

Angular 1.x



## Термины

Модули — библиотеки взаимосвязанных элементов приложения;

Контроллеры — классы описывающие поведение;

Директивы — виджеты в Angular;

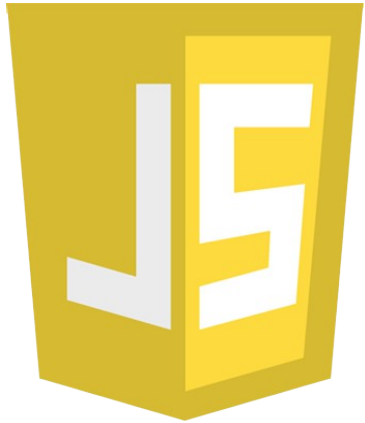
Фильтры — функции которые можно применять во view;

Scope – пространство имен / область видимости;

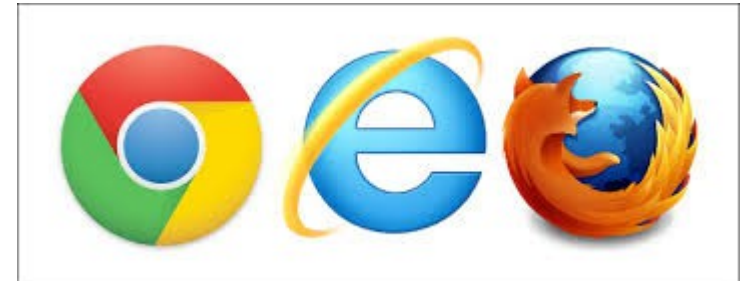
Views – “шаблон интерфейса” (в реальности нечто большее);

Модель — объект, содержащий состояние интерфейса (не путать с доменным объектом!!!);

## Что такое Angular?



- Это JavaScript фреймворк;
- Он работает в броузере на front-end-е;



Что нужно знать чтобы начать писать на Angular?

**HTML**



**CSS**



**JS**



## Какие сильные стороны есть у Angular-a?

- Делает сложное простыми средствами;
- Делает программу структурированной;
- Позволяет писать тестируемый код;

# Какие сильные стороны есть у Angular-a?

## jQuery

```
jQuery(function ($) {
    'use strict';

    var App = {
        init: function () {
            this.count = 0;
            this.bindEvents();
            this.render();
        },
        bindEvents: function () {
            $('.CounterContainer')
                .on('click', 'a', this.anchorClick.bind(this));
        },
        render: function () {
            $(".CounterContainer .value").html(this.count);
        },
        anchorClick: function () {
            this.count++;
            this.render();
        }
    };

    App.init();
});
```

