AngularJS--第八天 课程

自定义服务

案例一

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <script src="js/angular.min.js"></script>
  <script>
  var m1 = angular.module('myModule',[]);
  m1.factory("myServics",function(){
        return {
            name:"hello",
            run : function(){
              return this.name +':angular';
            }
        }
  });
  // m1被定义成一个模块，m2引用这个模块的名字。就被加载成功了
  var m2 = angular.module('myApp',['myModule']);

  m2.controller('Aaa',['$scope','myServics',function($scope,myServics){

      console.log(myServics.run());
  }]);

</script>
</head>

<body>
<div ng-controller="Aaa">

</div>
</body>
</html>
```

案例二

```
<!DOCTYPE HTML>
<html ng-app="myApp">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>无标题文档</title>
<script src="js/angular.min.js"></script>
<script src="js/service.js"></script>
<script>
```

```
var m2 = angular.module('myApp',['module1']);

// m1.config(['myServiceProvider',function(myServiceProvider){

//     myServiceProvider.name = 'hi';

// }]);

m2.controller('Aaa',['$scope','myService',function($scope,myService){

    console.log(myService.show());

}]);



</script>
</head>

<body>
<div ng-controller="Aaa">

</div>
</body>
</html>
```

js部分

```
// JavaScript Document

var m1 = angular.module('module1',[]);
m1.factory('myService',function(){

 return {
  name : 'hello',
  show : function(){
   return this.name + ':angular';
  }
 };

});
// m1.provider('myService',function(){

//  return {
//   name : 'hello',
//   $get : function(){
//    return {
//     name : this.name,
//     show : function(){
//      return this.name + ':angular';
//     }
//    };
//  }
// };

// });
```

案例三

```
<!DOCTYPE HTML>
<html ng-app="myApp">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>无标题文档</title>
<script src="js/angular.min.js"></script>
<script src="js/service.js"></script>
<script>


var m1 = angular.module('myApp',[]);
// m1.service('myService',show);
//    // 针对构造函数的形式
//    function show(){
//        this.name='fengniao';
//    }
//    show.prototype.num='100';
m1.constant('myService','fengniao');
m1.config(['myService',function(myService){
    console.log(myService);
}]);
// 可以配置操作
// 设置常量
// m1.value('myService','蜂鸟科技');
// 不可以配置操作

m1.controller('Aaa',['$scope','myService',function($scope,myService){

    // console.log(myService.name);
    // console.log(myService.num);

    console.log(myService);

}]);




</script>
</head>

<body>
<div ng-controller="Aaa">

</div>
</body>
```

```
</html>
```

ngRoute

ng-view：切换的主体内容

$routeProvider：在供应商中

when：接受两个参数一个是哈希值

另一个是template或templateUrl

otherwise：初始是和哈希值错误情况下的默认处理

redirectTo：默认的哈希值和模板

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <style>
        .oDiv{
            width: 700px;
            height: 300px;
            border:2px solid purple;
            margin-top: 30px;
        }
    </style>
    <script src="js/angular.min.js"></script>
    <script src="js/angular-route.min.js"></script>
    <script>
    var m1 = angular.module('myApp',['ngRoute']);
    m1.config(['$routeProvider',function($routeProvider){
            $routeProvider
            .when('/tp',{
                // when：接受两个参数一个是哈希值
                // 另一个是template或templateUrl
                template:'<p>图片图片图片图片</p>'
                })
            .when('/qc',{
                template:'<p>汽车汽车汽车汽车</p>'
                })
            .when('/bk',{
                template:'<p>博客博客博客博客</p>'
                })
            .otherwise({
                    redirectTo:'/tp',
                    template:'<p>图片图片图片图片</p>'
                    // otherwise：初始是和哈希值错误情况下的默认处理
                    // redirectTo：默认的哈希值和模板
            });

    }]);
    m1.controller('Aaa',["$scope",function($scope){

    }]);
    </script>
</head>
<body>
    <div ng-controller='Aaa'>
        <a href="#tp">图片</a>
        <a href="#qc">汽车</a>
        <a href="#bk">博客</a>
        <div class="oDiv" ng-view></div>
    <!-- ng-view：切换的主体内容 -->
  </div>
</body>
</html>
```

