而相关的框架Nuxt.js、Next.js也大行其道,当然对于已经采用客户端渲染的应用这个成本太高了。

于是有人想到了办法,谷歌开源了一个库Puppeteer,这个库其实是一个无头浏览器,通过这个无头浏览器我们能用代码模拟各种浏览器的操作,比如我们就可以用 node 将 html 保存为 pdf,可以在后端进行模拟点击、提交表单等操作,自然也可以模拟浏览器获取首屏的 HTML 结构。

**prerender-spa-plugin**  就是基于以上原理的插件,此插件在本地模拟浏览器环境,预先执行我们的打包文件,这样通过解析就可以获取首屏的 HTML,在正常环境中,我们就可以返回预先解析好的 HTML 了。

- 关键: router mode History

因为 `hash` 有浏览器解析。服务器根本接收不到服务器发送的任何 `hash` 信息，所以服务器端无法进行渲染。

比如客户端发送请求：

- `http://justjavac.com#about`
- `http://justjavac.com#home`
- `http://justjavac.com#tags`
- `http://justjavac.com`

上面 4 个请求，服务器都只接收到了 `http://justjavac.com`

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-09-09 15:14:28

# 1. HTTP

## 1.1. 状态码

- 1xx 临时响应（服务端响应，并等待中）
- 2xx 成功
- 3xx 重定向
- 4xx 请求错误（web服务器错误）
- 5xx 服务器错误

参考：<https://blog.csdn.net/u014346301/article/details/53995333>

## 1.2. HTTP1、HTTPS版本区别[安全点]

https是基于ssl加密的http协议

## 1.3. HTTP1、HTTP2版本区别[优化点]

- HTTP2压缩消息头，减少带宽，提高传输速率
- 允许多路复用

HTTP1.1:

浏览器客户端在同一时间，针对同一域名下的请求有一定数量限制（连接数量），超过限制会被阻塞

HTTP2:

允许同时通过单一的HTTP/2连接发送多重请求-响应信息

- 二进制分帧  
将所有的传输信息分割为更小的信息或者帧，并对他们进行二进制编码

## 1.4. 浏览器访问页面过程

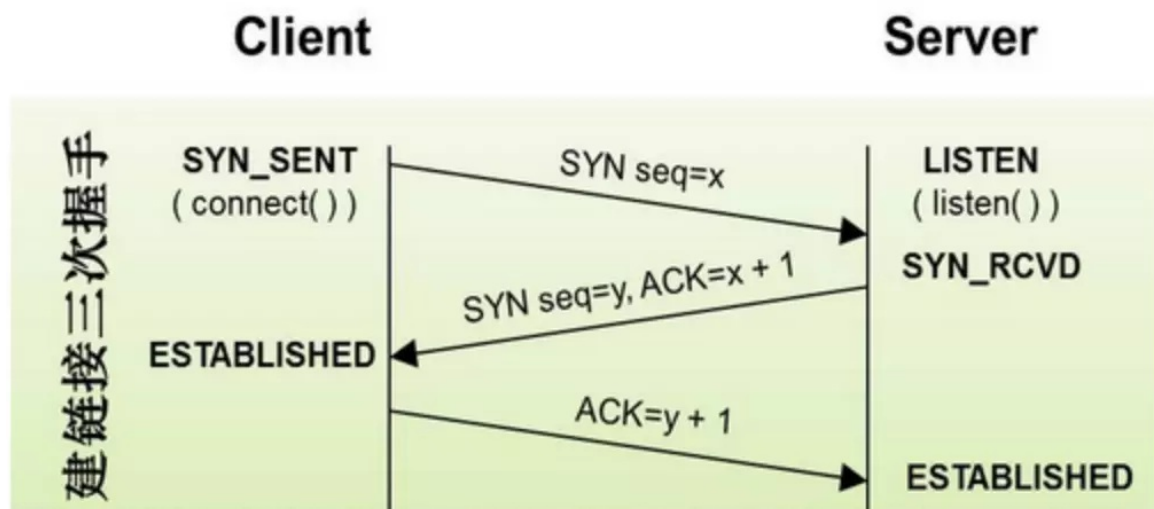
### 1.4.1. 步骤：

- DNS 解析:将域名解析成 IP 地址
- TCP 连接：TCP 三次握手
- 发送 HTTP 请求
- 服务器处理请求并返回 HTTP 报文
- 浏览器解析渲染页面
- 断开连接：TCP 四次挥手

### 1.4.2. 详解：

- TCP 三次握手





客户端发送一个带  $\text{SYN}=1$ ,  $\text{Seq}=\text{X}$  的数据包到服务器端口（第一次握手，由浏览器发起，告诉服务器我要发送请求了）

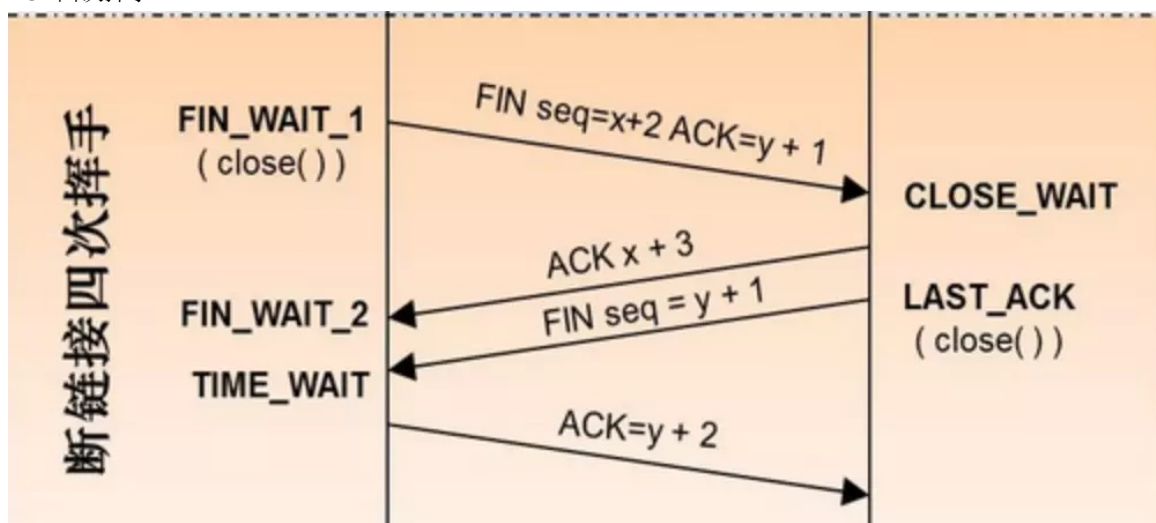
服务器发回一个带  $\text{SYN}=1$ ,  $\text{ACK}=\text{X}+1$ ,  $\text{Seq}=\text{Y}$  的响应包以示传达确认信息（第二次握手，由服务器发起，告诉浏览器我准备接受了，你赶紧发送吧）

客户端再回传一个带  $\text{ACK}=\text{Y}+1$ ,  $\text{Seq}=\text{Z}$  的数据包，代表“握手结束”（第三次握手，由浏览器发送，告诉服务器，我马上就发了，准备接受吧）

- 浏览器解析渲染页面
  - 浏览器内核拿到内容后，渲染步骤如下：
  - 解析HTML，构建DOM树
  - 解析CSS，生成CSS规则树
  - 合并DOM树和CSS规则，生成render树
  - 布局render树（Layout/reflow，布局/回流），负责各元素尺寸、位置的计算
  - 绘制render树（paint），绘制页面像素信息

注意：

- 布局/回流，早于绘制
- TCP四次挥手



参考:

<https://juejin.im/post/5c7646f26fb9a049fd108380>

## 2. 算法

运行了几次时间复杂度就为多少，如运行了n次，则时间复杂度为O(n)

### 2.1. 递归

- 特点:
  - 自调用
  - 明确结束条件（避免死循环）
  - 消耗大量内存
- 例子:

```
function abc(n){
  if(n===1)return "1" //结束条件
  return n+'*'+abc(n-1)
};
abc(10); //"10*9*8*7*6*5*4*3*2*1"
```

