



# 2016年信天翁训练营

# 深入学习JavaScript

李丁辉-转转事业部-前端技术部





# 自我介绍



人力资源部



神奇学院  
Magic College

## 李丁辉

- 6年前端开发
- 伪果粉、工具控
- 目前在转转事业部-前端技术部，负责转转App FE团队
- 关注前端工程、前端性能



# 主要内容



人力资源部



神奇学院  
Magic College

- JS那些事儿 – 语言的由来
- 庄周梦蝶 – 语言精髓
- 九阴真经 – 修炼之道
- 打死小强 – 调试和性能分析



# JS那些事儿



# JS之父?



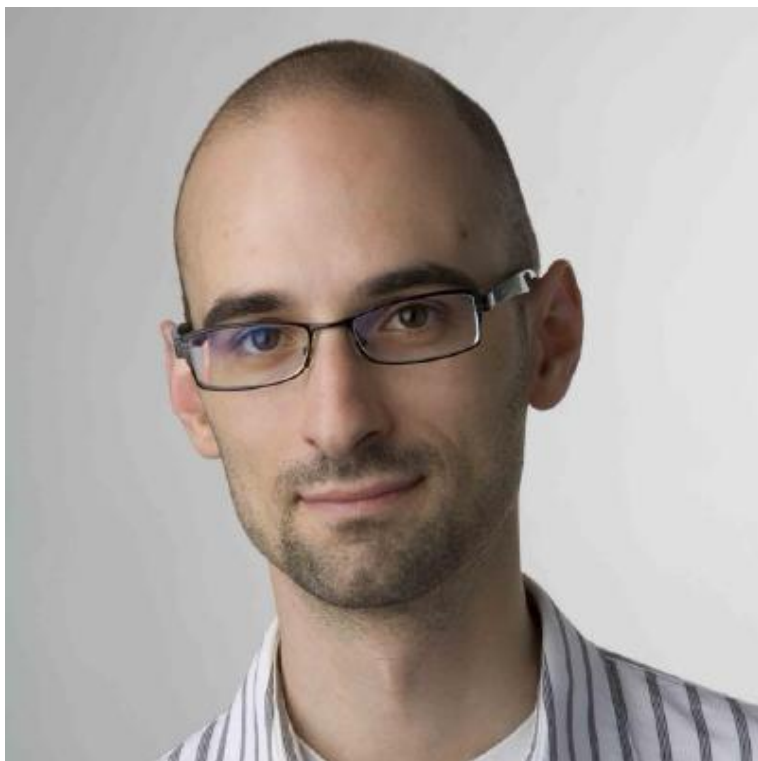
1



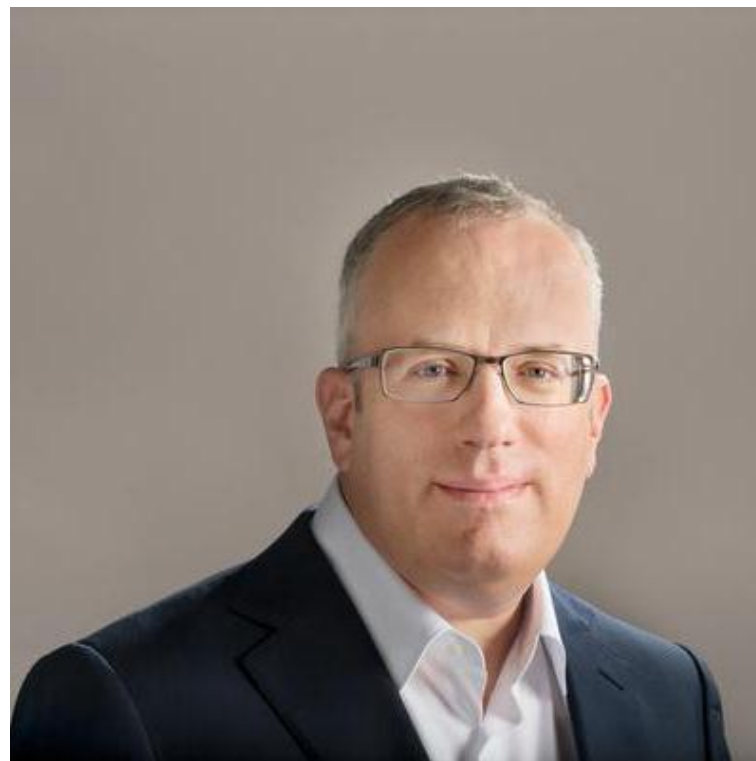
2



# JS之父?



3



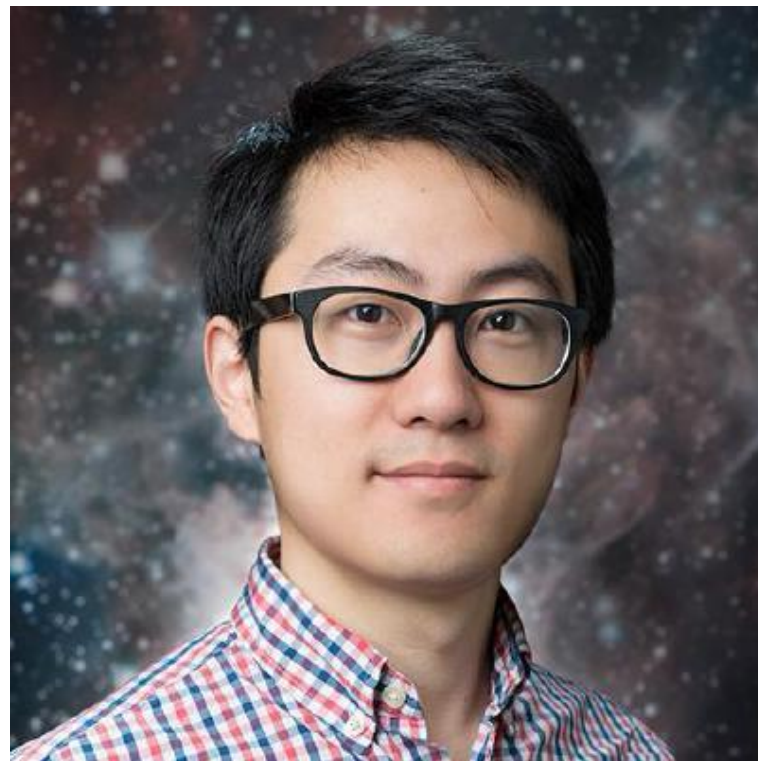
4



# JS之父?



5



6





# JS之父?



7



8





# JS之父?



John Resig

jQuery 之父



Douglas Crockford

- JSON 之父
- 《JavaScript 语言精髓》作者



Nicholas c. Zakas

《JavaScript高级程序设计》作者



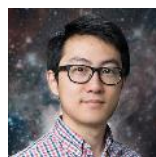
Brendan Eich

JavaScript之父



Ryan Dahl

NodeJS 之父



尤雨溪

Vue 之父



Marc Andreessen

Browser 之父

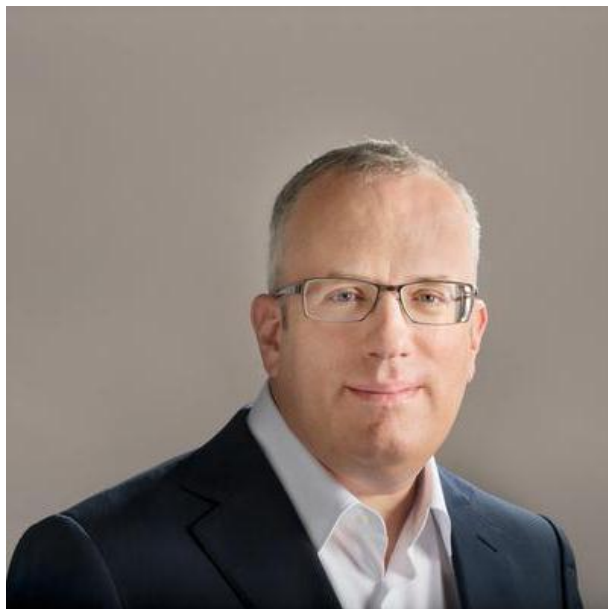


李晓肆

□ □ □



# 布兰登·艾奇



Brendan Eich

<https://brendaneich.com>

- 1995年供职于网景，开发JavaScript
