AngularJS

JavaScript
Courses

vk.com/**js.courses**

js.courses.dp.ua/files

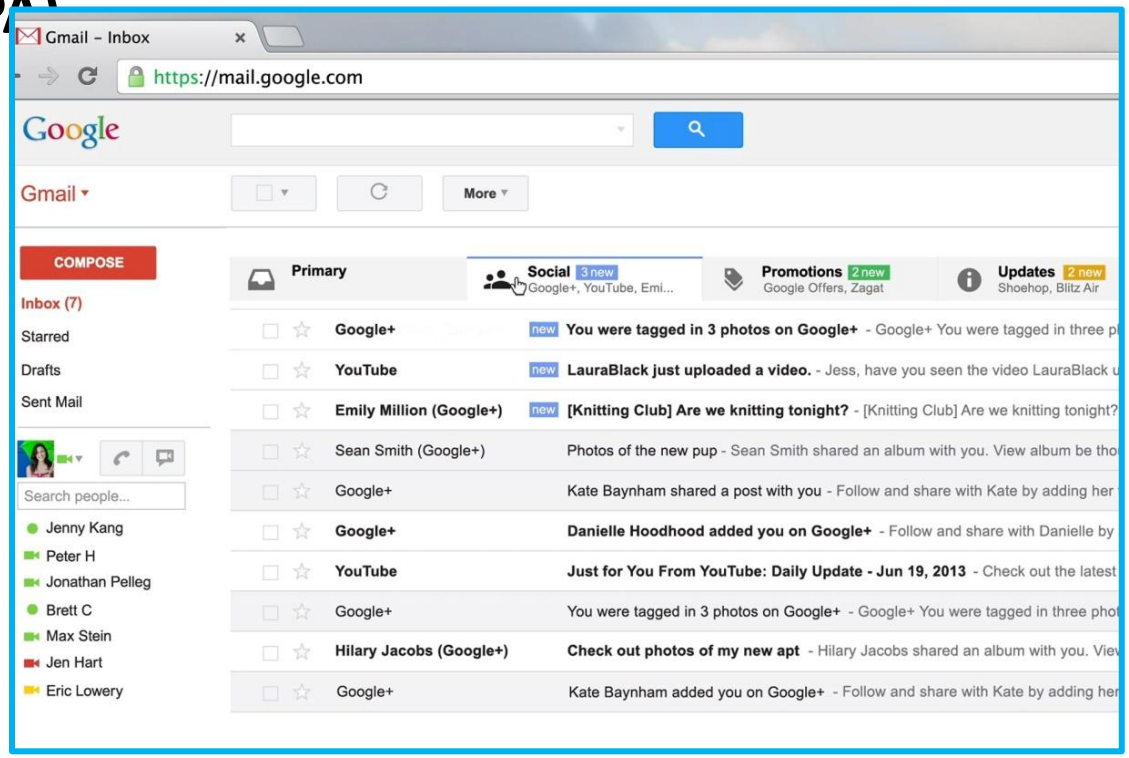
Еще раз про прототипы, или как библиотеки оборачивают объекты

```
<html>
<head>
    <script>
        function $$(element){
            element.make_red = function(){
                this.style.color = "red";
                return this;
            };
            return element;
        }

        window.onload = function(){
            var h1  = document.querySelector("h1");
            var div = document.querySelector("div");

            $$(h1).make_red();
            div.make_red();
        }
    </script>
</head>
<body>
    <h1>Header</h1>
    <div>Text</div>
</body>
</html>
```

???

Как js-библиотеки оборачивает своими функциями объекты

Одностраничное приложение (single page application - SPA)



Gmail *типичный пример SPA*

SPA *– это веб-приложение, размещенное на одной странице, которая для обеспечения работы загружает все JS и CSS файлы вместе с загрузкой самой страницы (или по ходу работы AJAX-ом).*

https://ru.wikipedia.org/wiki/Одностраничное_приложение

Императивный подход в программирование и декларативный

Императивный подход *– состоит в описании того как достигнуть требуемый результат, т.е. необходимо написать последовательность действий, выполнив которую будет получен результат.* ***JavaScript*** *– императивный язык.*

Декларативный подход *– состоит в описании того что необходимо получить, оставив порядок выполнения и способ достижения результата «на усмотрение» компьютера.* ***HTML&CSS*** *– представители декларативного подхода.*

AngularJS

попытка внести в JS декларативные

```
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js">
```

AngularJS *–* ***Javascript*** *библиотека, и как любая* ***JS****-библиотека не привносит ничего такого, чего нельзя написать на чистом* ***JS****, но как и все библиотеки направлена на уменьшение объема кода который пишет разраб*

v.1 https://angularjs.org/ **v.2 Beta**

Директивы – основа AngularJS

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
5  </head>
6  <body ng-app="">
7      Ваше имя: <input type="text" ng-model="user_name">
8      <h1>Привет {{user_name}}!</h1>
9  </body>
10 </html>
```

???

*Директива **ng-app** – определяет тег в рамках которого будет «крутится» Angular приложение.*

*Директива **ng-model** – определяет имя под которым будет доступного значение из поля ввода (модель).*

{{x}} – директива вставляет данные из модели в тело документа.

http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js

Связка данных (data

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js">
   </script>
5  </head>
6  <body ng-app="" ng-init="price=75">
7      Выберите цену: <input type="range" ng-model="price">
8      <h1>Цена: {{price}}$</h1>
9  </body>
10 </html>
```

