$q.defer() //创建并返回一个deferred对象

$q.all(promise对象数组)

$q.when(value) //将value转换为promise对象

$q.reject(rejectReson) //简化说明 创建该deferred对象的**promise**的then的失败调用函数参数

$q.resolve(value) //简化说明 配置该deferred对象的**promise**的then的成功调用函数参数

promise.then(successFun,errorFun,noticeFun)

promise.catch(fun)

promise.finally(fun)

var deferred=$q.defer();

deferred对象的所有方法和属性：

**promise** : promise对象

**resolve** : 方法，对应success

**reject** : 方法，对应error

**notify**: 方法，对应notice

$q.all(promise对象数组) //返回一个promises对象，可以通过.then(function(data){}) 处理数据

$q.when(value)，value可以是一个值，可以是一个符合promise标准的外部对象。

$q.when的作用是将里面的参数封装成一个$q.defer().promise对象，因此可以通过then方法获取when里面的值。

var funcA = function(){

console.log("funcA");

return "hello,funA";

}

$q.when(funcA()) .then(

function(result){

console.log(result);

});