- 2005年至2014年，任Mozilla CTO
- 2015年，任Mozilla CEO，十天后被迫辞职





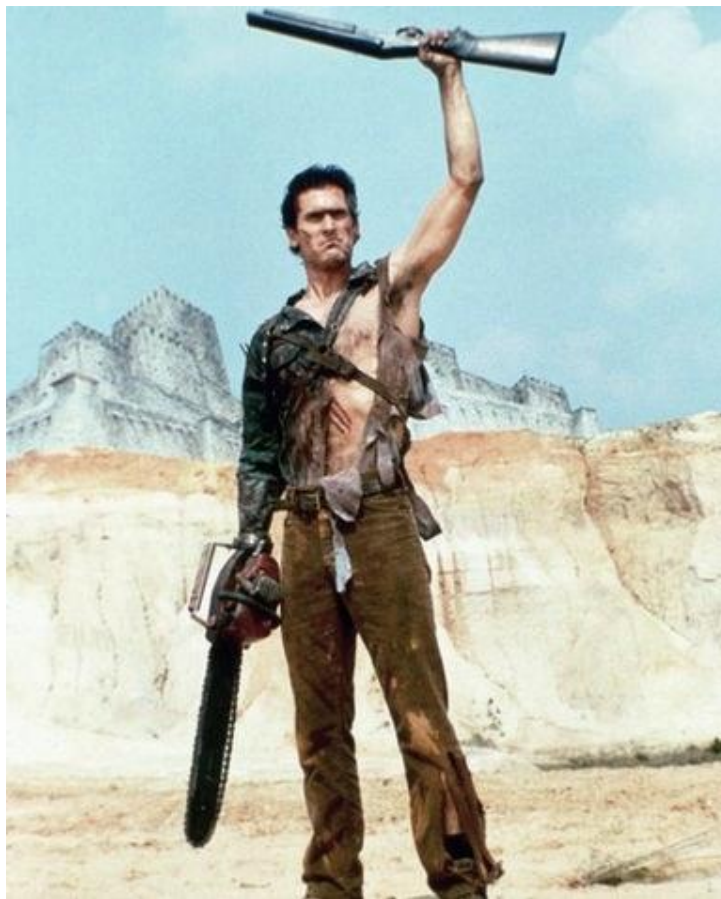
1995: Eich recruited to Netscape

**"Come and put Scheme in the browser!"**

<http://brendaneich.github.io/BrazilJS-2015/#3>



# JS的由来



1995, after I was done creating JS in 10 days

<http://brendaneich.github.io/ModernWeb.tw-2015/#5>



So in **10 days** in May 1995, I wrote

- A lexical scanner and parser for early JS
- The parser emitted stack-machine bytecode
- Which ran in a bytecode interpreter
- Function.prototype.toString bytecode decompiler
- The standard library was poor
- Array was Object with .length property
- Date hand-ported (h/t ksmith@netscape.com) from java.util.Date

<http://brendaneich.github.io/ModernWeb.tw-2015/#20>



# Java与JavaScript

人力资源部

神奇学院  
Magic College







# Java与JavaScript



浏览器大战



比尔.乔伊



马克.安德森



Java's bytecode design influenced my work on JS.

