



ES6 | 与前端工程化入门

By SonderLau / 刘柏田

前言

- ES6
- 前端工程化

ECMAScript
6

**2015年6月发
布**

从制定到公布
耗时**15年**

区别

分不清 E S 与 J S ？

历史

实际关系

前者是后者的规格，后者是前者的一种实现

- JScript

- ActionScript

ES6与ECMAScript2015

2011年

ECMAScript 5.1 经常有新的提案 以年号为版本号
发布

ES6在2015年6月诞生了

ES6

历史名词

泛指


ES2015

正式名称

特指正式版本

不太懂？

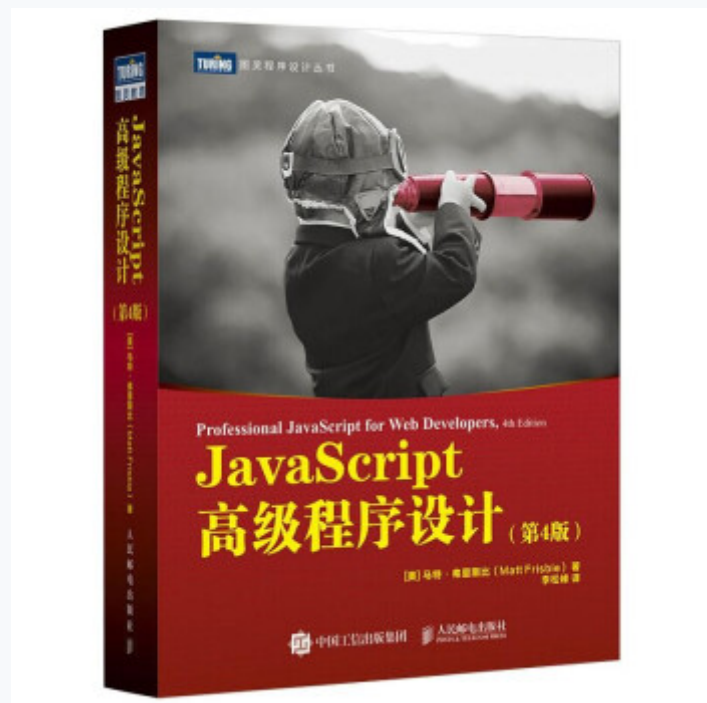
比上一个版本多了点新语法

- 《ES6入门教程》
- 不讲语法
- 不影响理解

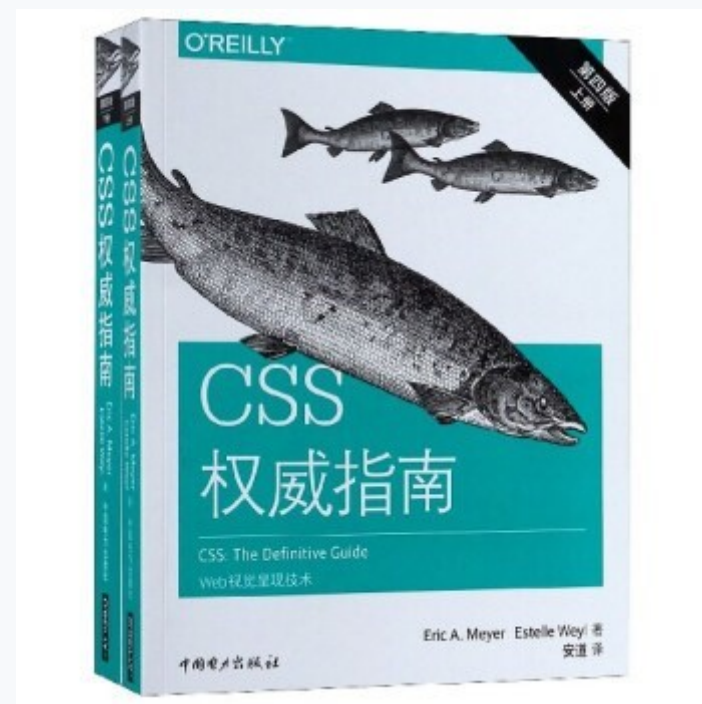
开讲前

- 现在有什么样的问题
- 我们想要实现怎么样的解决效果
- 目前比较成熟的解决方案

提出问题-解决问题-现有解答



《JavaScript 高级程序设计》第四版
红宝书



《CSS权威指南》第四版
2本 分为上下册 / 开源非官方译版

- 《CSS权威指南 第四版》 非官方中文译本 
- 《现代 JavaScript 教程》 
- 《JavaScript教程 - 阮一峰》 

工程化

三剑客



首先感觉到不舒服的地方

在两个组件内容几乎差不多的情况下，代码的复用性很低

动画

换一换 更多



B站高等级用户现状

UP 进击的金厂长



【天气愈报大合集】2020年汇总

UP 劲小郑



【海贼王】这就是世界第一大剑豪-乔拉可尔·米霍克

UP 叫我绿藻头



小丑竟是我自己? 2020年10月新番吐槽打脸大总结!

UP 泛式



爆龙专车——爆龙战队腕龙万代日版DX 合体爆炎王 对比...

UP 大鹏质检员



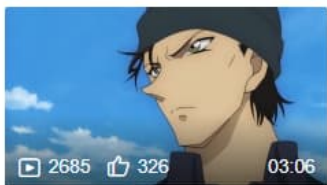
没有计时器的新奥特曼? 泽塔荣登N站年度网络流行语第...

UP 木南工坊



【SBR合集】『JOJO的奇妙冒险第7部飙马野郎』漫画解说

UP 竖笛小魔王



【赤井秀一/个人向】他顺光而行, 只为迎战黑暗

UP 唐先生的猫



【JOJO/吉良吉影】遇事不决·败者食尘

UP 无头叔



【jojo】用31个替身名唱一首稻香

UP 怠惰的一休



【配音手书】猜猜我是谁 (德安)

