AngularJS--第六天 课程

自定义指令

指令是可以复用的

scope：独立作用域、让当前这个标签不会互相影响

scope:{}隔离作用域、让数据隔离。

```
scope:{
            myClass:"@",
            // @绑定的是字符串，输入什么、解析出来什么。
            myName:"=",
            // =绑定的是数据，是作用域下的变量名。
            myFn:"&"
            // &绑定的是方法。
        }
```

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <script src="js/angular.min.js"></script>
    <style>
        .oDiv1 div,.oDiv2 div{
            display: none;
            width:200px;
            height: 200px;
            border: 2px solid purple;
        }
        .red{
            background: red;
        }
    </style>
    <script>
    var m1 = angular.module('myApp',[]);
    m1.directive('myHi',function(){
        return{
                restrict:"E",
                replace:true,
                // scope:true,
                // 独立作用域、让当前这个标签不会互相影响
                scope:{
                    myClass:"@",
                    // @绑定的是字符串，输入什么、解析出来什么。
                    myName:"=",
```

```
                // =绑定的是数据，是作用域下的变量名。
                myFn:"&"
                // &绑定的是方法。
            },
            // 隔离作用域、让数据隔离。
            controller:["$scope",function($scope){
                $scope.name="蜂鸟~"
                // 这里面写共享的数据。
            }],
            template:'<div class="{{myClass}}">\
                        <span class="red" ng-click="myFn({num:123})">图片</span>\
                        <span>微博</span>\
                        <span>汽车</span>\
                        <div style="display:block">{{myName}}</div>\
                        <div>微博微博微博微博</div>\
                        <div>汽车汽车汽车汽车</div>\
                    </div>'
        };
    });
    m1.controller("Aaa",["$scope",function($scope){
        $scope.name="hi";
        $scope.show = function(a){
            alert(a);
        }
    }]);
    </script>
</head>
<body ng-controller="Aaa">
    <my-hi my-class="oDiv1" my-name="name" my-fn="show(num)"></my-hi>
    <my-hi my-class="oDiv2" my-name="name"></my-hi>
    <!-- 指令是可以复用的。参考angular自带指令 -->
    <!-- ng-init="name='我是修改的数据'"我们的数据是从内到外搜索的 -->

</body>
</html>
```

link

scope

element：元素

attr

reController

选项卡完成版

demo.html

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
```

```
    <style>
        #oDiv1 div ,#oDiv2 div{
            width: 200px;
            height: 200px;
            border: 1px solid red;
            display: none;
            text-align: center;
        }
        #oDiv1 input.active ,#oDiv2 input.active{
            background: red;
        }
    </style>
    <script src="js/jquery-1.11.2.js"></script>
    <script src="js/angular.min.js"></script>
    <script>
    var m1 = angular.module('myApp',[]);

    m1.directive('myTab',function(){
        return{
            restrict:'E',
            replace:true,
            scope:{
                myId:'@',
                myData:'='
            },
            controller:["$scope",function($scope){
                    $scope.name="fengniao";
            }],
                templateUrl:'te1.html',
                link: function(scope,element,attr){
                  // console.log(scope.name);
                  // console.log(element);
                  // console.log(attr.myId);
                  element.delegate('input','click',function(){
                      $(this).attr('class','active').siblings('input').attr('class','');
                      $(this).siblings('div').eq($(this).index()).css('display','block').siblings('div').css('display','none');
                  })
                }
        };
    });
    m1.controller('Aaa',["$scope",function($scope){
        $scope.data1=[
            {title:"汽车",content:"汽车汽车汽车"},
            {title:"风景",content:"风景风景风景"},
            {title:"手机",content:"手机手机手机"}
        ];
        $scope.data2=[
            {title:"杂志",content:"杂志杂志杂志"},
            {title:"小说",content:"小说小说小说"}
        ];
    }]);
    </script>
</head>
<body>
    <div ng-controller='Aaa'>
        <my-tab my-id="oDiv1" my-data="data1"></my-tab>
        <my-tab my-id="oDiv2" my-data="data2"></my-tab>
    </div>
</body>
</html>
```

te1.html

```
<div id="{{myId}}">
                        <!-- <input class="active" type="button" value="1">
```

```
                    <input type="button" value="2">
                    <input type="button" value="3">
                    <div style="display:block">11111111</div>
                    <div>22222222</div>
                    <div>33333333</div> -->
    <input type="button" ng-repeat=" data in myData" ng-value="data.title" ng-class="{active:$first}">
    <div ng-repeat=" data in myData" ng-style="{display:$first?'block':'none'}">{{ data.content}}</div>
</div>
```