## Java Intermediate Bytecodes

### ACM SIGPLAN Workshop on Intermediate Representations (IR '95)

James Gosling <jag@eng.sun.com>  
Sun Microsystems Laboratories

*Java<sup>†</sup> is a programming language loosely related to C++. Java originated in a project to produce a software development environment for small distributed embedded systems. Programs needed to be small, fast, “safe” and portable. These needs led to a design that is rather different from standard practice. In particular, the form of compiled programs is machine independent bytecodes. But we needed to manipulate programs in ways usually associated with higher level, more abstract intermediate representations. This lets us build systems that are safer, less fragile, more portable, and yet show little performance penalty while still being simple.*

<http://brendaneich.github.io/BrazilJS-2015/#23>



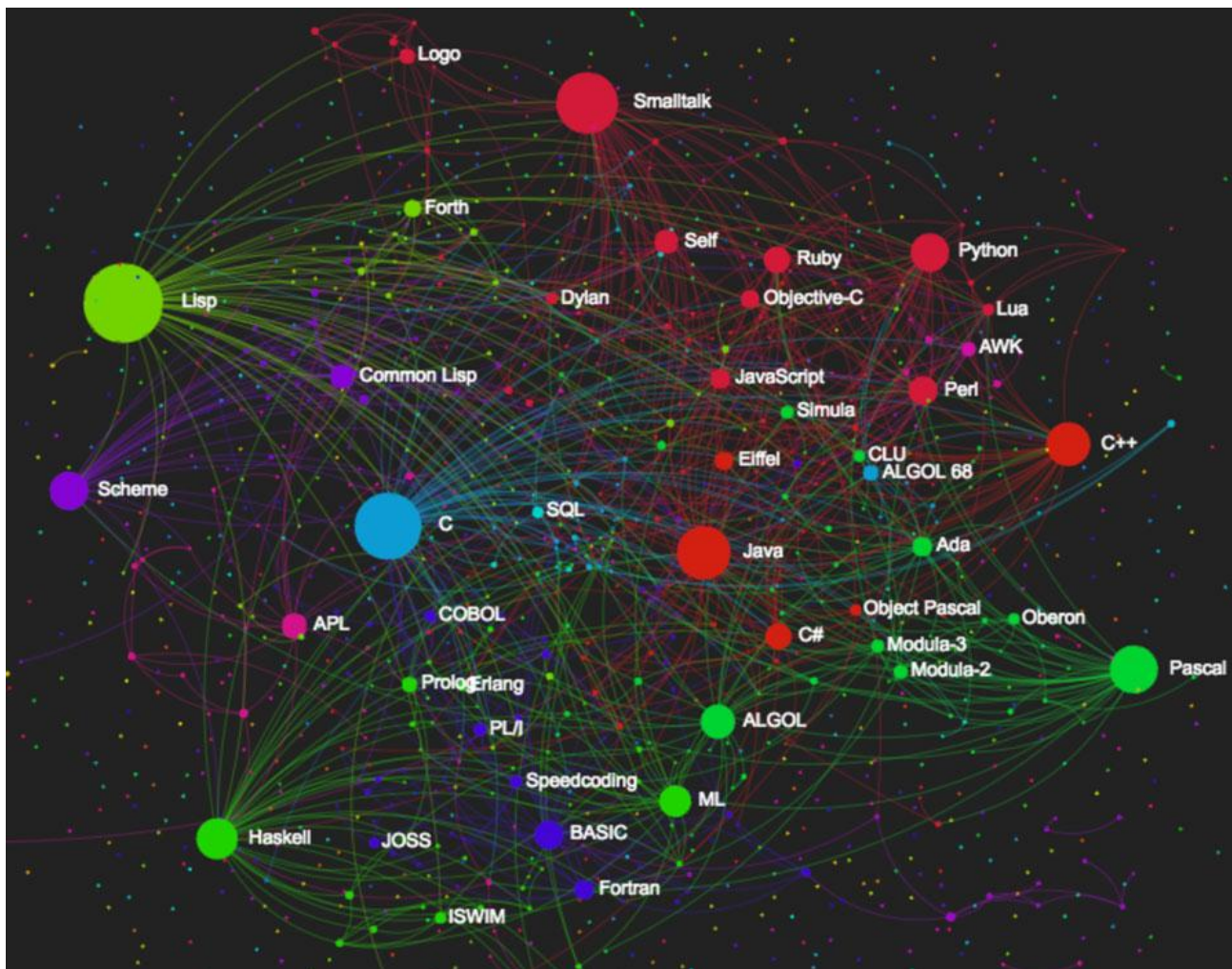
“它是披着C外衣的Lisp”



《JavaScript语言精髓》(道格拉斯.克利斯朵夫)