UP 忆梦君__



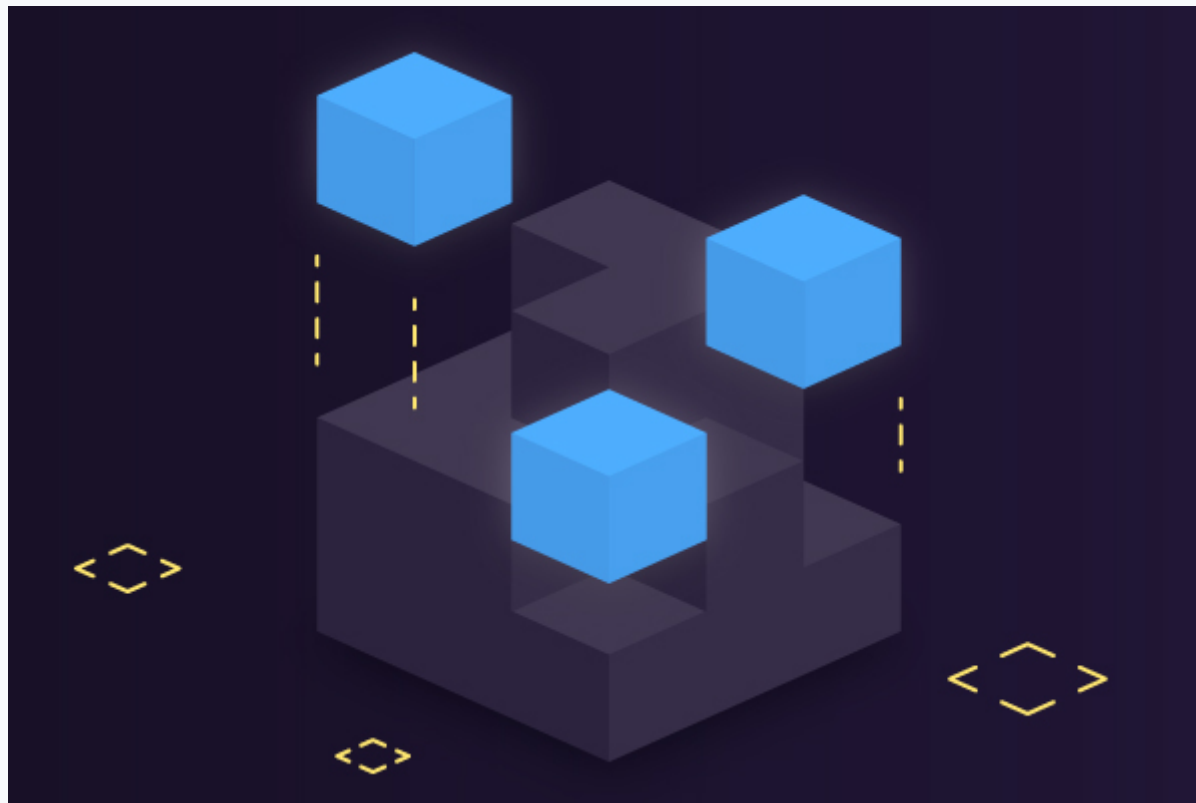
满级好感度迪卢克老爷已解锁

UP 玟_Amz

模板

模块化

模块化是一种设计思想，并不是前端领域独有的



拆分整合的文件 分别开发
不需要过多阅读 **上下文** 代码

编程语言中的模块化

Java

接口 Interface

类 Class

JavaScript

直到 ES6 才有实现

前端领域的模块化

JS 模块化

CSS 模块化

资源模块化

Node.js

- 一个文件就是一个模块 Module
- 每个模块都有单独的作用域
- 通过 `module.exports` 导出成员
- 通过 `require` 函数载入模块
- CommonJS 是以同步(Sync)模式加载模块

局限性

```
var math = require('math');  
math.add(2,3);
```

AMD

Asynchronous Module Definition

```
require([module], callback);
```

```
// define
define('math',['base'], function(){
  function add(a, b){
    return a + b;
  }
  return {
    add: add
  }
})

// load
require(['math'], function(math){
  math.add(2, 3);
})
```

- require.js
- curl.js

CMD

Common Module Definition

就近依赖 用什么才引什么

执行时机不同

• Sea.js

```
// define
define(function(require, exports, module){
  function add(a, b){
    return a+b;
  }
  exports.add = add;
})

// load
seajs.use(['math.js'], function(math){
  var sum = math.add(1+2);
})
```

“ AMD用户体验好，因为没有延迟，依赖模块提前执行了。CMD性能好，因为只有用户需要的时候才执行。

— 网评

ES6 Module

编译时引入模块代码

```
// define math.js
var basicNum = 0;
var add = function(a, b){
  return a + b;
}
export {basicNum, add};