拖拽元素

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <script src="js/jquery-1.11.3.min.js"></script>
  <script src="js/angular.min.js"></script>
  <style>
    #oDiv{
      width:100px;
      height: 100px;
      background: red;
      position: absolute;
    }
  </style>
  <script>
  var m1 = angular.module('myApp',[]);
  m1.directive('myRun',function(){
    return{
          restrict:"A",
          link : function(scope,element,attr){
            // console.log(element);
            // 要拖拽的元素
            element.on('mousedown',function(e){
              var This = this;
              divX = e.pageX - $(this).offset().left;
              divY = e.pageY - $(this).offset().top;

              $(document).on('mousemove',function(e){
                $(This).css('left',e.pageX-divX);
                $(This).css('top',e.pageY-divY);
              });
              $(document).on('mouseup',function(){
                $(document).off();
              });
              return false;
            });
          }
    };
  });
  m1.controller("Aaa",["$scope",function($scope){
  }]);
  </script>
```

```
</head>
<body ng-controller="Aaa">
    <div id="oDiv" my-run></div>
</body>
</html>
```

拖拽元素-1

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <script src="js/jquery-1.11.3.min.js"></script>
    <script src="js/angular.min.js"></script>
    <style>
        #oDiv{
            width:100px;
            height: 100px;
            background: red;
            position: absolute;
        }
    </style>
    <script>
    var m1 = angular.module('myApp',[]);
    m1.directive('myRun',function(){
        return{
                restrict:"A",
                link : function(scope,element,attr){
                    // console.log(element);
                    // 要拖拽的元素
                    //
                    // console.log(attr);
                    // console.log(attr.myRun);
                    // 得到是的一个字符串的值string
                    // console.log(typeof  attr.myRun);

                    attr.myRun = angular.equals(attr.myRun,true);
                    // console.log(typeof  attr.myRun);
                    // 用equals()工具方法，把字符串类型转换成布尔值。
                    // equals()：把两个值比对。得到的结果是布尔值
                    //



                    element.on('mousedown',function(e){
                        var This = this;
                        divX = e.pageX - $(this).offset().left;
                        divY = e.pageY - $(this).offset().top;
```

```
                $(document).on('mousemove',function(e){
                    $(This).css('left',e.pageX-divX);
                    $(This).css('top',e.pageY-divY);
                });
                $(document).on('mouseup',function(){
                    $(document).off();
                });
                return false;
            });
        }
    };
});
m1.controller("Aaa",["$scope",function($scope){
}]);
</script>
</head>
<body ng-controller="Aaa">
    <div id="oDiv" my-run="true"></div>
</body>
</html>
```

拖拽元素-2

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <style>
      #oDiv{
        width: 100px;
        height: 100px;
        background: red;
        position: absolute;
      }
    </style>
    <script src="js/jquery-1.11.2.js"></script>
    <script src="js/angular.min.js"></script>
    <script>
    var m1 = angular.module('myApp',[]);

    m1.directive('myDrag',function(){
        return{
            restrict:'A',
                link: function(scope,element,attr){
                    var disX = 0;
                    var disY = 0;
                    // console.log(attr.myDrag)
                    attr.myDrag = angular.equals(attr.myDrag,'true');
                    // console.log(attr.myDrag)
                    element.on('mousedown',function(ev){
                        var This = this;
```

```
                disX = ev.pageX - $(this).offset().left;
                disY = ev.pageY - $(this).offset().top;

                if(attr.myDrag){
                  var $line=$('<div>');
                  $line.css({ width : $(this).outerWidth() , height : $(this).outerHeight() , position: 'absolute' , left :
$(this).offset().left , top : $(this).offset().top , border : '1px black dotted'});
                  $('body').append($line);
                }
                $(document).on('mousemove',function(ev){
                   if(attr.myDrag){
                      $line.css('left',ev.pageX-disX);
                      $line.css('top',ev.pageY-disY);
                   }else{
                      $(This).css('left',ev.pageX-disX);
                      $(This).css('top',ev.pageY-disY);
                   }