参考: <https://blog.csdn.net/yikewang2016/article/details/81115636>

### 2.2. 二分查找

- 特点:
  - 高效
  - 适用于有序排列的数组
- 例子:

```
function abc(arr,target,mid){
  var begin = 0;
  var end = arr.length-1;
  mid = mid || parseInt(end/2);
  if(target===arr[mid]){
    return mid;
  }else if(target>arr[mid]){
    mid++;
    return abc(arr,target,mid)
  }else if(target<arr[mid]){
    mid--;
    return abc(arr,target,mid)
  }
}
abc([1,4,6,12,44,98],44,0) //4
```

参考: [https://blog.csdn.net/qq\\_38021852/article/details/82318006](https://blog.csdn.net/qq_38021852/article/details/82318006)

### 2.3. 数组排序

sort（条件函数）

- 条件函数:
  - 若 a 小于 b，在排序后的数组中 a 应该出现在 b 之前，则返回一个小于 0 的值。若 a 等于 b，则返回 0。若 a 大于 b，则返回一个大于 0 的值。
- 例子:

```
[6,12,44,1,4,98].sort((a,b)=>a-b) //[1, 4, 6, 12, 44, 98]
```

## 2.4. 快速排序

## 2.5. 冒泡法

# 3. 前端原理

## 3.1. javascript预解释

- 读取var a
- 查看当前作用域（集合）是否有当前变量声明
  - 没有，则当前作用域（集合）声明此变量
  - 有，则忽略

# 4. 浏览器原理

## 4.1. 进程

## 4.2. V8引擎（javascript引擎）

- javascript执行
  - 读取a = 2
    - 查找当前作用域（集合）是否当前变量声明
      - 有，则赋值
      - 没有，则向上级作用域（集合）查找

## 4.3. 安全

- 参考:  
[https://blog.csdn.net/qq\\_36367995/article/details/8027921](https://blog.csdn.net/qq_36367995/article/details/8027921)

### 4.3.1. XSS攻击（跨站脚本攻击）

- 原理

利用html的一些漏洞来进行注入脚本

- 例子

- (1) 插入一个script标签<script>,或者直接进行页面弹窗
- (2) 通过你的input植入脚本，获取数据库内容

- 防御[安全点]

- 输入过滤
  - 输入侧过滤
    - 方法一

```
escape('XXXX')
```

- 方法二  
转义HTML  
采用完善的转义库
- 结论  
解决特定的XSS问题
- 纯前端渲染  
静态HTML、明确告诉浏览器设置的内容（比如：文本、样式、属性）不让浏览器被欺骗、不包含业务数据

### 4.3.2. CSRF攻击（跨站点伪造请求）

- 原理

攻击者通过一些技术手段欺骗用户的浏览器去访问一个自己曾经认证过的网站并运行一些操作（如发邮件，发消息，甚至财产操作如转账和购买商品）。

- 例子

【来源】假如一家银行用以运行转账操作的URL地址下：

```
http://www.examplebank.com/withdrawaccount=AccoutName&amount=1000&for=PayeeName
```

那么，一个恶意攻击者可以在另一个网站上放置如下代码：

```
<imgsrc="http://www.examplebank.com/withdrawaccount=Alice&amount=1000&for=Badman">
```

如果有账户名为Alice的用户访问了恶意站点，而她之前刚访问过银行不久，登录信息尚未过期，那么她就会损失1000资金。

- 防御[安全点]

- （1）判断http的header中Referer字段，是否与目标地址同域。
- （2）请求入参，增加token字段校验。token由账户+密码+时间戳+随机数组成，它具有时效性。

### 4.3.3. SQL 注入

- 原理
- 解决

## 5. 主流框架

### 5.1. vue

#### 5.1.1. 实现原理

#### 5.1.2. 数据双向绑定

- 实现原理

## 5.2. react

- 实现原理

### 5.2.1. 数据单向绑定

- 实现原理

## 5.3. router

- 实现原理

## 5.4. 小程序

### 5.4.1. 微信小程序

参考:

<https://juejin.im/entry/581db98fa0bb9f0058abffea>

### 5.4.2. 支付宝小程序

### 5.4.3. 头条小程序

### 5.4.4. 京东小程序

### 5.4.5. 百度小程序

## 6. 状态管理器

### 6.1. redux

- 实现原理

### 6.2. vuex

- 实现原理

## 7. 语言

### 7.1. 前端

#### 7.1.1. CSS

- 优先级

1. 基本公式:

内联 > ID选择器 > 类选择器 > 标签选择器

## 2. 复杂样式的计算方法：

A 的值等于 1 的前提是存在内联样式，否则  $A = 0$   
B 的值等于 ID选择器 出现的次数  
C 的值等于 类选择器 和 属性选择器 和 伪类 出现的总次数  
D 的值等于 标签选择器 和 伪元素 出现的总次数

## 3. 计算出：

{A,B,C,D}

## 4. 排序方法：

A比较，谁大谁优先级高  
A相等时，  
B比较，谁大谁优先级高  
B相等时，  
C比较，谁大谁优先级高  
C相等时，  
D比较，谁大谁优先级高  
D相等时，  
谁位置靠后谁优先级高

- 伪类、伪元素的区别

- 单位

- px

- 概念：  
精准像素

- em

- 概念：  
相对单位
    - 计算：  
基准点为父节点的字体大小，如果自身节点定义字体大小，则优先基于自身

- rem

- 概念：  
相对单位
    - 计算  
基准点为根节点html的字体大小
    - 兼容  
css3新增属性，支持chrome/firefox/IE9

## 7.1.2. JavaScript

- ES6特性

- ES7特性

- ES8特性

- ES9特性

### 7.1.3. TypeScript

## 7.2. 后端

### 7.2.1. NodeJS

- 模块

异步编程【重点】

**koa**

- 原理机制

**co**

- 原理机制

**async**

- 原理机制

- v8引擎

- 特点
- 好处

- 疑难点

性能、稳定性、内存瓶颈优化

- 描述

node.js以及V8的性能和稳定性优化，能对系统整体性能评估，解决内存瓶颈

高并发解决方案

## 8. 备注

### 8.1. 伪元素

```
(常见)
::before (:before)
::after (:after)
::first-letter (:first-letter)
::first-line (:first-line)
::placeholder
::selection
::backdrop
(不常见)
::grammar-error
::marker
::slotted()
::spelling-error
::cue (:cue)
```

## 8.2. 伪类

```
(切换元素状态)
:focus
: hover
: active
: visited
: link
(元素位置)
: first-child
: last-child
: nth-child()
(元素状态)
: checked
: disabled
: read-only
: required
: read-write
(不常见)
: nth-last-child()
: any
: default
: dir()
: empty
: enabled
: first
: first-of-type
: fullscreen
: indeterminate
: in-range
: invalid
: lang()
: last-of-type
: left
: not()
: nth-last-of-type()
: nth-of-type()
: only-child
: only-of-type
: optional
: out-of-range
: right
: root
: scope
: target
: valid
```

汇总参考: <https://www.cxymsg.com/guide/>

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-09-09 17:42:04

- 1. HTTP
  - 1.1. 状态码
  - 1.2. HTTP1、HTTPS版本区别[安全点]
  - 1.3. HTTP1、HTTP2版本区别[优化点]
  - 1.4. 浏览器访问页面过程
    - 1.4.1. 步骤 :
    - 1.4.2. 详解 :
- 2. 算法
  - 2.1. 递归
  - 2.2. 二分查找
  - 2.3. 数组排序



- 2.4. 快速排序
  - 2.5. 冒泡法
- 3. 前端原理
  - 3.1. javascript预解释
- 4. 浏览器原理
  - 4.1. 进程
  - 4.2. V8引擎（javascript引擎）
  - 4.3. 安全
    - 4.3.1. XSS攻击（跨站脚本攻击）
    - 4.3.2. CSRF攻击（跨站点伪造请求）
    - 4.3.3. SQL 注入
- 5. 主流框架
  - 5.1. vue
    - 5.1.1. 实现原理
    - 5.1.2. 数据双向绑定
  - 5.2. react
    - 5.2.1. 数据单向绑定
  - 5.3. router
  - 5.4. 小程序
    - 5.4.1. 微信小程序
    - 5.4.2. 支付宝小程序
    - 5.4.3. 头条小程序
    - 5.4.4. 京东小程序
    - 5.4.5. 百度小程序
- 6. 状态管理器
  - 6.1. redux
  - 6.2. vuex
- 7. 语言
  - 7.1. 前端
    - 7.1.1. CSS
    - 7.1.2. JavaScript
    - 7.1.3. TypeScript
  - 7.2. 后端
    - 7.2.1. NodeJS
- 8. 备注
  - 8.1. 伪元素
  - 8.2. 伪类

