Работа c DOM

|  |  |  |  |
| --- | --- | --- | --- |
| № урока: | 3 | Курс: | AngularJS |
|  | | |  |
| Средства обучения: | | | Visual Studio 2013 Web Express или коммерческая версия |

# Обзор, цель и назначение урока

Научится работать со встроенными директивами AngularJS для работы с DOM, манипулированием элементами разметки, обработкой событий и валидацией пользовательского ввода.

**Изучив материал данного занятия, учащийся сможет:**

* Использовать встроенные директивы AngularJS для манипулирования элементами DOM.
* Понимать работу этих директив и их ключевые отличия.
* Обрабатывать стандартные события JavaScript используя AngularJS.
* Создавать свои директивы angular для обработки событий.
* Использовать специальные атрибуты AngularJS.

# Содержание урока

1. Работа над добавлением, удалением, отображением DOM элементов.
2. Работа со стилями.
3. Обработка событий.
4. Работа с формами.
5. Валидация пользовательского ввода.

# Резюме

* Для отображения или скрытия элементов используются директивы **ng-show** и **ng-hide**. В зависимости от условия они добавляют или удаляют у элемента css атрибут display: none. В таком случае элемент просто скрывается от пользователя, но продолжает содержатся в DOM что может повлечь за собой некоторые последствия. Основное различие между ng-show и ng-hide состоит в том, что ng-show прячет элемент, когда выражение равно false, а ng-hide, когда выражение равно true.
* Для удаления или добавления элемента в DOM в зависимости от определенного условия, значение которого может определятся привязкой данных следует использовать директиву **ng-if**. Эта директива в отличии от ng-show & ng-hide директив не скрывает видимость элемента, а попросту удаляет или добавляет его в DOM.
* Для управления стилями в AngularJS используются директивы **ng-class** и **ng-style**. Директива ng-class используется для применения css класса к элементу, ng-style применяет стиль непосредственно к элементу, но при этом предоставляет возможность динамически изменять значения как css атрибутов, так и их содержимого. Так же существуют вспомогательные директивы **ng-class-even** и **ng-class-odd** для указания css класса четным или нечетным элементам.
* AngularJS предоставляет множество оберток над стандартными событиями JavaScript которые позволяют обрабатывать эти же события в приложении angular. Но в AngularJS не предусмотрена событийная модель DOM level 2 что может показаться странным, так что здесь придётся задавать обработчики на события прямо в разметке.

<button ng-click=“handler ($event)”>

Так же angular предоставляет вам возможность создавать свои директивы для обработки событий.

* В angular присутствует ряд оберток над стандартными атрибутами: readonly, disabled, checked и т.д. Это сделано для того чтобы дать возможность использовать Boolean значения с этими атрибутами, так как стандартный функционал этого не предусматривает. Теперь появляется возможность применять тот или иной атрибут в зависимости от значения в модели

(true/false) что является очень удобно.

* Для валидации в AngularJS существует несколько переменных:
* $pristine возвращает true если пользователь не взаимодействовал с элементом
* $dirty возвращает true если пользователь взаимодействовал с элементом
* $valid возвращает true если форма и ее элементы валидны
* $invalid возвращает true если форма не валидна
* $error содержит информацию об ошибках валидации
* AngularJS для валидации применяет к элементам классы в зависимости от их состояния:
* ng-pristine в этот класс добавляются все элементы с которыми пользователь еще не взаимодействовал
* ng-dirty содержит элементы с которыми пользователь взаимодействовал
* ng-valid класс валидных элементов
* ng-invalid класс не валидных элементов

Далее определив стили в этих классах можно задавать желаемые стили для каждого этапа валидации

# Закрепление материала

* Объясните ключевые различия в работе ng-show/hide и ng-if?
* Как можно управлять стилями в angular?
* Перечислите специальные атрибуты angular?
* Какой атрибут нужно обязательно задавать для корректной валидации в Angular?
* Как работает применение определенных стилей к валидируемым элементам?

# Дополнительное задание

Задание

Создайте приложение в котором используется 3 темы для оформления контента, предоставьте пользователю возможность выбирать тему.

# Самостоятельная деятельность учащегося

Задание 1

Доработайте предыдущий пример добавив возможность при наведении курсора мыши на цвет темы демонстрировать применение этого цвета к контенту, но как только курсор выйдет за элемент в котором отображается цвет темы цвет контента должен вернутся обратно. Изменить цветовую схему можно по-прежнему только при нажатии на тему. Суть этого примера в создании preview при наведении на цветовые схемы.

Задание 2

Создайте форму для регистрации нового пользователя. Форма должна содержать поля:

ФИО, email, телефон, пароль. Реализуйте real-time валидацию пользовательского ввода.

# Рекомендуемые ресурсы

Ng-show:

<https://docs.angularjs.org/api/ng/directive/ngShow>

Ng-hide:

<https://docs.angularjs.org/api/ng/directive/ngHide>

Ng-class:

<https://docs.angularjs.org/api/ng/directive/ngClass>

Ng-style:

<https://docs.angularjs.org/api/ng/directive/ngStyle>

Ng-if:

https://docs.angularjs.org/api/ng/directive/ngIf