0.回顾

BOM == js将Browser理解成对象 = 浏览器对象模型

DOM == js将HTML文档理解成对象 = 文档对象模型

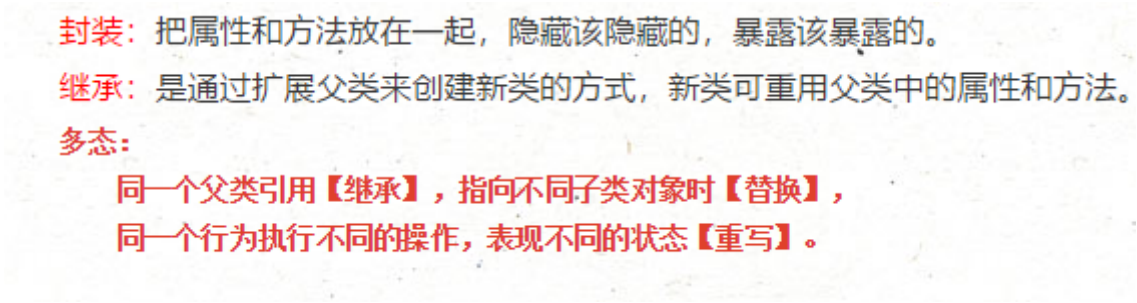
core-dom ： 节点的导航

html-dom： 节点的CRUD 【重点和难点】

css-dom ： style和className

1.JS对象【JS是基于对象，不是面向对象】

1.1.Java对象【各编程语言的面向对象的三大特征】



面向过程：没有封装的概念，方法归方法，属性作为全局变量

面向对象：属性和方法放在类里，方法可以直接方问该类中的相关属性

方法访问其它类的属性，需要实例化。

在现实世界中，万物皆对象。

继承：出于蓝，胜于蓝部分（重写）

Son extends Father，重写biz()方法

Daught extends Father，重写biz()方法

父.biz()　　开个线下便利店

子.biz()重写　开个网上商城

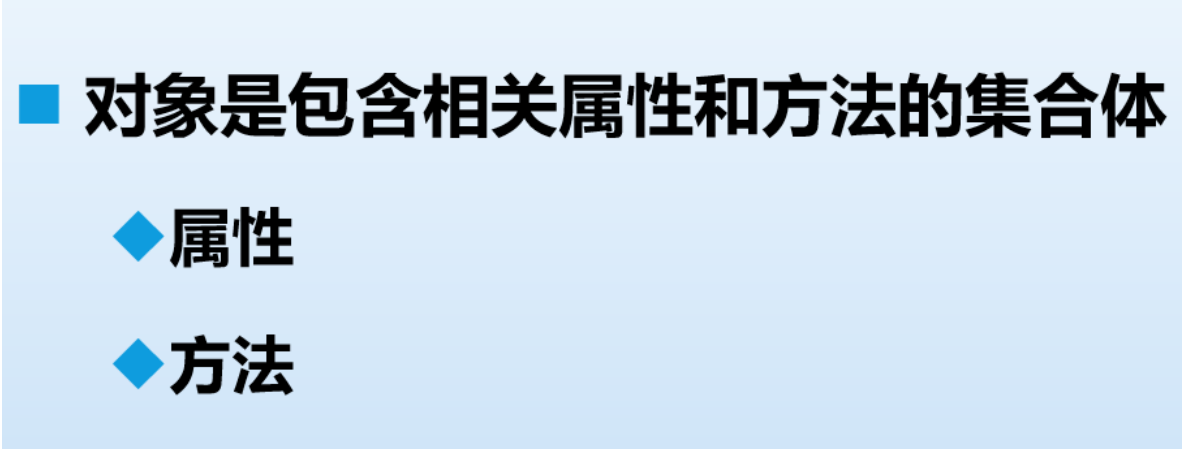
女.biz()重写　开个服装店

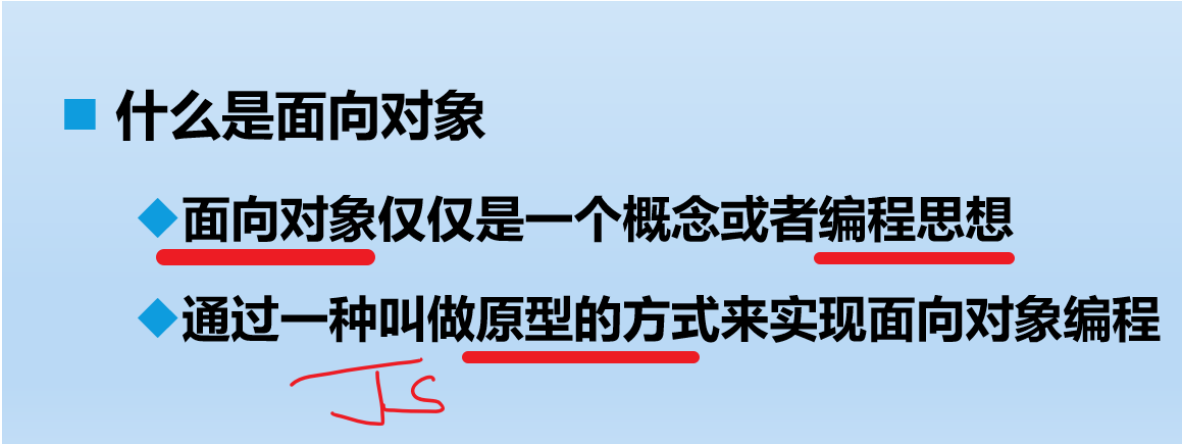
多态：多个子类【继承】，子类必须有重写相关方法【重写】，变量指向子类对象

```
Father s1 = null;
s1 = new Father();
s1.biz();  // ??
```

```
s1 = new Son();
s1.biz();   // ??

s1= new Daught ();
s1.biz(); // ?
```