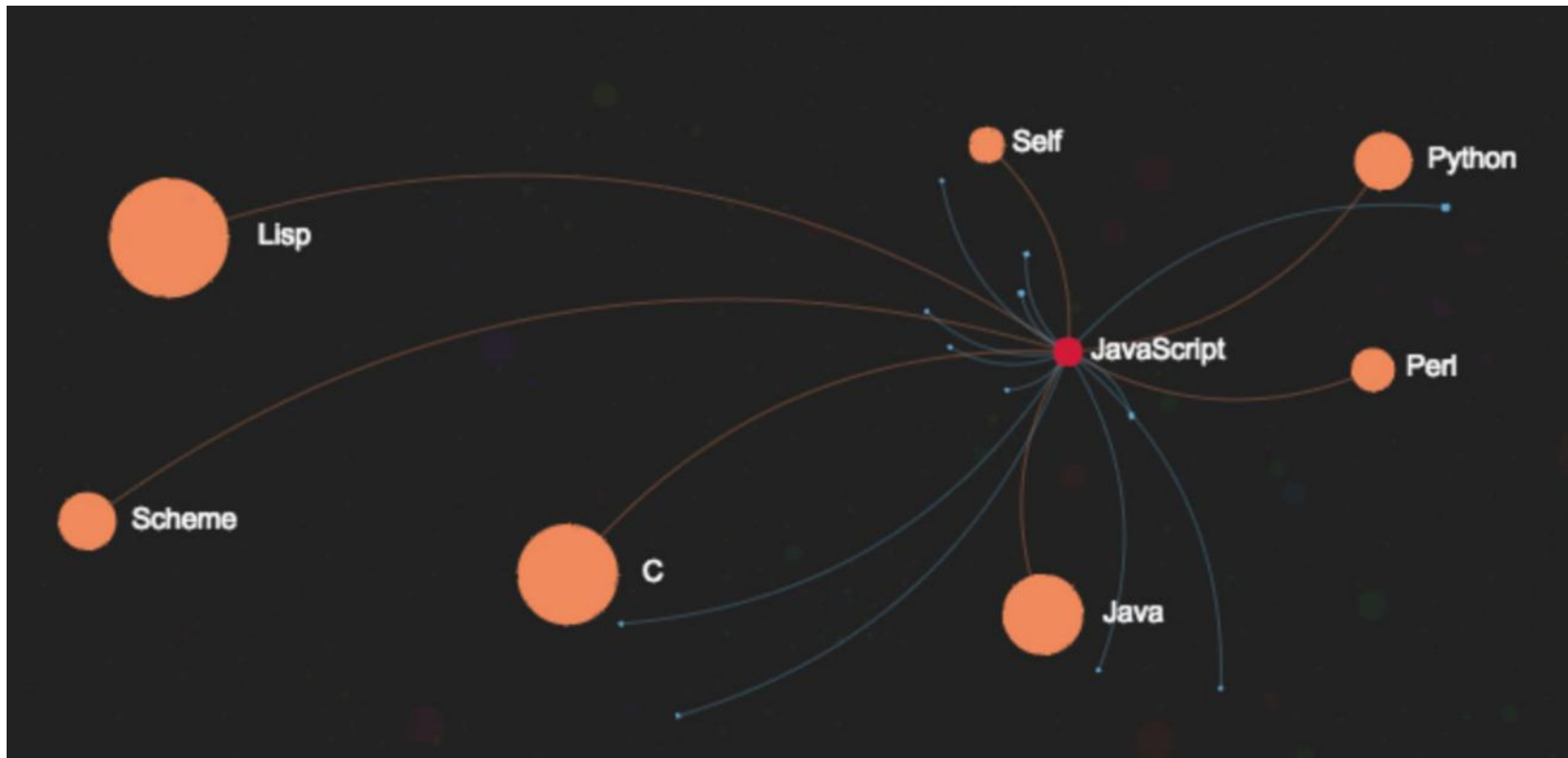
# JS借鉴的语言



<https://exploringdata.github.io/vis/programming-languages-influence-network/>



# JS借鉴的语言



<https://exploringdata.github.io/vis/programming-languages-influence-network/#JavaScript>



- 借鉴C语言的基本语法
- 借鉴Java语言的数据类型和内存管理
- 借鉴Scheme语言，将函数提升到"第一等公民"（ first class ）的地位
- 借鉴Self语言，使用基于原型（ prototype ）的继承机制

《Javascript诞生记》（ 阮一峰 ）

[http://www.ruanyifeng.com/blog/2011/06/birth\\_of\\_javascript.html](http://www.ruanyifeng.com/blog/2011/06/birth_of_javascript.html)





# JS社区与标准



人力资源部



神奇学院  
Magic College



*CoffeeScript*

<http://coffeescript.org/>



<http://clojurescript.org/>



TypeScript

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



Dart

<https://www.dartlang.org/>

facebook



<https://flowtype.org/>



# JS标准

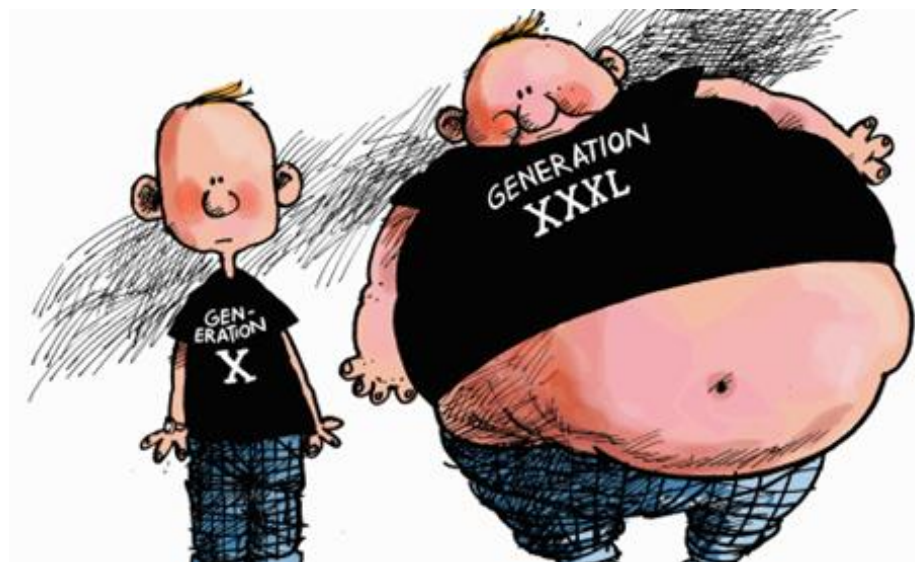


人力资源部



神奇学院  
Magic College

- 1999 ECMAScript 3
- 2004 ECMAScript 4(放弃)
- 2009 ECMAScript 5(ECMAScript 3.1)
- 2015 ECMAScript 6(ECMAScript 2015)
- 2016 ECMAScript 7(ECMAScript 2016)
- 2017 ECMAScript Next





# TC39



## Ecma TC39



Ecma International, Technical Committee 39 - ECMAScript

The web [http://ecma-international.org/memento/...](http://ecma-international.org/memento/)

### Repositories

People 18

Filters ▾

Find a repository...

