



真传X

IT前沿技术在线大学

www.zhenchuanx.com

课程介绍

第一期跟StudQ合作：9月-12月

- 有同学翻番了，从7500涨到15000
- 有同学成功拿到58同城的offer
- 学员还有苏宁、美团、爱奇艺的同学也来学习了
- 也有同学面进了阿里

课程是怎么帮助大家？

- 课程内容：基础知识 & 应用 & 成长面试
- 练习作业：测试驱动
- 面试跟踪：目前仅针对符合到课率要求人群，预期大概是10%-20%

所以面试跟踪到底包含什么？

- 价值500块的BAT工程师模拟面试
- STAR面试准备辅导
- 一次面试的跟踪辅导

学习更多是靠自己也是为了自己

- 能自己解决问题往往比别人告诉你学到的东西更多
- 自发的去学东西往往比老师逼你学学到更多东西
- 跟人讨论往往比被输入能得到更多东西
- 还有就是我是一线工程师，平时时间有点窘迫

上课时间怎么安排？

- 我们尽量放在一天，希望大家能全天投入
- 上课时间大家可以一起商量，比如放假之类的
- 正课2小时，答疑1小时，优先答疑上周内容

课程有多难？

- 我们尝试把2-3年所学在三个月过一遍
- 有大量东西需要自己去文章、文档和实践
- 我很喜欢讲超纲的东西或者说最近一些有意思的技术

希望大家有所收获！

市场价算法

Level3：大公司胜人者

能独立负责一个产品并利用技术降低其成本

$60w * 2 = 120w$

Level2：大公司熟练工

能独立完成项目并利用技术降低其成本

$30w * 2 = 60w$

Level1：大公司毕业生

输入输出明确，可完成任务

30w (来自offershow)



前端测试

填鸭式 → 测试驱动

测试驱动强制要求你：

- 单元尽量解耦，否则单元不可测
- 开发前先设计接口，再实现细节
- 便于回归和内部代码重构

Node里面的测试库

- 请打开fork exercise1
- 按照exercrise1要求安装mocha
- 尝试跑mocha

如果事先没做练习的话

Array

#indexOf()

1) should return -1 when the value is not present

assert

2) a和b应当深度相等

3) 可以捕获并验证函数fn的错误

0 passing (14ms)

3 failing

1) Array #indexOf() should return -1 when the value is not present:

AssertionError: -1 == [1, 2, 3]

at Context.<anonymous> (test/test.js:6:14)

2) assert a和b应当深度相等:

AssertionError: { c: { e: 1 } } == { c: { e: 1 } }

+ expected - actual

at Context.<anonymous> (test/test.js:24:12)

3) assert 可以捕获并验证函数fn的错误:

ReferenceError: xxx is not defined

at fn (test/test.js:29:7)

at Context.<anonymous> (test/test.js:32:5)

我们先一起看看 assert 有啥东西

`assert.ok(value[, message])`

```
1  const assert = require('assert');
2
3  assert.ok(true);
4  // 测试通过。
5  assert.ok(1);
6  // 测试通过。
7  assert.ok(false);
8  // 抛出 "AssertionError: false == true"
9  assert.ok(0);
10 // 抛出 "AssertionError: 0 == true"
11 assert.ok(false, '不是真值');
12 // 抛出 "AssertionError: 不是真值"
```

我们先一起看看 assert 有啥东西

`assert.equal(actual, expected[, message])`

```
1  const assert = require('assert');
2
3  assert.equal(1, 1);
4  // 测试通过, 1 == 1。
5  assert.equal(1, '1');
6  // 测试通过, 1 == '1'。
7
8  assert.equal(1, 2);
9  // 抛出 AssertionError: 1 == 2
10 assert.equal({ a: { b: 1 } }, { a: { b: 1 } });
11 // 抛出 AssertionError: { a: { b: 1 } } == { a: { b: 1 } }
```

我们先一起看看 assert 有啥东西

`assert.strictEqual(actual, expected[, message])`

```
1  const assert = require('assert');
2
3  assert.strictEqual(1, 2);
4  // 抛出 AssertionError: 1 === 2
5
6  assert.strictEqual(1, 1);
7  // 测试通过。
8
9  assert.strictEqual(1, '1');
10 // 抛出 AssertionError: 1 === '1'
```

我们先一起看看 assert 有啥东西

`assert.throws(block[, error][, message])`

```
1  assert.throws(  
2    () => {  
3      throw new Error('错误信息');  
4    },  
5    Error  
6  );
```