公司招聘信息汇总 <https://github.com/ruanyf/weekly/issues/692>

Copyright (c) 2019 Fit Web Front End Tobayang. all right reserved, powered by Gitbooktime: 2019-08-28  
12:00:09

# 广发证券（深圳）- 前/后端开发工程师

## 1. 关于我们

互联网金融技术研发组是广发证券2014年在深圳布局互联网金融技术研发最重要，也是人数最多的群组之一。我们群组核心成员大部分来自于腾讯、百度、网易有道、OPPO以及华为等知名互联网公司，有着浓厚的技术氛围和底蕴。群组负责开发广发证券易淘金APP作为公司王牌产品，在“2017券商APP风云榜”上荣获“年度最佳APP”、“最具AI范APP”以及“业内人士最任何的APP”三项大奖，在近百个券商APP中脱颖而出。除此之外，我们还积极推动技术的学习和分享，每年都会组织小组核心成员参加前端开发者大会、全球架构师峰会、中国软件开发者大会等多个业界知名的会议，同时也在诸如Qcon、全球移动技术大会等会议上担任讲师。由群组自主编撰的书籍《揭秘Angular》销量也达数万册，得到了国内多名技术大拿的认可和推荐，二期修订版也将于近期推出。广发证券2019年将持续加大在IT部门的投入，不要犹豫了，平台已经为你准备好，赶快加入我们，一展胸中抱负吧！

## 2. 前端开发工程师

### 2.1. 岗位描述

开发酷炫、具有创新交互体验的金融网站/各类APP，向投资者用户提供最佳投资、理财的Web应用服务。广发证券在金融圈内的IT技术实力数一数二，如果你想了解金融领域业务，又想实践前沿的Web技术，这里是你不容错过的平台。

### 2.2. 岗位要求

- 计算机相关专业本科或以上学历；有强烈技术好奇心，对研究各种Web前端技术具有浓厚兴趣；
- 熟悉JavaScript/AJAX/HTML5/CSS等前端开发技术；
- 熟悉各浏览器的调试工具，以及各浏览器间的差异；
- 有Angular/React/Vue等工业框架的实践经验者为佳；
- 任何客户端技术包括iOS、Android、JFC/Swing、ActionScript的认识或开发背景，我们都认为是加分；
- 至少熟悉一种后台开发技术（Node.js/PHP/Python/JSP或其他类似）；
- 了解HTTP和其他Web相关的网络协议，了解前后端联合开发的技术原理；
- 有RN开发经验者优先；

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-28 12:00:09

广发证券（深圳）- 前/后端开发工程师

- 1. 关于我们
- 2. 前端开发工程师
  - 2.1. 岗位描述
  - 2.2. 岗位要求









# 1. view in browser(支持所有文件在浏览器预览)

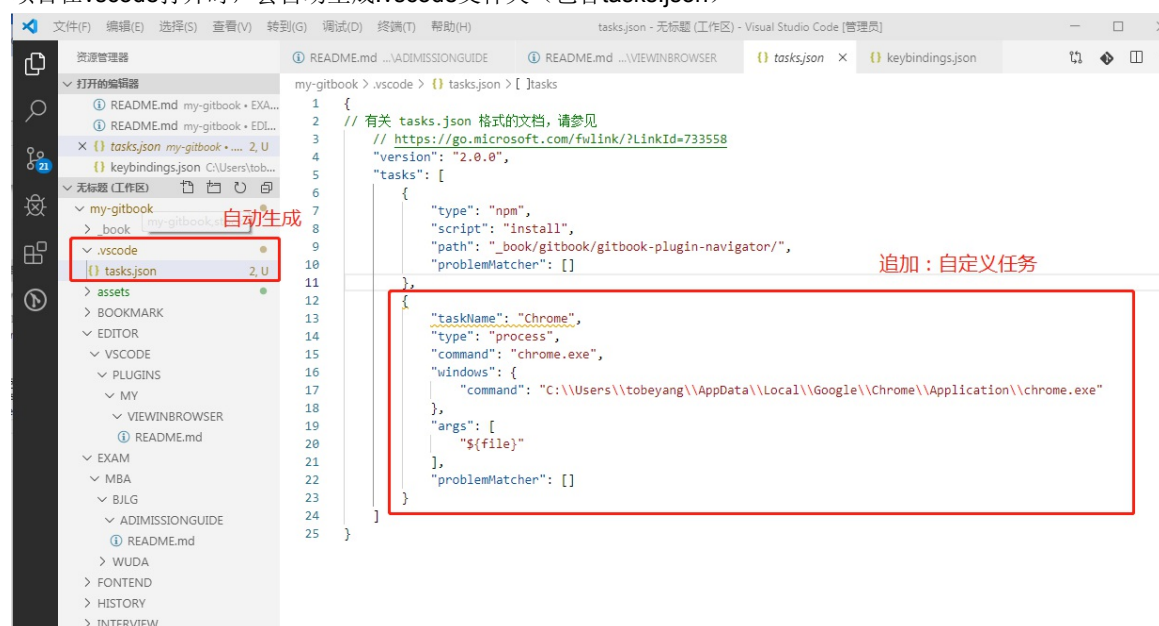
## • 1.1. step1: 设置vscode快捷键

keybinding.json:



## • 1.2. step2: 设置vscode任务（configure task）

项目在vscode打开时，会自动生成.vscode文件夹（包含tasks.json）



## • 1.3. step3: 触发自定义插件

ctrl+shift+t



- 1. [view in browser](#)(支持所有文件在浏览器预览)
  - 1.1. [step1](#): 设置vscode快捷键
  - 1.2. [step2](#): 设置vscode任务 ([configure task](#))
  - 1.3. [step3](#): 触发自定义插件

# 1. 【vscode插件】Markdown-AutoLinefeed

让markdown文件自动换行

- 等价于:  
空格\*2+回车
- 快捷键:
  - 自动换行:  
ctrl+alt+a
  - 自动空格替换:  
ctrl+alt+f

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-09-09 14:38:51

- [1. 【vscode插件】Markdown-AutoLinefeed](#)







# 1. 考试流程

## 1.1. step1: 网上报名

意指：联考/统考的报名，包含初试（笔试）、复试（面试）

- 网报时间：  
2019年10月中下旬
- 网报地址（学信网）：  
<https://yz.chsi.com.cn/user/center.jsp>
- 操作内容：
  - （1）学历查验
  - （2）支付报名费

## 1.2. step2: 现场确认

- 时间：  
2019年11月中旬
- 操作内容：
  - （1）考生携带本人有效证件、毕业证原件，进行照片采集及现场验证
  - （2）未通过学历校验考生，需携带学历认证报告等文件。
  - （3）打印准考证：2019年12月中下旬

## 1.3. step3: 初试

- 初试时间：  
2019年12月底（具体时间以教育部文件规定为准）
- 初试科目：  
综合能力（数学、逻辑、写作），英语
- 初试方式：  
统一笔试，参加工商管理硕士（MBA）全国联考

## 1.4. step3: 复试

- 复试科目：  
综合素质面试、政治笔试、英语口语、听力测试
- 复试时间：  
2020年3月中旬，3月下旬出复试结果（具体时间以我校专业学位教育联合中心通知为准）

# 2. 参考：

- 全文：<http://pdc.bit.edu.cn/MBA/zsjz/120187.htm>

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-09-09 14:30:45

- 1. 考试流程
  - 1.1. step1: 网上报名

- 1.2. step2: 现场确认
  - 1.3. step3: 初试
  - 1.4. step3: 复试
- 2. 参考:







## 1. 招生简章:

<http://pdc.bit.edu.cn/MBA/zsjz/120187.htm>

## 2. 考试流程

### 2.1. 网上报名:

2019年10月中下旬, 考生须在研招网 (<http://yz.chsi.com.cn/>) 进行网上报名 (学历查验), 网上支付报名费。

### 2.2. 现场确认:

2019年11月中旬, 考生携带本人有效证件、毕业证原件, 进行照片采集及现场验证。

未通过学历校验考生, 需携带学历认证报告等文件。

打印准考证: 2019年12月中下旬

### 2.3. 初试

初试时间: 2019年12月底 (具体时间以教育部文件规定为准)