                })
                $(document).on('mouseup',function(){
                   $(document).off();
                   if(attr.myDrag){
                      $(This).css('left',$line.offset().left);
                      $(This).css('top',$line.offset().top);
                      $line.remove();
                   }
                })
                return false;
             })
          }
      };
   });
   m1.controller('Aaa',["$scope",function($scope){

   }]);
   </script>
</head>
<body>
   <div ng-controller='Aaa'>
      <div id="oDiv" my-drag="true"></div>
   </div>
</body>
</html>
```

当我们需要嵌套自定义指令的时候

transclude

ng-transclude

```
<!DOCTYPE html>
<html ng-app="myApp">
```

```
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <script src="js/angular.min.js"></script>
    <style>
        .oDiv1 div,.oDiv2 div{
            display: none;
            width:200px;
            height: 200px;
            border: 2px solid purple;
        }
        .red{
            background: red;
        }
    </style>
    <script>
    var m1 = angular.module('myApp',[]);
    m1.directive('hi',function(){
        return{
                restrict:"E",
                replace:true,
                transclude:true,
                // 当我们配置项中transclude为true时，可以在模板中引用其他自定义的指令内容。放入ng-transclude的标签中进行覆盖。
                template:'<div>hi 你好<h1 ng-transclude></h1></div>'
        };
    });
    m1.directive('hello',function(){
        return{
                restrict:"E",
                replace:true,
                template:'<div>hello 你好</div>'
        };
    });
    m1.controller("Aaa",["$scope",function($scope){
            $scope.name="hi";
    }]);
    </script>
</head>
<body ng-controller="Aaa">
    <hi>
        <hello></hello>
    </hi>
</body>
</html>
```

require

$http：类似jQ中的ajax

method:传递数据的方式

url：路径

success：成功后

error：失败后

```
<!DOCTYPE html>
<html lang="en" ng-app="myApp">
    <head>
        <meta charset="utf-8">
        <script src="js/angular.min.js"></script>
```

```
    <script>
      var m1 = angular.module("myApp",[]);
      m1.controller("Aaa",["$scope","$http",function($scope,$http){
        // $http({
        //     method:'GET',
        //     // GET     POST   JSONP
        //     url:'data.php'
        // }).success(function(data,state,headers,config){
        //     // data : 数据   state : 状态   headers : 头部信息  config : 配置数据
        //     console.log(data);
        // }).error(function(){
        //     console.log(data);
        // });
            //简写格式
        $http.get('data.php').success(function(data,state,headers,config){
          console.log(data);
        });
      }]);
    </script>
  </head>
  <body>
    <div ng-controller="Aaa">

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

简写方式

案例 -百度搜索功能

```
<!DOCTYPE html>
<html lang="en" ng-app="myApp">
  <head>
    <meta charset="utf-8">
    <script src="js/angular.min.js"></script>
    <style>
      li{
        width: 600px;
        height: 30px;
        list-style: none;
        background: #b6b6b6;
        color:purple;
        margin-top: 20px;
      }
    </style>
    <script>
      var m1 = angular.module("myApp",[]);
      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'
          // jQuery110200530641299525833 6_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 ng-repeat="da in data">{{ da }}</li>
      </ul>
    </div>
  </body>
</html>
```

$location

absUrl()：编码后的绝对地址

```
            var a = $location.absUrl();
          // $location.absUrl()：编码后的绝对地址
    console.log(a);
```

path()：配合路由使用

```
            $location.path('aaa');
            var a = $location.path();
            // 配合路由使用，改变网址路径
           console.log(a);
```

replace()：可以替换保留、历史路径方式，这样就没有历史路径了。

```
            $location.path('aaa/bbb/ccc').replace();
            // replace()可以替换保留、历史路径方式，这样就没有历史路径了。
            var a = $location.path();
           console.log(a);
```

hash():哈希值

```
            $location.hash('hi');
            // $location.hash('hi');可以得到相应的哈希值。
            var a = $location.hash();
           console.log(a);
```

search():搜索

```
            $location.search({'name':'boy'});
            // $location.search({'name':'boy'});可以得到相应的search。
            var a = $location.search();
           console.log(a);
```

url()：地址

```
                var a = $location.url();
            // $location.url();得到一个并不是绝对路径的地址
```

host()：主题名

localhost

port()：端口

```
var a = $location.port();
            // 得到端口
```

protocol()：协议

```
var a = $location.protocol();
            // 得到当前协议
```