

# 聊聊前端新技术

屈光宇



<https://imququ.com>



***ES6 / ES7***

# 新特性

let和const命令  
变量的解构赋值  
字符串的扩展  
正则的扩展  
数值的扩展  
数组的扩展  
函数的扩展  
对象的扩展  
Symbol  
Proxy和Reflect

二进制数组  
Set和Map数据结构  
Iterator和for...of循环  
Generator函数  
Promise对象  
异步操作和Async函数 \*  
Class  
Decorator \*  
Module

```
"use strict";
```

```
for(var i = 0; i < 10; i++) {  
    setTimeout(function() {  
        console.log(i);  
    }, 1);  
}
```

```
for(let i = 0; i < 10; i++) {  
    setTimeout(function() {  
        console.log(i);  
    }, 1);  
}
```

```
"use strict";
```

```
var obj = {  
  name : 'world',  
  hello : function() {  
    setTimeout(function() {  
      console.log('hello ' + this.name);  
    }, 10);  
  }  
};  
obj.hello();
```

```
var obj = {  
  name : 'world',  
  hello : function() {  
    setTimeout(() => {  
      console.log('hello ' + this.name);  
    }, 10);  
  }  
};  
obj.hello();
```

# 方案



<https://babeljs.io/>

# TypeScript

<http://www.typescriptlang.org/>

强类型、TypeScript Definitions File

# CSS 预 / 后处理器

# 新特性

变量、函数、运算符；

嵌套、作用域；

混入、继承；

文件组织；

减少重复劳动（自动前缀）；

拥抱未来；

... ..



CSS input

```
:fullscreen {  
}
```

CSS output

```
:-webkit-:full-screen {  
}  
:-moz-:full-screen {  
}  
:full-screen {  
}
```

CSS input

```
:root {  
  --red: #d33;  
}  
a {  
  &:hover {  
    color: color(var(--red) a(54%))  
  }  
}
```

CSS output

```
a:hover {  
  color: #dd3333;  
  color: rgba(221, 51, 51, 0.54);  
}
```

# 方案



<http://sass-lang.com/>



<http://postcss.org/>

**cssnext** {|||||}

<http://cssnext.io/>

# 异步编程

# 方案

Promise;

Async functions;

Generators; //不推荐使用

# Callback

```
1
2 var db = new Db('test', new Server('localhost', 27017));
3 // Establish connection to db
4 db.open(function(err, db) {
5   // Create a test collection
6   db.createCollection('test_group', function(err, collection) {
7     // Perform a simple group by on an empty collection
8     collection.group([], {}, {"count":0}, "function (obj, prev) { prev.count++; }", function(err, results) {
9       assert.deepEqual([], results);
10      // Trigger some inserts on the collection
11      collection.insert([{'a':2}, {'b':5}, {'a':1}], {w:1}, function(err, ids) {
12        // Perform a group count
13        collection.group([], {}, {"count":0}, "function (obj, prev) { prev.count++; }", function(err, results) {
14          assert.equal(3, results[0].count);
15          // Perform a group count using the eval method
16          collection.group([], {}, {"count":0}, "function (obj, prev) { prev.count++; }", false, function(err, results) {
17            // Group with a conditional
18            collection.group([], {'a':{'$gt':1}}, {"count":0}, "function (obj, prev) { prev.count++; }", function(err, results) {
19              collection.group([], {'a':{'$gt':1}}, {"count":0}, "function (obj, prev) { prev.count++; }", false, function(err, results) {
20                collection.insert([{'a':2}, {'b':3}], {w:1}, function(err, ids) {
21                  collection.group(['a'], {}, {"count":0}, "function (obj, prev) { prev.count++; }", function(err, results) {
22                    collection.group({'a':true}, {}, {"count":0}, "function (obj, prev) { prev.count++; }", true, function(err, results) {
23                      // Correctly handle illegal function
24                      collection.group([], {}, {}, "5 ++ 5", function(err, results) {
25                        assert.ok(err.message != null);
26                        // Use a function to select the keys used to group by
27                        var keyf = function(doc) { return {a: doc.a}; };
28                        collection.group(keyf, {a: {$gt: 0}}, {"count": 0, "value": 0}, function(obj, prev) { prev.count++; });
29                      });
30                    });
31                  });
32                });
33              });
34            });
35          });
36        });
37      });
38    });
39  });
40 });
```



# Promise

```
var self = this;
domainInstance.run(function() {
  return tag('app_init', http).then(function() {
    return Dispatcher(http).run();
  }).then(function() {
    return tag('app_begin', http);
  }).then(function() {
    return tag('action_init', http);
  }).then(function() {
    return self.exec(http);
  }).then(function() {
    return tag('app_end', http);
  }).catch(function(err) {
    self.sendError(http, err);
  })
});
```

# Generators

```
var co = require('co');  
function * gen() {  
  var data = yield Promise.resolve(1111);  
  return data;  
}  
co(gen).then(function(data) {  
  //data is 1111  
}))
```