初试科目: 综合能力 (数学、逻辑、写作), 英语

初试方式: 统一笔试, 参加工商管理硕士 (MBA) 全国联考

### 2.4. 复试

复试科目: 综合素质面试、政治笔试、英语口语、听力测试

复试时间: 2020年3月中旬, 3月下旬出复试结果 (具体时间以我校专业学位教育联合中心通知为准)

### 2.5. ※ 不接受非第一志愿考生调剂

## 3. 北京理工在职班的授课方式

类 别	全日制		非全日制	
班 型	固定课程班	选课班	在职班	国际班
授课时间	周一至周五	全周学习	周六日+平时晚上	周六日+平时晚上 (全英文授课)
课程学期	2 个学期	2 个学期	3 个学期	3 个学期
预计招生(以 9 月份研究生院公布为准)	40	160	340	50
学费	9.6 万	11.6 万	15.6 万	
学制	2 年 (最长学制 2.5 年)		2 年 (最长学制 4 年)	
档案政策	按学校相关政策执行; 如转入档案,毕业后可按学校相关政策派遣		户档不转入	

## 4. 沟通

- 电话咨询: 86-10-68945687, 68918820
- 微信平台: BIT-PDC
- MBA联合会公众号: MBABIT
- 官方QQ备考群: 662683401

## 5. 参考:

- 全文: <http://pdc.bit.edu.cn/MBA/zsjz/120187.htm>

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-09-09 17:39:47

- 1. 招生简章:
- 2. 考试流程
  - 2.1. 网上报名:
  - 2.2. 现场确认:
  - 2.3. 初试
  - 2.4. 复试
  - 2.5. ※ 不接受非第一志愿考生调剂
- 3. 北京理工在职班的授课方式
- 4. 沟通
- 5. 参考:









## 1. 华中科大

面试有风险，准备需充分！！

- 1、学习经历描述（填写学习期间的学习成绩表现、个人主要学习与锻炼的事项与成果等，100字以内）
- 2、当前工作内容及业绩描述（100字以内）
- 3、自荐信（重点列举互选的理由，300字以内）
- 4、请论述攻读MBA对你职业发展的重要性和必要性。
- 5、请举例说明，当你确定目标后，发现有多种途径或平台都可以帮助你实现目标，你是如何决策的，并阐述理由。（300字以内）。
- 6、假设你已经考上华中科技大学MBA，作为华中科技大学MBA学生，你将如何利用学校及MBA的平台资源，促进自己的学习和事业发展。

参考：

<http://www.kybang.com/?p=18120>

## 2. 浙大

- 1.请补充可以帮助我们了解您的其他信息。（400字以内）
- 2.您的业余爱好、特长。
- 3.您的入学期望。
- 4.如果您加入浙大MBA大家庭后，对这个大家庭的贡献。

## 3. 上海外国语大学

1. 请描述一下你的职业目标，为什么上外MBA项目能够帮助你达到这个目标。（请详细描述，不可少于300字）
2. 以下二道题请选择一题进行回答（请详细描述，不可少于300字）
  - a) 请描述一个让你印象最深刻的工作失败经历，你从中学到什么？
  - b) 请描述一件你所在的团队曾经成功解决的、让你印象最深刻的难题，并告诉我们你在其中所起到的关键性作用。

## 4. 上海对外经贸

- 1.自我介绍和对自己优势和劣势的认识和分析（600字以内）。2.本人报考MBA的主要目的、原因或动机。3.对未来3—5年的职业发展规划。

参考：

[各大院校的提前面试题](#)

## 5. 武大以前的提面材料：

<https://zhuanlan.zhihu.com/p/79688957>

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-09-09 14:56:58

- [1. 华中科大](#)
- [2. 浙大](#)



- 3. 上海外国语大学
- 4. 上海对外经贸
- 5. 武大以前的提面材料：



提面的分析:

<http://www.jinciwei.cn/i253411.html>

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-09-09  
14:58:59

提面报考地址：

[http://ems.whu.edu.cn/memonline/e/member/Index\\_file/?urd=b6db3485ab4d2d89aa270b4b13a1748e](http://ems.whu.edu.cn/memonline/e/member/Index_file/?urd=b6db3485ab4d2d89aa270b4b13a1748e)

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-09-09  
14:58:59

**【提前面试-2020年真题】**请论述攻读**MBA**对你职业发展的重要性和必要性。(300字以内)

<https://www.jianshu.com/p/e94a04ecd14a>

**【提前面试-2020年真题】**请举例说明，当你确定目标后，有多种途径或平台都可以帮助你实现目标，你是如何决策的并且阐述理由。

<https://www.jianshu.com/p/11dd3c0b34b8>

## 1. 粤港澳大湾区成立的意义？

### • 1.0.1. 答案：

1. 加强基础设施互联互通，形成与区域经济社会发展相适应的基础设施体系，重点共建“一中心三网”，形成辐射国内外的综合交通体系；
2. 打造全球创新高地，合作打造全球科技创新平台，构建开放型创新体系，完善创新合作体制机制，建设粤港澳大湾区创新共同体，逐步发展成为全球重要科技产业创新中心；
3. 携手构建“一带一路”开放新格局，深化与沿线国家基础设施互联互通及经贸合作，深入推进粤港澳服务贸易自由化，打造升级版；
4. 培育利益共享的产业价值链，加快向全球价值链高端迈进，打造具有国际竞争力的现代产业先导区；
5. 共建金融核心圈，推动粤港澳金融竞合有序、协同发展，培育金融合作新平台，扩大内地与港澳金融市场要素双向开放与联通，打造引领泛珠、辐射东南亚、服务于“一带一路”的金融枢纽；
6. 共建大湾区优质生活圈，以改善社会民生为重点，打造国际化教育高地，完善就业创业服务体系，促进文化繁荣发展，共建健康湾区。

### • 1.0.2. 概括：

- ① 加强基础设施互联互通
- ② 打造全球创新高地
- ③ 携手构建“一带一路”开放新格局
- ④ 培育利益共享的产业价值链
- ⑤ 共建金融核心圈
- ⑥ 共建大湾区优质生活圈

参考：

<https://wenku.baidu.com/view/f69b07a4a66e58fafab069dc5022aaea988f414c.html>

## 2. 一国两制的理解？

### • 2.0.1. 答案：

#### 1. 一国两制的含义

“一个国家，两种制度”是我国实现祖国统一的基本国策。“一个国家”是指中国的主权和领土必须完整，不容分割。中国是一个统一的国家，在国际上，代表中国的只能是中华人民共和国。“两种制度”是指在中华人民共和国这个统一主权的国家内，在祖国统一的前提下，大陆实行社会主义制度，香港、澳门和台湾实行资本主义制度。

关键词：

- ① “一个国家，两种制度”是我国实现祖国统一的基本国策
- ② “一个国家”是指中国的主权和领土必须完整，不容分割
- ③ “两种制度”是指在中华人民共和国统一主权国家内，大陆实行社会主义制度，香港、澳门和台湾实行资本主义制度

#### 2. 一国两制的基本内容

实行“一国两制”的前提和基础是一国，即中华人民共和国，国家主权由中华人民共和国统一行使，在国际上代表中国的只能是中华人民共和国及其中央政府，香港、澳门和台湾是中华人民共和国的组成部分；在这一前提下，大陆实行社会主义制度，而香港、澳门及台湾的则保留原有的资本主义制度和生活方式。

香港、澳门作为我国的特别行政区，是我国的地方行政区域，享有高度的自治权。除了外交和国防事务属于中央政府管理外，享有行政管理权、立法权、独立的司法权和终审权。台湾还可以享有更多的自治权。

关键词：

- ① 国家主权由中华人民共和国统一行使
- ② 国际上代表中国的只能是中华人民共和国及其中央政府
- ③ 香港、澳门和台湾是中华人民共和国的组成部分
- ④ 大陆实行社会主义制度，香港、澳门及台湾保留原有的资本主义制度和生活方式
- ⑤ 香港、澳门作为我国的特别行政区
- ⑥ 是我国的地方行政区域
- ⑦ 享有高度的自治权