其实assert也就这么多要记住而已

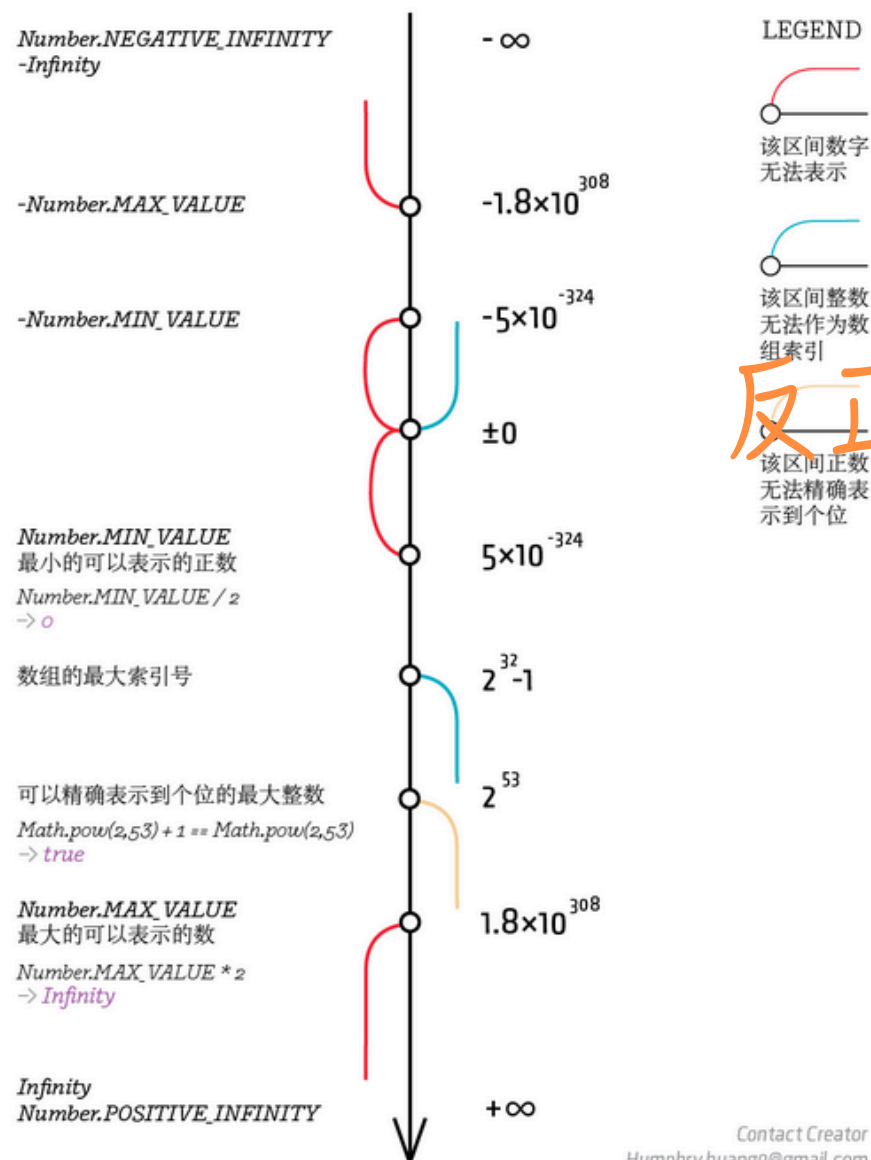
OK, 给没有事先完成的同学
10分钟做一下!

大数🌲相加

- 请打开fork exercise2
- 按照exercrise2要求完成题目

JavaScript Number (伪)数轴

JavaScript 采用 IEEE 754 标准中的浮点数算法来表示数字 Number。
这里展示了一系列的表示上下限。



反正就是大数 🌲 不精准

OK, 给没有事先完成的同学
10分钟做一下!

大家觉得TDD和BDD的差别是什么？

找一找，刚刚assert里面的方法should.js里怎么写？

找一找，刚刚assert里面的方法should.js里怎么写？

别人问我源代码怎么看？

- 比如先想想should.js的API要是我，我怎么实现
- 看看别人怎么实现的？
- 我的实现好，还是别人的实现好？

Travis CI 持续集成

- 持续集成的理论在这里不讲，大家自己查资料
- 我们在这里主要说Travis CI怎么用

跑🏃在浏览器里

- 请打开fork exercise3
- 按照exercrise3要求完成题目

OK, 给没有事先完成的同学
10分钟做一下!



IT前沿技术在线大学