

高级1

问题1： OOP 指什么？有哪些特性

- **OOP**

Object-oriented programming的缩写，即面向对象程序设计，其中两个最重要的概念就是类和对象。类只是具备了某些功能和属性的抽象模型，而实际应用中需要一个一个实体，也就是需要对类进行实例化，类在实例化之后就是对象。

- **特性**

1. 继承性：子类自动继承其父级类中的属性和方法，并可以添加新的属性和方法或者对部分属性和方法进行重写。继承增加了代码的可重用性。
2. 多态性：子类继承了来自父级类中的属性和方法，并对其中部分方法进行重写。
3. 封装性：将一个类的使用和实现分开，只保留部分接口和方法与外部联系。

问题2： 如何通过构造函数的方式创建一个拥有属性和方法的对象？

```
function Animal(name, sound){
  this.name = name;
  this.sound = sound;
}
Animal.prototype.bark = function(){
  console.log(this.sound)
}
var cat = new Animal('mimi', 'Miaow');
cat.bark();//Miaow
```

问题3： prototype 是什么？有什么特性？

所有的构造方法都有一个prototype属性，并且可以为其添加方法。添加给构造方法的prototype属性的任何方法都可以作为该构造方法所创建的所有对象的一个方法来使用。

问题4： 画出如下代码的原型图

```
function People (name){
  this.name = name;
  this.sayName = function(){
    console.log('my name is:' + this.name);
  }
}
```

```

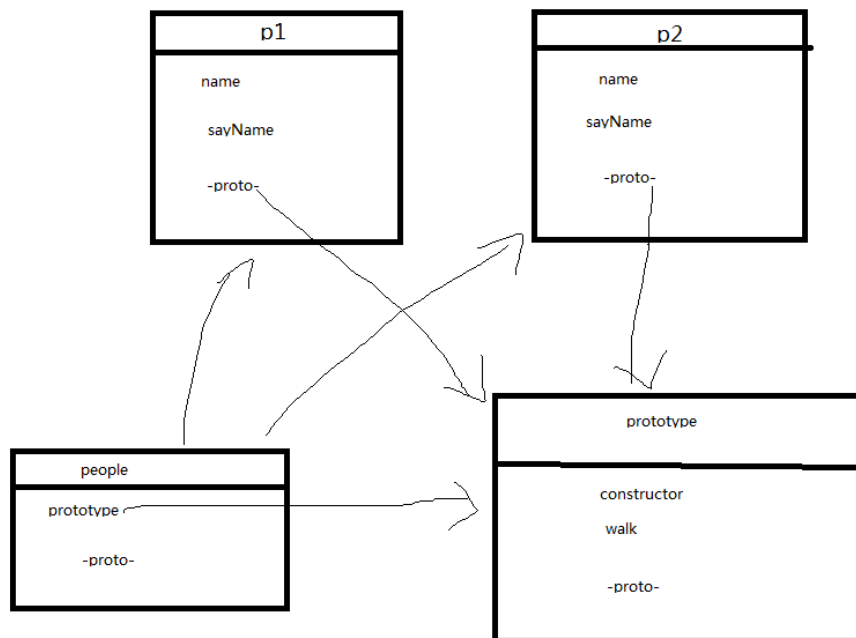
People.prototype.walk = function(){
  console.log(this.name + ' is walking');
}

```

```

var p1 = new People('饥人谷');
var p2 = new People('前端');

```



问题5：创建一个 Car 对象，拥有属性name、color、status；拥有方法run，stop，getStatus

```

function Car(name,color,status){
  this.name=name;
  this.color=color;
  this.status=status;
}
Car.prototype.run=function(){
  console.log('running');
};
Car.prototype.stop=function(){
  console.log('stop');
};
Car.prototype.getStatus=function(){
  console.log(this.status);
};
var Benz = new Car('Benz','black','running');
var BYD = new Car('BYD','red','stopped');
Benz.run(); // "running"

```

```
BYD.stop();    //"stop"  
BYD.getStatus();    //"stopped"
```

问题6：创建一个 GoTop 对象，当 new 一个 GoTop 对象则会在页面上创建一个回到顶部的元素，点击页面滚动到顶部。拥有以下属性和方法

1. `ct` 属性，GoTop 对应的 DOM 元素的容器
2. `target` 属性，GoTop 对应的 DOM 元素
3. `bindEvent` 方法，用于绑定事件
4. `createNode` 方法，用于在容器内创建节点

[代码地址](#)