# Ch04 JavaScript对象及初识面向对象

## 练习题

#### —、选择题

1、阅读下面的代码，在横线处填写（ A、C ）可输出的值为"小花猫"。（选择两项）

var Animal=new Object();

Animal.name="小花猫";

Animal.age=3;

Animal.showName=function(){

return this.name;

}

alert (\_\_\_\_\_\_\_\_\_\_\_\_);

A. Animal.showName() B. Animal. showName C. Animal. name D. Animal. name ()

2、Array对象的（ B ）方法可以向数组末尾添加一个元素。

A. match () B. push() C. call() D. apply ()

3、有两个构造函数Andutomobile和Car，下面（ A ）能够实现Car继承Automobile的方法。

A. Car.prototype=new Automobile();

B. Car.\_proto\_=new Automobile ();

C. var Car= new Automobile();

D. var carl = new Car();

4、JavaScript的面向对象是使用（ A ）体现继承关系的。

A. prototype B. \_proto\_ C. constructor D. call

5、在JavaScript 中有如下代码，alert(dog1.showFoot())和 alert(dog1.name)分别输出（ C ）。

function Animal(foot, name){

this.foot=4;

this.name="动物";

}

Animal.prototype.showFoot=function(){

return this.foot;

};

function Dog(foot,name){

Animal.call(this,foot,name);

}；

Dog.prototype=new Animal();

var dogl=new Dog (3,"大黄");

alert(dogl.showFoot ());

alert(dogl.name);

A. 4 大黄 B. 3 大黄 C. 4 动物 D. 3 动物

#### 二、简答题

1、简述使用原型链实现继承的思想。

【参考答案】回答要点：

1. 原型对象都包含一个指向构造函数的指针（constructor）；
2. 原型对象X.prototype将包含一个指向原型Y.prototype的指针；
3. 一个原型实际上可以变成另一个类型的实例；
4. 各个原型之间的层层递进，就实现了继承；

2、简述借用构造函数技术在实现继承的过程中的作用。

【参考答案】回答要点：

在子类型构造函数的内部调用父类型构造函数，即在子类型构造函数的内部通过apply()或call( )方法调用父类型的构造函数，也可以在将来新创建的对象上执行构造函数。

3、创建一个对象student，表示一个学生对象，要求如下：

* 添加属性：姓名、年龄和自我介绍。
* 添加方法：输出该学生的姓名、年龄和自我介绍。
* 在页上显示该方法输出的信息，如图5-13所示。

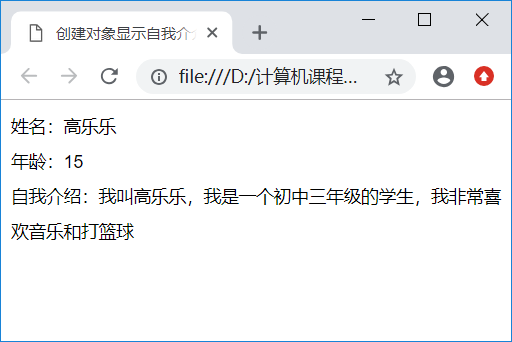
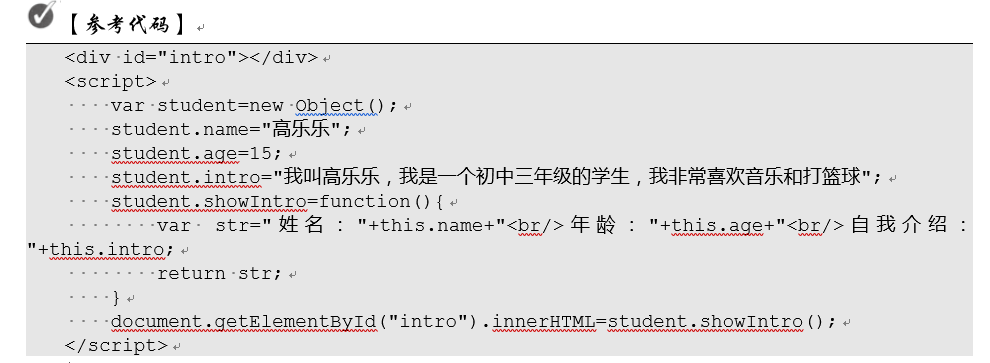


图5-13 显示学生的信息



* 使用new创建对象student，添加属性。
* 添加方法，在方法中使用字符串把学生的信息使用字符“+”串起来；使用return返回该字符串。
* 使用getElementById()和innerHTML把方法中的字符串显示在页面中。