$routeParams：接受传参

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    .oDiv{
      width: 700px;
      height: 300px;
      border:2px solid purple;
      margin-top: 30px;
    }
  </style>
  <script src="js/angular.min.js"></script>
  <script src="js/angular-route.min.js"></script>
  <script>
  var m1 = angular.module('myApp',['ngRoute']);
  m1.config(['$routeProvider',function($routeProvider){
      $routeProvider
      .when('/tp/:num',{
        // when：接受两个参数一个是哈希值
        // 另一个是template或templateUrl
        template:'{{name}}<p>图片图片图片图片</p>',
        controller:'Aaa'
        })
      .when('/qc',{
        template:'{{name}}<p>汽车汽车汽车汽车</p>',
        controller:'Bbb'
        })
      .when('/bk',{
        template:'{{name}}<p>博客博客博客博客</p>',
        controller:'Ccc'
        })
      .otherwise({
          redirectTo:'/tp',
          template:'<p>图片图片图片图片</p>'
          // otherwise：初始是和哈希值错误情况下的默认处理
          // redirectTo：默认的哈希值和模板
      });

  }]);
  m1.controller('Aaa',["$scope","$location","$routeParams",function($scope,$location,$routeParams){
    $scope.name="hello";
    $scope.$location=$location;
    console.log($routeParams);
  }]);
  m1.controller('Bbb',["$scope",function($scope){
    $scope.name="蜂鸟";
  }]);
  m1.controller('Ccc',["$scope",function($scope){
    $scope.name="科技";
  }]);
  </script>
</head>
<body>
  <div ng-controller='Aaa'>
    <!-- <a href="#tp">图片</a>
    <a href="#qc">汽车</a>
    <a href="#bk">博客</a> -->
```

```
        <a href="#tp" ng-click="$location.path('tp/111')">图片</a>
        <a href="#qc" ng-click="$location.path('qc')">汽车</a>
        <a href="#bk" ng-click="$location.path('tp/222')">博客</a>
        <div class="oDiv" ng-view></div>
        <!-- ng-view：切换的主体内容 -->
    </div>
</body>
</html>
```

事件

$on：绑定监听方法

$routeChangeStart：路由切换之前触发的内容

$routeChangeSuccess/Error：路由切换成功、失败触发的内容

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <style>
        .oDiv{
            width: 700px;
            height: 300px;
            border:2px solid purple;
            margin-top: 30px;
        }
    </style>
    <script src="js/angular.min.js"></script>
    <script src="js/angular-route.min.js"></script>
    <script>
    var m1 = angular.module('myApp',['ngRoute']);
    m1.config(['$routeProvider',function($routeProvider){
            $routeProvider
            .when('/tp/:num',{
                // when：接受两个参数一个是哈希值
                // 另一个是template或templateUrl
                template:'{{name}}<p>图片图片图片图片</p>',
                controller:'Aaa'
                })
            .when('/qc',{
                template:'{{name}}<p>汽车汽车汽车汽车</p>',
                controller:'Bbb'
                })
            .when('/bk',{
                template:'{{name}}<p>博客博客博客博客</p>',
                controller:'Ccc'
                })
            .otherwise({
                    redirectTo:'/tp',
                    template:'<p>图片图片图片图片</p>'
                    // otherwise：初始是和哈希值错误情况下的默认处理
                    // redirectTo：默认的哈希值和模板
            });

    }]);
    m1.controller('Aaa',["$scope","$location","$routeParams",function($scope,$location,$routeParams){
        $scope.name="hello";
        $scope.$location=$location;
        console.log($routeParams);
    }]);
    m1.run(['$rootScope',function($rootScope){
                $rootScope.$on('$routeChangeStart',function(event,current,pre){
                    // $on：绑定监听方法
                // $routeChangeStart：路由切换之前触发的内容
                // console.log('start');
                console.log(event);
                console.log(current);
                console.log(pre);
                // event：event对象    current：现在的路径地址    pre：上一个路径地址

                });

    }])
    m1.controller('Bbb',["$scope",function($scope){
        $scope.name="蜂鸟";
    }]);
    m1.controller('Ccc',["$scope",function($scope){
        $scope.name="科技";
    }]);
    </script>
</head>
<body>
    <div ng-controller='Aaa'>
        <!-- <a href="#tp">图片</a>
        <a href="#qc">汽车</a>
        <a href="#bk">博客</a> -->
        <a href="#tp" ng-click="$location.path('tp/111')">图片</a>
        <a href="#qc" ng-click="$location.path('qc')">汽车</a>
        <a href="#bk" ng-click="$location.path('bk/222')">博客</a>
        <div class="oDiv" ng-view></div>
        <!-- ng-view：切换的主体内容 -->
    </div>
</body>
</html>
```

