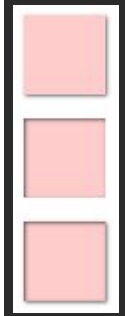
Занятие 7

CSS3

Обновления в CSS3

```
// В CSS3 тоже появилось множество новых свойств и возможностей.

// Скругленные углы
border-radius: value | % | auto;

// Тени для блока (внешние и внутренние).
box-shadow: none | inset <смещение по X> <смещение по Y> <радиус размытия> <растяжение> <цвет>;
```



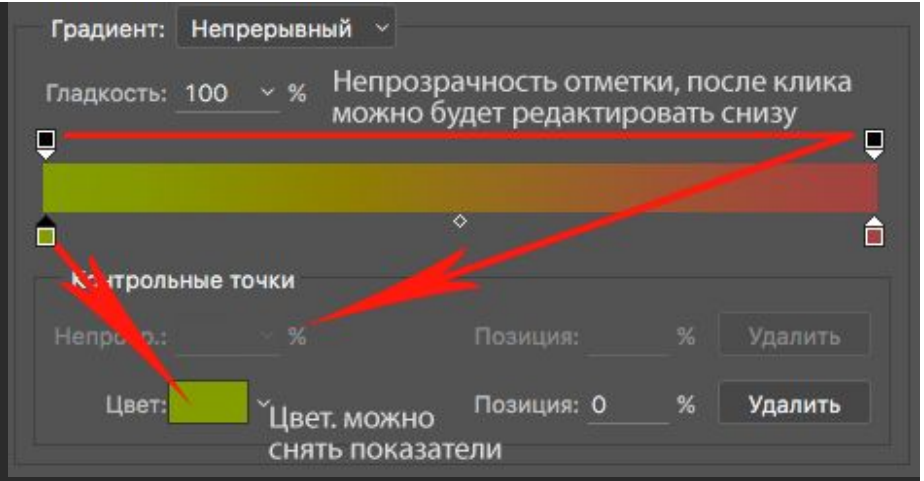
```
// Следующий пример добавляет полупрозрачную тень черного цвета выступающую снизу и справа.
box-shadow: 1px 1px 3px rgba(0,0,0,.5);

// Для внутренней тени достаточно добавить вначале параметр inset.
box-shadow: inset 1px 1px 3px rgba(0,0,0,.5);

// Для добавления множества теней на один элемент просто перечисляйте их через запятую.
box-shadow: inset 1px 1px 3px rgba(0,0,0,.5), 1px 1px 3px rgba(0,0,0,.5);

// У тени для текста нет параметра "растяжение", также у текста не может быть внутренней тени.
text-shadow: none | <смещение по X> <смещение по Y> <радиус размытия> <цвет>;

// Тень добавляется по тому же принципу, что и к box-shadow.
text-shadow: 1px 1px 3px rgba(0,0,0,.3);

// Примеры теней можно найти тут http://jsfiddle.net/r_ageychik/1db36Len.
```

Градиенты

// В окне редактора градиента нам важно знать сколько есть точек, какой у каждой из них цвет и прозрачность.

// Точек в теории может быть много, но обычно получается две.

// Заходим с этими данными на сайт http://www.colorzilla.com/ выбираем “Gradient Generator”.



// Интерфейс приложения очень сильно похож на интерфейс окна редактора градиентов photoshop, поэтому сложностей вызвать не должен. В окне справа мы увидим кроссбраузерные CSS стили этого градиента.

// Рекомендую убрать галку “Comments” и удалить последнюю строку (начиная от filter: … и до конца) если, конечно нет необходимости поддерживать IE6 - IE9.

