指令：

Ng-app

Ng-controller

Ng-init

Ng-model

Ng-bind

Ng-repeat

主要对象：

$http：用户Http请求

ng-show接收boolean类型的值以及计算结果为boolean类型的表达式。

自定义过滤器：

app

.filter('typeFilter', function () {

return function (input) {

return input == '1' ? '单选题' : '多选题';

}

})

过滤器使用：{{expression|typeFilter:arg1:arg2...}}

自定义指令

app.directive('factionNum', function () {

return {

link: function (scope, elements, attrs, controller) {

elements[0].onkeyup = function () {

if (isNaN(this.value) || this.value < 1 || this.value > 10) {

this.style.borderColor = 'red';

console.log(this.value);

}

else {

this.style.borderColor = '';

}

};

}

}

})

使用：在标签中使用 faction-num特性；

## 依赖注入：

定义函数的参数要与框架预定义的服务或模块名字相对应；

## 定义服务：

框架提供的服务名字都是由$开头的，所以我们自己定义的最好不要用$开头，防止发生命名冲突。

## NG模板：

可以使用：

指令：ng-controller、ng-app、ng-model等等

标记：{{expression[|filterType]}} 用于数据单项绑定

过滤器：

表单控制

### 表达式：

#### 插值表达式：{{varName}}固有字符串

字符串表达式：ng-model=”person.name” ， person.name属于字符串表达式，在框架内部，字符串不会使用eval()来执行，而是有一个专门的$parse服务来处理

ng表达式中不可以使用循环语句、判断语句

## 2015/10/30

AngularJs的一些疑问，

1. 数据绑定是，试图绑定到一个不存在的model会怎么样?
2. <p>
3. <span>Name</span>
4. <span>**{{**person.name**}}**</span>
5. </p>
6. <p>
7. <span>Age</span>
8. <span>**{{**person.age**}}**</span>
9. </p>

当$scope.person=null or $scope.person=undefined时候视图并不会显示什么东西，

1. 什么是实时模板？
2. Angular脚步是什么时候执行的

这行代码载入angular.js脚本，当浏览器将整个HTML页面载入完毕后将会执行该angular.js脚本，angular.js脚本运行后将会寻找含有ng-app指令的HTML标签，该标签即定义了AngularJS应用的作用域。

Tip：

模型中的数据可能是Javascript对象、数组或基本类型，这都不重要，重要的是，他们都属于AngularJS作用域对象。

## 2015/11/9

$resource，创建资源对象的接口，返回接口对象（资源类对象），用于访问后端的接口；

正确定义控制器的方式是：把它定义成模块的一部分

什么是模块？

什么是模板？

模板是HTML片段

模板边界：

Ng-app指令的标签；angularjs是在页面加载完成时候运行的。？那么什么才叫做页面加载完呢？加载了图片，还是只是加载了页面和脚本。

模板可以缓存，所以可以提升应用性能。

Ng-bind与{{}}的选择：

两者效果是一样的，使用场景不一样，{{}}语法阅读起来自然，但是在页面加载时候，angularjs在使用数据替换{{}}之前，用户会看到模板中的数据绑定。而ng-binding不存在这种情况，ng-bind适合使用在首页模板中，此方式可以在数据加载完成之前保证用户看不到绑定的信息；

类似的问题有：img 与a 标签中src和href属性值的绑定；在src或href中使用{{}}无法很好的运行；解决方法：使用ng-src指令 <img ng-src=”/images/cats/{{exp}}”，对于a标签使用:

<a ng-href=”[http://www.baidu.com/{{exp}}](http://www.baidu.com/%7b%7bexp%7d%7d)；”

在form中使用ng-submit可以自动阻止浏览器执行POST操作；

Angularjs有很好的容错性，在模板中的表达式遇到错误，不会报错，所以可以安全的使用未经过初始化的数据模型值；

控制器应该保持小巧和可管理状态；可以为视图中的每一块功能区域创建一个控制器；如菜单控制器，面包屑导航控制器；

控制器的嵌套：子控制器可以访问父控制器的$scope中的所有属性；

$scope.$watch函数

Var dereg=$watch(‘exp’,function(newValue,oldValue,$scope));

Dereg() ;销毁监听数据

## 指令：

App.directive(‘myDirective’,function(){

Return {

Restrict:’EACM’,

Replace:true,

Scope:{

innerAttr:’@’ //将内部作用域的innerAttr绑定到Dom中inner-attr的属性值

},

Controller:function($scope){ //指令可以用自己的控制器，这里可以初始化参数

},

Template:’<tag {{innerAttr}}>...</tag>’

}

})

Restrict:

E:Element,指令可以写成标签：<my-driective></my-directive>

A:Attribute:指令可以写在属性中：<a my-directive></a>

C:Class,指令写在类中：<a class=”my-directive”></a>

M:写在注释中

尽量使用属性方式定义指令

指令内部作用域

给指令内部传值

Scope:{

innerAttr:’@’

}

指令内部作用域：将dom中 inner-attr属性复制到内部作用域的innerAttr属性中；进而更新{{innerAttr}}表达式的值

Angularjs如何动态创建指令，如何放置。

# Angular Guide：

In Angular, the only place where an application should access the DOM is within directives. This is important because artifacts that access the DOM are hard to test.