???

ng-init – *директива задающая стартовые значения для модели.*

Связка данных (data

```
<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js">
</script>
</head>
<body ng-app="" ng-init="price=75; quantity=1">
    Цена: <input type="range" ng-model="price"><span> {{price}}$</span>
    <br>
    Кол-во: <input type="range" ng-model="quantity"> <span>{{quantity}} ед.</span>
    <h1>Итого: {{price*quantity}}$</h1>
</body>
</html>
```

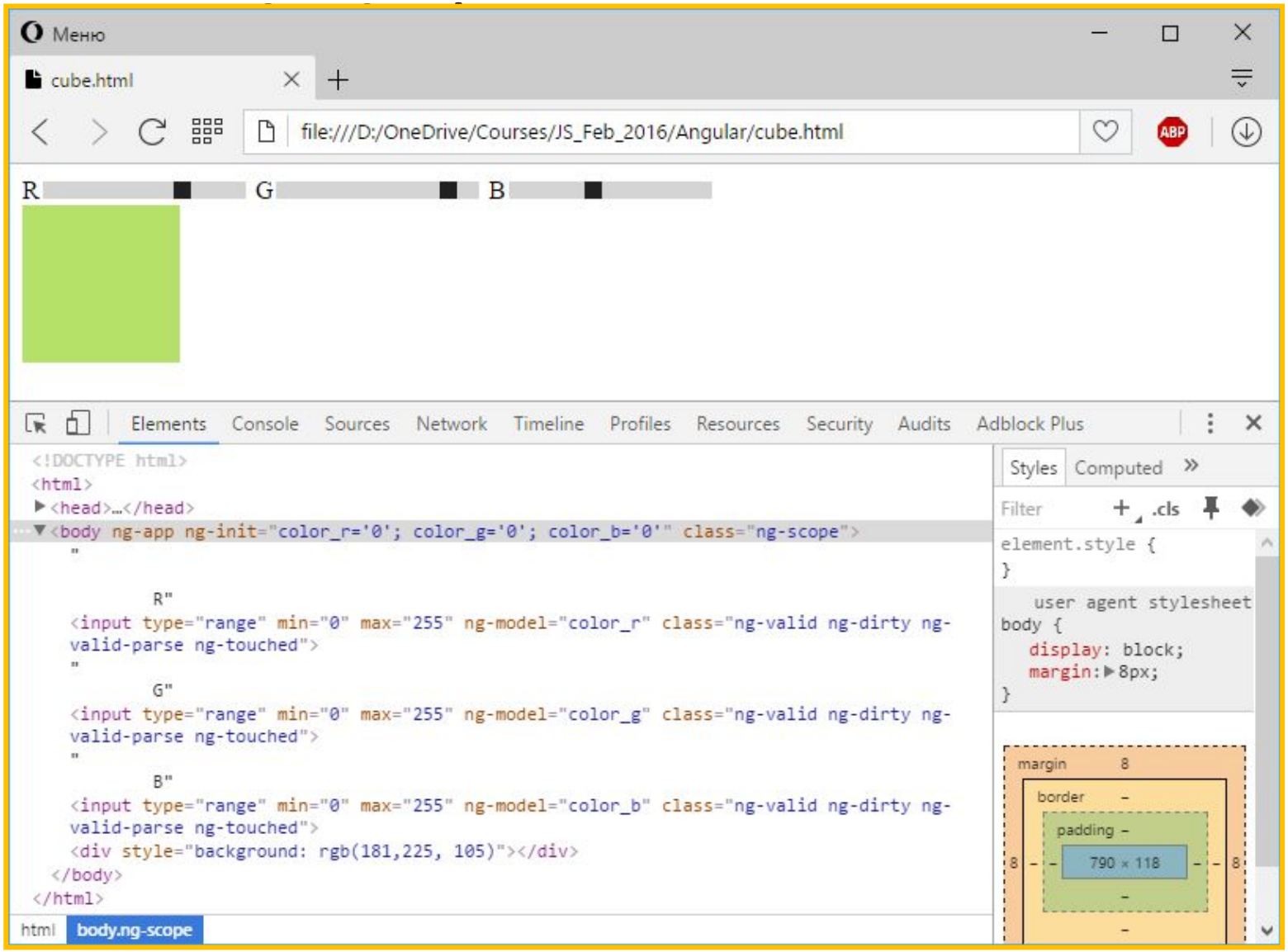
???

Связка данных (data binding)

```
<!DOCTYPE html>
<html>
<head>
    <style>
    div {
        width: 100px;
        height: 100px;
        background: rgb(0, 0, 0)
    }
    </style>
    <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js
    "></script>
</head>
<body ng-app="" ng-init="color_r='0'; color_g='0'; color_b='0'">

        R<input type="range" min="0" max="255" ng-model="color_r" >
        G<input type="range" min="0" max="255" ng-model="color_g" >
        B<input type="range" min="0" max="255" ng-model="color_b" >

<div style="background: rgb({{color_r}},{{color_g}}, {{color_b}})"></div>
</body>
</html>
```

