

Трансформация и анимация, Градиенты

№ урока: 6 **Курс:** HTML5&CSS3 Advanced

Средства обучения: Visual Studio, Visual Studio Code, NotePad++

Обзор, цель и назначение урока

Цель урока – ознакомить студентов с новыми возможностями **CSS3** – трансформациями и переходами. Обозначить различие между переходом и анимацией. Разобрать как использовать анимацию на странице вместе с трансформацией. Выучить понятие градиента и научиться его применять в стилях.

Изучив материал данного занятия, учащийся сможет:

- Использовать резкие переходы на странице.
- Применять плавные анимации, используя задержки и функции скорости.
- Применять градиенты для заливки фона.
- Использовать **2D** и **3D** трансформации элементов.
- Подключать сторонние библиотеки **CSS** для работы с анимациями.

Содержание урока

1. Свойство **transition**;
2. Создание переходов;
3. Виды трансформации;
4. Матрица трансформации;
5. Совместное использование трансформации с переходом;
6. **3D** трансформации;
7. Анимации;
8. Использование градиентов.

Резюме

- **Transition** – свойство **CSS3**, описывающее перевод из одного состояния в другое. Для определения состояния после перехода – необходимо использовать псевдоклассы (**hover**, **active**).
- Свойства для использования перехода:
 - **transition-property** – определяет имя (имена) свойств **CSS**, к которым должны быть применены переходы.
 - **transition-duration** – определяет время, за которое должен произойти переход.
 - **transition-timing-function** – определяет скорость выполнения перехода. В **CSS3** существуют несколько видов таких функций: **linear**, **ease**, **ease-in-out**, **ease-in**, **ease-out**, **cubic-bezier** – ручное определение значения перемещения.
 - **transition-delay** – определяет задержку перед выполнением перехода.
- **Transitionend** – событие, определяемое после окончания выполнения перехода.
- Для выполнения трансформации используются следующие функции:
- **Scale(x,y)** – увеличивает или уменьшает элемент в зависимости от выбранного элемента.
- **Translate**: перемещает элемент вдоль трёх осей (**x**, **y** и **z**);
- **Rotate** – поворачивает элемент по углам вдоль осей;
- **Skew** – Используется для деформирования (искажения) сторон элемента относительно координатных осей.
- **Transform: matrix(a,b,c,d,x,y)** – матрица трансформации: **a** - масштаб по горизонтали, **d** - изменяет масштаб по вертикали; **b** - деформирует (сдвигает) стороны элемента по оси **X**, **c** - деформирует стороны элемента по оси **Y**; **x** - смещение элемента по оси **X**, **y** - смещение элемент по оси **Y**.

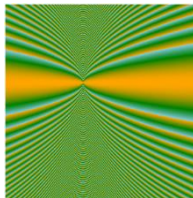
- **Transform-origin** – позволяет изменить центр трансформации, относительно которого, будут происходить изменения.
- **Perspective** – свойство активирует 3D-пространство для элемента. Свойство **perspective** добавляет глубину элементу, увеличивая его размеры по оси Z, сам элемент при этом становится визуально меньше. Чем меньше значение, тем ближе Z-пространство к зрителю и тем больше эффект, заданный с помощью свойства **transform**. Чем больше значение, тем менее выражен эффект. 0 означает отсутствие перспективы.
- Для создания анимации в **CSS3** используется свойство **@keyframes**.
- **Animation-name** – свойство, что задаёт имя анимации. Имя анимации создаётся в правиле **@keyframes**;
- **Animation-duration** - свойство устанавливает продолжительность анимации;
- **Animation-timing-function** - свойство определяет изменение скорости от начала до конца анимации с помощью временных функций;
- **Animation-delay** – задает задержку времени перед началом анимации;
- **Animation-iteration-count** - свойство позволяет определить количество повторений анимации;
- **Animation-direction** – свойство задает направление повтора анимации;
- **Animation-play-state** – свойство управляет проигрыванием и остановкой анимации;
- **Animation-fill-mode** – состояние элемента до и после воспроизведения анимации.
- **Transform-style** Свойство определяет, как вложенные элементы отрисовываются в трехмерном пространстве.
- **Linear-gradient** – линейный градиент создается с помощью двух и более цветов, для которых задано направление, или линия градиента.
- 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

- Назовите основные способы создания переходов?
- За что отвечает свойство **transform-origin**?
- За что отвечает свойство **animation-play-state**?
- Какое событие добавляется при реализации перехода в **CSS3**?
- Как добавить на один элемент несколько анимаций?
- Как можно определить линейный градиент для блока на странице?
- Как создать повторяющийся радиальный градиент?

Дополнительное задание

Используя полученные знания, выполните следующую заливку элемента.



Самостоятельная деятельность учащегося

Задание 1

Выучить основные понятия, рассмотренные на уроке.

Задание 2

Создайте горизонтальное меню и разработайте анимированные выпадающие подменю при наведении мыши.

Задание 3

Создайте приложение по следующему макету.

На странице должна быть отображена карта. На карте реализуйте 4-5 точек, с городами.

При загрузке страницы – каждая точка города пульсирует цветом.

При наведении – пульсация останавливается, и выводится маленькое окошко, предлагающее купить билеты. При отведении мышки – окошко прячется.

Для остановки анимации используйте свойство: **animation-play-state**.



Рекомендуемые ресурсы

<http://htmlbook.ru/>

<https://html.com/>

<https://www.html5rocks.com/>

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

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