$emit：类似冒泡方式往上传播

$broadcast：类似广播方式往下传播

event

targetScope：要传播的位置

currentScope：目前的位置

name：打印方法名称

stopPropagation()：阻止事件冒泡

内部传播方式

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <script src="js/angular.min.js"></script>
    <script>
    var m1 = angular.module('myApp',[]);
    m1.controller('Aaa',['$scope',function($scope){
        $scope.count=0;
        $scope.$on('myStart',function(event){
            $scope.count++;

            // console.log(event.targetScope==event.currentScope);
            // console.log( event.name );
            // targetScope：要传播的位置
            // currentScope：目前的位置
            // 让两个位置进行对比，第一个值为false第二个为ture

            // console.log(event.name);
            //  打印方法名称
            event.stopPropagation();
            // 阻止事件冒泡
        })
    }]);

</script>
</head>

<body>
<div ng-controller="Aaa">
        {{count}}
        <div ng-controller="Aaa" ng-click="$emit('myStart')">
            <!-- 默认情况是不会影响父级和子级的 -->
            <!-- $emit('start') : 往上传播 -->
            <!-- $broadcast('start') : 往下传播 -->
                {{count}}
                <div ng-controller="Aaa">
                        {{count}}
                </div>
        </div>
</div>
</body>
</html>
```

ngAnimate：

ng-enter

ng-enter-active

ng-leave

ng-leave-active

案例

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <script src="js/angular.min.js"></script>
    <script src="js/angular-animate.min.js"></script>
    <style>
        .red{
            width: 200px;
            height: 200px;
            background: red;
            transition:1s all;
        }
        .red.ng-enter{
            width: 0;
            height: 0;
            opacity: 0;
        }
        .red.ng-enter-active{
            width: 200px;
            height: 200px;
            opacity: 1;
        }
        .red.ng-leave{
            opacity: 1;
        }
        .red.ng-leave-active{
            width: 0;
            height: 0;
            opacity: 0;
        }
    </style>
    <script>
    var m1 = angular.module('myApp',['ngAnimate']);
    m1.controller('Aaa',["$scope",function($scope){
        $scope.bBtn=true;


    }]);
    </script>
</head>
<body>
    <div ng-controller='Aaa'>
        <input type="checkbox" ng-model="bBtn">
        <div ng-if="bBtn" class="red"></div>
    </div>
</body>
</html>
```

支持的指令：if、repeat、view、swtich

ng-enter-stagger

animation-delay：延迟多久

跟angular版本有很大关系

```
<!DOCTYPE html>
<html lang="en" ng-app="myApp">
    <head>
        <meta charset="utf-8">
       <script src="http://cdn.bootcss.com/angular.js/1.3.8/angular.min.js"></script>
        <script src="http://cdn.bootcss.com/angular.js/1.3.8/angular-animate.min.js"></script>
        <style>
            li{
                width: 600px;
                height: 30px;
                list-style: none;
                background: #b6b6b6;
                color:purple;
                margin-top: 20px;
            }
            .red{
            transition:1s all;
        }
            .red.ng-enter{
            opacity: 0;
        }
        .red.ng-enter-active{
            opacity: 1;
        }
        .red.ng-leave{
             display: none;
        }
        .red.ng-enter-stagger{
            animation-delay:100ms;
        }

        </style>
        <script>
            var m1 = angular.module("myApp",['ngAnimate']);
            m1.controller("Aaa",["$scope","$http","$timeout",function($scope,$http,$timeout){
                var timer=null;
                $scope.data = [];
                $scope.change = function(name){
                    $timeout.cancel(timer);
                    // 清除延迟器
                    timer=$timeout(function(){
                        // 加上延迟效果，不然在输入文字过程中会多次请求重复刷新
                        $http({
                    method:'JSONP',
                    url:'https://sp0.baidu.com/5a1Fazu8AA54nxGko9WTAnF6hhy/su?wd='+name+'&cb=JSON_CALLBACK'
                    // jQuery1102005306412995258336_1477293454578&_=1477293454581
                    // 百度用的jQ的方式，在angular中要改成JSON_CALLBACK才能收到回调信息。
                }).success(function(data){
                    // console.log(data);
                    $scope.data = data.s;
                   });
                 },400);
                };

            }]);
        </script>
    </head>
    <body>
        <div ng-controller="Aaa">
            <input type="text" ng-model="name" ng-keyup="change(name)" placeholder="请输入搜索内容">
            <input type="button" ng-click="change(name)" value="搜索">
            <ul>
                <li class="red" ng-repeat="da in data">{{ da }}</li>
            </ul>
        </div>
    </body>
</html>
```