### ecmascript\_sharedmem

HTML ★ 135 7

Shared memory and atomics for ECMAScript

Updated 7 hours ago

### agendas

★ 138 55

TC39 meeting agendas

Updated a day ago

### People

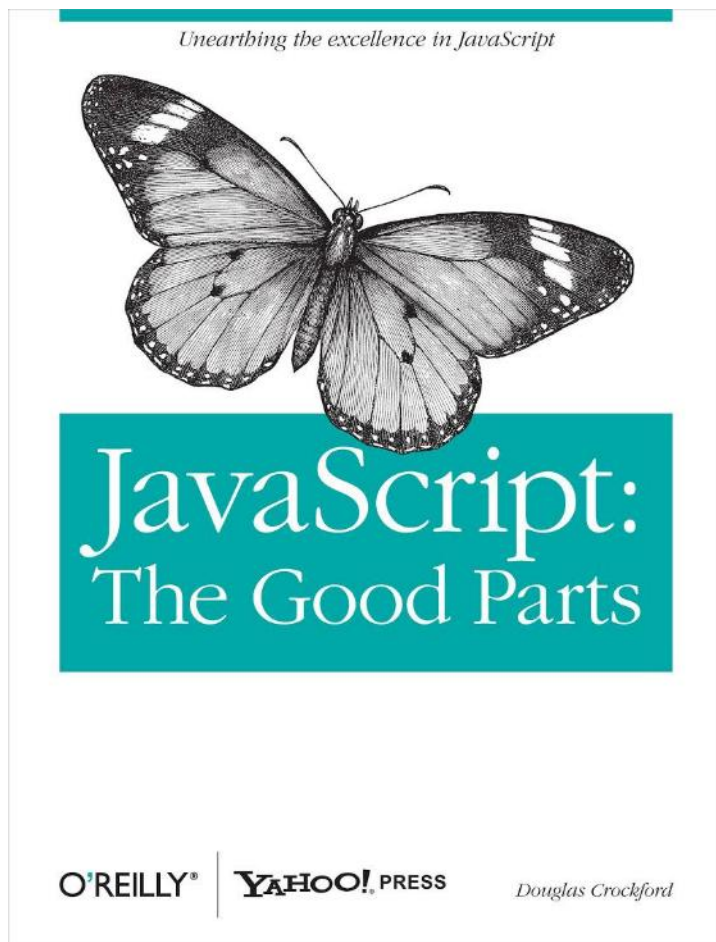
18 >



<https://github.com/tc39>



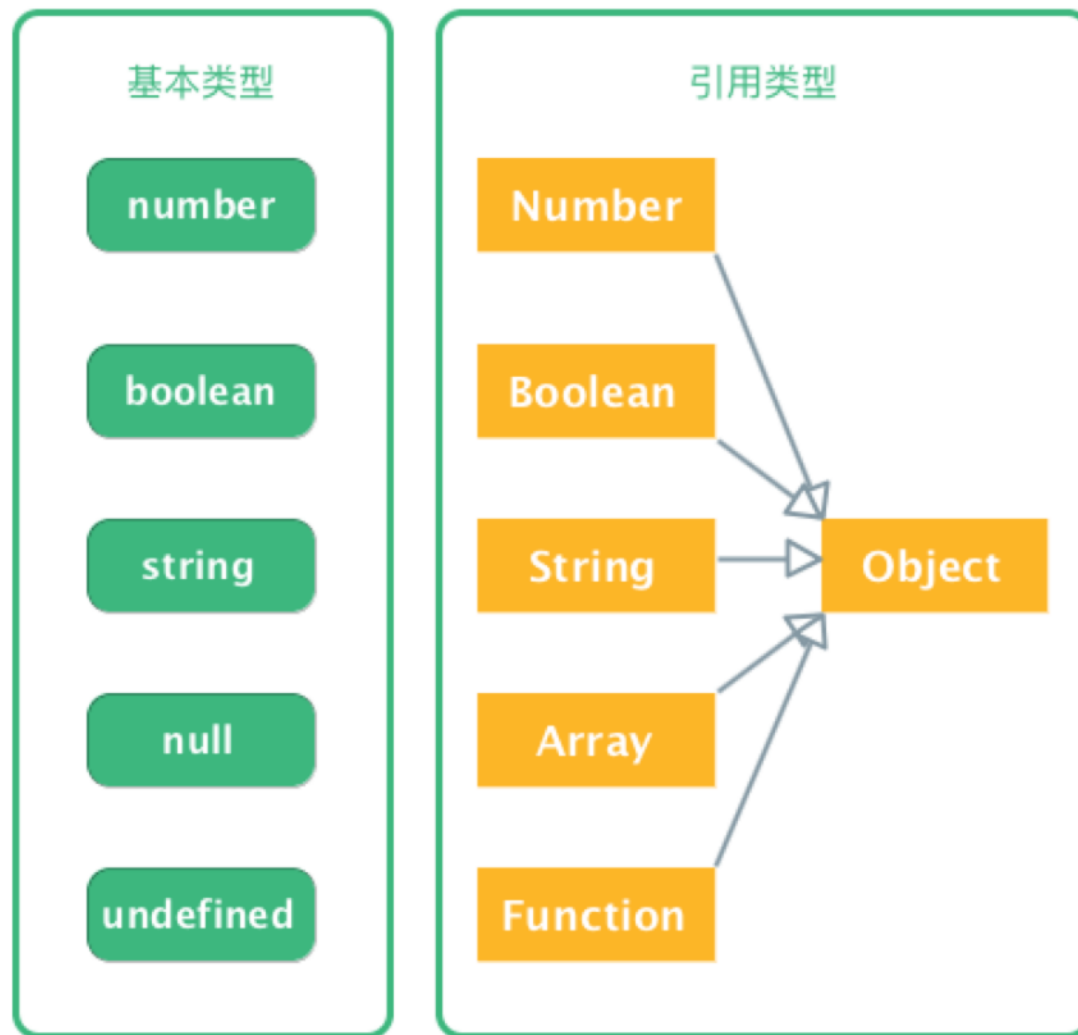
# 庄周梦蝶



- 对象
  - 类型
  - 反射
  - 原型
  - 对象的CRUD
- 函数
  - 第一等公民
  - 动态绑定
  - 作用域
  - 闭包



# JS类型







# 反射

```
typeof 1  
typeof 'hello'  
typeof true  
typeof undefined  
typeof null  
typeof Array  
typeof {}  
typeof []  
typeof /^1$/i
```





# 反射



人力资源部



神奇学院  
Magic College

```
typeof 1 //number
typeof 'hello' //string
typeof true //boolean
typeof undefined //undefined
typeof null //object
typeof Array //function
typeof {} //object
typeof [] //object
typeof /^1$/i //object
```



# 原型



人力资源部



神奇学院  
Magic College

```
(1).__proto__  
'hello'.__proto__  
(true).__proto__  
([]).__proto__  
({}).__proto__  
Array.__proto__  
Function.__proto__  
Object.prototype.__proto__
```