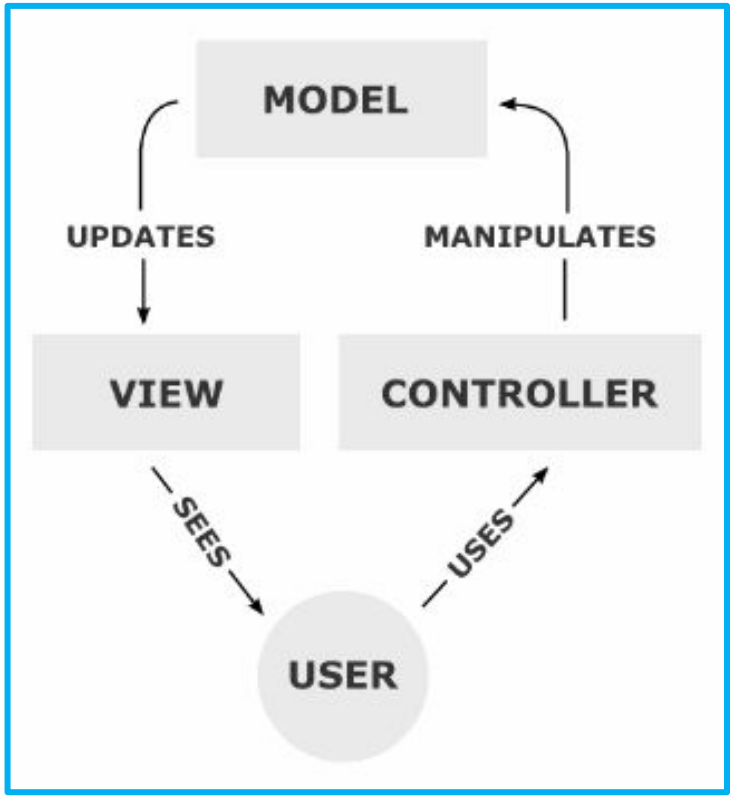
ColorPicker с тонкой настройкой без единой строчки **JS**-кода.

Связка данных (data



ColorPicker с тонкой настройкой без единой строчки **JS**-кода.

Model-view-controller (**MVC**, «модель-представление-контроллер», «модель-вид-контроллер»).



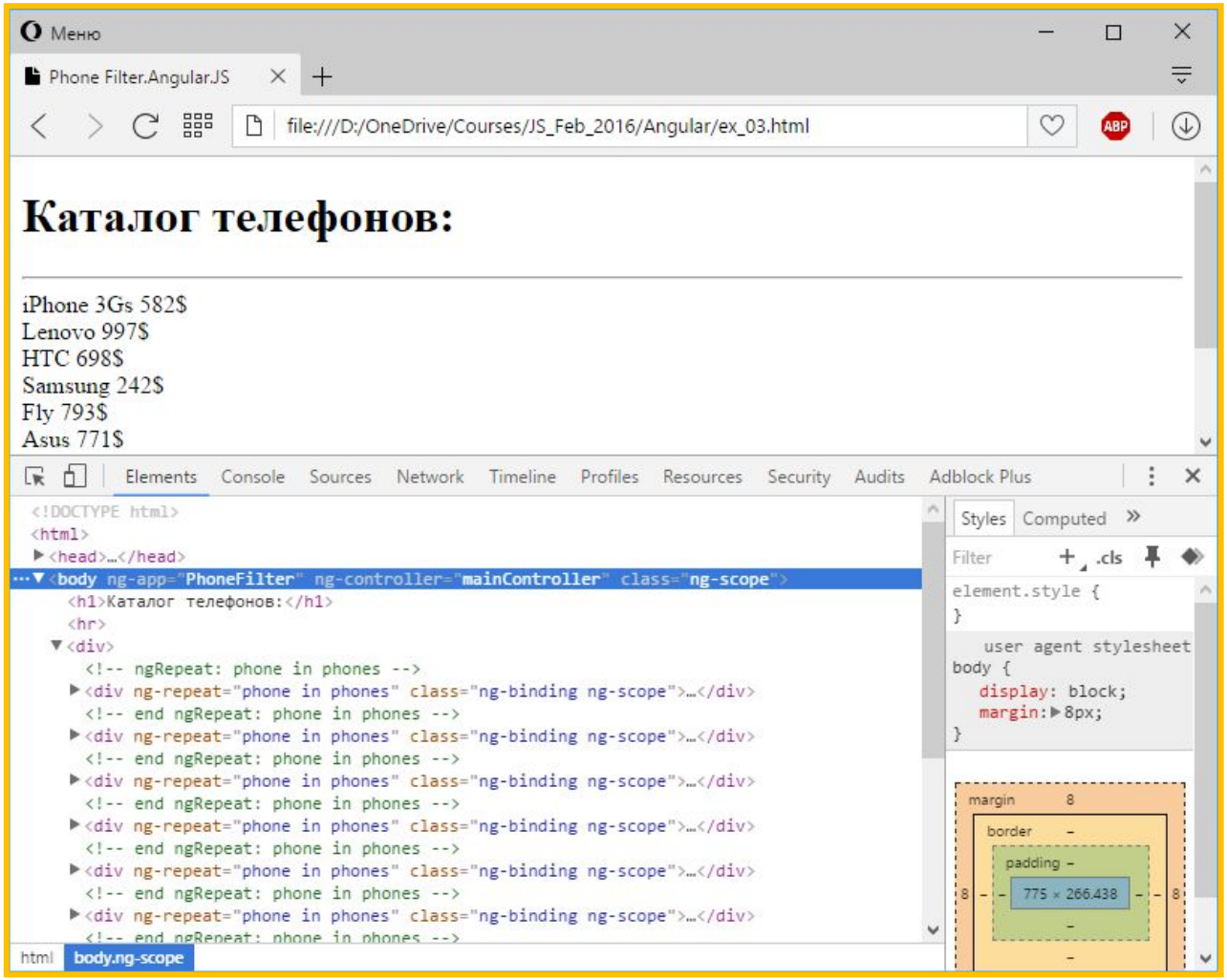
Основная цель применения этой концепции состоит в отделении бизнес-логики (модели) от её визуализации (представления, вида).