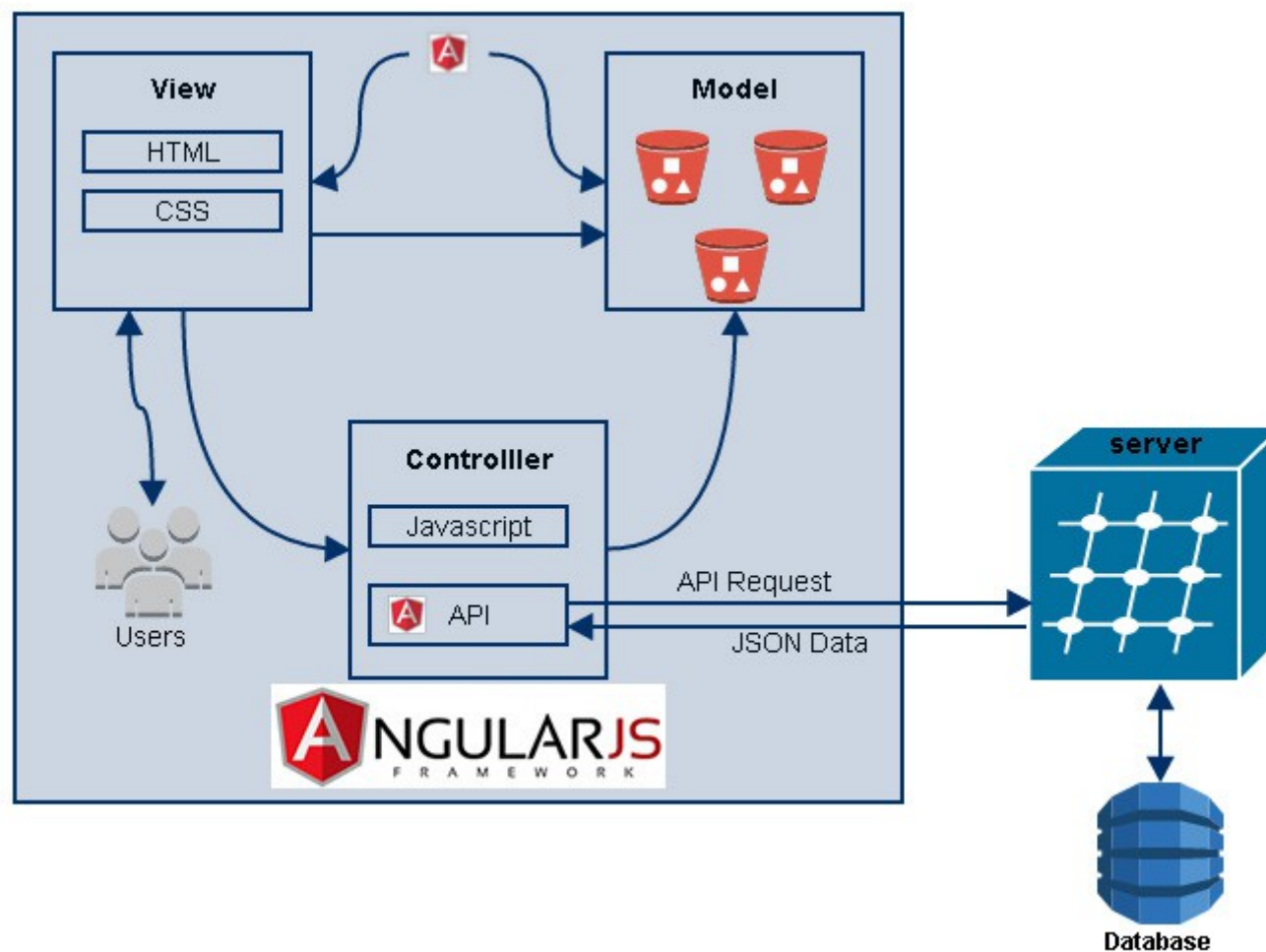
## AngularJS

```
function CounterController($scope) {
    $scope.count = 0;

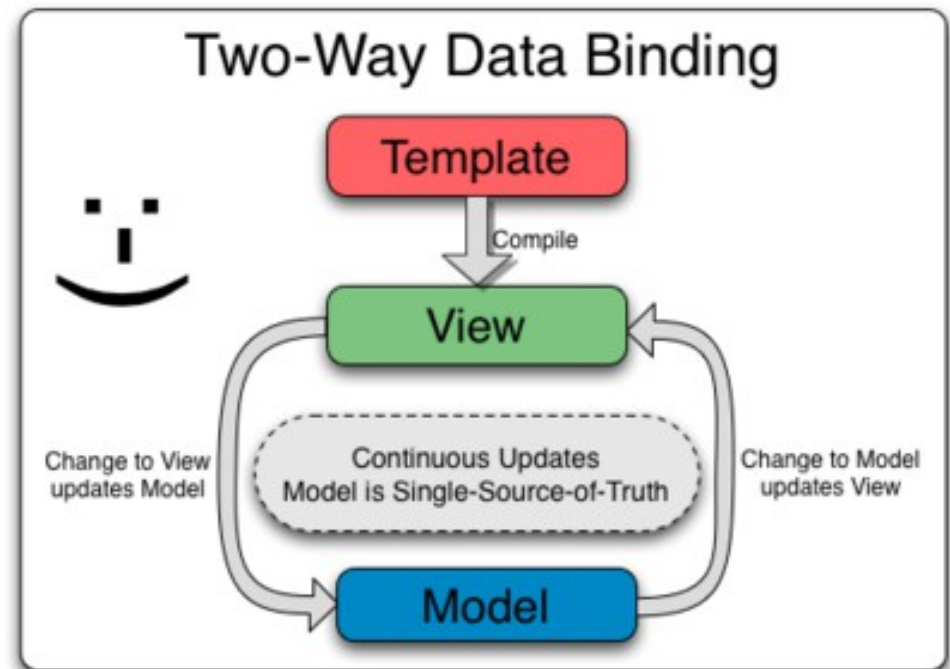
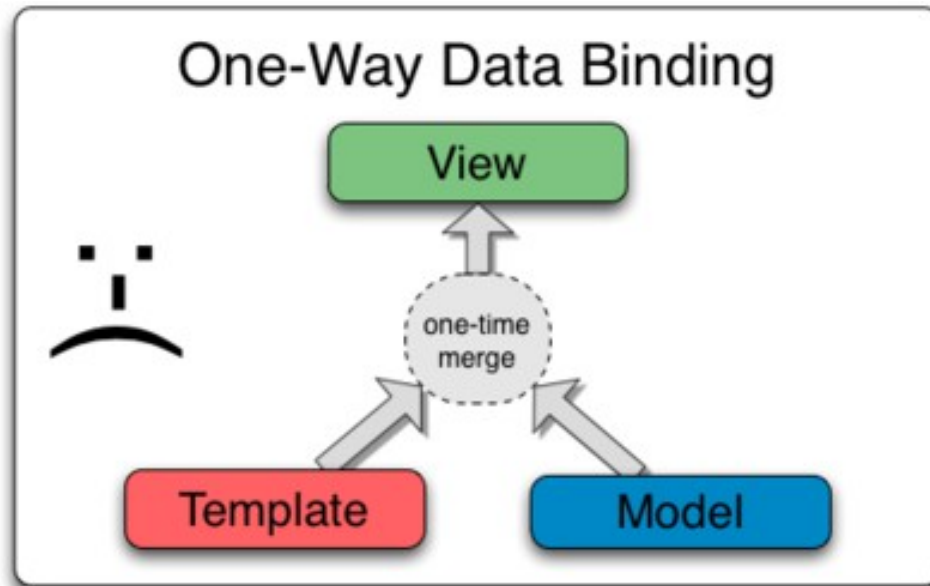
    $scope.click = function(){
        $scope.count++;
    };
}

var myApp = angular
    .module('CounterApp',[])
    .controller('CounterController', ['$scope', CounterController]);
```