// load
import {basicNum, add} from './math';
add(basicNum, 2);
```

CSS 模块化

- Sass
- Less
- Stylus
- PostCSS
-

原生 CSS

- style 标签

- ```
<head>
 <style>
 div {
 margin: 20px;
 }
 </style>
</head>
```

- style 行内

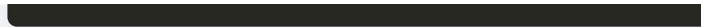
- ```
<div style="margin:20px">
  <!-- something here -->
</div>
```

- link 外部文件

- ```
<link href="./main.css" rel="stylesheet" >
```

- @import 导入

- ```
@import url(common.css) all;
```

有不少的问题

- 行内样式

- 不能复用
- 权重高
- 结构没分离
- 不能缓存

- 导入样式

- 导入语法位置
- 不能利用并发请求
- 页面体验不佳

"原生CSS 苦天下久矣"

预处理器 Sass / Less

解决了

变量

嵌套

父选择器

效果

书写方便

结果仍是原生

```
// variable
@width: 10px;
@height: @width + 10px;
```

```
#header{
  width: @width;
  height: @height;
}
```

```
// nesting
#header {
  .navigation {
    font-size: 12px;
  }
  .logo {
    width: 300px;
  }
}
```

```
// father selector
&:nth-child(2) {
  margin: 20px;
}
}
```

后处理器 PostCSS

- 抽象语法树 Absctrac Syntax Tree
- 分析源码
- 提供API供修改内容

Autoprefixer



CSS的样式在每个浏览器的不同版本上的实现是不同的

Can I Use

```
<input placeholder="我应该是灰色的! ">
```

```
input::placeholder {  
  color: grey;  
}
```



CSS selector: `::placeholder`



Usage % of **all users** ?

