



前端JSer装逼手册

原创

2016年07月17日 10:29:25

标签: [前端](#) / [javascript](#) / [装逼](#) / [逼格](#) / [奇技淫巧](#)

925

前端JSer装逼手册

在装逼成本越来越高的JS圈，是时候充值一下了 —— 题记

§ 开发

Macbook Pro是标配，美其名曰“提高开发体验”
什么？你还在用Spotlight？赶紧给我换Alfred！

编辑器，Sublime / Atom / VS Code 三选一
虽然很想用IDE，但一定要忍住，并且与人解释道：
“启动速度慢，消耗资源多，不适合我这种完美主义者
如果不是为了美观，我宁愿使用 Vim / Emacs”

命令行iTerm2 + Oh-my-zsh
二逼青年用bash，普通青年用zsh
我们也只是想做一名普通人罢了

查资料虽然都是百度
但一定要称都是用Google
且要说英文而不是中文的“谷歌”
使用美式发音，当自己是湾区老司机

尽管四级飘过，六级没过
在Stack Overflow上点数也低
但也要说每天都与各国程序员谈笑风生

§ 语言

这年头如果还不用Babel + ES6
都不好意思说自己是JSer
当然还有 TypeScript / CoffeeScript / Dart ...
没学过没关系
对外人说自己“略懂”即可
反正最后都是编译为ES5，你懂的

为了避免对方深入问
此时你应该继续发表高见：
“JS是基于原型的函数式弱类语言
引入类与强类真的是不伦不类”
说到此，顿一下，表现出百感交集
随后继续徐徐道：
“可大势所趋，吾等小辈惟随波逐流”
说罢，即可挥挥衣袖转身离去

在这里不得不提一下，虽然使用Bable转码可以尽情装逼
但其对某些新特性的转换相当二逼（详情请看[这篇文章](#)）
一句话：Babel虽好，但别贪杯哦（推荐[Babel在线实时编译](#)）

联系我们



请扫描二维码联系?

✉ webmaster@c

☎ 400-660-010

💬 QQ客服 🗣 客

[关于](#) · [招聘](#) · [广告服务](#) · [用户](#)

©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

广告

他的最新文章

substring和substr，slice和splice/
不清楚？进来！

有逼格地判断数组是否为空

由bind引发的血案

2016春季阿里游戏前端实习生笔试

文章分类

JavaScript奇技淫巧与坑

面试笔试整理

文章存档

2016年7月

2016年3月

他的热门文章

前端JSer装逼手册

📖 922

2016春季阿里游戏前端实习生笔试

📖 656

substring和substr，slice和splice/
不清楚？进来！

📖 592

由bind引发的血案

📖 212

有逼格地判断数组是否为空

📖 202

§ 代码风格

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

✖ 册

摒弃JSLint / JSHint / JSCS, 拥抱ESLint
尽管平时只是个搬砖的
但时刻以世界顶级企业的规范约束自己
于是`eslint-config-airbnb`成了我们的标配

一般新手是这样写的:

```
1  /* Low */
2  if (a) {
3      return b;
4  } else {
5      return c;
6  }
```



0



逼格稍微高一点的这样写:

```
1  /* Bigger */
2  if (a) return b; // 提前结束, 免用大括号与else
3  return c;
```

实际上还能更进一步:

```
1  /* Bigger than bigger */
2  ① return a ? b : c // 不要写分号, 留白予人想象的空间
3  ② return a && b || c
```



总而言之, 代码越短, 可读性越差, 逼格越高
不能让人随便看懂, 就像人不能轻易让人看透

§ 奇技淫巧

罄竹难书

§ 常用库

DOM库

标配是jQuery, 手机端有Zepto作为替代品
想要装逼且不怕坑, 那就上Mootools
Prototype? 嗯, 复古的逼格都是很高的

一定要说自己纯粹为了优雅简洁, 不得不用jQuery
(如何做到jQuery-free, 请看[这篇文章](#))

当然, 就算是写jQuery

也能体现出逼格

我们来看看新手一般是怎么写的:

```
1  /* Low */
2  var value = $(".container .myInput1").val();
3  $(".container .myInput2").val(value);
4  $(".container .myInput3").attr("disabled", "disabled");
```

用双引号, 以及对选择器性能认知不足, 是新手的特征
一般直接使用类选择器的, 都是对用户体验很有自信的

```
1  /* Bigger */
2  // 把div.container命名为myDiv
3  var $myDiv = $('#myDiv'), // 缓存DOM
4      v = $myDiv.find('.myInput1').val();
```

联系我们



请扫描二维码联系2

✉ webmaster@c

☎ 400-660-010

🗣 QQ客服 🗣 客

关于 · 招聘 · 广告服务 · 站

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

```
7 .find('.myInput2').val(v)
8 .end() // 坚持链式调用
9 .find('.myInput3').attr('disabled', 'disabled');
```

(有关jQuery选择器的性能以及最佳实践, 请看[这篇文章](#))

UI

Bootstrap烂大街

不是我们的菜

我们选择的标准是门槛要高

于是

Foundation6 / Ant Design

映入眼帘

请谨慎使用

Semantic UI / UIKit / Amaze UI ...