Вывод данных (директива

```
<!DOCTYPE html>
<html>
<head>
    <title>Phone Filter.Angular.JS</title>
    <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
    <script>
        var phones_arr = [
            { "name": "iPhone 3Gs", "price": "582$" },
            { "name": "Lenovo", "price": "997$" },
            { "name": "HTC", "price": "698$" },
            { "name": "Samsung", "price": "242$" },
            { "name": "Fly", "price": "793$" },
            { "name": "Asus", "price": "771$" },
            { "name": "Nokia", "price": "108$" },
            { "name": "LG", "price": "214$" },
            { "name": "IPhone 5s", "price": "216$" },
            { "name": "IPhone 6+", "price": "326$" },
            { "name": "IPhone 6s+", "price": "799$" }
        ];

        var app = angular.module('PhoneFilter', []);
        app.controller('mainController', function($scope){
            $scope.phones = phones_arr;
        });
    </script>
</head>
<body ng-app="PhoneFilter" ng-controller="mainController">
    <h1>Каталог телефонов:</h1>
    <hr>
    <div>
        <div ng-repeat="phone in phones">{{phone.name}} <span>{{phone.price}}</span></div>
    </div>
</body>
</html>
```

Возьмите шаблон с:
http://js.courses.dp.ua/files/angular/ex03_clear.html

Вывод данных (директива



*Директива **ng-repeat** позволяет многократно повторить вывод фрагмента разметки, вставляя в неё данные из указанного массива.*

Асинхронная загрузка и вывод данных

```
<!DOCTYPE html>
<html>
<head>
    <title>Phones Filter.Angular.JS</title>
    <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
    <script>
        var json_url = 'http://js.courses.dp.ua/files/angular/get_phones_json.php';
        var app = angular.module('PhoneFilter', []);
        app.controller('mainController', function($scope, $http){
            $http.get(json_url).then(function(res){
                $scope.phones = res.data;
                });
            });
    </script>
</head>
<body ng-app="PhoneFilter" ng-controller="mainController">
    <h1>Каталог телефонов:</h1>
    <hr>
    <div>
        <div ng-repeat="phone in phones">{{phone.name}} <span>{{phone.price}}</span></div>
    </div>
</body>
</html>
```

В

???

$http – объект для выполнения **ajax** запросов.