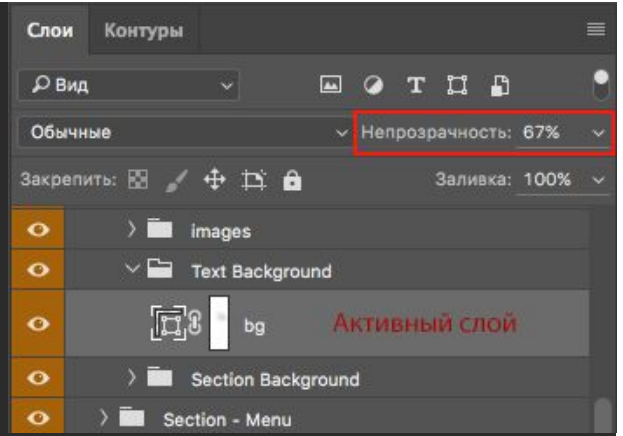
Градиенты

```
// background-image: linear-gradient(45deg, #EECFBA, #C5DDE8);
// background: linear-gradient(to right, #F6EFD2, #CEAD78);
```

https://developer.mozilla.org/en-US/docs/Web/CSS/linear-gradient

Прозрачность в CSS3

```
// В CSS3 была появилась возможность устанавливать полупрозрачность заливки и прозрачность элемента.

// Узнать процент прозрачности в макете можно в свойствах слоя.

// Устанавливает уровень прозрачности блока.
opacity: значение от 0 до 1;

// Сделает блок полупрозрачным.
opacity: 0.5; || opacity: .5;
```




```
// Также стоит помнить, что прозрачность блока будет наследоваться, поэтому в случае с одноцветным или градиентным блоком с указанной в макете прозрачностью лучше искать решение с полупрозрачной заливкой (RGBA).

// RGBA формат похож на RGB, но в него также включен альфа-канал, который передается четвертым параметром.
background-color: rgba(red, green, blue, прозрачность от 0 до 1);

// Получим наполовину прозрачный блок, залитый красным цветом.
background-color: rgba(255, 0, 0, .5);
```

Трансформация элементов

```
// Была добавлена возможность трансформировать элемент в 2D и 3D. Что позволяет изменять размер, форму и положение элемента. Для этого используется свойство transform.

// transform может принимать сразу несколько параметров, которые перечисляются через пробел.
// Что может transform:
// // translate(X, Y) | translateX(value) | translateY(value) | translate3d(x,y,z) - сдвиг элемента относительно своего первоначального положения;
// // rotate(deg) | rotate3d(x, y, z, deg) - проворачивает элемент на заданное количество градусов;
// // scale(value-x, value-y) | scaleX(value) | scaleY(value) - масштабирует элемент, задается в относительных величинах, где 1 - 100%, а 0 - 0%, чтобы получить 50% в значении надо указать 0.5 || .5;
// // skew(deg-x, deg-y) | skewX(deg) | skewY(deg) - деформирует стороны элемента;
// // perspective(value) - Задает перспективу для элемента, измеряется в px. Чем меньше значение - тем ближе элемент находится к зрителю и наоборот. Работает только совместно с rotate();

// Необходимо помнить, что при такой записи при наведении на элемент изменится не только его translate, но и rotate, тк в hover мы его не перезаписали, свойство transform обновилось, и данные о rotate не применялись для hover, чтобы этого избежать при изменении трансформации переносите все параметры transform.
.elem{transform: translate(30px, 0) rotateY(45deg);}
.elem:hover{transform: translate(0);}

// На примеры трансофрмации можно посмотреть тут http://css3.bradshawenterprises.com/transforms/.
```

Анимация элементов

```
// В CSS3 анимация была реализована для двух разных целей:
// // Для анимированной смены CSS свойств (например, плавная реакция на ховер)
// // Для более сложных циклических анимаций.

// Для простого перехода состояния используется свойство transition, которое позволяет, путем смены параметров за указанную единицу времени плавно изменить стили на необходимые.
// Надо понимать, что анимироваться будут только свойства с числовым значением, те display: none в display: block анимировать невозможно, а вот например opacity: 0 в opacity: 1 - запросто.
// Обычно используется при наведении, либо с помощью js скрипта, после добавления класса-модификатора.

// Анимация задается следующими свойствами:
// // transition-property: none | all | свойство - указывает какое свойство будет анимироваться.
// // transition-duration - устанавливает время, за которое должна закончится анимация. Устанавливается в секундах (s) или миллисекундах (ms), по умолчанию 0s;
// // transition-timing-function - Задает кривую скорости изменения параметров свойства. Подробнее можно найти тут https://html5book.ru/css3-transition/#transition-timing-function;
// // transition-delay - Позволяет изменять свойство после задержки. Устанавливается в секундах (s) или миллисекундах (ms);

// Также можно использовать короткую запись:
transition: [ none | <transition-property> || <transition-duration> || <transition-timing-function> || <transition-delay>];
transition: all 1s ease 0s; // Все измененные свойства будут анимироваться 1 секунду без задержки.
transition: transform .5s;  // Свойство transform будет анимироваться в течении половины секунды.
// Для использования нескольких transition, их стоит перечислять через запятую.
// Обязательным параметром тут является только transition-property
```