A filter formats the value of an expression

a controller code we also added an [ng-controller](https://docs.angularjs.org/api/ng/directive/ngController) directive to the HTML. This directive tells Angular that the new InvoiceController is responsible for the element with the directive and all of the element's children.

<div ng-app="invoice1" ng-controller="InvoiceController as invoice">

The syntax InvoiceController as invoice tells Angular to instantiate the controller and save it in the variable invoice in the current scope.

we are also able to bind the result of a function to DOM using {{ invoice.total(...) }}.

the DOM will be automatically updated whenever the result of the function changes.

So if the data binding function is complexed,it will take a long time to calculate.

When the application grouws it is a good pratice to move view-independent logic from controller into a service.a service can be reused by other parts of the application.

Dependency Injection (DI) is a software design pattern that deals with how objects and functions get created and how they get a hold of their dependencies

可以绑定一个函数到视图；

DI container：injector

When Angular starts, it will use the configuration of the module with the name defined by the ng-app directive, including the configuration of all modules that this module depends on.

ng-app="invoice2". This tells Angular to use the invoice2 module as the main module for the application.

that controllers are created using a factory function

controller 有创建controller 的构造函数(controller factory function)，service有创建service的构造函数(service factory function).

Pass an arrry to factory to prevent errors if code is minified

## DataBinding



First the template (which is the uncompiled HTML along with any additional markup or directives) is compiled on the browser. The compilation step produces a live view

## Controller

When a Controller is attached to the DOM via the [ng-controller](https://docs.angularjs.org/api/ng/directive/ngController) directive, Angular will instantiate a new Controller object, using the specified Controller's **constructor function**.

在module中声明的controller可以附加到多个DOM中；

If the controller has been attached using the controller as syntax then the controller instance will be assigned to a property on the new scope.

**inline injection annotation**

All the $scope properties will be available to the [template](https://docs.angularjs.org/guide/templates) at the point in the DOM where the Controller is registered

 Any methods assigned to the scope are available in the template/view, and can be invoked **via angular expressions** and **ng event handler directives** (e.g. [ngClick](https://docs.angularjs.org/api/ng/directive/ngClick)).

ng-controller directive is used to (implicitly) create a scope for our template,

The $scope that each Controller receives will have access to properties and methods defined by Controllers higher up the hierarchy.

## Services:

* Lazily instantiated – Angular only instantiates a service when an application component depends on it.
* Singletons – Each component dependent on a service gets a reference to the single instance generated by the service factory.

he **service factory function** generates the single object or function that represents the service to the rest of the application

myModule.factory('serviceId', function() {

var shinyNewServiceInstance;

// factory function body that constructs shinyNewServiceInstance

return shinyNewServiceInstance;

});

Note that you are not registering a **service instance**, but rather a **factory function** that will create this instance when called.

* The order of identifiers in the array is the same as the order of argument names in the factory function.

### Registering

#### Factory

moduleName.factory(‘ServiceName’,[‘arg1’,’arg2’,function(arg1,arg2){

......

Return object;

}])

#### $provide

angular.module('myModule', []).config(['$provide', function($provide) {

$provide.factory('serviceId', function() {

var shinyNewServiceInstance;

// factory function body that constructs shinyNewServiceInstance

return shinyNewServiceInstance;

});

}]);

模板是由compiler编译的。

## Scope:

创建时间：template linking!

$scope存在于声明ng-controller 或 ng-repeat的DOM节点上，如果想从DOM中获取其$scope,可以调用angular.element(domElement).scope);

The [root scope](https://docs.angularjs.org/api/ng/service/$rootScope) is created during the application bootstrap by the [$injector](https://docs.angularjs.org/api/auto/service/$injector). During template linking, some directives create new child scopes.

You can think of the scope and its properties as the data which is used to render the view.

Each Angular application has exactly **one**[**root scope**](https://docs.angularjs.org/api/ng/service/$rootScope), but may have several child scopes.

When new scopes are created, they are added as children of their parent scope.

Angular automatically places ng-scope class on elements where scopes are attached

Ng-controller,ng-repeat

some directives, such as [ng-controller](https://docs.angularjs.org/api/ng/directive/ngController)and [ng-repeat](https://docs.angularjs.org/api/ng/directive/ngRepeat), create new child scopes and attach the child scope to the corresponding DOM element.