http://js.courses.dp.ua/files/angular/get_phones_json.php

Фильтрация

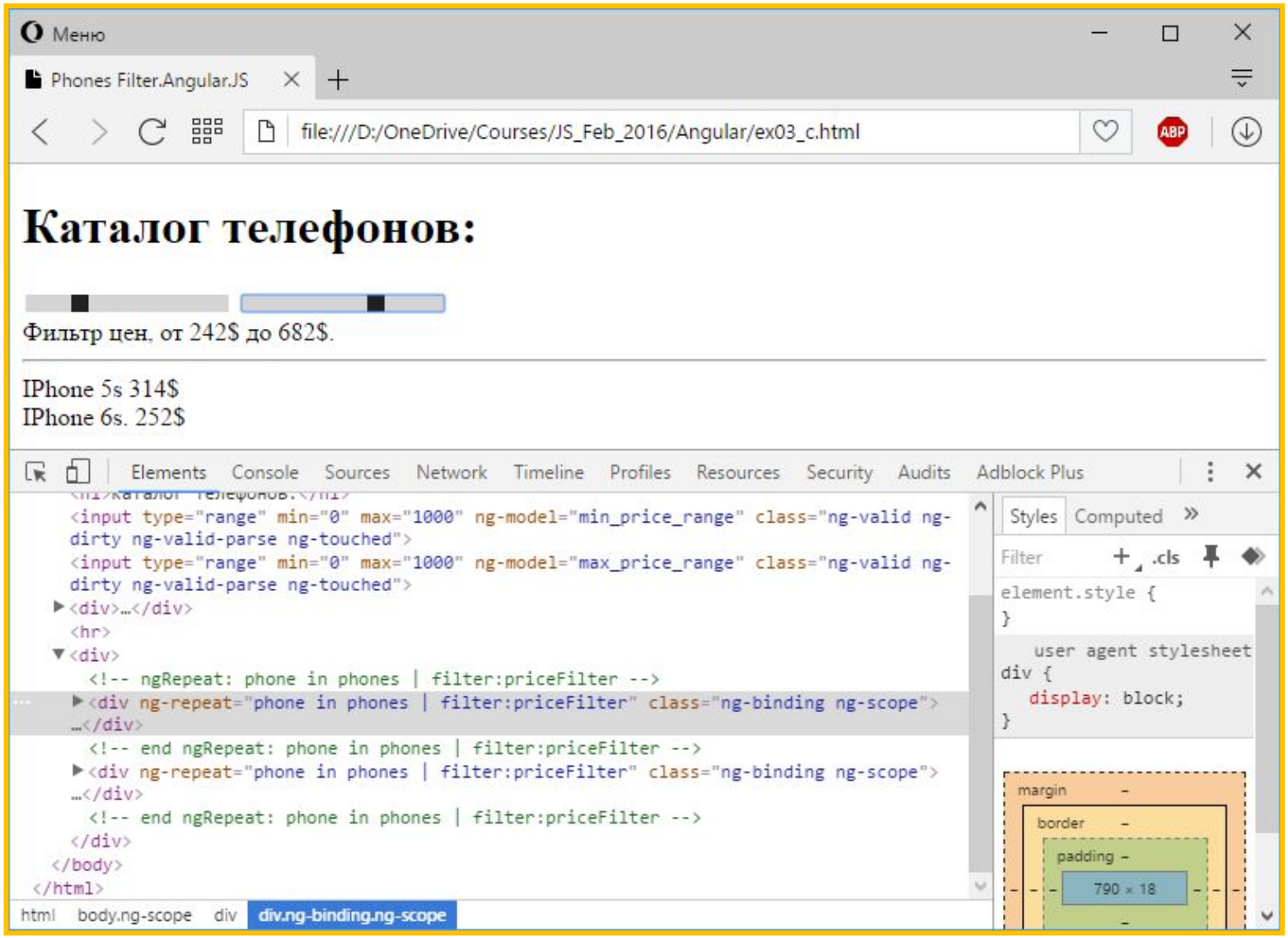
```
<!DOCTYPE html>
<html>
<head>
    <title>Phones Filter.Angular.JS</title>
    <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
    <script>
        var json_url = 'http://js.courses.dp.ua/files/angular/get_phones_json.php';
        var app = angular.module('PhoneFilter', []);
        app.controller('mainController', function($scope, $http){

            $scope.min_price_range = 0;
            $scope.max_price_range = 1000;

            $http.get(json_url).then(function(res){
                $scope.phones = res.data;
            });

            $scope.priceFilter = function(data){
                return (parseInt(data.price) >= parseInt($scope.min_price_range))
                        && (parseInt(data.price) <= parseInt($scope.max_price_range));
            }
        });
    </script>
</head>
<body ng-app="PhoneFilter" ng-controller="mainController">
    <h1>Каталог телефонов:</h1>
        <input type="range"  min="0" max="1000" ng-model="min_price_range">
        <input type="range"  min="0" max="1000" ng-model="max_price_range">
        <div>
            Фильтр цен, от <span ng-bind="min_price_range">0</span>$
            до <span ng-bind="max_price_range">1000</span>$.
        </div>
    <hr>
    <div>
        <div ng-repeat="phone in phones | filter:priceFilter">{{phone.name}} <span>{{phone.price}}</span></div>
    </div>
</body>
</html>
```

С

???

*Директива **filter** позволяет фильтровать выводимы данные, по каким-либо правилам.*

Фильтрация



*Директива **filter** позволяет фильтровать выводимы данные, по каким-либо правилам.*

Сортировка

```
<!DOCTYPE html>
<html>
<head>
    <title>Phones Filter.Angular.JS</title>
    <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
    <script>
        var json_url = 'http://js.courses.dp.ua/files/angular/get_phones_json.php';
        var app = angular.module('PhoneFilter', []);
        app.controller('mainController', function($scope, $http){

            $scope.min_price_range = 0;
            $scope.max_price_range = 1000;

            $http.get(json_url).then(function(res){
                $scope.phones = res.data;
            });

            $scope.priceFilter = function(data){
                return (parseInt(data.price) >= parseInt($scope.min_price_range))
                        && (parseInt(data.price) <= parseInt($scope.max_price_range));
            }
        });
    </script>
</head>
<body ng-app="PhoneFilter" ng-controller="mainController">
    <h1>Каталог телефонов:</h1>
        <input type="range"  min="0" max="1000" ng-model="min_price_range">
        <input type="range"  min="0" max="1000" ng-model="max_price_range">
        <div>
            Фильтр цен, от <span ng-bind="min_price_range">0</span>$
            до <span ng-bind="max_price_range">1000</span>$.
        </div>
    <hr>
    <div>
        <div ng-repeat="phone in phones | orderBy:'price' | filter:priceFilter">{{phone.name}}
            <span>{{phone.price}}</span>
        </div>
    </div>
</body>
</html>
```

D

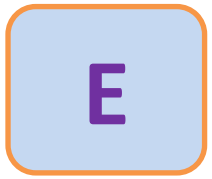
???

Директива orderBy позволяет указать критерий сортировки

Сортировка данных

Добавьте следующие стили:

```
<style>
    button:after { content: '\25b2'; }
    button.toggleclass:after { content: '\25bc'; }
</style>
```



Добавьте следующий код в кон

```
<script>
    var json_url = 'http://js.courses.dp.ua/files/angular/get_phones_json.php';
    var app = angular.module('PhoneFilter', []);
    app.controller('mainController', function($scope, $http){

        $scope.min_price_range = 0;
        $scope.max_price_range = 1000;

        $http.get(json_url).then(function(res){
            $scope.phones = res.data;
        });

        $scope.priceFilter = function(data){
            return (parseInt(data.price) >= parseInt($scope.min_price_range))
                    && (parseInt(data.price) <= parseInt($scope.max_price_range));
        }

        $scope.reverse = false;
        $scope.order_reverse = function(){
            $scope.reverse = !$scope.reverse;
        }

    });
</script>
```

Директива orderBy позволяет указать критерий сортировки. И порядок сортировки можно динамически