# 原型



人力资源部



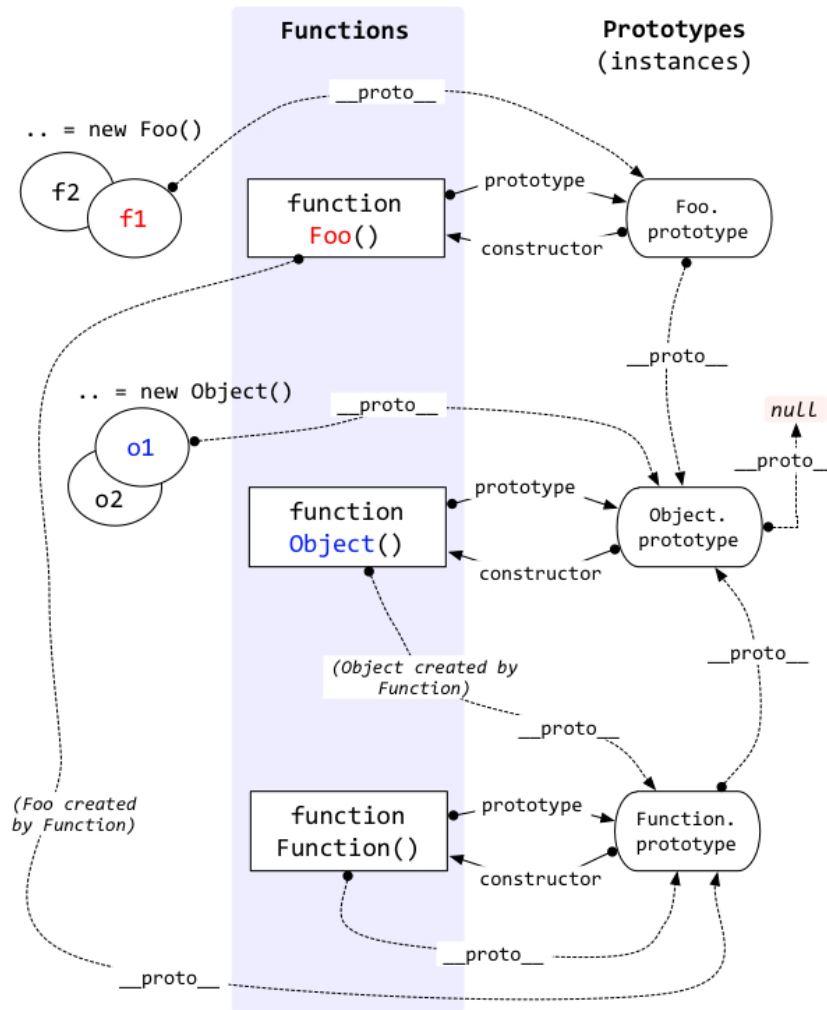
神奇学院  
Magic College

```
(1).__proto__           // Number.prototype
'hello'.__proto__       // String.prototype
(true).__proto__        // Boolean.prototype
([]).__proto__          // Array.prototype
({}).__proto__          // Object.prototype
Array.__proto__         // Function.prototype
Function.__proto__      // Function.prototype
Object.prototype.__proto__ // null
```



# 原型

JavaScript Object Layout [Hursh Jain/mollypages.org]





# 原型



\_\_proto\_\_

prototype

constructor







# 对象的CRUD



人力资源部



神奇学院  
Magic College

```
function Person(){}  
var p = new Person();           // Create  
console.log(p.name);           // Read  
  
Person.prototype.name = 'person';  
console.log(p.name);           // Read  
  
delete p.name                   // Delete  
console.log(p.name)            // Read  
  
p.name = 'p';                   // Update  
console.log(p.name);           // Read  
  
delete p.name;                  // Delete  
console.log(p.name);           // Read
```



# 装箱

```
Object.prototype.getInstance = function(){  
    return this;  
}
```

```
var numberInstance = (1).getInstance();  
//Number {[[PrimitiveValue]]: 1}
```

```
var booleanInstance = (true).getInstance();  
//Boolean {[[PrimitiveValue]]: true}
```

```
console.log(typeof numberInstance); //object  
console.log(typeof booleanInstance); //object
```

**true === new Boolean(true)**



# 函数是第一等公民



人力资源部



神奇学院  
Magic College

- 可以用变量存储
- 可以作为参数传递
- 可以作为返回值
- 具备对象的一切特征



# 函数是第一等公民

```
function consoleFactory(prefix){  
    return function(msg){  
        console.log(prefix + ':' + msg)  
    }  
}  
  
var log = consoleFactory('log'),  
    error = consoleFactory('error');  
log('success');           // log:success  
error('no param');        // error:param
```



# 动态绑定

```
function person(nickname){  
    if(nickname){ this.nickname = nickname; }  
    console.log(this.nickname);  
}  
person.prototype.nickname = 'Tom';  
var human = {  
    person : person,  
    nickname : 'Cat'  
}  
person();  
person('Jack');  
new person();  
new person('Rose');  
human.person();  
person.call(window);
```



## this执行时候才能确定绑定的值

- 函数调用, this指向global
- 方法调用, this指向所有者
- 构造函数, this指向新实例
- call/apply/bind, this指向传入的实参对象



# 作用域

```
var a = '58';  
function hello(){  
    console.log(a);  
    var a = 'FE';  
    console.log(a);  
}  
hello();
```





# 作用域

- JavaScript只有函数作用域
- 检索本地var,function定义的变量,生成本地作用域
- 检索是否使用包裹的祖先函数的本地作用域,生成闭包
- 检索全局定义的变量,生成全局作用域