Global 95.17%

unprefixed: 89.74%

Current aligned Usage relative Date relative Filtered All ⚙

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser	KaiOS Browser
		2-18	4-5	3.1-4	10-12.1	3.2-4.3										
		19-50	6-56	5-10	15-43	5-10.2							4-6.4			
6-10	12-86	51-83	57-86	10.1-13.1	44-71	10.3-13.7		2.1-4.4.4	12-12.1				7.2-12.0			
11	87	84	87	14	72	14.2	all	81	59	87	83	12.12	13.0	10.4	7.12	2.5
		85-86	88-90	TP												

Notes

Test on a real browser **NEW**

Feedback

See full reference on [MDN Web Docs](#).

Support data for this feature provided by:



MDN browser-compat-data

#

CSS selector: `::placeholder`

Current aligned

Usage relative

Date relative

Filtered

All



IE	Edge	Firefox
		2-18
		19-50
6-10	12-86	51-83
11	87	84
		85-86

Firefox 19 - 50

Support info

✓ Supported with prefix: `-moz-`

Browser versions

Released Jan 8, 2013 - Nov 15, 2016

Total usage

Global: 0.25%

[Test on Firefox 19 - 50](#)

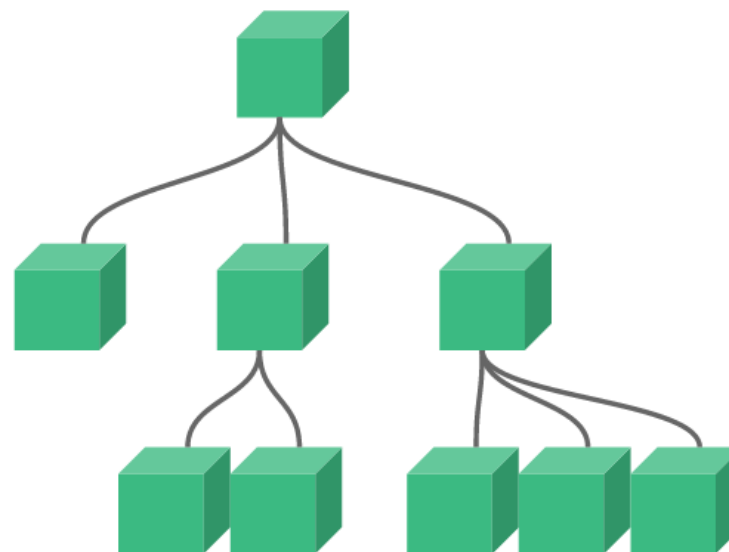
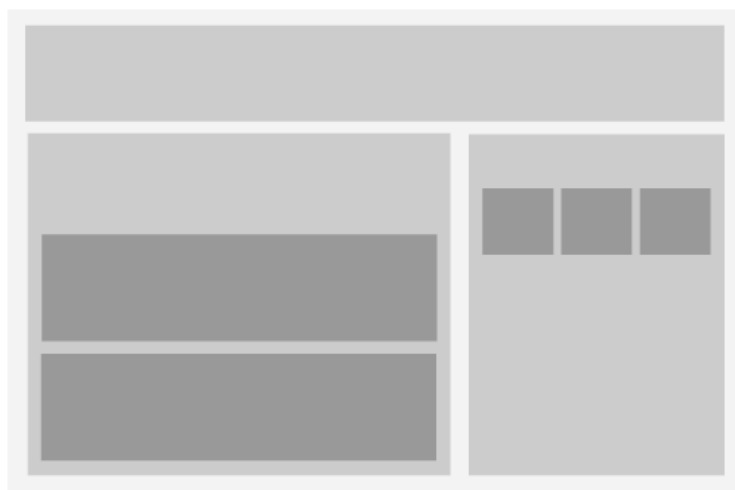
```
input::-moz-placeholder {  
  color: gray;  
}  
input:-ms-input-placeholder {  
  color: gray;  
}  
input::placeholder {  
  color: gray;  
}
```

