



Директивы стилизации и управления элементами

Последнее обновление: 1.11.2015



Изучай английский по Skype!

В особую группу директив можно выделить директивы, которые позволяют стилизовать элементы или управлять ими:

- **ng-style**: устанавливает у элемента стили CSS
- **ng-class**: задает элементу определенный класс
- **ng-class-even**: устанавливает класс для четных элементов
- **ng-class-odd**: устанавливает класс для нечетных элементов
- **ng-hide**: скрывает элемент
- **ng-show**: делает элемент видимым
- **ng-if**: задает для элемента некоторое условие

ng-class и ng-style

Данные директивы служат стилизации элементов. Объявим в контроллере два объекта:

```
$scope.somestyle={background: '#eee', color: '#333'};  
$scope.someclass="myclass";
```

myclass в данном случае - это название класса, который содержит определение в файле стилей CSS. И на страничке будет такая разметка:

```
<div ng-controller="phonesController">  
  <div ng-style="somestyle">Директива ng-style</div>  
  <div ng-class="someclass">Директива ng-class</div>  
</div>
```

Директива `ng-style`
Директива `ng-class`

ng-show и ng-hide

Директивы `ngshow` и `nghide` служат для отображения и скрывания элемента соответственно. Например, создадим в контроллере еще один объект для хранения булевых значений:

```
$scope.data={visible : false};
```

В разметке страницы создадим два параграфа и флажок, переключая который, мы будем управлять видимостью параграфов:

```
<div ng-controller="phoneController">  
<p ng-show="data.visible">Первый блок</p>  
<p ng-hide="data.visible">Второй блок</p>  
<input type="checkbox" ng-model="data.visible">Открыть/скрыть  
</div>
```

Первоначально у нас объект `data.visible` равен `false`, поэтому первый блок будет не виден. Изменяя значение флажка, который привязан к объекту `visible` опять же через директиву `ng-model`, мы также изменяем значение `data.visible`, что изменяет видимость параграфов.

Первый блок
☒ Открыть/скрыть

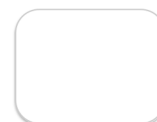
ng-if

Действие `ng-if` во многом похоже на действие `ng-show/ng-hide`. Если `ng-if` принимает значение `false`, то элемент с директивой удаляется из структуры DOM, а при значении `true`, наоборот, добавляется:

```
<div ng-controller="phoneController">  
<input type="checkbox" ng-model="data.visible">Открыть/скрыть  
<p ng-if="data.visible">Первый блок</p>  
<p ng-if="!data.visible">Второй блок</p>  
</div>
```

Galaxy S7 till superpris

Samsung Galaxy S7 med Fastpris. Allt ingår för 335 kr/mån!



[Назад](#) [Содержание](#) [Вперед](#)



4 Комментариев

metanit.com

 Войти Рекомендовать Поделиться

Лучшее в начале



Присоединиться к обсуждению...

**Илья** • 3 месяца назад

В чем преимущества ng-style перед атрибутом style и ng-class перед атрибутом class?

  • Ответить • Поделиться ›**Metanit** Модератор → Илья • 3 месяца назад

ng-style и ng-class позволяют динамически задавать стили, например, на основе данных в scope

  • Ответить • Поделиться ›**Dmitry Luchkin** • 6 месяцев назад

а когда имеет смысл просто скрывать, а когда полностью удалять?

  • Ответить • Поделиться ›**apchehov** → Dmitry Luchkin • 6 месяцев назад

Скорее всего смысл скрывать есть в том случае, когда представление на странице может быть отображено пользователем несколько раз (например, содержимое табов).

Удалять стоит, соответственно, когда пользователь видит представление лишь единожды (например, заглушка для последующего вывода результата асинхронного запроса).

Необязательные к прочтению размышления:

Даже если Ангулар кэширует содержимое блока ng-if, то его повторное внедрение в DOM (после удаления) - операция более ресурсоёмкая, чем скрытие/отображение содержимого. Плюс, когда я использовал backbone.js и его форки было такое: после внедрения в DOM новых объектов (уже после генерации страницы) требовалось отслеживать этот момент и заново оборачивать элементы плагинами (например, бутстрэпом). В случае с Ангулар и его системой мониторинга, этого "головняка" может и не быть, хотя я не проверял...

  • Ответить • Поделиться ›

ТАКЖЕ НА METANIT.COM

UWP | История навигации

2 комментария • 3 месяца назад

Yury Yatskov — Отбой! В новых шаблонах в xaml установлено свойство NavigationCacheMode = "Enabled". В старых

Паттерны в C# и .NET | Принцип инверсии зависимостей

1 комментарий • 2 месяца назад

андрій саврій — Всі статті по SOLID дуже цікаві!

IWP | CommandBar

ASP.NET MVC 5 | Действия по



[r](#) | [Google+](#) | [Канал сайта на youtube](#) | [Помощь сайту](#)

м: metanit22@mail.ru

t.com, 2012-2016. Все права защищены.