**How to get $scope from DOM:**

**Angular.element(domElement).scope();**

Scopes are attached to the DOM as $scope data property

$rootscope is attached to the Dom with ng-app attribute

### Child scope

Can inherits properties from its parent scope

### Isolate scope

Can’t not inherits from other scope

How to look for a property

When Angular evaluates {{name}}, it first looks at the scope associated with the given element for the name property. If no such property is found, it searches the parent scope and so on until the root scope is reached. In JavaScript this behavior is known as prototypical inheritance, and child scopes prototypically inherit from their parents.

### Scope 生命周期

1,创建：template linking, some directives create new scope;

2,注册watcher:template linking, directives register watches on scope;

3,模型改变：调用$scope.$apply().

4,通知

5销毁：$scope.$destroy();

## Event propagation

 Only model modifications which execute inside the $apply method will be properly accounted for by Angular.

For mutations to be properly observed, you should make them only within the [scope.$apply()](https://docs.angularjs.org/api/ng/type/$rootScope.Scope#$apply).

Broadcasted:from parent to children

Emitted:from children to parents

不管是broadcast还是emit，触发事件所在的$scope可以接收到事件通知；

The directives usually fall into one of two categories:

Observing directive:{{expr}} using $watch()

Listener directive:ng-click $apply()

两类指令：

观察者指令：{{expr}}

监听者指令：ng-click 监听DOM事件

## Dirty checking

dirty checking function does not do any DOM access

**三种方式**：

1. by reference
2. by collection

detects changes that occur inside an **array** or an **object**

When items are added, removed, or reordered.  it does not reach into nested collections

copies of the collection contents need to be maintained ?? what for

1. by value

A full traversal of the nested data structure is needed on each digest, and a full copy of it needs to be held in memory



Watch 是干嘛用的？

register [watches](https://docs.angularjs.org/api/ng/type/$rootScope.Scope#$watch) on the scope. These watches will be used to propagate model values to the DOM？ 怎么理解呢？

Controllers may register [watches](https://docs.angularjs.org/api/ng/type/$rootScope.Scope#$watch) on the model.

## Watch and apply

Watch : observe model mutations , allows the directives to be notified of property changes

$digest phase the scope examines all of the $watchexpressions and compares them with the previous value

Apply函数中是调用$rootScope.$digest();During the $digest cycle, all $watched expressions or functions are checked for model mutation and if a mutation is detected, the $watch listener is called.

Apply: propagate any model changes

# Dependency Injection

Components such as services, directives, filters, and animations are defined by an injectable factory method or constructor function

Unlike services, there can be many instances of the same type of controller in an application. 意味着在同个页面中可以同时声明几个ng-controller=”someController” 的标记，DI会创建几个someController 标记；

additional dependencies are made available to Controllers:

1，$scope :与controller 对应得Dom总的scope对象；而在service中只能获取$rootScope;

2，[resolves](https://docs.angularjs.org/api/ngRoute/provider/$routeProvider#when): If a controller is instantiated as part of a route, then any values that are resolved as part of the route are made available for injection into the controller. ？？？？不懂

3种注入方式：

1,inline array annotation

App.controller(‘controllerName’,[‘arg1’,’arg2’,function(arg1,arg2){

}]);

2, $inject property annotation

allow the minifiers to rename the function parameters and still be able to inject the right services

var MyController = function ($scope) {

$scope.name = "黄清露";

}

MyController.$inject = ['$scope'];

app.controller('MyController', MyController);

 the ordering of the values in the $inject array must match the ordering of the parameters in MyController.

3, Implicitly from the function parameter names

each Angular application has an [injector](https://docs.angularjs.org/api/ng/function/angular.injector). The injector is a [service locator](http://en.wikipedia.org/wiki/Service_locator_pattern) that is responsible for construction and lookup of dependencies.

You can add an ng-strict-di directive on the same element as ng-app to opt into strict DI mode:

Strict mode throws an error whenever a service tries to use implicit annotations.

When Angular compiles the HTML, it processes the ng-controller directive, which in turn asks the injector to create an instance of the controller and its dependencies.

<div ng-controller="MyController">

<button ng-click="sayHello()">Hello</button>

</div>

function MyController($scope, greeter) {

$scope.sayHello = function() {

greeter.greet('Hello World');

};

}

injector.instantiate(MyController);

# Templates

## Angular Expressions

JavaScript-like code snippets that are usually placed in bindings such as {{ expression }}.

与javascript表达式的区别：

**Context**:angular exprs are evaluated against a scope object;

**Forgiving:**expression evaluation is forgiving to undefined and null

**No Control Flow Statements:** You cannot use the following in an Angular expression: conditionals, loops, or exceptions. 可以使用**条件运算符 (boolean? Exp1 :exp2)**

Apart from the ternary operator (a ? b : c)

Angular expressions do not have access to global variables like window, document or location.