# Идеология Angular

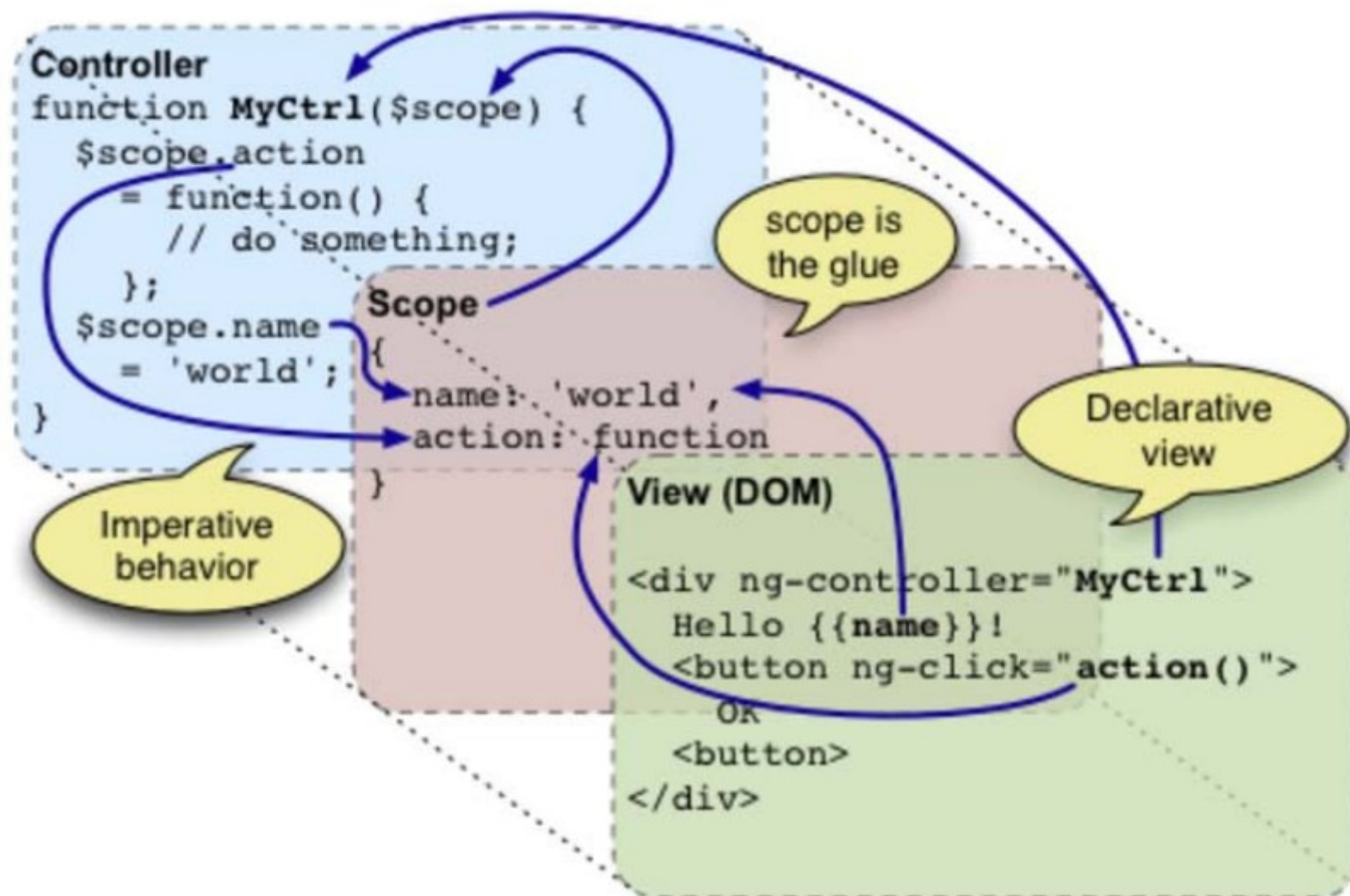


## Data binding-и





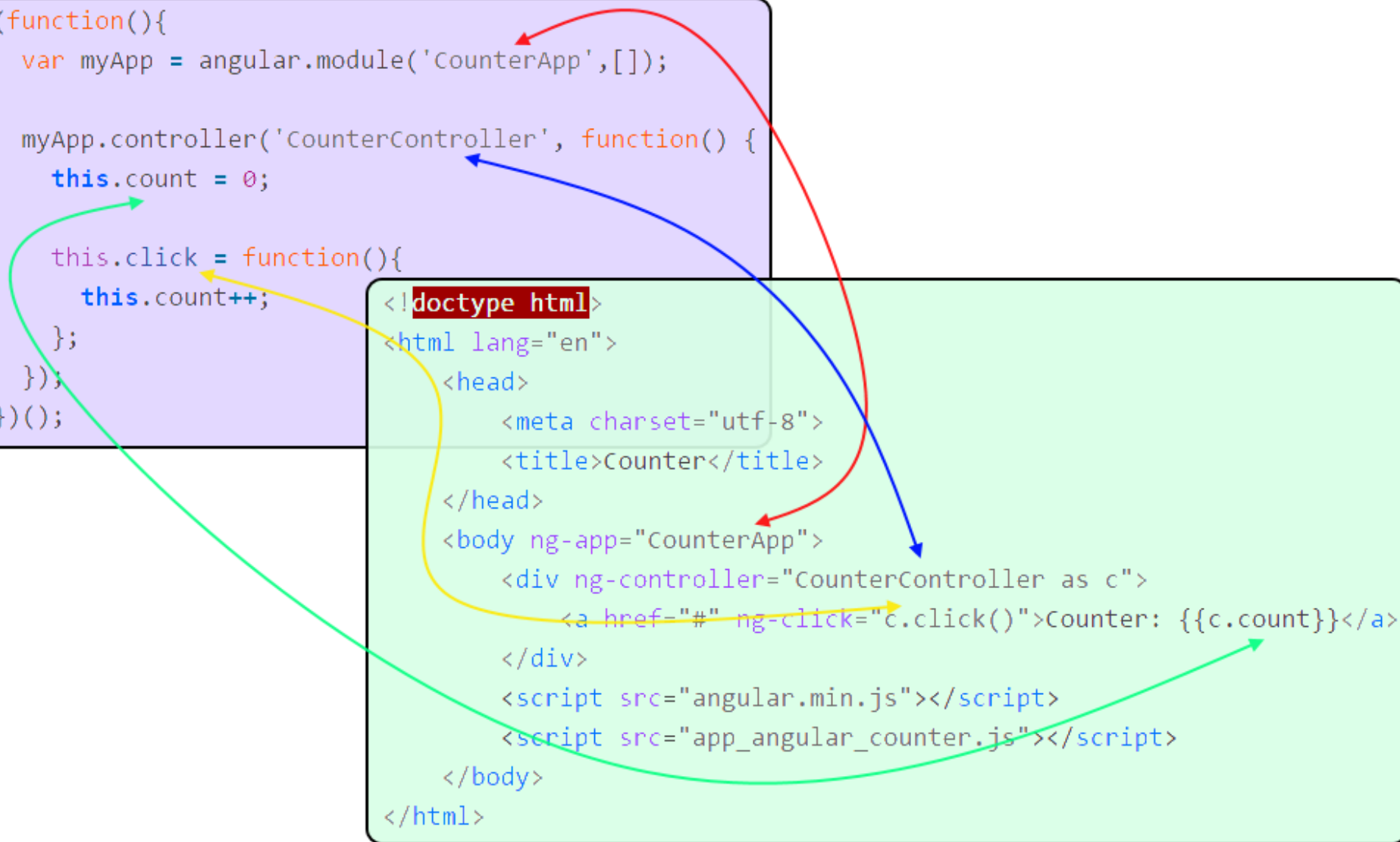
## Идеология Angular (как оно планировалось)



