

`obj.__proto__ === Fn.prototype;`

原型链一定是找实例化对象的，构造函数下的原型；

`Fn -> Function` 实例化对象；

除了`Function` 别的实例化对象前面`new`都报错！

面向对象最擅长复杂的东西简单化，越复杂越能感受到他的便利；

写法极为优雅，条理清晰，藕合度不高（组合度较高）；

`let`:变量：

1. 声明变量的时候**不会在**`window`下注册这个变量；
 2. 暂存死区（**没有预解析，先声明在使用，不然会报错**）；
 3. 支持块级作用域{}；
-

面向对象写东西的方法步骤：

1. 把变量提取出来；
 2. 拆`function`；
 3. 变量变属性，函数变方法；
 4. **修正this**。指向问题（99%死在`this`上）
-

包装对象：

当简单类型去调用某个属性或者方法的时候，那么系统会偷偷的在内部将简单类型的系统对象的属性或者方法调用出来使用以后，这个对象会被自动销毁；

注意：

包装对象的属性只能读不能写；

`toString`

把当前的数据转成字符串的方法，每种属性类型都有`toString`方法；当你使用了`alert()`这个系统方法，那么会自动会调用这个数据的`toString`

`call`和`apply`

当声明一个函数的时候，就会有一些方法或者属性；

其中有两个方法，一个叫`call`， 一个叫`apply`；

`call`：

修改`this`的指向问题

第一个参数：

修改后的`this`；

第二个参数开始：

实参（可以有若干个）

`apply`，：

修改`this`的指向问题，只有两个参数

第一个参数:

修改后的this;

第二个参数开始:

数组 (将实参放到数组中)