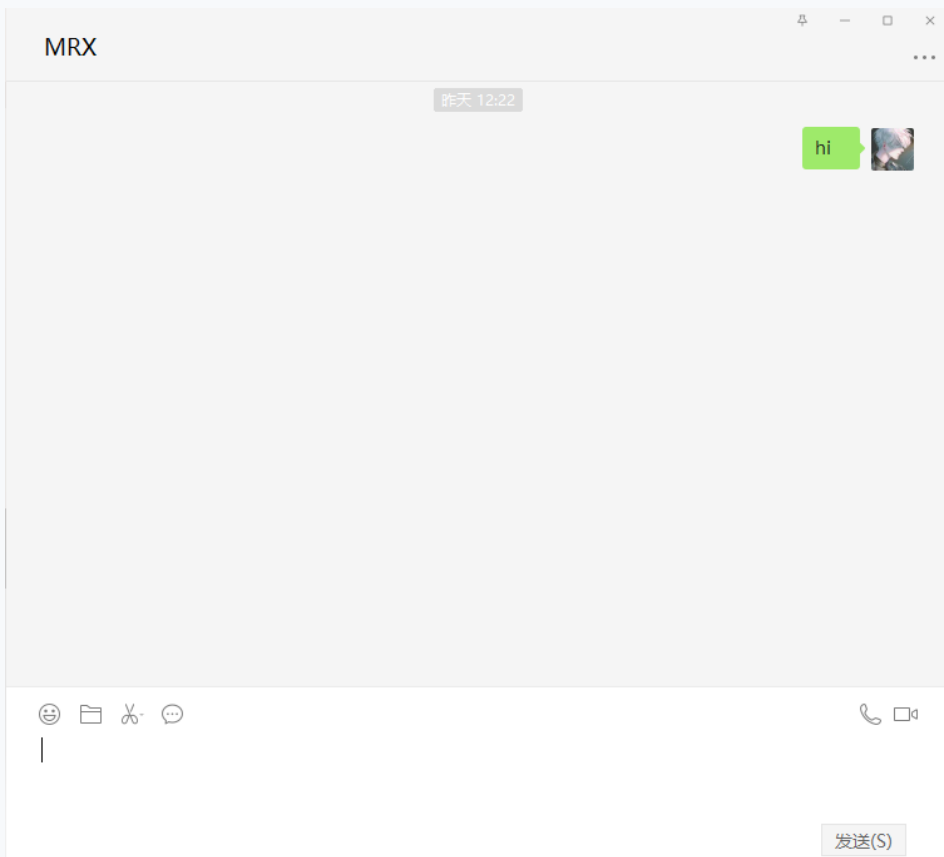
资源模块化

- 资源以 **模块** 的形式加载

组件化

UI层面的拆分





基础组件

业务组件

区块组件

页面组件

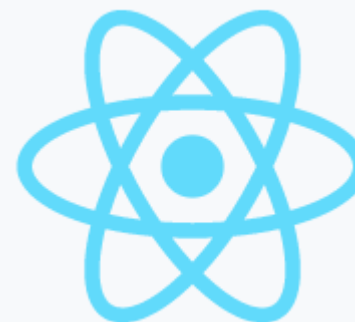
页面组件	Chat	
区块组件	MessageList	FunctionBar
业务组件	MessageItem	
基础组件	img、h3、p	input、button

组合

渐进式 & 离散式



Vue.js



React.js

Vue.js

渐进式框架

双向数据绑定机制

React.js

函数式编程

触发事件更新显示的内容

```
<ul>
  {this.props.children}
</ul>
```

```
<main>
  <MessageList>
    messages.map(message => <MessageItem message={message} /> )
  </MessageList>
  <FunctionBar />
</main>
```

