

## try catch

```
try{
  console.log('a');//执行
  console.log(b);//不执行
  console.log('c');//不执行
}catch(e){ //e是封装错误名字和信息的对象
  console.log(e.message + ' ' + e.name);//如果try部分代码错误执行
}finally{
  console.log('d');//无论如何都会执行的代码
}
console.log('e');//执行
//正常执行b报错，但是在try catch中报错部分不执行跳过
```

## 错误的类型

六种：

1. **EvalError: eval()的使用和定义不一致**
2. **RangeError:数值越界**
3. **ReferenceError:非法或者不能识别的引用数值（当一个变量或函数没经声明就使用）**
4. **SyntaxError:发生语法解析错误**
5. **TypeError:操作数类型错误**
6. **URIError:URI处理函数使用不当**

## es5严格模式

正常情况下是基于es3.0而使用es5.0

es5.0相比es3.0有一部分新增也有一部分冲突，

冲突部分就使用es3.0 的方法

//启动es5.0严格模式

"use strict" //写在最上面影响全局，但是推荐写在局部

//如果使用函数方法那么如果在老版本浏览器代码将不能执行

```
function test(){
  console.log(arguments.callee);
  //正常情况下可执行，打印函数整体，但是在es5.0不能执行
}
```

# with

改变作用域链作用域链的最顶端变成with括号里的对象

**但是在es5.0中就不能使用**

```
var obj = {  
  name : 'obj'  
}  
var name = 'window';  
function test (){  
  var name = 'scope';  
  with(obj){  
    console.log(name);  
  }  
}  
test();  
//obj 也就是说打印name先在obj中找，然后在test最后在window  
with(document){  
  write("asdf"); //通过此方式也可以实现document.write  
}
```

**es5.0严格模式必须声明才能使用变量**

**局部的this必须赋值，被赋什么就是什么，不赋就没有，拒绝重复属性和参数**

# eval()

eval() 能将字符串转化为代码来使用

**es3.0不能使用**

```
eval('console.log(123)');
```