#### 3. 一国两制的意义

“一个国家，两种制度”是我国实现祖国和平统一的基本国策。完成祖国统一大业，是民主根本利益所在，是全体中国人民的共同愿望，也是维护国家领土主权和民族尊严的重大原则问题。实行“一国两制”，有利于实现祖国的统一，有利于促进我国社会主义现代化建设，有利于港澳台稳定、繁荣和发展，有利于中华民族的伟大复兴。

关键词：

- ① 是我国实现祖国和平统一的基本国策
- ② 完成祖国统一大业，是民主根本利益所在
- ③ 是全体中国人民的共同愿望
- ④ 是维护国家领土主权和民族尊严的重大原则问题
- ⑤ 有利于实现祖国的统一
- ⑥ 有利于促进我国社会主义现代化建设
- ⑦ 有利于港澳台的稳定、繁荣和发展
- ⑧ 有利于中华民族的伟大复兴

参考：<https://wenku.baidu.com/view/2b0fb9e77e21af45b307a8dd.html>

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-09-09

15:05:18

【提前面试-2020年真题】请举例说明，当你确定目标后，有多种途径或平台都可以帮助你实现目标，你是如何决策的并且阐述理由。

- 1. 粤港澳大湾区成立的意义？
  - none
    - 1.0.1. 答案：
    - 1.0.2. 概括：
- 2. 一国两制的理解？
  - none
    - 2.0.1. 答案：







# 1. 报考截止时间

2020年9月1日

## 2. 报名前有两个阶段：

- 网上报名
- 网上确认

报名时间、考试时间、《准考证》打印，详见： 中国研究生招生信息( <http://yz.chsi.com.cn>)和武汉大学研究生院网站( <http://www.gs.whu.edu.cn>)

正式考试-笔试： 12月21日（周六）或 12月28日（周六）

录取方式： 提面+正式考试-笔试

提面申报： <http://ems.whu.edu.cn/memonline/e/member/login/>

提面考试时间-深圳班： 2019.09.06晚-2019.09.08

参考：【重要】<https://www.jianshu.com/p/4e4f2223524>

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-09-09 14:58:14

- [1. 报考截止时间](#)
- [2. 报名前有两个阶段：](#)



# CRUD

- 概念

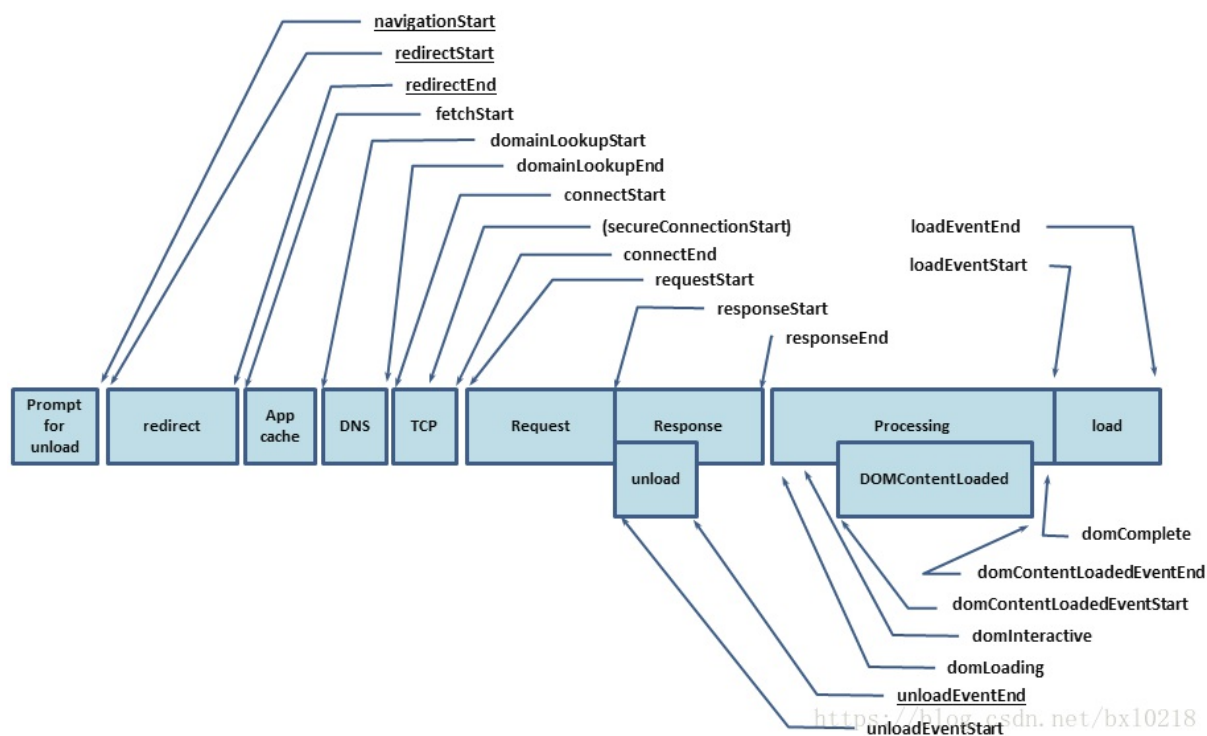
指在做计算处理时的增加(Create)、读取查询(Retrieve)、更新(Update)和删除>Delete)几个单词的首字母简写，即：CRUD=Create(增加)+Retrieve(读取查询)+Update(更新/更改)+Delete(删除)

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-28 12:00:09

## 1. chrome的控制台/开发者工具

- 英文  
chrome devtools
- 源码  
<https://github.com/ChromeDevTools>
- 教程文档（中文）  
<https://www.html.cn/doc/chrome-devtools/network-performance/resource-loading/>  
（英文）  
<https://developers.google.cn/web/tools/chrome-devtools/javascript/>
- [Tab] performance monitor（实时性能监控） DEMO（官方示例）
  - 实时的检测你的CSS性能  
“Layouts / sec”  
“Style recalcs / sec”
  - 性能差的样式写法:  
感谢 [csstriggers.com](https://csstriggers.com)，我们知道，改变 CSS 的 top 和 left 属性会触发整个像素渲染流程：绘制，布局 and 组合。如果我们将这些属性用于动画，它将每秒触发几十次/上百次操作。
  - 性能好的样式写法:  
但是如果你使用 CSS 的 transform 属性的 translateX/Y 来切换动画，你将会发现，这并不会触发绘制和布局，仅仅会触发组合这一阶段，因为这是基于 GPU 的，会将你的 CPU 使用率降低为基本为 0%。

## 2. performance



- 性能数据

```
window.performance.timing  
![Image Text](performance-timing.png)
```

- 性能指标

DNS查询耗时 = domainLookupEnd - domainLookupStart

TCP链接耗时 = connectEnd - connectStart

request请求耗时 = responseEnd - responseStart

解析dom树耗时 = domComplete - domInteractive

白屏时间 = domloadng - fetchStart

domready时间 = domContentLoadedEventEnd - fetchStart

onload时间 = loadEventEnd - fetchStart

- 总结

1.对于网页的测速上报需求，可以通过对Performance.timing对象的属性排列组合，计算出业务需要的测速数据。

2.对于网页的性能监测需求，可以通过对Performance.memory对象进行分析，得出内存使用情况等数据。

3.对于页面的其他业务监测需求，可以通过Performance提供的其他方法进行灵活使用，计算出业务所需数据

- 二、下面标注几个比较重要的统计阶段

- 页面加载完成的时间，这几乎代表了用户等待页面可用的时间

loadPage = loadEventEnd - navigationStart

- 解析 DOM 树结构的时间，反省下你的 DOM 树嵌套是不是太多了！

domReady = domComplete - responseEnd;

- 重定向的时间，拒绝重定向！比如，<http://fudao.qq.com/> 就不该写成 <http://fudao.qq.com>

redirect = redirectEnd - redirectStart;

- DNS 查询时间,页面内是不是使用了太多不同的域名导致域名查询的时间太长？

lookupDomain = domainLookupEnd - domainLookupStart;

详见 HTML5 prefetch

- 读取页面第一个字节的时间，可以理解为用户拿到你的资源占用的时间，  
加异地机房了么，加CDN 处理了么？加带宽了么？加 CPU 运算速度了么？

TTFB 即 Time To First Byte 的意思

ttfb = responseStart - navigationStart;