# Async functions

```
async exec(){  
    await this.hook('resource');  
    await this.hook('route_parse');  
  
    await this.hook('logic_before');  
    await this.execLogic();  
    await this.hook('logic_after');  
  
    await this.hook('controller_before');  
    await this.execController();  
    await this.hook('controller_after');  
  
    await this.hook('response_end');  
}
```



# 模块化 / 构建工具

# 方案

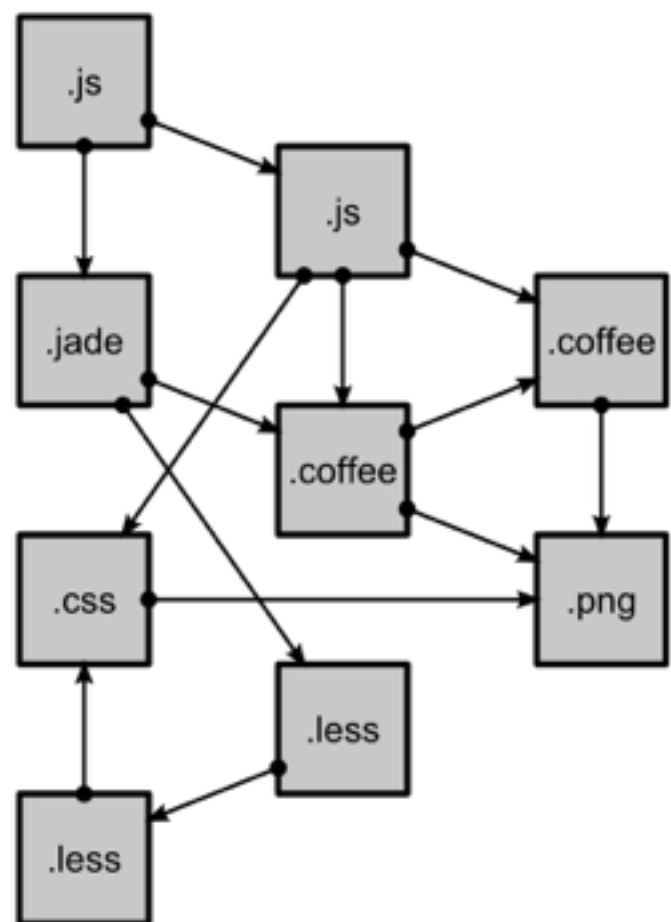
CommonJS (CJS);

Asynchronous Module Design (AMD);

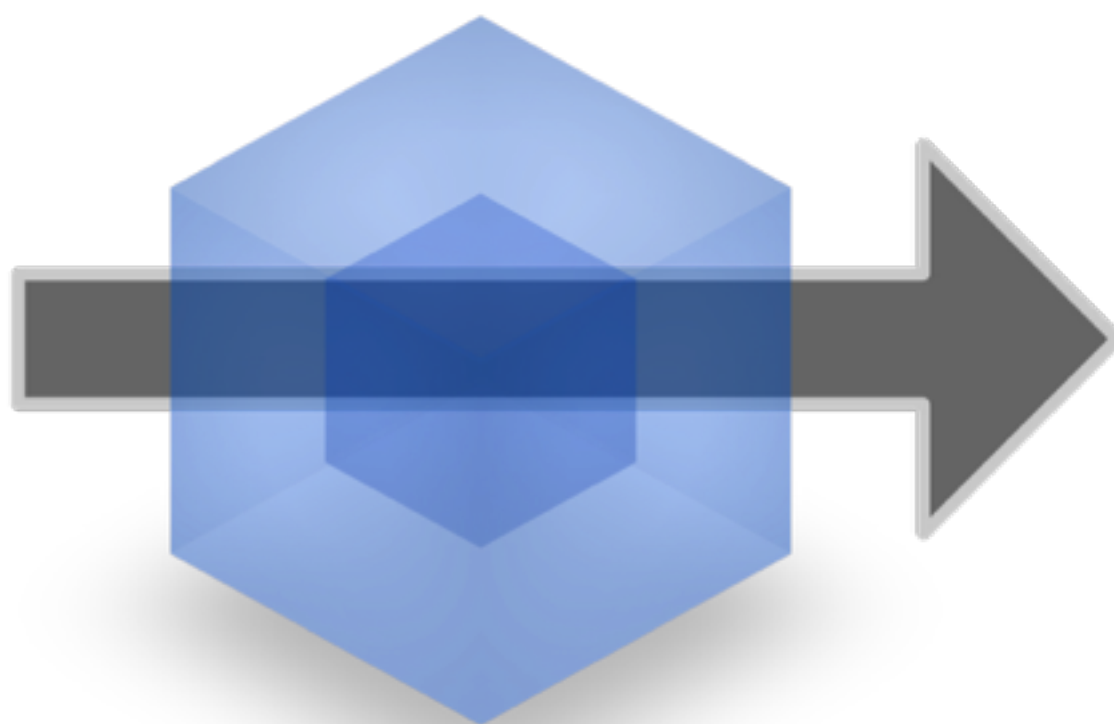
ES6 modules;



