

# JavaScript

## 生僻知识点

### 变量

- 变量提升 更多请参考[深入理解JavaScript的作用域与变量提升（hoisting）](#)
- "use strict"
- 

### isNaN

参考博客

```
isNaN(NaN);      // true
isNaN(undefined); // true
isNaN({});       // true

isNaN(true);     // false
isNaN(null);     // false
isNaN(37);       // false

// strings
isNaN("37");      // false: 可以被转换成数值37
isNaN("37.37");   // false: 可以被转换成数值37.37
isNaN("37,5");    // true
isNaN('123ABC');  // true: parseInt("123ABC")的结果是 123, 但是Number("123ABC")结果是 NaN
isNaN("");        // false: 空字符串被转换成0
isNaN(" ");       // false: 包含空格的字符串被转换成0

// dates
isNaN(new Date()); // false
isNaN(new Date().toString()); // true

isNaN("blabla")    // true: "blabla"不能转换成数值
                  // 转换成数值失败, 返回NaN
```

## 异步/同步

参考代码:

```
async function testAsync() {
  var i = 0;
  while (i<100){
    await sleep(1000)
    console.log('1 - testAsync inner ... ')
    i++
  }
  return '跑完啦。。。'
}

// 结合'await', 实现线程休眠
function sleep(ms) {
  return new Promise(resolve => setTimeout(resolve, ms))
}
```

```
console.log('start ----->')
testAsync().then(_re => {console.log(_re)})
console.log('end ----->')
```



不使用`setInterval`的原因:

在开发环境下，很少使用间歇调用（`setInterval`），原因是后一个间歇调用很可能在前一个间歇调用结束前启动

## 2. 字符串转数字 [JavaScript字符串转数字的5种方法及其陷阱](#)