“高内聚 低耦合” — 设计原则

规范化

目录结构

api views
components

编码规范

ESLint StyleLint

文件命名规范

略

样式管理规范


BEM Sass Stylus
CSS Modules

Block Element Modifier

```
<div class="menu">
  <div class="menu__header">
    <h2 class="menu__title">Title text here</h2>
  </div>
  <div class="menu__body">
    
    <p class="menu__text">text</p>
  </div>
</div>
```

gitflow workflow



- 分支命名规范
- 合并规范
- 计算机教育中缺失的一课 

定期的代码审查

- 对计算机源代码系统化地审查，常用软件同行评审的方式进行
- 提升软件质量以及开发者技术

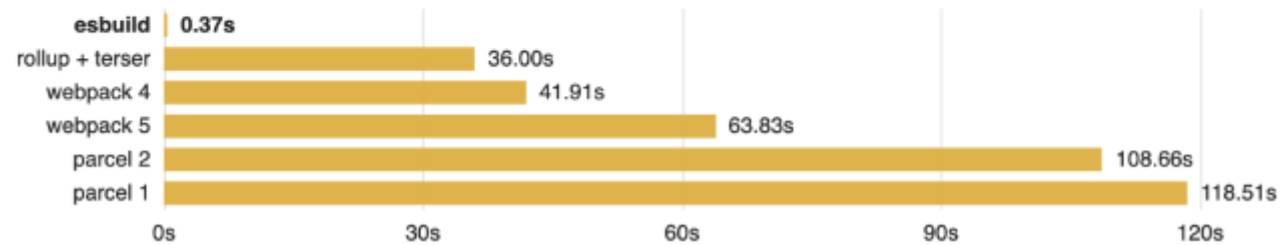
打包 与 整合





esbuild

An extremely fast JavaScript bundler



H Z I E E . S I T E



Hziee 计科生存指南

我什么都不会...ohh 还是会一点点的咯！

开始摸鱼 →

Github | [🔗](#)

真实有效

由在校学生上传并提供 具有时效性和可参考性

丰富多样

覆盖大部分学科的笔记/大作业/
实践作业

更新迅速

上完课就更呗 才不会拖更呢！
QAQ

Hziee.site 自动构建并部署

```
name: Build and Deploy
on: [push]
jobs:
  build-and-deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - uses: borales/actions-yarn@v2.0.0
        with:
          cmd: install
      - uses: borales/actions-yarn@v2.0.0
        with:
          cmd: docs:build
      - uses: SamKirkland/FTP-Deploy-Action@3.1.1
        with:
          ftp-server: ftp://ftB.cname-cdn.com:1041/wwwroot/
          ftp-username: sonderlau
          ftp-password: ${ secrets.FTP_PW }
          local-dir: docs/.vuepress/dist/
          git-ftp-args: --insecure # if your certificate is setup correctly this can be removed
```

定时全盘同步

```
name: CI
on:
  schedule:
    - cron: 0 21 * * *
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout master
        uses: actions/checkout@v2
        with:
          ref: master
      - name: Setup node
        uses: actions/setup-node@v1
        with:
          node-version: "10.x"
      - name: Upload COS
        uses: zkqiang/tencent-cos-action@v0.1.0
        with:
          args: delete -r -f / && upload -rs ./assets/ /
          secret_id: ${ secrets.SECRET_ID }
          secret_key: ${ secrets.SECRET_KEY }
          bucket: ${ secrets.BUCKET }
          region: ap-shanghai
```

推送时更新

name: CI

on:

push:

branches:

- master

jobs:

build:

runs-on: ubuntu-latest

steps:

- **name:** Checkout master

uses: actions/checkout@v2

with:

ref: master

- **name:** Tencent Cloud COS Action

uses: TencentCloud/cos-action@v1

with:

secret_id: \${{ secrets.SECRET_ID }}

secret_key: \${{ secrets.SECRET_KEY }}

cos_bucket: \${{ secrets.BUCKET }}

cos_region: ap-shanghai

local_path: assets

remote_path: /

clean: **true**

一点建议

1. 观摩其他人的作品，研究源码
2. 按类别实现常见的UI
3. 多练，多练，多练
4. 了解HTTP(S) 相关原理
5. 学习一点设计理论

“真相不曾放过我，我也不会放过真相。”

— Thanks 