- 内容加载完成的时间，页面内容经过 gzip 压缩了么，静态资源 css、js 等压缩

request = responseEnd - requestStart;

- 执行 onload 回调函数的时间 是否太多不必要的操作都放到 onload 回调函数里执行了，考虑过延迟加载、按需加载的策略么？

loadEvent = loadEventEnd - loadEventStart;

- DNS 缓存时间

appcache = domainLookupStart - fetchStart;

- 卸载页面的时间

times.unloadEvent = unloadEventEnd - unloadEventStart;

- TCP 建立连接完成握手的时间

times.connect = connectEnd - connectStart;

- 加载时间

```
// 计算加载时间  
function getPerformanceTiming () {  
    var performance = window.performance;  
  
    if (!performance) {  
        // 当前浏览器不支持
```

```

    console.log('你的浏览器不支持 performance 接口');
    return;
}

var t = performance.timing;
var times = {};

//【重要】页面加载完成的时间
//【原因】这几乎代表了用户等待页面可用的时间
times.loadPage = t.loadEventEnd - t.navigationStart;

//【重要】解析 DOM 树结构的时间
//【原因】反省下你的 DOM 树嵌套是不是太多了！
times.domReady = t.domComplete - t.responseEnd;

//【重要】重定向的时间
//【原因】拒绝重定向！比如，http://example.com/ 就不该写成 http://example.com
times.redirect = t.redirectEnd - t.redirectStart;

//【重要】DNS 查询时间
//【原因】DNS 预加载做了么？页面内是不是使用了太多不同的域名导致域名查询的时间太长？
// 可使用 HTML5 Prefetch 预查询 DNS，见：[HTML5 prefetch](http://segmentfault.com/a/1190000000633364)

times.lookupDomain = t.domainLookupEnd - t.domainLookupStart;

//【重要】读取页面第一个字节的时间
//【原因】这可以理解为用户拿到你的资源占用的时间，加异地机房的么，加CDN 处理了么？加带宽了么？加 CPU 运算速度了么？
// TTFB 即 Time To First Byte 的意思
// 维基百科：https://en.wikipedia.org/wiki/Time_To_First_Byte
times.ttfb = t.responseStart - t.navigationStart;

//【重要】内容加载完成的时间
//【原因】页面内容经过 gzip 压缩了么，静态资源 css/js 等压缩了么？
times.request = t.responseEnd - t.requestStart;

//【重要】执行 onload 回调函数的时间
//【原因】是否太多不必要的操作都放到 onload 回调函数里执行了，考虑过延迟加载、按需加载的策略么？
times.loadEvent = t.loadEventEnd - t.loadEventStart;

// DNS 缓存时间
times.appcache = t.domainLookupStart - t.fetchStart;

// 卸载页面的时间
times.unloadEvent = t.unloadEventEnd - t.unloadEventStart;

// TCP 建立连接完成握手的时间
times.connect = t.connectEnd - t.connectStart;

return times;
}

```

参考：

<http://www.alloyteam.com/2015/09/explore-performance/>

参考：

<https://blog.csdn.net/bxl0218/article/details/80813699>

Copyright (c) 2019 Fit Web Front End Tobayang. all right reserved, powered by Gitbooktime: 2019-08-29 10:47:02

- [1. chrome的控制台/开发者工具](#)
- [2. performance](#)





## 建立容易迁移环境

docker

## 搭建本地gitlab网站

docker安装配置gitlab

## mac搭建docker的教程

重要教程:

- <https://www.runoob.com/docker/macos-docker-install.html>
- <https://blog.51cto.com/13673090/2092462>

## docker官网

<https://hub.docker.com/>

## 使用步骤（成功）

- STEP 1 安装docker

```
brew cask install docker
```

- STEP 2 安装目录:

```
/Applications/Docker.app
```

- STEP 3 在docker.app登录账号

docker.app登录后，docker指令才会生效

- STEP 4 操作指令

docker ps 查看正在运行的容器

docker stop 停止正在运行的容器

docker start 启动容器

docker ps -a 查看终止状态的容器

docker rm -f webserver 命令来移除正在运行的容器

docker list 列出本地镜像

## docker.app之preferences全解:

<https://blog.csdn.net/w275840140/article/details/80158972>

## docker容器的创建与操作

<https://www.linuxidc.com/Linux/2017-12/149575.htm>

### 【问题】Unable to find image 'hello-world'

- 分析:  
这个时候不要害怕, docker只是还没有下载这个镜像, 所以我们重复命令, 这时我们就会发现, Docker从 Docker Hub上获取到最新的Hello World镜像, 下载到了本地
- 参考: [https://blog.csdn.net/CaoMei\\_HuaCha/article/details/87544109](https://blog.csdn.net/CaoMei_HuaCha/article/details/87544109)
- 解答:
- 设置docker的镜像加速:
- 阿里 (成功)  
<https://alzgoonw.mirror.aliyuncs.com>
- 网易  
<http://hub-mirror.c.163.com>
- 重启电脑, 让它生效
- docker run hello-world

## 后台运行

关键词:

```
-d
```

例子:

```
docker run -d ubuntu:15.10
```

## 查看

```
docker ps
```

## 关闭运行中的进程

```
docker stop xxx
```

备注:

**xxx**, 指容器启动后生成的进程ID字符串。一定要有开头, 可以是片段

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-28  
12:00:09

## 更换git账户的密码

- 没用做法：让系统重新弹框给用户重新填写用户名和密码，进行提交

```
git config --system --unset credential.helper
```

window系统（有用）

<https://my.oschina.net/Jackcrj/blog/2991933>

注意：

修改sourcetree,是无用的。

## 【问题】gitbook在window下，liveload失败的解决方案

修改，gitbook serve就崩了的解决：gitbook serve执行后，删除\_book

<https://cloud.tencent.com/developer/article/1426724>

复制粘贴，gitbook serve就崩了的解决：修改以下文件：

C:\Users\tobeyang.gitbook\versions\3.2.2\lib\output\website\copyPluginAssets.js 文件中的 112 行

将 confirm: true 改为 confirm: false

参考：

<https://www.cnblogs.com/wenhui92/p/9482629.html>

## github的fork

拷贝源目标git仓库到自己Git仓库

## 回滚到某个版本

前提条件：已远程提交

操作：重置到当前分支

结论：虽然可以回到之前的版本，但一更新就又回到未重置前的状态。（无用功）

前提条件：已远程提交

操作：回滚提交

结论：回滚的是要回滚当前版本提交的内容，不能达到目的

参考：

<https://www.cnblogs.com/hopeway-shaon/p/5740280.html>

命令回滚：

git reflog

git reset --hard commit\_id git push origin HEAD --force

前提条件：当远端设置了分支保护，则首先要将其取消，再强推；备注：

HEAD就代表当前

强推之后，之前的版本被覆盖了，无法再查看到。

参考：

<https://jacheng.top/2017/02/16/Git-%E4%BB%A3%E7%A0%81%E5%9B%9E%E6%BB%9A/>

## 修改远程仓库地址

- 例：

远程仓库迁移，本地提交地址需要更新

操作：

```
git remote set-url origin [url]
```

参考：

<https://ddnode.com/2015/04/14/git-modify-remote-responsity-url.html>

## 1. git命名规则

Project 名称 全部小写，并且以横杠（-）作为连字符。命名规则：[产品名称]-[项目类型]-(自定义名称)-[日期]

<http://res.nie.netease.com/comm/doc/tutorial/git-naming.html>

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-09-05 16:40:10

[修改远程仓库地址](#)

- [1. git命名规则](#)

## 1. gitbook块用法: />

标签需正常闭合。

## gitbook资料:

<https://www.imoooc.com/article/22889>

## gitbook用例:

<https://docs.gitbook.com/account> <https://www.jianshu.com/p/421cc442f06c>

## 阮一峰的EcmaScript6文档

也是gitbook写成的。

- 网址  
<http://es6.ruanyifeng.com/#docs/set-map#Map>
- github  
<https://github.com/ruanyf/es6tutorial>

