**题目**

**第一题：使用面向对象思想，创建任意5个类，并生成对应的对象**

**作业目标及能力提升**

1. 掌握JavaScript之面向对象
2. 掌握JavaScript之类的封装
3. 掌握JavaScript之对象的创建

**解题思路**

1. 通过构造函数创建对象，并添加对应的属性
2. 在原型上添加方法
3. 通过new创建对象

* 参考答案

首先创建HTML文件，创建script标签，通过构造函数创建对象，并添加对应的属性，在原型上添加方法，通过new创建对象。实现此作业需要按照如下步骤进行。

步骤一：新建HTML文件

首先新建一个名为index的html文件

代码如下所示：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

</body>

</html>

步骤二：编写js代码，并生成类

创建script标签，使用构造函数，创建类。代码如下所示：

function Family() {

}

步骤三：添加属性

在构造方法内，添加属性，并为属性赋值。代码如下所示：

function Family(father, mother) {

this.father = father;

this.mother = mother;

};

步骤四：添加方法

在原型上添加方法，实现开party的方法。代码如下所示：

Family.prototype.party = function() {

console.log(this.father + "和" + this.mother + "开派对");

};

步骤五：完成其他类

根据上面的步骤，再完成其他几个类。代码如下所示：

Family.prototype.party = function() {

console.log(this.father + "和" + this.mother + "开派对");

};;

var family = new Family("李雷", "韩梅梅");

family.party();

//车类

function Car(color, length) {

this.color = color;

this.length = length;

}

Car.prototype.run = function() {

console.log(this.color + "的汽车在跑");

}

var car = new Car("红色", "9");

car.run();

//人类

function Person(name, gender) {

this.name = name;

this.gender = gender;

}

Person.prototype.say = function() {

console.log("大家好，我叫" + this.name);

}

var p = new Person("凯撒", "男");

p.say();

//猫

function Cat(name) {

this.name = name;

}

Cat.prototype.makeSound = function() {

console.log(this.name + "在喵喵喵");

};

var cat = new Cat("咪咪");

cat.makeSound();

//苹果

function Apple(type, color) {

this.type = type;

this.color = color;

}

Apple.prototype.grow = function() {

console.log(this.type + "的苹果慢慢长大");

};

var apple = new Apple("红富士", "红色");

apple.grow();

然后运行HTML文件, 可以在终端里看到结果

本案例的完整代码如下所示：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

<script type="text/javascript">

//家庭类

function Family(father, mother) {

this.father = father;

this.mother = mother;

};

Family.prototype.party = function() {

console.log(this.father + "和" + this.mother + "开派对");

};;

var family = new Family("李雷", "韩梅梅");

family.party();

//车类

function Car(color, length) {

this.color = color;

this.length = length;

}

Car.prototype.run = function() {

console.log(this.color + "的汽车在跑");

}

var car = new Car("红色", "9");

car.run();

//人类

function Person(name, gender) {

this.name = name;

this.gender = gender;

}

Person.prototype.say = function() {

console.log("大家好，我叫" + this.name);

}

var p = new Person("凯撒", "男");

p.say();

//猫

function Cat(name) {

this.name = name;

}

Cat.prototype.makeSound = function() {

console.log(this.name + "在喵喵喵");

};

var cat = new Cat("咪咪");

cat.makeSound();

//苹果

function Apple(type, color) {

this.type = type;

this.color = color;

}

Apple.prototype.grow = function() {

console.log(this.type + "的苹果慢慢长大");

};

var apple = new Apple("红富士", "红色");

apple.grow();

</script>

</body>

</html>

**第二题：为字符串类String扩充方法，实现以下功能**

1. **统计指定字符出现的次数**
2. **比较两个字符串是否相等，不区分大小写**
3. **判断字符串是否数字串**

**作业目标及能力提升**

1. 掌握JavaScript之面向对象
2. 掌握JavaScript之原型上添加方法
3. 掌握JavaScript之字符串的处理

**解题思路**

1. 在原型上添加方法
2. 通过原有的字符串方法完成现有的功能

* 参考答案

首先创建HTML文件，创建script标签，在原型上添加方法，判断字符串是否数字串。实现此作业需要按照如下步骤进行。

步骤一：新建HTML文件

首先新建一个名为index的html文件

代码如下所示：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

</body>

</html>

步骤二：编写js代码，完成统计指定字符出现的次数

创建script标签，在String的原型上，添加Occurs方法，实现思路：以某个字符进行分割成数组，数组的元素个数 – 1 就是字符出现的次数。代码如下所示：

String.prototype.Occurs = function(ch) {

return this.split(ch).length - 1;

}

步骤三：完成比较两个字符串是否相等，不区分大小写

在String的原型上，添加equalIgnoreCase方法，实现思路：把字符串统一先变成小写，再进行比较。代码如下所示：

String.prototype.equalIgnoreCase = function(str) {

var temp1 = this.toLowerCase();

var temp2 = str.toLowerCase();

return temp1.equal(temp2);

}

步骤四：完成判断字符串是否数字串

在String的原型上，添加isAllNumber方法，实现思路：遍历字符串的每个字符，判断字符是否在‘0’-‘9’之间，如果全部字符都在这个范围内，即为数字串。代码如下所示：

String.prototype.isAllNumber = function() {

for(var i = 0; i < this.length; i++) {

if(this.charAt(i) < '0' || this.charAt(i) > '9') {

return false;

}

}

return true;

}

本案例的完整代码如下所示：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

<script type="text/javascript">

/\*\*\* 统计指定字符出现的次数 \*\*\*/

String.prototype.Occurs = function(ch) {

return this.split(ch).length - 1;

}

/\*\* 比较两个字符串是否相等，不区分大小写 \*\*/

String.prototype.equalIgnoreCase = function(str) {

var temp1 = this.toLowerCase();

var temp2 = str.toLowerCase();

return temp1.equal(temp2);

}

/\*\* 判断字符串是否数字串 \*\*/

String.prototype.isAllNumber = function() {

for(var i = 0; i < this.length; i++) {

if(this.charAt(i) < '0' || this.charAt(i) > '9') {

return false;

}

}

return true;

}

</script>

</body>

</html>