 If you want to eval() an Angular expression yourself, use the [$eval()](https://docs.angularjs.org/api/ng/type/$rootScope.Scope#$eval) method.？？？how

Angular's [$parse](https://docs.angularjs.org/api/ng/service/$parse) service evaluate expressions in template

表达式不能访问global var

## $event

Directives like [ngClick](https://docs.angularjs.org/api/ng/directive/ngClick) and [ngFocus](https://docs.angularjs.org/api/ng/directive/ngFocus) expose a $event object within the scope of that expression. The object is an instance of a [jQuery Event Object](http://api.jquery.com/category/events/event-object/) when jQuery is present or a similar jqLite object.

事件: <button ng-click="clickMe($event)">Event</button>

$event is outside the scope of that binding.

### One time binding:

One-time expressions will stop recalculating once they are stable, which happens after the first digest if the expression result is a non-undefined value

## Filter

Filters can be applied to the result of another filter. This is called "chaining" and uses the following syntax:

{{ expression | filter1 | filter2 | ... }}

Filters may have arguments. The syntax for this is

Expression作为fitler的第一个参数，argument1第二个

{{ expression | filter:argument1:argument2:... }}

use filters in controllers, services, and directives

当filter作为依赖时候必须用filterNameFilter 引入；

在模板中使用过滤器的劣势：

using a filter in a view template will reevaluate the filter on every digest, which can be costly if the array is big.

在编程中.一个幂等操作的特点是其任意多次执行所产生的影响均与一次执行的影响相同。幂等函数，或幂等方法，是指可以使用相同参数重复执行，并能获得相同结果的函数。这些[函数](http://baike.baidu.com/view/15061.htm)不会影响系统状态，也不用担心重复执行会对系统造成改变。

## Stateful filters

If you however do need to write a stateful filter, you have to mark the filter as $stateful

## Form Validation

The value of ngModel won't be set unless it passes validation for the input field.

Ng-touched什么时候被添加？当有焦点进入control之后都会add ng-touched

ng-pending: any $asyncValidators are unfulfilled

A form is an instance of [FormController](https://code.angularjs.org/1.4.7/docs/api/ng/type/form.FormController). The form instance can optionally be published into the scope using the name attribute.意思说在form标签的外部，可以通过form的name属性来引用form instance

an input control that has the [ngModel](https://code.angularjs.org/1.4.7/docs/api/ng/directive/ngModel) directive holds an instance of [NgModelController](https://code.angularjs.org/1.4.7/docs/api/ng/type/ngModel.NgModelController). Such a control instance can be published as a property of the form instance using the name attribute on the input control.意思是可以通过formname.inputname 来引用这个control instance

### update model and validate non-immediate:

一、event trigger:By default, any change to the content will trigger a model update and form validation.有性能问题？

You can override this behavior using the[ngModelOptions](https://code.angularjs.org/1.4.7/docs/api/ng/directive/ngModelOptions) directive to bind only to specified list of events. I.e. **ng-model-options**="{ updateOn: 'blur' }" will update **and validate** only after the control loses focus

二：delay

ng-model-options="{ debounce: 500 }" will wait for half a second since the last content change before triggering the model update and form validation.

If those attributes are added to an element, they will be applied to all the child elements and controls that inherit from it unless they are overridden.

**Ng-model-options 属性可以被继承**

**验证器:**

Validation functions在 formname.inputName.$validators 对象中；是一个函数

ngModelController.$validators

 Failed validators are stored by key in [ngModelController.$error](https://code.angularjs.org/1.4.7/docs/api/ng/type/ngModel.NgModelController#$error)

ngModelController.$asyncValidators

 In-progress async validations are stored by key in [ngModelController.$pending](https://code.angularjs.org/1.4.7/docs/api/ng/type/ngModel.NgModelController#$pending)

The validation functions are executed every time an input is changed ($setViewValue is called) or whenever the bound model changes

Validation happens after successfully running$parsers and $formatters ????

###### How to add custom validate?

app.directive('integer', function() {

return {

require: 'ngModel',

link: function(scope, elm, attrs, ctrl) {

var INTEGER\_REGEXP = /^\-?\d+$/;

ctrl.$validators.integer = function(modelValue, viewValue) {

if (ctrl.$isEmpty(modelValue)) {

// consider empty models to be valid

return true;

}

if (INTEGER\_REGEXP.test(viewValue)) {

// it is valid

return true;

}

// it is invalid

return false;

};

}

};

});

###### How to delay update model and validate?

###### What is FormController and NgModelController?

Every form is an instance of FormController.

Every form control with ng-model attribute is an instance of NgModelController.

Something special in NgModelController:

$parsers

$formatters

$validators

$asyncValidators

$error

$setValidity

$setViewValue

$render

###### How to custom own form control (using ngModel)