1.2.JS对象概念：JS是基于对象

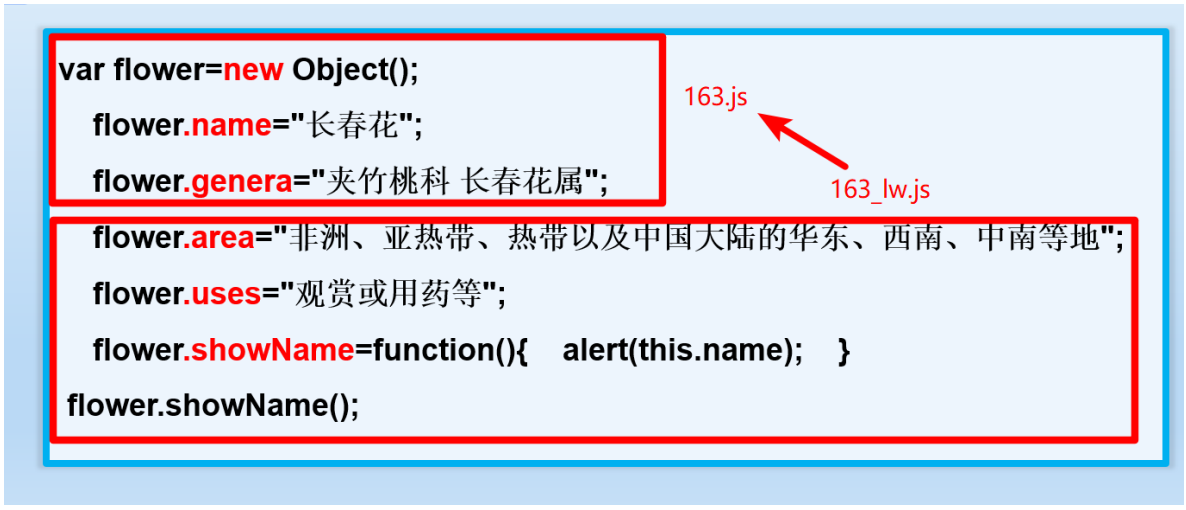




1.3.基于Object创建一个对象【优点：容易扩展】

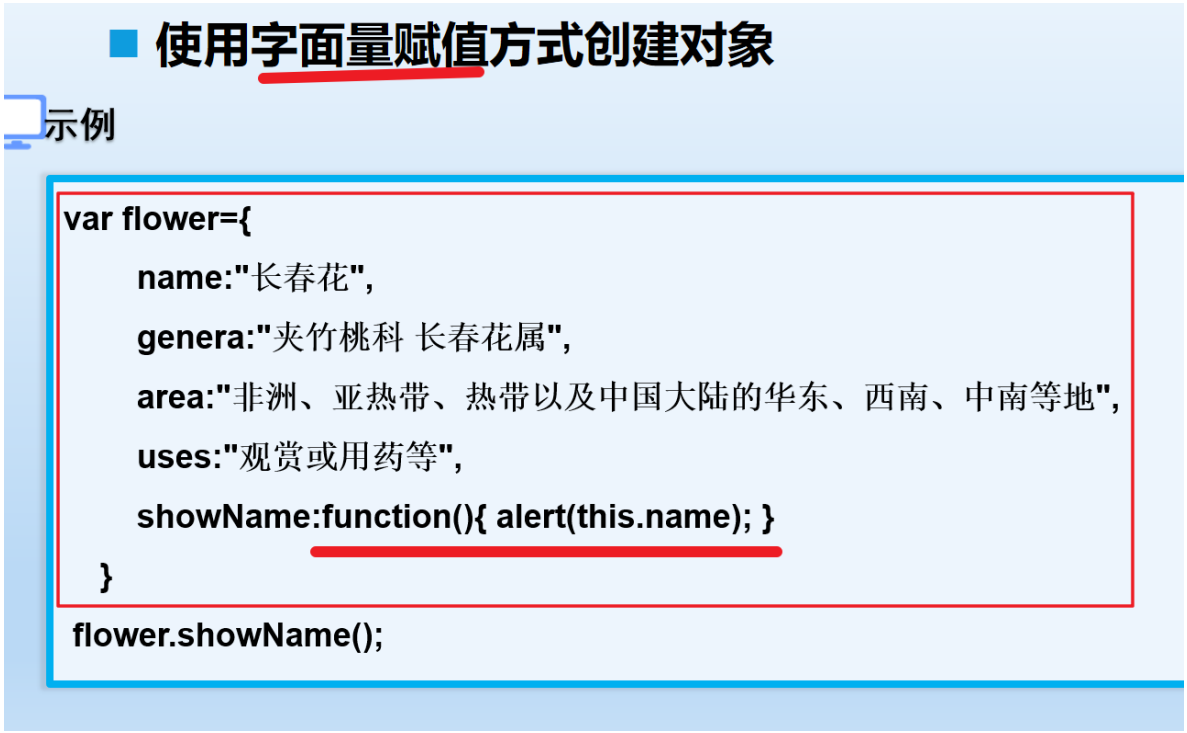
【代码不可重用】

l63_lw.js是对163.js框架的一个扩展



1.4.基于字面量赋值方式创建一个对象

【代码不可重用】



1.5.常见的内置对象

2.j04_a01_创建person对象

3.JS对象2【JS构造函数=Java的类】

3.0.Java的构造函数

```
public class Father{

    private int age;
```

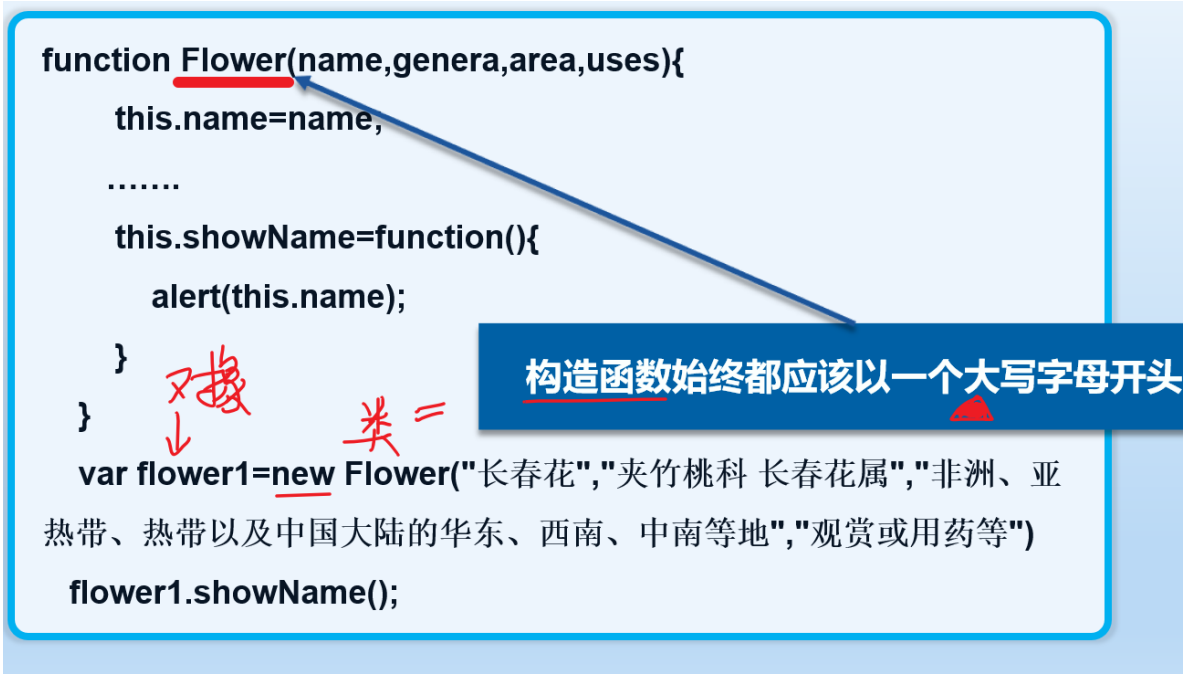
```
    public Father(int age){
        this.age = age;
    }

}


public class Entry{

    public static void main(String[] args){

        Father f1 = new Father(58);
        // (1) new ：代表在heap堆中开辟一个对象空间
        // (2) Father ：对象空间的规格大小，通过Father类计算
        // (3) (58)  ： 调用并执行Father类的构造函数，该构造函数只能一个整型参数

        // 最后，将该空间的地址引用赋值给f1变量，f1是存放在栈中

    }

}
```

