# Занятие #7 text-shadow, box-shadow, градиентный фон, кнопки, использование псевдоклассов :hover, :focus, :active,:link, visited

В современных веб-дизайнах достаточно часто используются тени для оформления блоков и тени для текста. Текстовые тени добавляются при помощи CSS правила text-shadow, блочные тени - при помощи box-shadow.

## **text-shadow**

Свойство text-shadow добавляет тень тексту и устанавливает параметры тени: цвет тени, смещение относительно надписи и радиус размытия. Пример записи: text-shadow: 2px 5px 10px #000;

1-е значение (2px) – сдвиг тени по оси Х т.е. по горизонтали, положительное значение сдвиг вправо, отрицательное – влево;

2-е значение (5px) – сдвиг по оси Y т.е. по вертикали, положительное значение – вниз, отрицательное – вверх;

3-е значение (10px) – размытие тени

4-е значение (#000;) – цвет тени.

Можно цвет поставить на первое место: text-shadow: #000 2px 5px 10px; - так тоже будет верно.

Браузер Internet Explorer понимает свойство text-shadow только с версии 10.0. Для IE 9 и IE8 можно использовать свойство filter: Shadow (параметры).

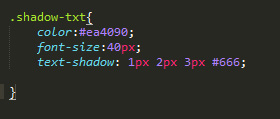
К примеру, следующая конструкция задает цвет тени (#666666), её направление (45° от вертикали) и величину смещения (4 пиксела): filter: Shadow(Color=#666666, Direction=45, Strength=4);

Создадим в html див с классом shadow-txt

****

И зададим параметры тексту (цвет розовый #ea4090 и размер текста 40px) и тени для текста (сдвиг по горизонтали влево 1px, по вертикали вниз 2px, размытие тени 3px,

цвет тени светло-серый (#666)

****

В браузере текст пример вид:

****

Допускается указывать несколько параметров тени, разделяя их между собой запятой. В CSS3 учитывается следующий порядок: первая тень в списке размещается на самом верху, последняя в списке — в самом низу.

Также как и для border-radius для того чтобы быстро и правильно подобрать тень для текста можно пользоваться генераторами кода:

<http://css3gen.com/text-shadow/>

<http://www.css3generator.in/text-shadow.html>

Заходите на страницу генератора тени, подбираете нужный вам вариант, копируете код и вставляете в Ваш css файл для нужного блока, параграфа, заголовка и т.д.

## box-shadow

Добавляет тень к элементу. Допускается использовать несколько теней, указывая их параметры через запятую, при наложении теней первая тень в списке будет выше, последняя ниже.

Если для элемента задается радиус скругления через свойство border-radius, то тень также получится с закругленными уголками. Также нужно учитывать, что тень увеличивает, размеры элемента.

Safari до версии 5.1, Chrome до версии 10.0, Android до версии 4.0 и iOS Safari до версии 5.0 поддерживают свойство -webkit-box-shadow.

Firefox до версии 4.0 поддерживает свойство -moz-box-shadow.

Internet Explorer до версии 9.0 не поддерживает свойство box-shadow, взамен можно использовать нестандартное свойство filter:

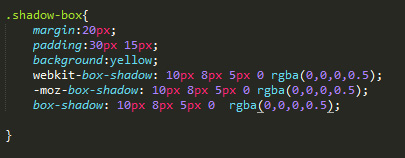
filter: progid:DXImageTransform.Microsoft.shadow(direction=120, color=#000000, strength=10);

direction — угол направления тени от 0 до 360°; color — цвет тени; strength — смещение тени в пикселях.

Создадим блок с классом shadow-box

****

Зададим для него следующие свойства:

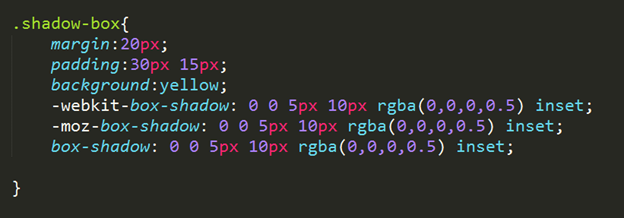
****

В браузере мы увидим следующую картинку:



Блок желтого цвета имеет тень, которая сдвинута по оси Х влево на 10 пикслей, по оси Y вниз на 8 пикселей., тень имеет размытие 5 пикселей и она не растянута (растяжение 0), цвет тени черный с прозрачностью 50% - rgba (0,0,0,0.5).