# 闭包



人力资源部

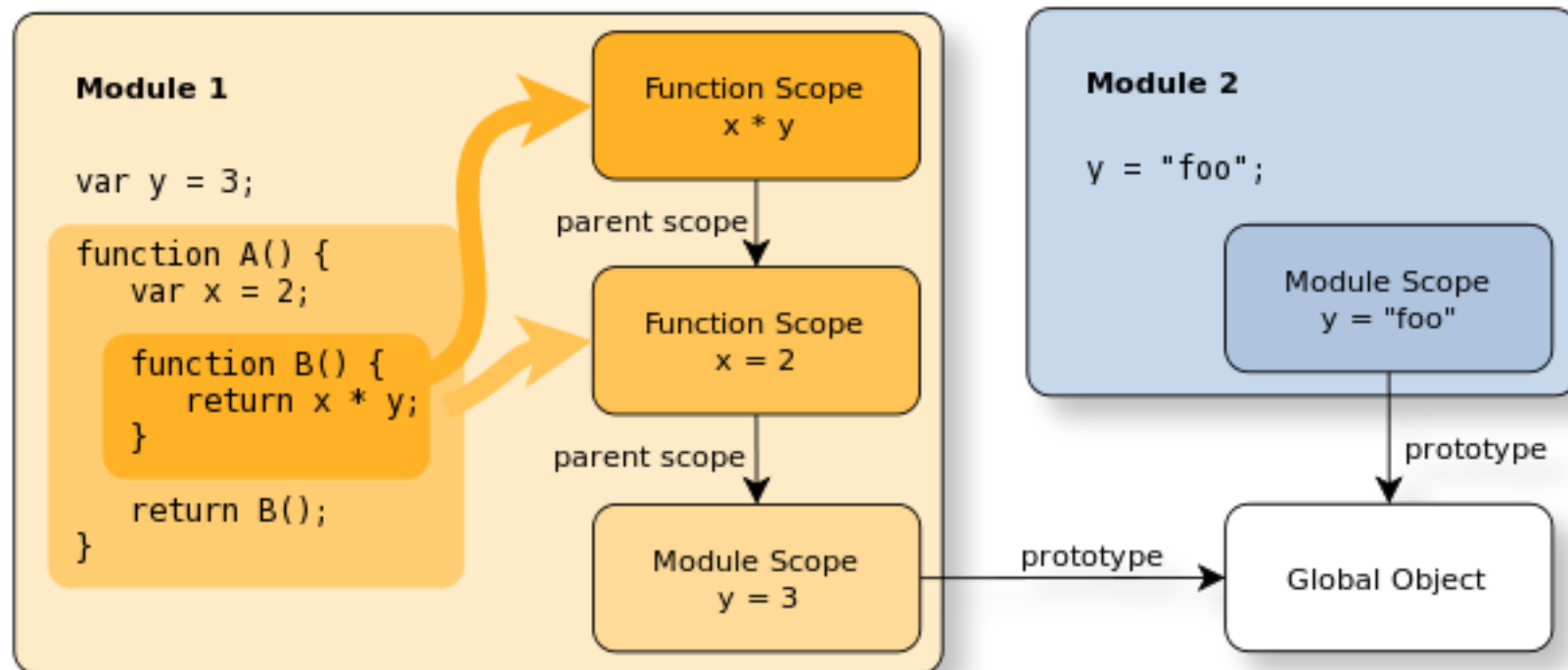


神奇学院  
Magic College

一个函数定义在另一个函数内部,可以访问包裹函数的本地作用域



# 闭包





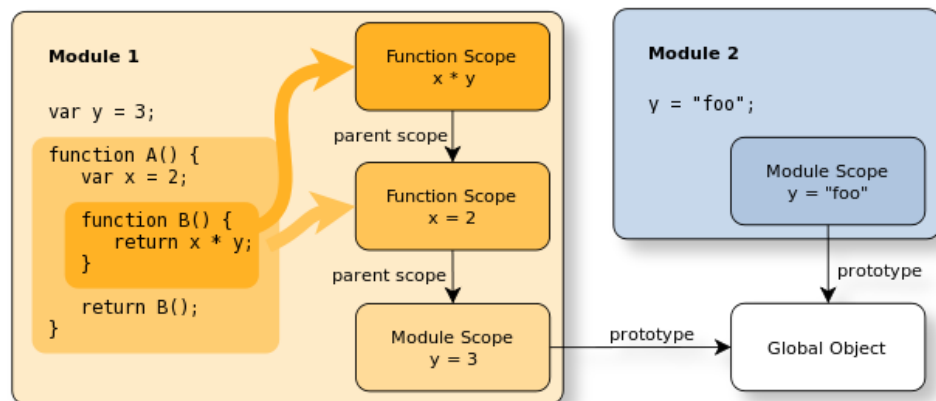
# 闭包



人力资源部



神奇学院  
Magic College



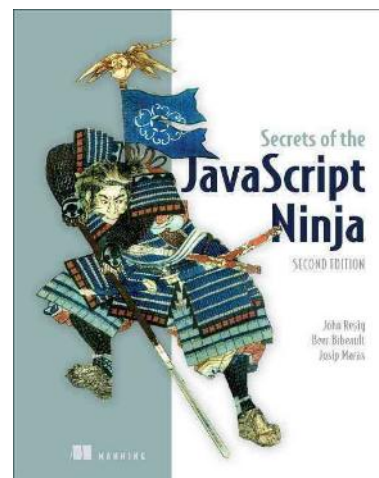
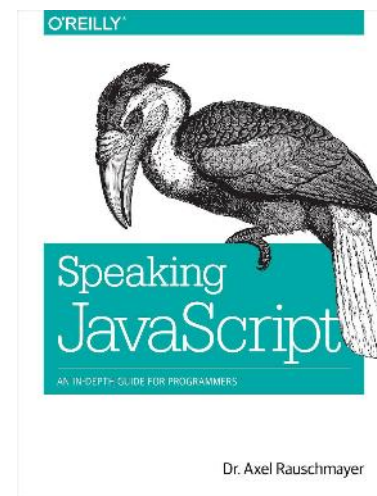
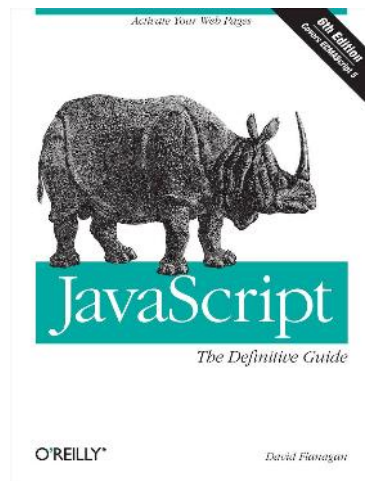
```
var BScope = {} // B函数本地作用域
var AScope = { x:2, B:B } // A函数本地作用域
var GScope = { y:3, A:A } // 全局作用域
BScope.__proto__ = AScope;
AScope.__proto__ = GScope;
// x * y ==> BScope.x * GScope.y
```



# 九阴真经



# 优质书籍





# Github开源代码



人力资源部



神奇学院  
Magic College



Personal

Open source

Business

Explore

Pricing

Blog

Support

Search GitHub



Your dashboard

Explore GitHub

Showcases

Integrations

Trending

## Trending in open source

See what the GitHub community is most excited about this month.