3.1.定义构造函数，再用构造函数创建对象

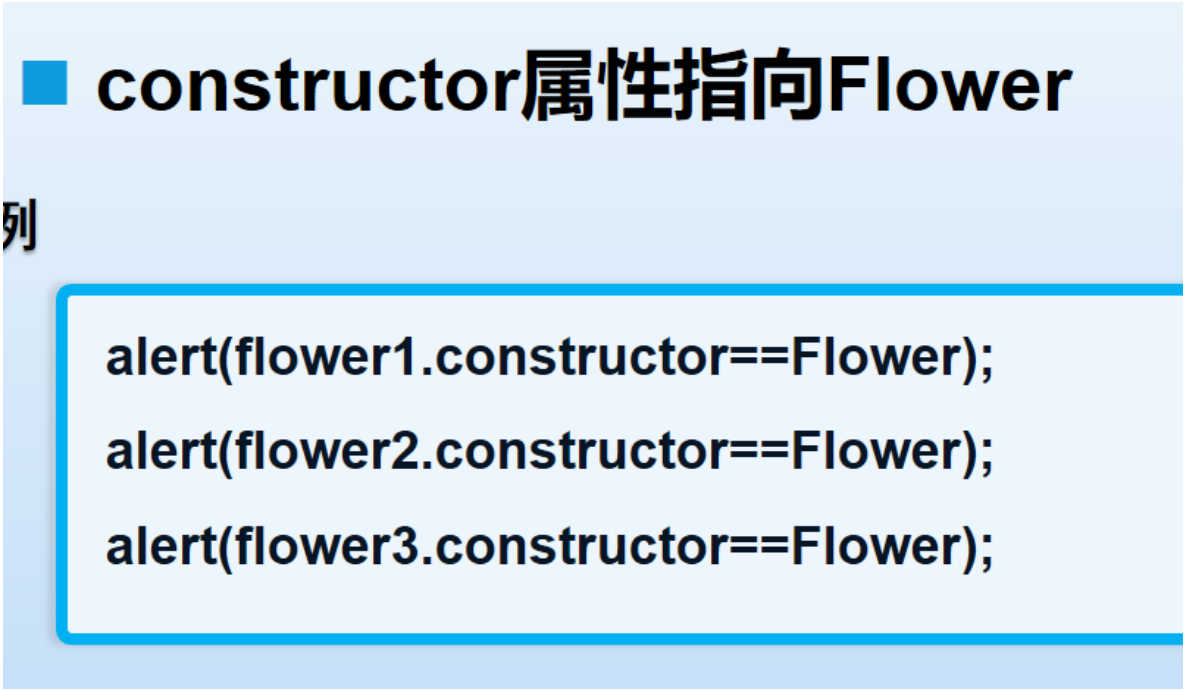
【JS构造函数 == Java的类 】

优点：【代码可重用】

缺点：【代码不容易扩展——添加属性或方法只能到定义的地方】



为什么JS中称为 构造函数创建对象？原因：flower1.constructor属性指向构造函数名



3.2.原型prototype

【空构造函数，配合prototype属性，定义属性或方法】

优点：【代码可重用】

优点：【代码容易扩展——添加属性或方法可以在另一个js中】

- 每个函数都有一个prototype属性，这个属性是一个指针，指向一个对象
- prototype就是通过调用构造函数而创建的那个对象实例的原型对象

- 每个函数都有一个prototype属性，这个属性是一个指针，指向一个对象
- prototype就是通过调用构造函数而创建的那个对象实例的原型对象

var flower1 = new Flower(...);