ng-hide-add

ng-hide-add-active

ng-hide-remove

ng-hide-remove-active

支持的指令：show、hide、class、model等

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <script src="js/angular.min.js"></script>
    <script src="js/angular-animate.min.js"></script>
    <style>
        .red{
            width: 200px;
            height: 200px;
            background: red;
            transition:1s all;
        }
        .red.ng-hide-remove{
            width: 0;
            height: 0;
            opacity: 0;
        }
        .red.ng-hide-remove-active{
            width: 200px;
            height: 200px;
            opacity: 1;
        }
        .red.ng-hide-add{
            opacity: 1;
        }
        .red.ng-hide-add-active{
            width: 0;
            height: 0;
            opacity: 0;
        }

    </style>
    <script>
    var m1 = angular.module('myApp',['ngAnimate']);
    m1.controller('Aaa',["$scope",function($scope){
        $scope.bBtn=true;


    }]);
    </script>
</head>
<body>
    <div ng-controller='Aaa'>
        <input type="checkbox" ng-model="bBtn">
        <div ng-show="bBtn" class="red"></div>
    </div>
</body>
</html>
```

JS方式

animation()

enter/leave：适用于if 、repeat

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <script src="js/angular.min.js"></script>
    <script src="js/angular-animate.min.js"></script>
    <script src="js/jquery-1.11.3.min.js"></script>
    <style>
        .red{
            width: 200px;
            height: 200px;
            background: red;
        }

    </style>
    <script>
    var m1 = angular.module('myApp',['ngAnimate']);

    m1.animation(".red",function(){
            return{
                leave:function(element,done){
                    // console.log(element);
                    // element：是元素本身
                    // console.log(done);
                    // done是函数、触发指令
                    // $(element).animate({width:0,height:0},1000);
                    // 在没有done的情况只是把宽高改变成0
                    $(element).animate({width:0,height:0},1000,done);
                    // 执行ng-if的指令

                },
                 enter:function(element,done){
                    $(element).css({width:0,height:0});
                    // ng-if指令让div的remove了，所以需要从新创建一个div的宽和高
                    $(element).animate({width:200,height:200},1000,done);

                }
            };
    });
    m1.controller('Aaa',["$scope",function($scope){
        $scope.bBtn=true;


    }]);
    </script>
</head>
<body>
    <div ng-controller='Aaa'>
        <input type="checkbox" ng-model="bBtn">
        <div ng-if="bBtn" class="red"></div>
    </div>
</body>
</html>
```

removeClass/addClass：适用于ng-show、ng-hide

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <script src="js/angular.min.js"></script>
    <script src="js/angular-animate.min.js"></script>
    <script src="js/jquery-1.11.3.min.js"></script>
    <style>
        .red{
            width: 200px;
            height: 200px;
            background: red;
        }

    </style>
    <script>
    var m1 = angular.module('myApp',['ngAnimate']);

    m1.animation(".red",function(){
            // return{
            //      leave:function(element,done){
            //          // console.log(element);
            //          // element：是元素本身
            //          // console.log(done);
            //          // done是函数、触发指令
            //          // $(element).animate({width:0,height:0},1000);
            //          // 在没有done的情况只是把宽高改变成0
            //          $(element).animate({width:0,height:0},1000,done);
            //          // 执行ng-if的指令

            //      },
            //       enter:function(element,done){
            //          $(element).css({width:0,height:0});
            //          // ng-if指令让div的remove了，所以需要从新创建一个div的宽和高
            //          $(element).animate({width:200,height:200},1000,done);

            //      }
            // };
            return{
                addClass:function(element,sClass,done){
                    // console.log(element);
                    // element：是元素本身
                    // console.log(sClass);
                    // sClass是样式
                    // console.log(done);
                    // done是函数、触发指令
                    // $(element).animate({width:0,height:0},1000);
                    // 在没有done的情况只是把宽高改变成0
                    $(element).animate({width:0,height:0},1000,done);
                    // 执行ng-if的指令

                },
                 removeClass:function(element,sClass,done){
                    $(element).css({width:0,height:0});
                    // ng-if指令让div的remove了，所以需要从新创建一个div的宽和高
                    $(element).animate({width:200,height:200},1000,done);

                }
            };
    });
    m1.controller('Aaa',["$scope",function($scope){
        $scope.bBtn=true;


    }]);
    </script>
</head>
<body>
    <div ng-controller='Aaa'>
        <input type="checkbox" ng-model="bBtn">
        <div ng-show="bBtn" class="red"></div>
    </div>
</body>
</html>
```