## 【问题】Error: ENOENT: no such file or directory, open '\_book/index.html'

- 描述:  
``` warn: "options" property is deprecated, use config.get(key) instead warn: "options.generator" property is deprecated, use "output.name" instead use simple mode: true

Error: ENOENT: no such file or directory, open '\_book/index.html'

```
+ 解决方法:
gitbook update
tnpm run book-install

# gitbook写得好的作品
https://docs.cocos.com/creator/manual/zh/getting-started/
https://mta.qq.com/docs/?q=

# 【问题】Error loading version latest: Error: Cannot find module 'internal/util/types'
原因:
这个问题原因在于graceful-fs引入了node模块internal/util/types, 此模块用处是给vm引入内置js文件, 具体作用不可知也不想去考究了,
重点在于此法已被废除, 只有低版本nodejs和npm可用, 但是强行降低node版本有点削足适履的意思, 这种天怒人怨的低级bug, graceful-fs开发团队应该早就修复了才对, 为什么还会出现报错, 去gitbook的git仓库看了一下版本更迭, gitbook-cli默认的版本是2.6.9, 最新的版本3.2.2, 推测是旧版本没有使用最新版本的graceful-fs导致的问题, 果然。。更新之后即可正常运行
参考:
https://www.zhihu.com/question/270284604/answer/766617038
解决:
gitbook fetch 3.2.2

# 安装gitbook
```

```
tnpm install gitbook-cli -g
```

# 查看gitbook版本  
gitbook -V

## Gitbook个人不错的使用教程  
[https://skyao.gitbooks.io/learning-gitbook/content/creation/multiple\\_lang.html](https://skyao.gitbooks.io/learning-gitbook/content/creation/multiple_lang.html)

## gitbook Error: EPERM: operation not permitted, open  
原因:  
不是这台机器创建的文件, 由git工具拉取下来的新文件。  
在windows中, 创建的用户不一样, 权限不一, 导致部分文件没权限, 最终热加载失败  
解决办法:  
赋予当前用户的读写权限  
使用git-bash

chown -R 用户名 赋权文件夹/文件

```
## 【问题】Error: ENOENT: no such file or directory, stat 'E:\kaifa\My\Git\MyGitBook\_book\gitbook\gitbook-plugin-fontsettings\fontsettings.js'
```

+ 方法一  
切换低版本的就行了, 并且用Node V6  
在book.js中加入(使用3.0.0以下的版本)  
推荐是: 2.6.7~2.6.4

gitbook fetch 2.6.7

+ 方法二  
在book目录下, 如下代码新建book.json

```
{ "plugins": [ "fontsettings", "sharing", "lunr", "search", "highlight", "livereload" ] }
```

保存文件后, 运行gitbook install安装插件依赖  
重新执行gitbook build或gitbook serve  
+ 方法二

gitbook serve .

`` 参考: <https://segmentfault.com/q/1010000009569245>

Copyright (c) 2019 Fit Web Front End Tobayang. all right reserved, powered by Gitbooktime: 2019-09-09 17:34:55

【问题】Error: ENOENT: no such file or directory, open '\_book/index.html'

- 1. gitbook块用法: />

## 1. gitbook插件

logo: "insert-logo"

高级搜索（支持中文）: "search-pro"

代码复制按钮: "copy-code-button"

导航目录折叠: "expandable-chapters"

回到顶部: "back-to-top-button"

在右上角添加github图标: "github"

分享当前页面: "sharing-plus"

页面添加页脚: "tbfed-pagefooter"

打赏: "donate"

自动生成目录树: "atoc"

根据文件夹自动生成SUMMARY.md文件: "summary"

菜单可以折叠: "toggle-chapters"

主题插件, 修改标题和表格颜色: "theme-comscore"

侧边栏宽度可调节: "splitter" 参考:

<http://gitbook.zhangjikai.com/plugins.html>

## 2. 【插件】atoc

- 用法:

book.json

```
"plugins": [
  "atoc"
],
"pluginsConfig": {
  "atoc": {
    "addClass": true,
    "className": "atoc"
  }
}
```

需要生成目录树的文件:

```
<!-- toc -->
所有内容
<!-- endtoc -->
```

## 3. 【插件】theme-comscore

改变文章里标题的字体颜色

## 4. 【插件】theme-default

给文章里标题自动添加序号

具体如:

```
"theme-default": {
```



```
"showLevel": true  
},
```

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-09-09 17:09:35

- 1. [gitbook插件](#)
- 2. [【插件】atoc](#)
- 3. [【插件】theme-comscore](#)
- 4. [【插件】theme-default](#)

## 前端缓存策略

大家可以想象我们使用 **a.js** 的场景，我们一般都是输入网址，访问一个 **html** 文件，**html**文件中会引入 **js**、**css**、图片等资源。所以呢，我们在**html**上做些手脚。我们不让 **html** 文件缓存，每次访问 **html** 都去请求服务器。所以浏览器每次都能拿到最新的**html**资源。**a.js** 内容更新的时候，我们修改一下 **html** 中 **a.js** 的版本号。

第一次访问 **html**

```
<script src="http://test.com/a.js?version=0.0.1"></script>
```

浏览器下载**0.0.1**版本的**a.js**文件。浏览器再次访问 **html**，发现还是**0.0.1**版本的**a.js**文件，则使用本地缓存。某一天 **a.js**变了，我们的**html**文件也相应变化如下：

```
<script src="http://test.com/a.js?version=0.0.2"></script>
```

浏览器再次访问**html**，发现【**test.com/a.js?versio... a.js**。如此往复。所以，通过设置**html**不缓存，**html**引用资源内容变化则改变资源路径的方式，就解决了无法及时得知资源更新的问题。当然除了以版本号来区分，也可以以 **MD5hash** 值来区分。如