Тень может размещаться внутри блока, для этого в стилях нужно добавить inset. Например нам нужно чтобы тень была внутри и располагалась со всех сторон по контуру блока, изменим параметры тени: уберем смещение по осям Х и Y, добавим растяжение тени 10 пикселей:



И в браузере мы увидим, следующее:

****

## Псевдоклассы :link, :visited, :active, :focus

Как нам уже известно помимо обычных классов, в CSS можно задавать определенное оформление элементов используя псевдоклассы. Так, псевдокласс **:hover**  определяет стиль элемента при наведении на него курсора мыши.

Познакомимся еще с некоторыми псевдоклассами:

**:link** применяется к ссылкам, которые еще не посещались пользователем, и задает для них стилевое оформление.

**:visited** применяется к посещенным ссылкам.

**:active** определяет стиль для активной ссылки. Активной ссылка становится при нажатии на нее. Можно применять также для кнопок.

**:focus** определяет стиль для элемента получающего фокус. Например, им может быть текстовое поле формы, в которое устанавливается курсор.

Псевдоклассы :hover и :active или :focus широко применяются для того чтобы создавать эффект нажатия кнопок.

# Кнопки

Кнопки на страницах сайтов создаются тегами:

<button>текст кнопки</button> - универсальная кнопка

<input type=”button” value”текст кнопки”> - универсальная кнопка

<input type=”submit” value”текст кнопки”> - кнопка для отправки данных на сервер

<input type=”image” src="путь к картинке"> - выполняет те же функции, что и кнопка <input type=”submit”> , но ее внешний вид задается картинкой.

<input type=”reset” value”текст кнопки”> - кнопка для возвращения данных формы в первоначальное значение.

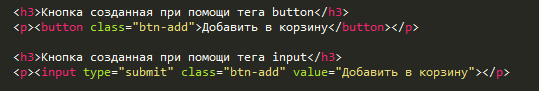
Также широко используется кнопка-ссылка (иными словами ссылка имеющая внешний вид кнопки). <a href=””> текст кнопки-ссылки </a>

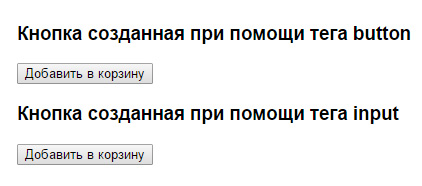
Кнопки <input type=”submit”> , <input type=”image”>, и <input type=”reset” > должны располагаться между открывающим и закрывающим тегами формы: <form> </form>

Теоретически, теги <button> и <input type=”button”> также должны располагаться внутри формы, устанавливаемой элементом <form>. Тем не менее, браузеры не выводят сообщение об ошибке и корректно работают с тегами <button>и <input type=”button”>, если они встречается самостоятельно. Однако, если необходимо результат нажатия на кнопку отправить на сервер, необходимо эти кнопки также обязательно помещать в контейнер <form>.

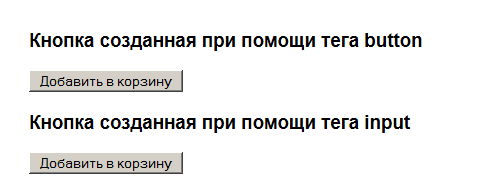
Кнопки по умолчанию имеют определенные стили, при этом в разных браузерах эти стили разные.

Создадим в браузере пару кнопок одну кнопку при помощи тега <button>, вторую при помощи тега <input type=”submit”>

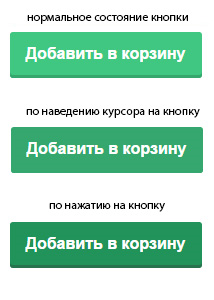


В хроме эти кнопки будут выглядеть одинаково таким образом:

В другом браузере эти кнопки могут выглядеть по другому, например так:



А по дизайну необходимо чтобы эти кнопки выглядели везде одинаково, например, так:

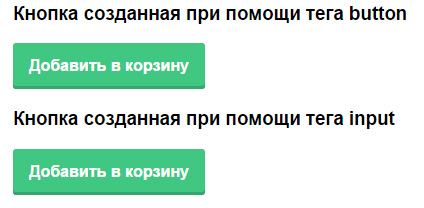
Проанализировав внешний вид кнопок мы видим, что кнопка имеет размеры: ширину 192 пикселя, высоту **46 пикселей**, шрифт - **Arial**, размер шрифта **16 пикселей**, вес шрифта – **bold**, цвет текста на кнопке - **белый**, фон у кнопки – в нормальном состоянии **#40c781**, и есть внутренняя тень снизу кнопки высотой 3 пикселя, цвет тени: **#35a76e**. По наведению на кнопку фон меняется на **#35a76e**, по нажатию фон становится #21935a и цвет тени меняется на #27734e. Также у кнопки есть скругление уголков радиусом 3 пикселя.

Добавим в css стили для кнопок:

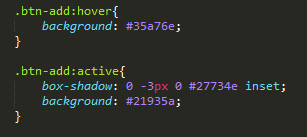


Мы обнулили border, т.к. у всех кнопок по умолчанию есть границы, задали нужный цвет фона, задали нужные размеры кнопке, задали стили для текста, обнулили паддинги, при помощи высоты строки (line-height) разместили текст по середине кнопки по вертикали. Задали тень, параметры тени сдвиг по горизонтали 0, сдвиг по вертикали -3px (т.е. вверх на 3 пикс.), размытие тени 0, цвет тени #35a76e, и указание что тень внутри блока inset. Добавили cursor:pointer (т.е. изменили вид курсора с дефолтного на пальчик, чтобы пользователю было поняытно что сюда можно кликнуть).

В браузере кнопки примут вид:



Добавим для кнопок стили по наведению и по нажатию:



И теперь кликнув на кнопку в браузере - мы увидим что появился эффект нажатия.

Мы можем добавить какую-нибудь иконку на кнопку, самый простой способ - это добавить иконку в фон кнопки.

Например нам нужно чтобы кнопка выглядела так:



Если на кнопке есть иконка то нужно чтобы для иконки было место (т.е. чтобы текст не наезжал на иконку) в этом случае можно:

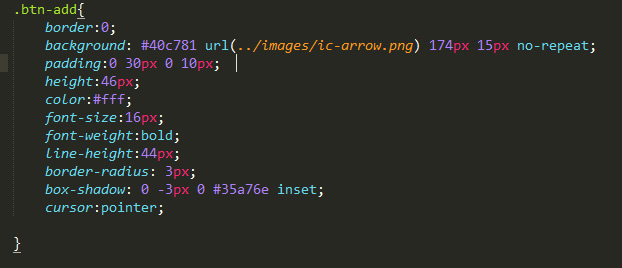
**вариант-1** задать кнопке ширину и паддинг справа или слева в зависимости от того где расположена иконка и желательно также добавить свойство box-sizing:border-box; или box-sizing:padding-box;

Добавление свойств box-sizing:border-box; или box-sizing:padding-box; нужно для того чтобы все браузеры одинаково считали ширину кнопок и чтобы во всех браузерах кнопки выглядели одинаково.

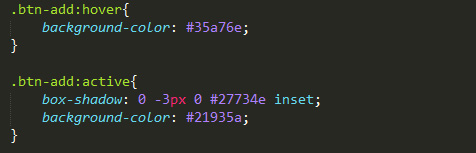
**вариант-2:** не задавать кнопке ширину, а создать нужную ширину при помощи паддингов справа и слева.

Изменим стили для кнопки: уберем ширину, зададим паддинги справа и слева (справа сделаем паддинг больше чем слева, чтобы было место для иконки).

В бэкграунд кроме цвета добавим иконку.



В стилях по наведению и для нажатой кнопки заменим background на background-color для того чтобы менялся только цвет фона, а иконка оставалась при всех состояниях кнопки.



Мы видим, что в браузере кнопка приняла нужный нам вид.

# Градиентный фон

Градиентный фон какому-либо элементу, например кнопке, можно задать либо картинкой, либо средствами CSS

Например, нам нужно сверстать кнопку-ссылку такого вида:





Для этого можно воспользоваться он-лайн генераторами градиента:

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

<http://angrytools.com/gradient/>

Сгенерированный код бэкграунда добавляем в CSS для нужного элемента.