# Идеология Angular (как правильно использовать)))

```
(function(){  
  var myApp = angular.module('CounterApp',[]);  
  
  myApp.controller('CounterController', function() {  
    this.count = 0;  
  
    this.click = function(){  
      this.count++;  
    };  
  });  
})();
```

```
<!doctype html>  
<html lang="en">  
  <head>  
    <meta charset="utf-8">  
    <title>Counter</title>  
  </head>  
  <body ng-app="CounterApp">  
    <div ng-controller="CounterController as c">  
      <a href="#" ng-click="c.click()">Counter: {{c.count}}</a>  
    </div>  
    <script src="angular.min.js"></script>  
    <script src="app_angular_counter.js"></script>  
  </body>  
</html>
```



# Scope

```
<div class="show-scope-demo">
  <div ng-controller="GreetController">
    Hello {{name}}!
  </div>
  <div ng-controller="ListController">
    <ol>
      <li ng-repeat="name in names">{{name}} from {{department}}</li>
    </ol>
  </div>
</div>
```

Hello World!

1. Igor from Angular
2. Misko from Angular
3. Vojta from Angular

# Scope

<pre>&lt;!DOCTYPE html&gt; &lt;html ng-app class="ng-scope"&gt;   &lt;head&gt;...&lt;/head&gt;   &lt;body&gt;     &lt;div&gt;       &lt;div ng-controller="GreetCtrl" class="ng-scope ng-binding"&gt;         Hello World!       &lt;/div&gt;       &lt;div ng-controller="ListCtrl" class="ng-scope"&gt;         &lt;ol&gt;           &lt;!-- ngRepeat: name in names --&gt;           &lt;li ng-repeat="name in names" class="ng-scope ng-binding"&gt;             Igor           &lt;/li&gt;           &lt;li ng-repeat="name in names" class="ng-scope ng-binding"&gt;             Misko           &lt;/li&gt;           &lt;li ng-repeat="name in names" class="ng-scope ng-binding"&gt;             Vojta           &lt;/li&gt;         &lt;/ol&gt;       &lt;/div&gt;     &lt;/div&gt;   &lt;/body&gt; &lt;/html&gt;</pre>	<div><div>\$scope</div><div><div>name='World'</div></div><div><div>names=[...]</div><div><div>name='Igor'</div><div>name='Misko'</div><div>name='Vojta'</div></div></div></div>
---	---

## Создание проекта

```
<!doctype html>
<html lang="en" ng-app="MyApp">
  <head>
    <meta charset="utf-8">
    <title>Title</title>
    <link rel="stylesheet" type="text/css" href="bootstrap.min.css" />
  </head>
  <body>

    <script type="text/javascript" src="angular.min.js"></script>
    <script type="text/javascript" src="app.js"></script>
  </body>
</html>
```

index.html

ng-app – может быть только один;

## Создание проекта

```
(function(){  
    angular.module('MyApp', []);  
  
})();
```

app.js

# Views-ы

```
<div class="show-scope-demo">
  <div ng-controller="GreetController">
    Hello {{name|uppercase}}!
  </div>
  <div ng-controller="ListController">
    <ol>
      <li ng-repeat="name in names">{{name + " from " + department}}</li>
    </ol>
  </div>
</div>
```

## Выражения (expressions)

- Это не JavaScript, хотя и похоже на него;
- Выполняются в рамках определенного scope;
- «Толерантные» к отсутствию данных:

`{{ a.b.c }}`

при `a = null` или `undefined`, просто ничего не выведет;



## Выражения (expressions)

Могут содержать:

- переменные;
- элементы массивов;
- свойства объектов;
- вызовы функций;
- операторы (+, -, \*, /, %, <, >, ==, !=, etc);
- фильтры;

## Выражения (expressions)

Не могут содержать:

- деклараций функций;
- операторов управления процессом выполнения (Control Flow Statements), кроме тернарных операторов (философия angular гласит что логика должна быть в контроллерах а не в шаблонах);
- регулярных выражений;
- создание объектов при помощи оператора new;

# Контроллеры

```
(function(){  
  var myApp = angular.module('CounterApp',[]);  
  
  myApp.controller('CounterController', ['$scope', function($scope) {  
    this.count = 0;  
  
    this.click = function(){  
      this.count++;  
    };  
  }]);  
})();
```

# Директивы

Element:

```
<my-dir></my-dir>
```

Attribute:

```
<span my-dir="exp"></span>
```

Comment:

```
<!-- directive: my-dir exp -->
```

Class:

```
<span class="my-dir: exp;"></span>
```

## Директивы

- Можно пробрасывать параметры;

```
<hello-tag some-attr="Всем"></hello-tag>
```

- Можно применять больше одной на элемент (тогда будет общий scope);

```

```

## Директивы из стандартной библиотеки биндинг данных (ng-model)

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Title</title>
    <link rel="stylesheet" type="text/css" href="bootstrap.min.css" />
  </head>
  <body ng-app>
    <br/>

    <form class="container">
      <div class="input-group">
        <input type="text" class="form-control" ng-model="inputText">
      </div><br/>
      <p>Input text: {{inputText}}</p>
    </form>

    <script type="text/javascript" src="angular.min.js"></script>
  </body>
</html>
```

## Директивы из стандартной библиотеки

### биндинг событий (ng-click, ng-mouseover, ...)

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Counter</title>
  </head>
  <body ng-app>
    <div ng-init="count = 0">
      <a href="#" ng-click="count = count + 1">Counter: {{count}}</a>
    </div>
    <script src="angular.min.js"></script>
  </body>
</html>
```

## Директивы из стандартной библиотеки

### корректный адрес (ng-src, ng-href)

The buggy way to write it:

```
<a href="http://www.gravatar.com/avatar/{{hash}}">link1</a>  

```

The correct way to write it:

```
<a ng-href="http://www.gravatar.com/avatar/{{hash}}">link1</a>  

```



# Директивы из стандартной библиотеки

## показ/сокрытие (ng-show / ng-hide)

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Counter</title>
  </head>
  <body ng-app>
    <div ng-init="count = 0">
      <a href="#" ng-click="count = count + 1">Counter: {{count}}</a>
      <p><span ng-show="count % 2 !== 0">no</span>even</p>
    </div>
    <script src="angular.min.js"></script>
  </body>
</html>
```

## Директивы из стандартной библиотеки

### цикл forEach (ng-repeat)

```
<div ng-repeat="n in [42, 42, 43, 43] track by $index">  
  {{n}}  
</div>
```

## Фильтры

$$Y = F(p_1, \dots, p_N, X)$$

Можно применять несколько сразу:

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

Можно параметризировать:

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

## Стандартные фильтры

currency — приведение аргумента к формату вывода валюты;

date - приведение аргумента к формату вывода даты;

filter — принимает массив;

json -превращает аргумент в форматированную json-строку;

limitTo — вырезает из массива подмассив;

number — задает формат вывода чисел;

orderBy — сортируют массив;

uppercase/lowercase — изменяют регистр входной строки;

## Пишем свою директиву

```
myApp.directive('lyricsList', function() {  
  return {  
    scope:{  
      lyrics:'=lyrics'  
    },  
    controller: function($scope) {  
      this.lyrics = $scope.lyrics;  
      this.tab = 0;  
  
      this.isSowTab = function(checkTab) {  
        return this.tab === checkTab;  
      };  
  
      this.setTab = function(activeTab) {  
        this.tab = activeTab;  
      };  
    },  
    controllerAs: "tabCtrl",  
    templateUrl:"lyricsList.html"  
  }  
});
```

## Полезняшки

Twitter Bootstrap;

NPM / NodeJS;

WebJars.ORG;