

第十二课 继承

学习目录

- 什么是继承
- 构造函数进阶应用
- 继承的方式

一 . 什么是继承

许多支持面向对象编程的语言都支持继承，继承分别有两种方式，一种是接口继承，另一种是实现继承，在 js 中只有实现继承，我们主要讨论的也是实现继承，实现继承的方式主要是依靠原型链来做到的。

其实继承简单来理解就是利用原型链让对象 a 拥有对象 b 的属性和方法。

```
var obj1 = {  
    a:100,  
    getNum:function(){  
        console.log(this.a);  
    }  
};  
  
var obj2 = obj1;  
  
function Fn1(){  
    this.num = 100;
```

```
}  
  
Fn1.prototype.getNum = function(){  
  
    console.log(this.num);  
  
}
```

```
var f1 = new Fn1();  
  
f1.getNum();
```

```
function Fn2(){  
  
    this.job = 'web 培训'  
  
}
```

```
Fn2.prototype = new Fn1();  
  
Fn2.prototype.getJob = function(){  
  
    console.log(this.job);  
  
}
```

```
var f2 = new Fn2();  
  
f2.getNum();  
  
f2.getJob();
```

1. 每个构造函数都有一个原型对象 Fn1.prototype。
2. 每个原型对象都包含一个指向该构造函数的指针 Fn1.prototype.constructor。
3. 每个 new 出来的实例都包含一个指向原型对象的隐式原型 f1.__proto__



学习前端，最快的进步是持续！

如果让一个构造函数的原型对象等于另一个构造函数的实例，根据原型链的这种关系，那么这个构造函数的 `Fn2.prototype = new Fn1()`，`var f1 = new Fn1()` 实际上是 `Fn2.prototype` 根据 `__proto__` 找到了 `f1.__proto__`，`f1.__proto__` 又找到了 `Fn1.prototype`，也就是 `Fn2.prototype.__proto__` 通过 `f1.__proto__` 隐式原型拥有了 `Fn1.prototype` 的属性和方法。

```
console.log(Fn1.prototype == f1.__proto__);//true
```

```
console.log(Fn2.prototype.__proto__ == f1.__proto__);//true
```

```
console.log(Fn2.prototype.__proto__ == Fn1.prototype);//true
```

谢谢观看！

我是星星课堂老师：周小周