

什么是原型？

原型是JavaScript中的继承的基础，JavaScript的继承就是基于原型的继承。

函数的原型对象？

在JavaScript中，我们创建一个函数A(就是声明一个函数)，那么浏览器就会在内存中创建一个对象B，而且每个函数都默认会有一个属性 `prototype` 指向了这个对象（即：`prototype`的属性的值是这个对象）。这个对象B就是函数A的原型对象，简称函数的原型。这个原型对象B 默认会有一个属性 `constructor` 指向了这个函数A（意思就是说：`constructor`属性的值是函数A）。

`hasOwnProperty`方法，可以判断一个属性是否来自对象本身。

`in`操作符用来判断一个属性是否存在于这个对象中（换句话说，只要对象和原型中有一个地方存在这个属性，就返回true）

// 动态原型模式创造对象

//构造方法内部封装属性

```
function Person(name, age) {
```

```
    //每个对象都添加自己的属性
```

```
    this.name = name;
```

```
    this.age = age;
```

```
    /*
```

判断this.eat这个属性是不是function，如果不是function则证明是第一次创建对象，

则把这个function添加到原型中。

如果是function，则代表原型中已经有了这个方法，则不需要再添加。

perfect！完美解决了性能和代码的封装问题。

```
    */
```

```
    if(typeof this.eat !== "function"){
```

```
        Person.prototype.eat = function () {
```

```
            alert(this.name + " 在吃");
```

```
        }
```

```
    }
```

```
}
```

```
var p1 = new Person("志玲", 40);
```

```
p1.eat();
```