Сортировка данных

Добавьте следующую разметку в

```
<body ng-app="PhoneFilter" ng-controller="mainController">
    <h1>Каталог телефонов:</h1>
        <input type="range"  min="0" max="1000" ng-model="min_price_range">
        <input type="range"  min="0" max="1000" ng-model="max_price_range">
        <div>
            Фильтр цен, от <span ng-bind="min_price_range">0</span>$
            до <span ng-bind="max_price_range">1000</span>$.
        </div>
    <hr>
    <div>
        Сортировать по цене:
        <button ng-click="order_reverse()" ng-class="{toggleclass:reverse}"></button>
    </div>
    <hr>
    <div>
        <div ng-repeat="phone in phones | orderBy:'price':reverse | filter:priceFilter ">
            {{phone.name}} <span>{{phone.price}}</span>
        </div>
    </div>
</body>
</html>
```

E

???

Директива orderBy позволяет указать критерий сортировки. И порядок сортировки можно динамически менять.

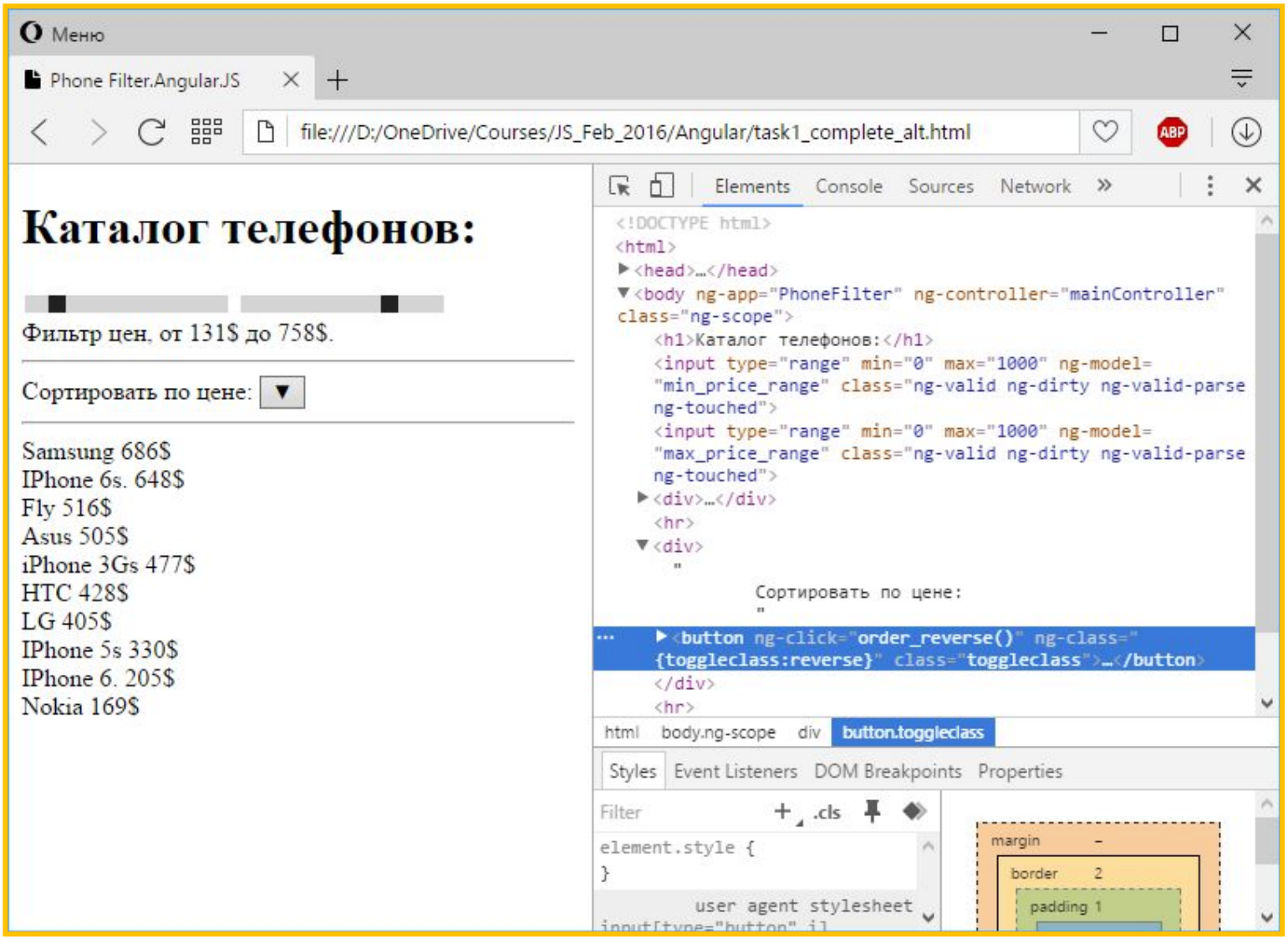
Сортировка данных (полный код)

```
<!DOCTYPE html>
<html>
<head>
    <title>Phone Filter.Angular.JS</title>
    <style>
        button:after { content: '\25b2'; }
        button.toggleclass:after { content: '\25bc'; }
    </style>
    <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
    <script>
        var json_url = 'http://js.courses.dp.ua/files/angular/get_phones_json.php';
        var app = angular.module('PhoneFilter', []);
        app.controller('mainController', function($scope, $http){
            $scope.min_price_range = 0;
            $scope.max_price_range = 1000;

            $http.get(json_url).then(function(res){
                $scope.phones = res.data;
            });

            $scope.priceFilter = function(data){
                return (parseInt(data.price) >= parseInt($scope.min_price_range))
                        && (parseInt(data.price) <= parseInt($scope.max_price_range));
            }

            $scope.reverse = false;
            $scope.order_reverse = function(){
                $scope.reverse = !$scope.reverse;
            }
        });
    </script>
</head>
<body ng-app="PhoneFilter" ng-controller="mainController">
    <h1>Каталог телефонов:</h1>
        <input type="range"  min="0" max="1000" ng-model="min_price_range">
        <input type="range"  min="0" max="1000" ng-model="max_price_range">
        <div>
            Фильтр цен, от <span ng-bind="min_price_range">0</span>$
            до <span ng-bind="max_price_range">1000</span>$.
        </div>
    <hr>
    <div>
        Сортировать по цене:
        <button ng-click="order_reverse()" ng-class="{toggleclass:reverse}"></button>
    </div>
    <hr>
    <div>
        <div ng-repeat="phone in phones | orderBy:'price':reverse | filter:priceFilter ">
            {{phone.name}} <span>{{phone.price}}</span>
        </div>
    </div>
</body>
</html>
```

