

## 接口和抽象类的比较

### ➤ 概念比较

	抽象类	接口
相同点	均可以包含抽象方法	
	都不能被实例化	
	都是引用数据类型，抽象类的子类或接口的实现类都可以向上转型	
不同点	一个类只能继承一个抽象类，即单继承	一个类可以实现多个接口
	抽象类可以包含抽象方法、静态方法、构造方法等，且抽象方法前必须加 <code>abstract</code> 关键字修饰。	接口中可以有抽象方法、默认方法、静态方法。接口中的抽象方法默认为 <code>public abstract</code> ，在实际编程中可以省略。接口中的默认方法和静态方法可以包含方法体。
	抽象类中可以定义实例属性，也可以定义静态常量属性	接口中只能定义静态常量属性，且定义时就初始化

### ➤ 代码层面比较

我们应该何时选用抽象类，何时使用接口呢？我们通过一个具体例子来理解一下。

---

比如超人 Superman 类继承自父类 People 类，People 类中除了包含超人，还包括一般人、蜘蛛侠、绿巨人等 这时便可以将 People 类声明为一个抽象类。而超人 Superman 不仅是拥有 People 走路和跑步的功能，他还拥有一般人类没有的功能“飞”，而很多东西都具有飞的功能，比如鸟飞、气球飞、飞机飞...而这些东西都抽象不出来一个公共的父类，所以就将“飞”这个功能定义为一个接口，以便其它生物如鸟去调用飞的功能。

具体代码：

```
//接口: fly
public interface IFly {
    void fly();
}
//父类: people类
public abstract class People {
    //声明抽象方法walk()和run()
    public abstract void walk();
    public abstract void run();
}
//子类superman类
public class Superman extends People implements IFly{
    public void walk(){
        System.out.println("超人可以走路");
    }
    public void run(){
        System.out.println("超人可以奔跑");
    }
    public void fly(){
        System.out.println("超人可以飞");
    }
}
```