Цикличная анимация @keyframes

```
// @keyframes - Правило, по которому будет анимироваться элемент. После ключевого слова @keyframes необходимо добавить имя анимации (рекомендуется использовать имя, описывающее тип анимации), которое будет использоваться как привязка элемента к данному правилу.
// Внутри правила должны находится ключевые кадры содержащие свойства для начала и конца анимации. Также можно использовать более двух кадров, в таком случае необходимо использовать проценты.
// Ключевые кадры from и to идентичны 0% и 100% соответственно.

 @keyframes move {
    from {transform: translateX(0);}
    to {transform: translateX(10%);}
 }
```

```
@keyframes slow-move {
   0% {transform: translateX(0);}
   50% {transform: translateX(10%);}
   100% {transform: translateX(30%);}
}
```

```
// После создания правила на него можно ссылаться любому элементу, благодаря свойству animation.
// // animation-name - Свойство для привязки правила к элементу.
// // animation-duration - Время выполнения анимации, задается в секундах или миллисекундах.
// // animation-delay - Задержка перед анимацией.
// // animation-iteration-count: [value | infinite] - Количество повторов анимации, задается числом, или infinite, по-умолчанию сработает только один раз.
// // animation-direction: [alternate | alternate-reverse] || [normal | reverse] - Задает направление анимации при повторе. alternate и alternate-reverse после завершения круга будут менять последовательность свойств, normal и reverse каждый раз проигрываются с начала.
animation: [none | <animation-name> || <animation-duration>  || <animation-delay> || <animation-iteration-count> || <animation-direction>];
animation: move 5s  infinite alternate;
```

Интересные решения с css3

```
// Танцующий Бендер, на чистом CSS3.
https://liveweave.com/GoGhKy;

// Игральные кости, на чистом CSS3.
https://codepen.io/tameraydin/pen/CADvB;

// Трехмерная карусель.
https://www.jqueryscript.net/demo/Minimal-3D-Perspective-Carousel-with-jQuery-CSS3-3D-Carousel/;

// Ещё один трехмерный слайдер.
http://www.bestjquery.com/?ynVfL3CE;

// Эффект книжки.
https://www.jqueryscript.net/demo/3D-Book-Flipping-Modal-Popup-With-jQuery-CSS3/;
```

CSS3 flexbox

```
// Кроме того, в CSS3 появился отличный инструмент для компоновки и настройки элементов - flex.
// Состоит из родительского flex-бокса и дочерних flex-элементов. Которые имеют свои свойства для гибкой настройки вывода содержимого.

// Благодаря flexbox можно:
// // Располагать элементы в одном из четырех направлений: слева направо, справа налево, сверху вниз или снизу вверх;
// // Переопределять порядок отображения элементов, что очень полезно при создании мобильной версии сайта;
// // Решать проблему с горизонтальным и вертикальным центрированием;
// // Создавать колонки одинаковой высоты;
// // Создавать прижатый к низу страницы подвал сайта.

// Частично или полностью реализовать все эти возможности получалось с помощью “костылей” из более старых версий CSS, но с приходом CSS3 удается избавиться от не семантических подходов.

// Подробнее о flexbox читай:
// // https://html5.by/blog/flexbox/;
// // https://html5book.ru/css3-flexbox/;
```