**$parse**

将Angular表达式转换为函数。

**使用**：$parse(expression);

expression：被编译的表达式。

**返回**：表示表达式编译后的函数。function(context,locals)

context：对象，含有需要解析的语句中的表达式（通常是一个scope对象）。

locals：对象，局部变量的上下文对象，用于重写上下文中的值。

**属性**：

literal：boolean，表达式的顶节点是否是一个javascript字面量。

constant：boolean, 表达式是否全部是由javascript的常量字面量组成。

assign：function(context,locals)，可以用来在给定的上下文中修改表达式的值。

使用代码：

<div ng-app="Demo" ng-controller="testCtrl as ctrl">

{{ctrl.ParsedValue}}

</div>

[IMG_256](http://www.cnblogs.com/ys-ys/p/javascript:void(0);)

(function(){

angular.module('Demo', [])

.controller('testCtrl',["$parse",testCtrl]);

function testCtrl($parse) {

var context = {

add: function (a, b) { return a + b; },

　　 mul: function (a, b) { return a \* b }

};

var data = {a: 2, b: 3, c: 4};

this.value = context.mul(data.a,context.add(data.b,data.c));// 平常的函数嵌套调用 2\*(3+4) = 14

var expression = "mul(a, add(b, c))";// 表达式，将被转换为函数

this.ParsedValue = $parse(expression)(context, data); //执行上一句的函数 2\*(3+4) = 14 }

}());