flower1.prototype == Flower.prototype属性

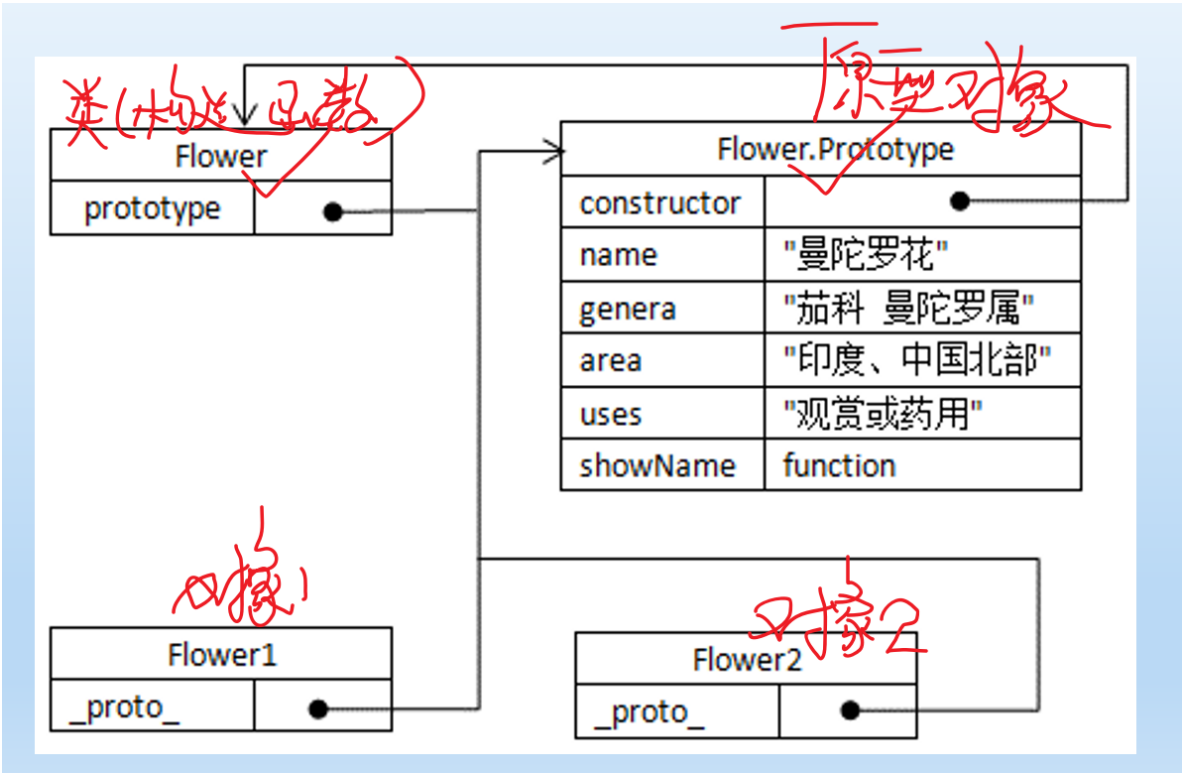
```
function Flower(){

}
Flower.prototype.name="曼陀罗花";
Flower.prototype.genera="茄科 曼陀罗属";
Flower.prototype.area="印度、中国北部";
Flower.prototype.uses="观赏或药用";
Flower.prototype.showName=function() {
    alert(this.name);
}
var flower1=new Flower();
flower1.showName();
var flower2=new Flower();
flower2.showName();
alert(flower1.showName==flower2.showName);
```

prototype方式容易扩展

```
function Flower(){

}
Flower.prototype.name="曼陀罗花";
Flower.prototype.genera="茄科 曼陀罗属";
Flower.prototype.area="印度、中国北部";
Flower.prototype.uses="观赏或药用";
Flower.prototype.showName=function() {
    alert(this.name);
}
var flower1=new Flower();
flower1.showName();
var flower2=new Flower();
flower2.showName();
alert(flower1.showName==flower2.showName);
```

163.js

163_lw.js



4.j04_a02_创建Person函数

【构造函数和原型方式实现】

参看本文的相关代码