IT前沿技术在线大学

www.zhenchuanx.com

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execrise1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])

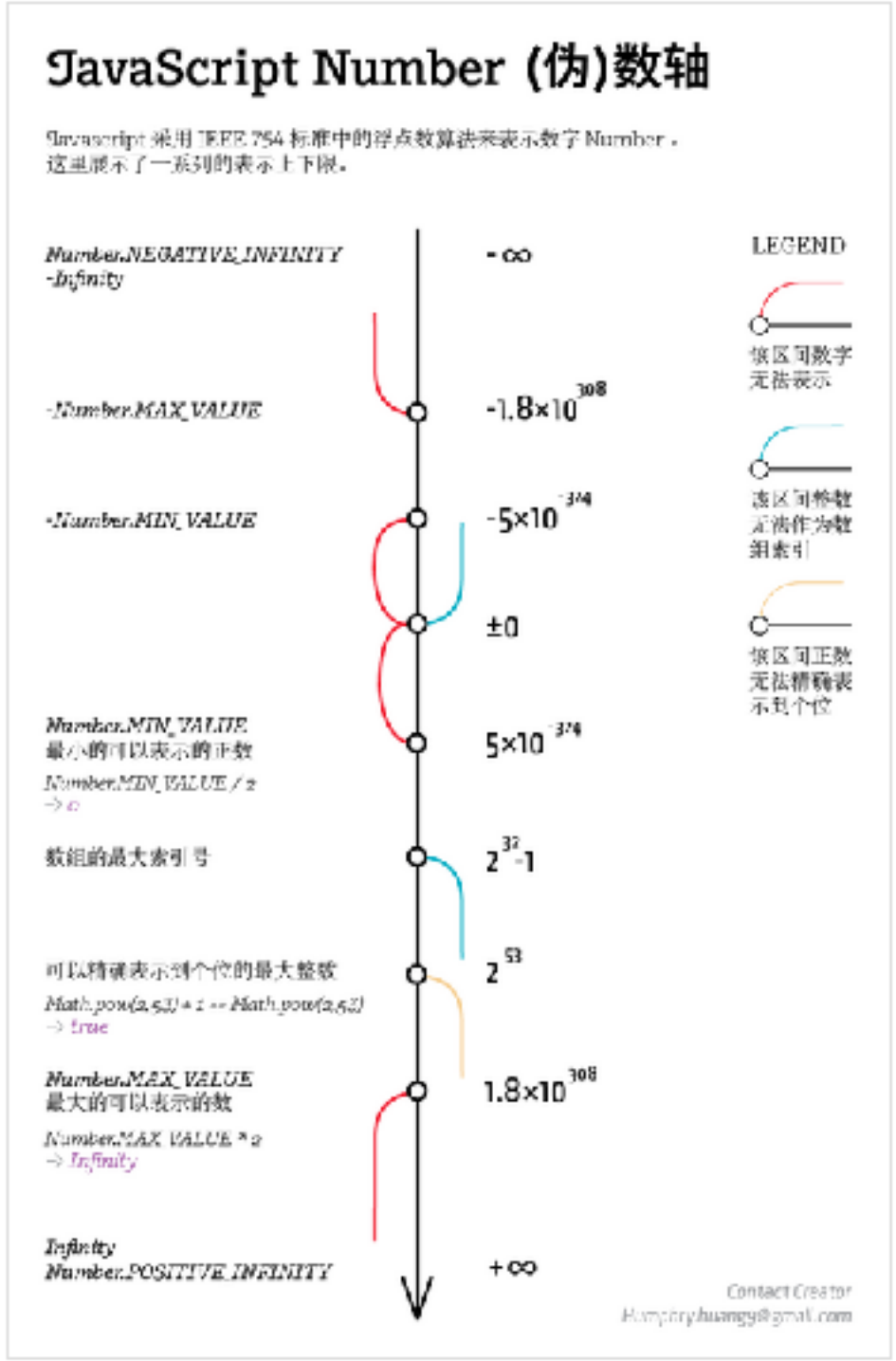
```
assert.throws(
    () => {
      throw new Error('错误信息');
    },
    Error
);
```

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execrise2要求完成题目



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execrise3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