

公有属性和公有方法

1. 公有属性和公有方法

```
function User(name,age){
    this.name = name;//公有属性
    this.age = age;
}
User.prototype.getName = function()//公有方法
    return this.name;
}
var user = new User('fire子海',26);
console.log(user.getName());//output:fire子海
```

2. 私有属性和方法

```
function User(name,age){
    var name = name;//私有属性
    var age = age;
    function alertAge()//私有方法
        alert(age);
    }
    alertAge(age); //弹出26
}
var user = new User('fire子海',26);
```

3. 静态属性和方法

在php中，无需实例化就可以调用的方法就叫静态方法，js也一样，无需实例化，即用new操作符实例化对象，就可调用对象的方法和属性。

```
function User(){}
User.age = 26;//静态属性
User.myname = 'fire子海';
User.getName =function()//静态方法
    return this.myname;//如果这里使用this.name，返回的将是User，所有改用了myname，
}
console.log(User.getName());//output:fire子海
```

4. 特权方法

```
function User(name,age){
    var name = name;//私有属性
    var age = age;
    this.getName = function() { //特权方法
        return name;//私有属性和方法不能使用this调用
    }
}
```

```

}
var user = new User('fire子海',26);
console.log(user.getName());//output:fire子海

```

5. 静态类

对于静态方法和静态属性，我们无需像第三步中那样去创建，如果网友看过我那篇“js如何制作图片轮播”，就知道可以使用字面量的方式来创建。

```

var user = {
  init:function(name,age){
    this.name = name;
    this.age = age;
  },
  getName:function(){
    return this.name;
  }
}
user.init('fire子海',26);
console.log(user.getName());//output:fire子海

```

6. 公有方法的调用规则

调用公有方法，我们必需先实例化对象

公有方法中通过不this调用公有属性和特权方法，不能使用this调用静态方法和属性，必需裁通过对象本身调用，即对象名。公有方法也不能调用私有方法

```

function User(){
  this.myname = 'fire子海';//公有属性
  this.age = 26;
  this.do = function(){//特权方法
    return this.myname+'学习js';
  }
}
User.eat = function(food){
  return '晚餐只有'+food;
}
User.prototype.alertAge = function(){
  alert(this.age);
}
User.prototype.alertDo = function(){
  alert(this.do());//调用特权方法
}
User.prototype.alertEat = function(food){
  alert(User.eat(food));//只能通过对象本身调用静态方法
  //alert(this.eat(food))这样调用将出错:this.eat is not a function
}
var user = new User();

```

```
user.alertAge();//alert:26
user.alertDo();//alert:fire子海学习js
user.alertEat('方便面')//alert:晚餐只有方便面
```

7. 静态方法的调用规则

使用静态方法时，无需实例化对象，便可以调用，对象实例不能调用对象的静态方法，只能调用实例自身的静态属性和方法

```
function User(){}
User.age = 26;//静态属性
User.myname = 'fire子海';
User.getName = function() {//静态方法
    return this.myname;
}
var user = new User();
console.log(user.getName);//TypeError: user.getName is not a function
user.supper = '方便面';
user.eat = function(){
    return '晚餐只有'+this.supper;
}
user.eat();//晚餐只有方便面
```

静态法无法调用公有属性、公有方法、私有方法、私有属性、特权方法和原型属性

```
function User(){
    this.myname = 'fire子海';//公有属性
    this.age = 26;
    this.do = function() {//特权方法
        return this.myname+'学习js';
    }
}
User.prototype.alertAge = function() {//公共方法，也叫原型方法
    alert(this.age);
}
User.prototype.sex = '男';//原型属性
User.getName = function() {//静态方法
    return this.myname;
}
User.getAge = function(){
    this.alertAge();
}
User.getDo = function(){
    return this.do();
}
//console.log(User.getName())//undefined
//console.log(User.getDo())//TypeError: this.do is not a function
```

```
//console.log(User.getAge())//TypeError: this.alertAge is not a function
```

8. 特权方法的调用规则

特权方法通过this调用公有方法、公有属性，通过对象本身调用静态方法和属性，在方法体内直接调用私有属性和私有方法

```
function User(girlfriend){
    var girlfriend = girlfriend;
    function getGirlFriend(){
        return '我女朋友'+girlfriend+'是美女! ';
    }
    this.myname = 'fire子海';//公有属性
    this.age = 26;
    this.do = function(){//特权方法
        return this.myname+'学习js';
    }
    this.alertAge = function(){
        this.changeAge();//特权方法调用公有方法
        alert(this.age);
    }
    this.alertGirlFriend = function(){
        alert(getGirlFriend());//调用私有方法
    }
}
User.prototype.changeAge = function(){
    this.age = 29;
}
var user = new User('某某');
user.alertAge();//alert:29
user.alertGirlFriend();//alert:我的女朋友某某是美女!
```

9. 私有方法

对象的私有方法和属性,外部是不可以访问的,在方法的内部不是能this调用对象的公有方法、公有属性、特权方法的

```
function User(girlfriend){
    var girlfriend = girlfriend;
    this.myname = 'fire子海';//公有属性
    this.age = 26;
    function getGirlFriend(){
        //this.myname ;//此时的this指向的window对象，并非User对象，
        // this.myname = 'fire子海',此时的this指向的是getGirlFriend对象了。
        //如果通过this调用了getGirlFriend中不存在的方法呀属性，this便会指向window 对象，只有this调用了getGirlFriend存在的方法和属性，this才会指定getGirlFriend;
        alert(User.eat('泡面'));//alert: 晚餐只有方便面
    }
}
```

```
this.do = function(){//特权方法
    return this.myname+'学习js';
}
this.alertAge = function(){
    this.changeAge();//特权方法调用公有方法
    alert(this.age);
}
this.alertGirlFriend = function(){
    getGirlFriend();//调用私有方法
}
}
User.eat = function(supper){
    return '晚餐只有'+supper;
}
var user = new User('某某');
user.alertGirlFriend();
```