```
<script src="http://test.com/a.【hash值】.js"></script>
```

使用**webpack**打包的话，借助插件可以很方便的处理。

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-28 12:00:09

## 【问题】在**Mac**里，**ntf**格式的移动硬盘不能正常写入

[Mounty](#)（免费软件），[下载地址](#)

[ntfs for Mac](#)（付费软件）

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-28  
12:00:09

## yarn的安装

`tnpm i yarn -g`

## yarn的使用

（错误）

`yarn i` （正确）

`yarn install`

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-28  
12:00:09

## 0.1. 阮一峰——科技爱好者周刊

## 0.2. alloyteam——技术日报

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-28 12:00:09

- none
  - 0.1. 阮一峰——科技爱好者周刊
  - 0.2. alloyteam——技术日报

# wiki

- 概述  
tapd系统里模块
- 作用： 文档

参考：

tapd的wiki模块

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-28  
15:40:40

<https://my.oschina.net/Jackcrj/blog/2991933>

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-28  
12:00:09

## 小马BI

- 概念  
智慧数据平台（驱动业务决策）
- 缺点
- 报表简单
- 可选择的报表类型少

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-28  
12:00:09



## 锁屏自动断开**wifi**

解决:

连接的**wifi**设置: 自动连接

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-28  
12:00:09





中国平安保险(集团)股份

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-29  
12:20:20

企业名称： 深圳市腾讯计算机系统有限公司 企业电话： 0755-86013388 统一社会信用代码：  
91440300708461136T

组织机构代码： 708461136 注册号： 440301103448669

经营状态： 登记 公司类型： 有限责任公司

成立日期： 1998-11-11 法定代表人： 马化腾

营业期限： 1998-11-11 - 2018-11-11 注册资本： 6500 万人民币

发照日期： 2016-01-19 登记机关： 深圳市市场监督管理局南山局 企业地址： 深圳市南山区高新区高新南一路飞亚达大厦5-10楼 经营范围： 计算机软、硬件的设计、技术开发、销售（不含专营、专控、专卖商品及限制项目）；数据库及计算机网络服务；国内商业、物资供销业（不含专营、专控、专卖商品）；第二类增值电信业务中的信息服务业务（不含固定网电话信息服务和互联网信息服务，并按许可证B2-20090028号文办）；信息服务业务（仅限互联网信息服务业务，并按许可证粤B2-20090059号文办）；从事广告业务（法律、行政法规规定应进行广告经营审批等级的，另行办理审批登记后方可经营）；网络游戏出版运营（凭有效的新出网证（粤）字010号互联网出版许可证经营）；货物及技术进出口。 参考： <http://www.xiechuangw.com/ask/question/9768.html>

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-29 12:20:04



**【思考】** 程序员**35**岁后做什么？

今年刚好40岁，算是非常幸运了，目前在一家世界100强的外企（欧美）做到技术专家的位置，工作强度还算合理，至少在保证健康的情况下，还有时间照顾家庭。

我在这家公司已经服务了10年，做过3年软件开发工程师，然后又做了5年的开发经理，现在又主动申请转到技术岗（技术专家）。坦率的讲，无论作管理还是做技术，过了35岁都一样焦虑，因为一旦失业，在就业市场上都一样没有竞争力，甚至管理岗更没有竞争力，毕竟一则管理岗本身就少，二则普通的中级管理岗更倾向于公司内部提拔，至少在目前的公司，10年了我都没见过一个空降的经理。

我所在的事业部目前盈利还不错，自我评估了下，3年内大概率不会失业，但更长远的未来就很难预料了，但唯一可以确定的下岗是早晚的事，一则这些年外企在国内一直处于颓势，二则放眼世界也没见过基业长青的科技公司。

人过了35，且不是说社会上赤裸裸的年龄歧视，就拿当前中国的劳动氛围来说，动辄就是996的工作强度，就算这些公司能够给你offer，以35岁之后每况愈下的身体状况，有几个人能吃得消，只怕是有命挣钱无命消受。而那些以为混到管理层就安全了的想法，也真够naive的了：底下的兄弟都996了，你不好意思让兄弟们在前面拼命，你一个人回家老婆孩子热炕头？你要敢这么干，只怕老板不让你分分钟滚蛋，底下兄弟的吐沫星子也早把你淹死了！

所以说“中年危机”的局怎么破？从个人来说很难，因为这本质上是一个社会问题。我们所能做的就是等待，等待在或远或近的将社会的劳动环境会变得更健康，更合法。等待将来整个社会都消灭了恶性加班，消灭了非法解雇，消灭了年龄歧视，消灭了...，只有那一天才是“中年危机”真正破局的时候。

尽管我相信社会的未来一定是美好的，“中年危机”一定有破局的一天，但活在当下，我们依然可以做一些事情，来最大限度的降低危机对我们的冲击：

1.35岁之前尽量进入一个温和的公司，避免盲目的跳槽。

正如我前面所说的，35岁之后身体机能急剧下降，高强度的工作（例如996）对健康的伤害是巨大的。在这种强度的工作压力下，即使公司不裁你，大概率的自己也得离开。因此，未雨绸缪，35岁之前就要可以考虑跳槽到那些工作环境健康的企业，例如外企，国企，事业单位等。而且要尽早，不要说过了35岁再行动，毕竟所有的企业都不是慈善家，都要盈利，自然都钟爱年轻人。

过了30岁，就要收敛下性格，不要一言不和就跳槽。我们都知道人年老了要靠养老金生活，其实年轻的时候，工龄也是一个巨大的财富。如果你能在一个公司服务10年，即使公司有一天要裁你，按照劳动法n+1的赔偿标准，在北京就会有40万左右的赔偿，这笔钱足够一个家庭3年的口粮了，3年的时间也能够让一个中年男人重整旗鼓再出发了。

职场上我见过不少35+的暴躁男一言不合就跳槽的，但大部分的境况越跳越不如前：新公司、新人、新环境，动辄6个月的试用期，于是每日战战兢兢，日夜加班，可谓疲惫不堪。甚至你一个中年油腻男，连中午吃饭年轻人都不愿意带你。在老公司，也许有小人伤害你，算计你。但谁能保证新公司就没有小人算计你呢？毕竟有人的地方就有江湖！

## 2.转行



我在这里说的转行，并不是说今天你是一个做软件开发的，明天跑去开出租车，而是指在当前的垂直行业内，转换一个方向，这样既能发挥多年累积的行业优势，又能平滑的过渡到新职业，一个可以想到的例子就是“软件开发转it讲师”。

目前社会上大有人人转码农的势头，看看遍地开花的it培训机构就知道当前的培训市场有多火，it培训师的需求自然也就水涨船高。而且和软件开发不一样，软件开发基本上好的就业岗位都在一线城市，但it培训机构即使在二三线城市也是遍地开花，所以说对很多年轻时在大城市奋斗，被高房价和户口所逼，中年想退守家乡的人来说，it讲师也不失为一个理想的选择。

和软件开发更青睐年轻人不一样，it培训机构更青睐那些有名企背景，有丰富一线开发经验的资深软件工程师，道理很明显：换做你要参加it培训，你希望教你的是一个毛头小伙？还是一个沉稳的中年男子？道理不言而喻嘛！再看看各个培训机构的宣传栏，都是把那些有名企背景，有丰富一线开发经验的讲师放在最显眼的位置。所以说，从职业生涯可持续性来说，it讲师强于软件开发。

### 3.主动降低生活的期望。

“中年危机”是个病，但病因是什么？你问问那些挣扎在社会底层的人（例如建筑工人，售货员，保安），问问他们有没有中年危机，估计人都会笑你矫情，生活对他们从来就没有轻松过。

说的不好听点，所谓的“中年危机”都是小布尔乔亚们臆想出来的。年轻的时候仗着身体好，加上被各种消费主义洗脑，出卖青春，挣些血汗钱，住着在大城市按揭的小房子，飘飘间恍然觉得自己已经实现了阶级跨越，成为了所谓的中产阶级。其实这一切都是自己的幻觉罢了。如果你明白了这一点，就知道你原来怎么活，现在依然可以那么活。

原来你给孩子报辅导班，现在失业了，上不起辅导班了，那就不上了呗，或者自己辅导（反正有的是时间），谁也没说只有上辅导班的孩子才会出息（教育要是这么简单那就太好了）。

父母病了，原来吃的是进口药，现在吃不起了，那就吃基本医保药就好了。看看你的周围，也没见隔壁的有钱人比咱普通老百姓健康多少，快乐多少。

所以说，在这世界上，如果解决了基本衣食住行，再往上的喜怒哀乐真的不是和钱成线性关系，更多在于对生活的态度！

参考：

[知乎：找工作时单位普遍要求 35 岁以下，那 35 岁以上的人都干嘛去了？](#)

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-28 12:00:09

# 通用电气公司(美国)

- 来源

爱迪生创建的爱迪生电力公司

- 历史

爱迪生发明直流电发电系统，但特斯拉发明了传输更远的交流电发电系统，爱迪生电力公司的股东们站在爱迪生的对立面：交流电发电，把爱迪生踢出决策圈。

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-08-28 15:22:52



# 美团技术团队

温馨提醒：文件过大，约350M，建议通过WIFI下载，或将地址复制到PC端，使用浏览器下载。

【前端系列】：<http://dpurl.cn/DWVhPpS>

【后台系列】：<http://dpurl.cn/obtIVn>

【系统系列】：<http://dpurl.cn/zTDWq8Q>

【算法系列】：<http://dpurl.cn/ODj5qAB>

【运维系列】：<http://dpurl.cn/6zjgHFj>

【测试系列】：<http://dpurl.cn/wfKQqy>

【工程师成长系列】：<http://dpurl.cn/p5sPoKT>

【2018美团点评技术文章合辑】：<http://dpurl.cn/iLGBtuS>

前端PDF文档的详细地址：<https://awps-assets.meituan.net/mit-x/2018-ebook-bundle3/2018-ebook-frontend.pdf>

Copyright (c) 2019 Fit Web Front End Tobayang. all right reserved, powered by Gitbooktime: 2019-08-28  
12:00:33



【用法】多层目录：

## 1. javascript

### • 1.0.1. ECMAScript

- ES6特性
- ES7特性
- ES8特性
- ES9特性

【用法】段首开头：

写法：（错误）

大  
幅  
度  
发  
到  
付  
写  
法：  
（  
正  
确  
）

大  
幅  
度  
发  
到  
付

【用法】正确的字体样式：

（正确）

【用法】错误的字体样式：

（错误）

- 【优化点】

- **【安全点】**

## 【用法】序号加粗

1.

## 【用法】加圈的序号

①

Copyright (c) 2019 Fit Web Front End Tobeyang. all right reserved, powered by Gitbooktime: 2019-09-09 13:02:01

**【用法】**加圈的序号

- 1. javascript
  - none
    - 1.0.1. ECMAScript