Repositories

Developers

Trending: this month

[yarnpkg/yarn](#)



Fast, reliable, and secure dependency management.

JavaScript • 12,773 stars this month • Built by



★ Star

[FreeCodeCamp/FreeCodeCamp](#)

The <https://FreeCodeCamp.com> open source codebase and curriculum. Learn to code and help nonprofits.

JavaScript • 11,576 stars this month • Built by



★ Unstar

[gavinkwoe/weapp-ide-crack](#)

【应用号】IDE + 破解 + Demo

JavaScript • 4,370 stars this month • Built by



★ Unstar

All languages

Unknown languages

CSS

HTML

Java

JavaScript

Objective-C

Python

Ruby

Other: Languages

**ProTip!** Looking for most forked JavaScript repositories?  
[Try this search](#)





- 看文档
- clone代码，安装依赖
- 运行单元测试
- 添加注释、console
- 断点调试
- 类图、顺序图



# Just Do It

JSFiddle online code editor interface showing HTML, JAVASCRIPT 1.7, and CSS panes. The HTML pane contains a script tag and a div with id 'container'. The JAVASCRIPT pane contains a React class definition for 'Hello' and a render call. The CSS pane is empty. The top bar includes buttons for 'Run', 'Update', 'Fork', 'Set as base', 'Tidy', 'Collaborate', and 'Embed'. The bottom bar shows the user 'leexiaosi'.

<https://jsfiddle.net/>



# 打死小强



# 善用断言



人力资源部



神奇学院  
Magic College

> console

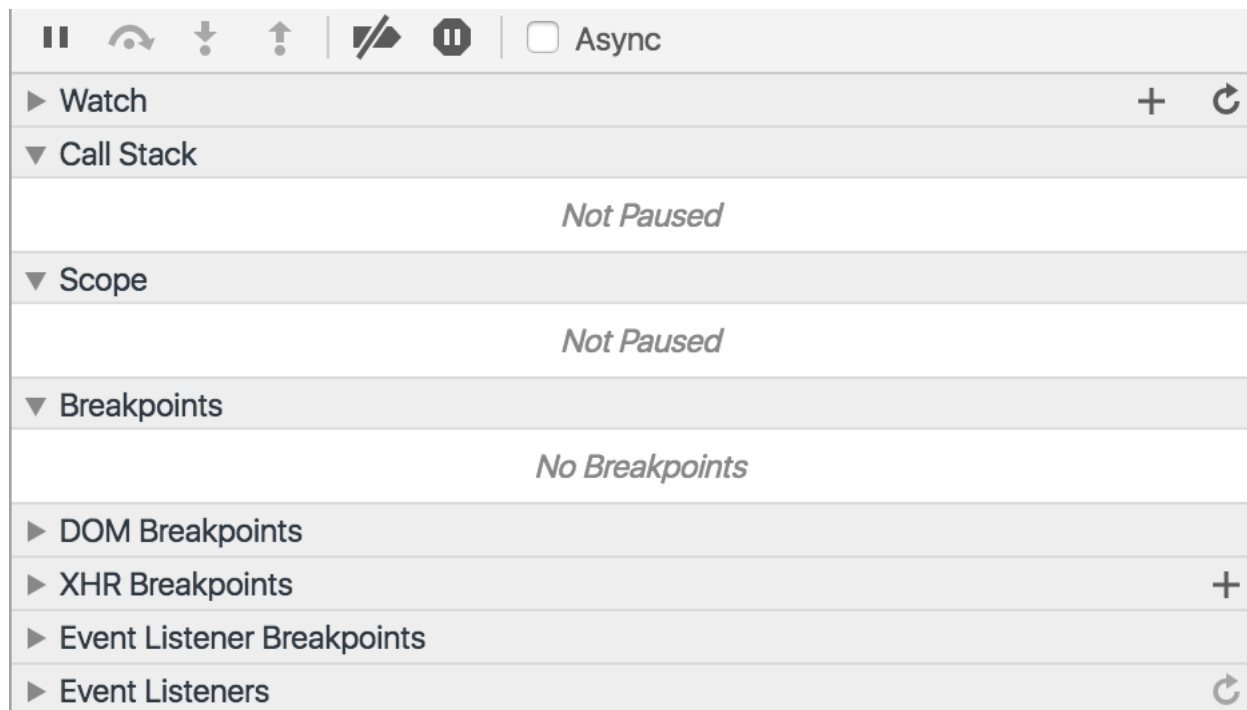
< ▼ Object {} ⓘ

- ▶ assert: function assert()
- ▶ clear: function clear()
- ▶ count: function count()
- ▶ debug: function debug()
- ▶ dir: function dir()
- ▶ dirxml: function dirxml()
- ▶ error: function error()
- ▶ group: function group()
- ▶ groupCollapsed: function groupCollapsed()
- ▶ groupEnd: function groupEnd()
- ▶ info: function info()
- ▶ log: function log()
- ▶ markTimeline: function markTimeline()
  - memory: (...)
- ▶ get memory: function ()
- ▶ set memory: function ()
- ▶ profile: function profile()
- ▶ profileEnd: function profileEnd()
- ▶ table: function table()
- ▶ time: function time()
- ▶ timeEnd: function timeEnd()
- ▶ timeStamp: function timeStamp()
- ▶ timeline: function timeline()
- ▶ timelineEnd: function timelineEnd()
- ▶ trace: function trace()
- ▶ warn: function warn()

console.assert()  
console.log()  
alert()  
debugger



# 调试工具





# NodeJS调试



人力资源部



神奇学院  
Magic College



## IntelliJ IDEA

The most intelligent Java IDE



## WebStorm

The smartest JavaScript IDE

<https://www.jetbrains.com/products.html#lang=js>



# 2016届信天翁训练营

**THANKS**  
@your name