4、创建一个构造函数Student，表示学生，要求如下：

* 添加属性：姓名、年龄和自我介绍。
* 添加方法：在一个P标签中显示该姓名、年龄和自我介绍。
* 创建两个学生对象，输入两个学生的姓名、年龄和自我介绍。
* 在页面中显示两个对象的方法输出的信息，如图5-14所示。



* 创建构造函数Student,添加属性和方法，在方法中使用createElement()创建p标签。使用innerHTML把姓名、年龄和自我介绍显示在此p标签中，使用return返回p标签。
* 使用.new创建两个学生对象。
* 使用getElementById()和appendChild()把创建的学生对象信息追加到页面中。

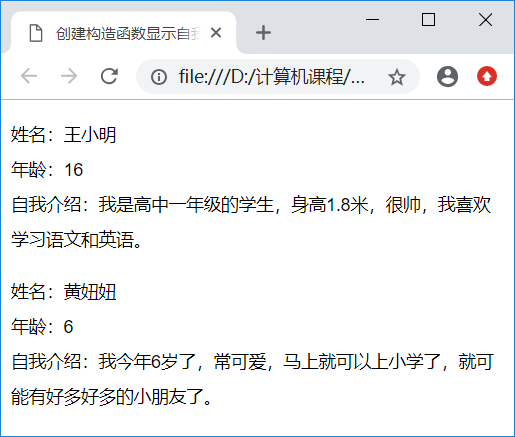
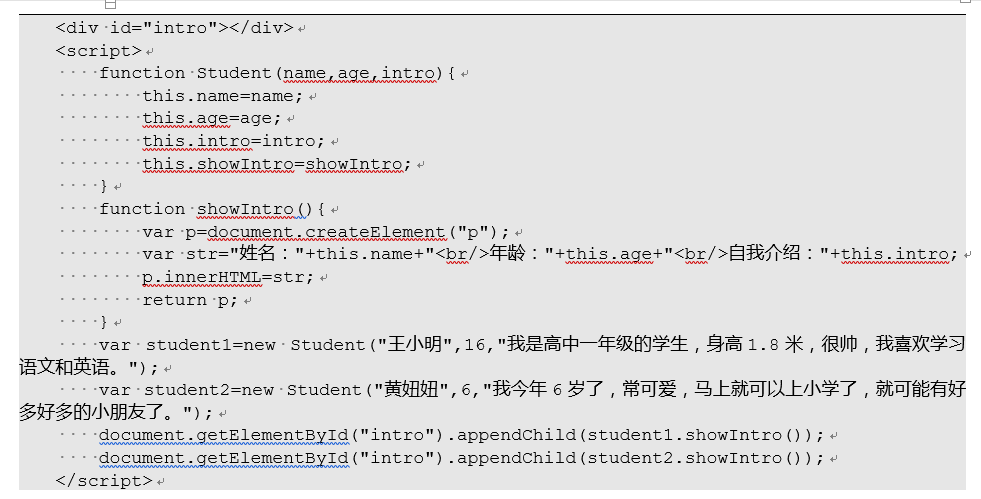


图5-14 显示两个学生的信息



5、创建一个构造函数Animal，使用继承完成如下要求：

* 构造函数Animal有自己的属性：名称（name)、年龄（age)和颜色（color)。
* 创建构造函数Poultry，继承Animal的属性，并且有属于自己的属性：腿（leg)。
* 创建Poultry的方法info,输出名称、年龄、颜色和腿等信息。
* 创建对象，把方法中的相关信息输出到页面中，如图5-15所示。

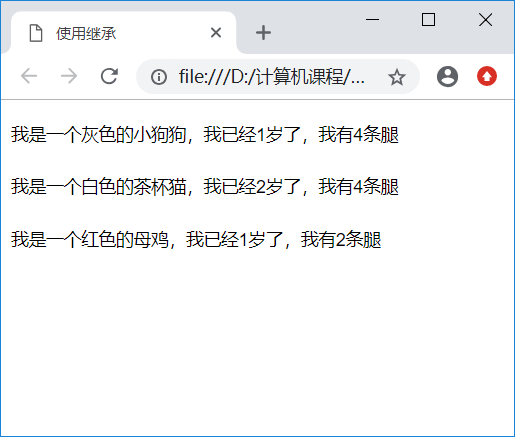


图5-15 显示对象信息



* 创建构造函数Animal，添加自己的属性。
* 创建构造函数Poultry，使用call()继承Animal的属性，并添加自己的属性。
* 创建方法info，使用createElement()创建p标签，使用innerHTML把名称、年龄、颜色和腿数显示在此p标签中，使用return返回p标签。
* 使用new创建两个对象。
* 使用getElementById()和appendChild()把创建的对象信息追加到页面中。