E

???

Сортировка



Директива orderBy позволяет указать критерий сортировки. И порядок сортировки можно динамически менять.

Помощь при валидации

```
<html>
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
<body ng-app="">
    <form name="my_form">
        <p>Name:
            <input name="user_name" ng-model="user_name" required>
            <span ng-show="my_form.user_name.$touched && my_form.user_name.$invalid">
                The name is required.
            </span>
        </p><p>Email:
            <input type="email" name="user_email" ng-model="user_email" required>
            <span ng-show="my_form.user_email.$touched && my_form.user_email.$invalid">
                The email is required.
            </span>
        </p><p>
            <input type="submit" ng-disabled="my_form.user_name.$invalid ||
            my_form.user_email.$invalid">
        </p>
    </form>
</body>
</html>
```

Name: The name is required.

Email: The email is required.

Отправить

*Директива **ng-show** – показывает либо скрывает элемент в заивисмости от значения данных в модели. **ng-disable** – выключает либо включает элемент ввода в заивисмости от значения данных в модели.*

Помощь при валидации данных

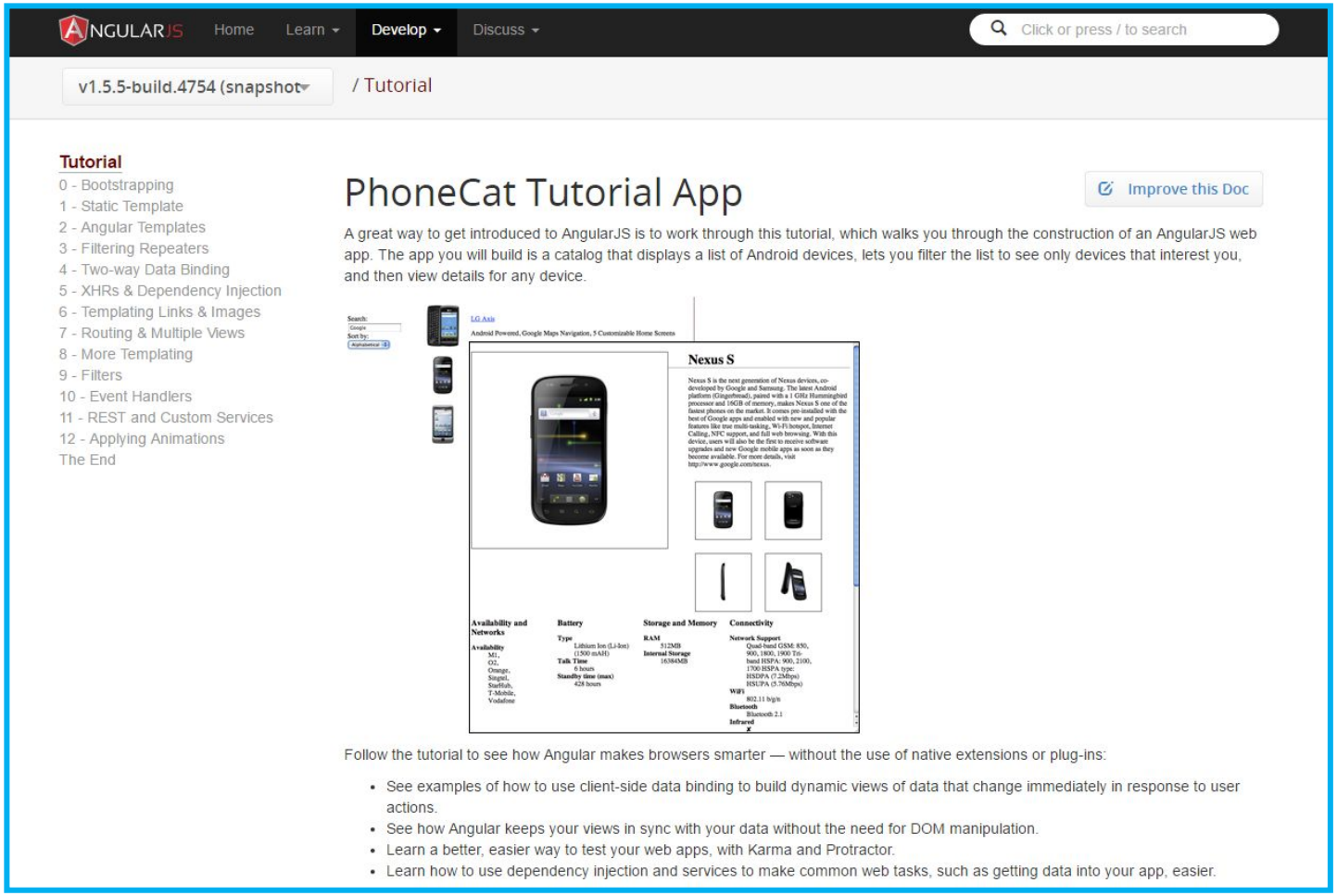
AngularJS добавляет к полям формы следующие свойства типа boolean:

```
ng-show="my_form.user_email.$touched && my_form.user_email.$invalid">
```