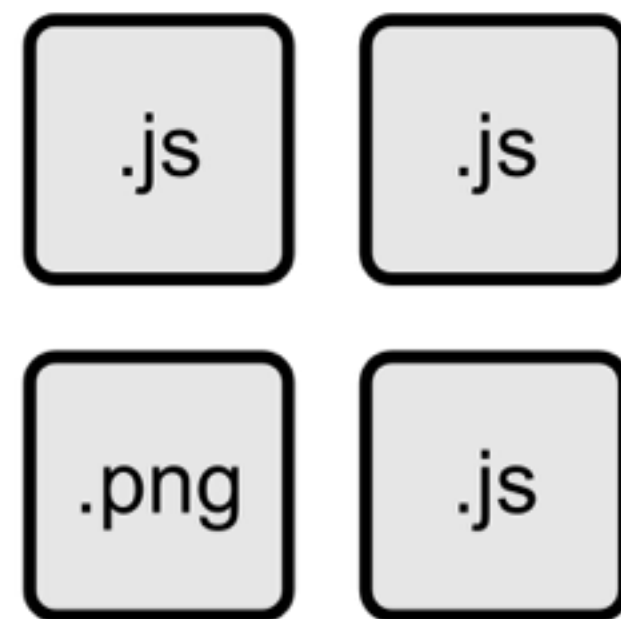
<https://webpack.github.io/>



modules  
with dependencies



**webpack**  
MODULE BUNDLER



static  
assets

框架

# 解决的问题

数据到 DOM 的声明式映射；

组件的组织方式；

组件之间如何组合与沟通；

组件的高内聚，高可移植性；

... ..

# 方案



<https://angular.io/>



<https://facebook.github.io/react/>



<http://vuejs.org/>

测试

# 代码规范



<http://eslint.org/>



# 单元测试



<https://karma-runner.github.io/0.13/index.html>

# 覆盖率测试

istanbul

<https://gotwarlost.github.io/istanbul/>

## 集成测试

# CasperJS

<http://casperjs.org/>

# Nightwatch.js

<http://nightwatchjs.org/>

# workflow

# 解决的问题

开发环境：

提供静态服务器；

基础编译；

实时编译；

实时更新（热更新）；

上线：

代码审查；

编译；

合并；

优化；

压缩；

发布到 CDN；

方案

*Gulp*

<http://gulpjs.com/>

# 网络协议

百度一下, 你就知道

← → ↺ 🏠 <https://www.baidu.com>

🔍 搜狗搜索引擎 - 上网从搜狗开

← → ↺ 🏠 <https://www.sogou.com>

🔍 360搜索, SO靠谱

← → ↺ 🏠 <https://www.so.com>

北京 🌤️ 霾3~17°C 空气质量: **重度污染** ▼

新闻 网页

JD 京东(JD.COM)-综合网购首选

← → ↺ 🏠 <https://www.jd.com>

🔍 淘宝网 - 淘! 我喜欢

← → ↺ 🏠 <https://www.taobao.com>

360导航\_新一代安全上网导航

← → ↺ 🏠 <https://hao.360.cn>

-  热门
-  直播
-  游戏

# 360导航

 360搜索

-  我的主页
-  新闻头条
- 

- |  |  |
|--|--|
|  新闻·视频  |  电影·电视剧 |
|  购物·海淘  |  彩票·理财  |
|  活期·定期  |  小说·商城  |
|  游戏·小游戏 |  动漫·直播  |



# HTTPS

了解 TLS 握手基本原理；

了解证书信任链基本原理；

了解 HTTP Strict Transport Security 等新增响应头；

理解 Mixed Content；

# HTTP/2

- 了解 HTTP/2 新增功能；
- 了解 HTTP/2 针对性能优化所做的改进点；
- 了解如何针对 HTTP/2 进行前端性能优化；

# 移动 *APP* 开发

# 难点

用户体验；

调试；

资源包更新；

调用系统功能；

# 方案



<https://facebook.github.io/react-native/>

# 桌面客户端开发

# 难点

安装包（windows）；

自动升级；

崩溃报告；

调试工具；

系统菜单、托盘图标等；

# 方案



<http://electron.atom.io/>





# 服务端开发

(Node.js)

# 特性

事件驱动，异步编程；

非阻塞 IO；

NPM 丰富的资源；

JS 知识无缝迁移；

适合 IO 密集型项目，不适合 CPU 密集型项目；

# 方案



<https://thinkjs.org/>

***Thank You!***

***Q & A***

 [quguangyu@gmail.com](mailto:quguangyu@gmail.com)