避免不能自拔

工具库

后浪lodash把前浪underscore拍死在沙滩上

于是它成了唯一的选择

不过为了保持逼格

我们要尽量使用原汁原味的ES6

就算要用也一定要注意素质:

```
1 /* Low */
2 import _ from 'lodash' // 把整个lodash打包进去了
```

```
1 /* Bigger */
2 import isEmpty from 'lodash/isEmpty' // 仅把个别函数打包
```

模板引擎

逼格最高显然是Jade

但改名为Pug (哈巴狗) 后

就像是小龙女被尹志平不可描述后

再也无爱了

从此以后

留了胡子 (Mustache)

扶着把手 (Handlebars)

默默耕耘

异步编程

这里不谈 Q / Bluebird / Async / co / then 等库

皆因Babel已经支持所有的异步编程解决方案

当前最常用的还是Promise

有些新手会写出这种代码:

```
1 /* Low */
2 // 找出与用户1同市的所有用户
3 User.findById(1).then((user) => {
4   User.find({ city: user.city }).then((users) => {
5     res.json(users.toJSON())
6   })
7 })
```

这属于Promise[反模式](#), 与回调函数无异

```
1 /* Bigger */
```

```
2 User.findById(1).then((user) => {
3   User.find({ city: user.city }).then((users) => {
4     res.json(users.toJSON())
5   })
6 })
```

加入CSDN, 享受更精准的内容推荐, 与500万程序员共同成长!

返回Promise

联系我们



请扫描二维码联系我们

✉ webmaster@c

☎ 400-660-010

💬 QQ客服 🗣 客

关于 · 招聘 · 广告服务 · 网站地图

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

注册

登录

```
4  }).then((users) => {
5    res.json(users.toJSON())
6  }).catch(next)
```

§ 包管理工具

如果你被

Bower / spm / Component / Duo ...

坑过

请回到npm的怀抱

什么? jspm? 有完没完...



0



§ 构建工具

想当年我不懂什么是自动构建工具

他们说：生命苦短，我们用Grunt

好不容易用上Grunt的时候

他们又说：Gulp基于流，符合Unix哲学

之后我虔诚地换上了Gulp

他们双说：Webpack最好用

最后终于用上了Webpack

他们聂说：FIS3约不约? ...

§ 模块化方案

无论是

* RequireJS (AMD)

* SeaJS (CMD)

* KMD.js (KMD)

* Browserify (CommonJS)

* ...

最后都庆幸回归到npm + Webpack

什么? SystemJS? 有完没完...

§ MV*框架 / 技术栈 / 大型框架

Backbone

每个人都有一段不堪回首的经历

就像当年在QQ空间发“你若安好便是晴天”的说说

Backbone就是这样子的存在

Angular

一定要边吐槽边用，不然就一点都不ng了

“学习曲线陡峭”不应从你口中说出

“学习过程趣味盎然”才是你的菜

Vue

一定要用“优雅”来形容

就像用ES6一定要“大胆”

React技术栈

联系我们



请扫描二维码联系我们

✉ webmaster@c

☎ 400-660-010

💬 QQ客服 🗣 客服

[关于](#) · [招聘](#) · [广告服务](#) · [网站地图](#)

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

React已经是前端高逼格的代名词

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

“React大法好”
因为这是一种信仰
称赞JSX的标新立异
谈谈 Flux / Redux
扯扯 Elm / RxJS
每到深入则戛然而止：
“太深入的太抽象，你们未必能理解”
由此，听者只会更加崇拜你

其他

还有国内相对小众的 Ember / Knockout / Avalon
(请别再把 YUI / Dojo / Ext / KISSY 扯进来了好伐)



0



§ 混合 / 原生开发

自从PhoneGap出来后
貌似我们也能抢安卓/iOS的饭碗了
Ionic更是将Hybrid APP推向高潮

不过混合始终比不上原生
于是React Native应运而生
最近多了一个新的选择：Weex

别忘了还有桌面的nw.js以及Electron

JSer从一入门开始，就掌握了改变世界的能力
也比其他程序员更容易走向人生的巅峰

§ 后端框架

我们一直标榜自己是全栈
不玩几下后端框架怎么行

快递员用Express
风湿患者用Koa
哲学家用ThinkJS
水手用Sails

还有全栈的Meteor
上述都用一遍
相信也快转行了

§ 服务器进程管理

既然都玩上了后端框架
不懂部署服务器怎么行

二逼青年用supervisor / nodemon
文艺青年用forever
普通青年用pm2
装逼青年用Tmux + node

§ 结语

```
import you, { isGoodPost, star } from 'you'  
import me, { thank } from 'me'
```

联系我们



请扫描二维码联系我们

✉ webmaster@c

☎ 400-660-010

💬 QQ客服 🗣 客

关于 · 招聘 · 广告服务 · 网站地图

©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心