

Псевдоклассы и градиенты

№ урока: 12 Курс: Fronted-basic

Средства обучения: Компьютер с установленным Sublime Text 3

Обзор, цель и назначение урока

Научиться использовать псевдоклассы CSS3 создавать линейные и радиальные градиенты. Создавать приложения позволяющие использовать возможности оформления через CSS3 в старых браузерах.

Изучив материал данного занятия, учащийся сможет:

Создавать сложные селекторы используя новые псевдоклассы и псевдоэлементы. Настраивать линейные и радиальные градиенты.

Содержание урока

1. Псевдоклассы CSS3
2. Селекторы CSS3
3. Градиенты
4. Использование CSS3 PIE

Резюме

- Псевдокласс **:checked** применяется к элементам интерфейса, таким как переключатели (checkbox) и флажки (radio), когда они находятся в положение «включено».
- Псевдокласс **:default** применяет стиль к элементам форм, которые установлены по умолчанию в группе похожих элементов.
- Псевдокласс **:disabled** используется для применения стиля к заблокированным элементам форм.
- Такие элементы не могут получить фокус, быть нажатыми или активированы, в текстовых полях нельзя набирать текст.
- Псевдокласс **:empty** представляет пустые элементы, иными словами такие, которые не содержат дочерних элементов, текста или пробелов.
- Псевдокласс **:enabled** используется для применения стиля к доступным (не заблокированным) элементам форм.
- По умолчанию, все элементы форм являются доступными, если в коде HTML к ним не добавляется атрибут **disabled**.
- Псевдокласс **:first-of-type** задает правила стилей для первого элемента в списке дочерних элементов своего родителя.
- Псевдокласс **:indeterminate** задает стиль для элементов форм, таким как флажки и переключатели, когда они находятся в неопределенном состоянии.
- К примеру, если из группы флажков ни один не помечен, то флажки находятся в указанном состоянии. В реальности, стиль применяется только к элементам, у которых DOM-атрибут **:indeterminate** через JavaScript установлен в значение **true**.
- **:required** – применяет стилевые правила к тегу **<input>**, у которого установлен атрибут **required**. Он позволяет выделять поля обязательные к заполнению перед отправкой формы.
- **:optional** – применяет стилевые правила к тегу **<input>**, у которого не задан атрибут **required**. Он позволяет выделять поля обязательные к заполнению перед отправкой формы. Таким образом, **:optional** применяется к необязательным полям формы.
- **:read-write** – применяется к полям формы, доступных для изменения.
- **:read-only** – применяется к полям формы, у которых задан атрибут **readonly**. Такое поле не может быть модифицировано и получить фокус.

- Псевдокласс **:root** определяет корневой элемент документа. В HTML этот селектор всегда соответствует элементу `<html>`.
- **:invalid** – применяется к полям формы, содержимое которых не соответствует указанному типу. Например, для `type="number"` должно вводиться число, а не буквы, для `type="email"` – корректный адрес электронной почты.
- **:valid** – применяется к полям формы, содержимое которых проходит проверку в браузере на соответствие указанному типу.
- Псевдокласс **:last-child** задает стилевое оформление последнего элемента своего родителя.
- Псевдокласс **:not** задаёт правила стилей для элементов, которые не содержат указанный селектор.
- Псевдокласс **:nth-child** используется для добавления стиля к элементам на основе нумерации в дереве элементов.
- Псевдокласс **:nth-last-child** используется для добавления стиля к элементам на основе нумерации в дереве элементов.
- В отличие от псевдокласса `:nth-child` отсчет ведется не от первого элемента, а от последнего.
- Псевдокласс **:only-child** применяется к дочернему элементу, только если он единственный у родителя.
- Псевдокласс **:only-of-type** применяется к дочернему элементу указанного типа, только если он единственный у родителя. Аналогичен использованию `:first-of-type:last-of-type` или `:nth-of-type(1):nth-last-of-type(1)`.
- Свойство **appearance** изменяет внешний вид элемента интерфейса, при сохранении его функции.
- **элемент1 ~ элемент2** селектор выберет все вхождения элемента2, которые находятся в разметке после элемента1.
- **[атрибут\$= значение]** селектор, который будет обращаться к каждому элементу, значение атрибута которого заканчивается с указанного значения.
- **[атрибут^= значение]** селектор, который будет обращаться к каждому элементу, значение атрибута которого начинается с указанного значения.
- Поддержка градиента браузерами:
возможность применения CSS-градиента была реализована Webkit несколько лет назад, но он редко использовался из-за несовместимости с большинством браузеров. Но теперь, в Firefox 3.6+, который начал поддерживать градиент, мы можем создавать градиент без обязательного использования изображения. Opera добавила поддержку CSS-градиента начиная с версии 11.10. Ниже показано, как создать CSS-градиент, который будет поддерживаться основными браузерами: IE 10+, Firefox 3.6+, Safari, Opera 11.10+ и Chrome.
- Градиент для Webkit: `background: -webkit-gradient(linear, left top, left bottom, from(#ccc), to(#000));`
- Mozilla Firefox 3.6+: `background: -moz-linear-gradient(top, #ccc, #000);`
- Для Opera 11.10+: `background: -o-linear-gradient(top, #ccc, #000);`
- Градиент для IE 10+: `background: -ms-linear-gradient(top, #ccc, #000);`
- С помощью метода **radial-gradient** Вы можете создавать сферические градиенты. Синтаксис определения сферических градиентов очень похож на синтаксис линейных, но требует также задания формы градиента (может быть сферической или эллипсоидной).
- Повторяющиеся градиенты задаются с помощью CSS3 методов **repeating-linear-gradient** (создает повторяющийся линейный градиент) и **repeating-radial-gradient** (создает повторяющийся сферический градиент).

Закрепление материала

- Какие специальные типы селекторов Вы запомнили?
- Какие псевдоклассы задают стилевое оформление последнего элемента своего родителя? Какие первого?
- С помощью какого псевдокласса можно определить стиль элементов форм в которых допущены ошибки?
- С помощью какого псевдокласса можно сделать поочерёдную заливку разным цветом строк таблицы?

- Как можно определить линейный градиент для блока на странице?
- Как сделать правило которое будет добавлять градиент для всех браузеров включая старые версии IE?

Дополнительное задание

Создайте страницу регистрации пользователя. От пользователя необходимо принять следующие параметры – логин, пароль, подтверждение пароля, email, дату рождения. Разработайте набор стилевых правил которые будут: А) отображать возле обязательных полей звездочки; Б) заливать красным цветом те поля вводов где значения введены не правильно; Г) заливать зеленым цветом те поля, которые заполнены правильно.

Самостоятельная деятельность учащегося

Задание 1

Изучите основные понятия, рассмотренные на уроке.

Задание 2

Используя градиенты и псевдоклассы создайте кнопку со следующим оформлением



Укажите смену стиля при наведении мыши на кнопку.

Рекомендуемые ресурсы

Псевдоклассы

<http://htmlbook.ru/css/cat/pseudoclass>

Линейные градиенты

<http://www.css3files.com/gradient/>

Создание CSS градиентов

<http://www.colorzilla.com/gradient-editor/>

Поддержка градиентов в IE

<http://css3pie.com/>

Генератор кнопок

<http://css3button.net/99268>