$untouched – поле еще не получало фокус;
$touched – поле минимум 1 раз получало фокус;
$pristine – поле не было изменено (в нём может быть значение по умолчанию);
$dirty – поле было изменено по сравнению с версией по умолчанию;
$invalid – содержимое поля не проходит валидацию;
$valid – содержимое поля проходит валидацию.

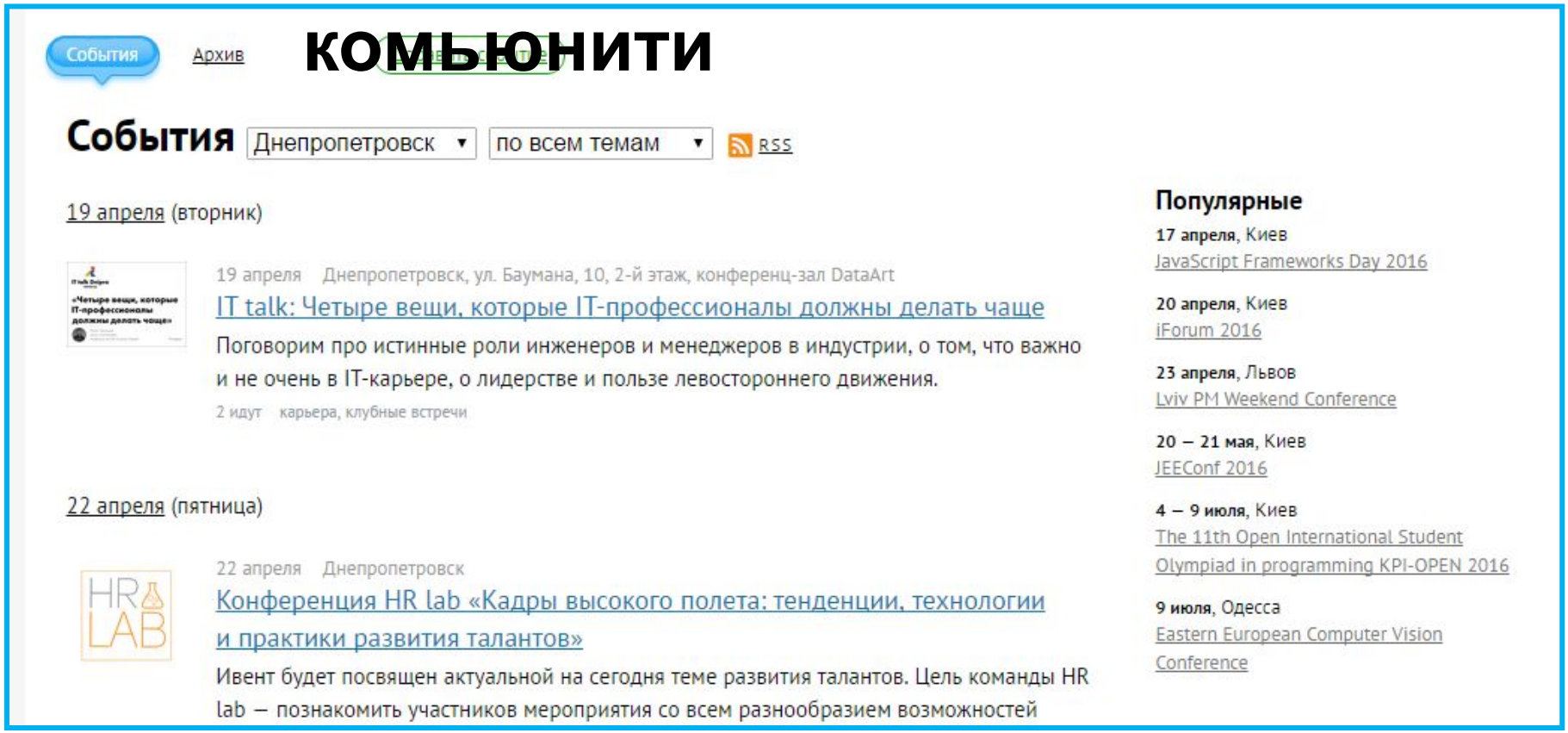
http://www.w3schools.com/angular/angular_validation.asp

По мнению google лучший способ освоить AngularJS пройти обучалку



https://docs.angularjs.org/tutorial

Пора вливаться в комьюнити



http://dou.ua/calendar/city/Днепропетровск

Если вы по описанию не понимаете о чём мероприятие – это верный признак того, что вам нужно на него сходить!!!