**Анимация**

**Анимация в веб-дизайне**

CSS анимации подходят для выполнения простых движений элемента и когда не требуется совместимость с устаревшими браузерами. JavaScript позволяет создавать сложные анимации, с возможностью интерактивного взаимодействия.

Эффект движущихся элементов на сайте, можно создать с помощью свойств CSS3:

* переходов — ***transition;***
* анимации — ***animation;***
* трансформации — ***transformation.***

Следует отметить, что трансформация не создаёт анимацию, а только трансформирует (перемещает, вращает, увеличивает) элемент. Трансформации часто используются совместно со свойствами ***animation и transition***.

**CSS-переходы. Свойство Transition**

CSS-переходы ***(transition),*** позволяют изменять CSS свойствам одно значение на другое, за определённый промежуток времени. Простейший пример подобного поведения — это изменение цвета фона ссылки-кнопки с помощью псевдокласса ***:hover***, при котором моментально происходит смена значений CSS-свойства ***background-color***.

transition

*transition: all 0.5s ease 0s;* — сокращенное свойство, определяющее переход.  
Первое значение определяет список свойств, чьи переходы должны анимироваться.  
Второе определяет время перехода.  
Третье значение определяет временную функцию.  
Четвертое определяет время задержки до начала анимации.

CSS-свойство ***transition*** необходимо применять для элемента в том состоянии, из которого будет осуществляться переход.

Объявить переход можно с помощью сокращенного свойства transition (как указано в примере выше) или с помощью 4-х отдельный свойств: ***transition-property, transition-duration, transition-timing-function, transition-delay***.

1. ***transition-property***

Определяет список изменяемых свойств. Можно указать одно свойство или список свойств, которые должны изменяться, через запятую.  
Изменению подлежат те CSS свойства, которые изменяют внешний вид элемента или его расположение в HTML документе. К таким свойствам относятся: ***width, height, left, top, right, bottom, color, opacity, background-color, margin, left, border*** и т.д.

1. ***transition-duration***

Определяет время перехода, указывается в s (секунды) или в ms (миллисекунды). Является обязательным свойством, при котором будет создаваться переход. Если в свойстве transition-property указано несколько свойств, длительность перехода можно указать разную для каждого свойства, через запятую.

1. ***transition-timing-function***

Временная функция, которая определяет, как будут изменяться, указанные свойства. Существует несколько стандартных функций. Также возможно написать свою собственную функцию, используя кривые Безье.

**Значение для собственной функции:**

* + сubic-bezier(x1,x2,x3,x4)

**Стандартные функции:**

* + ***ease*** — переход начинается быстро, к концу скорость увеличивается. Используется по умолчанию;
  + ***linear*** — скорость линейная, постоянная;
  + ***ease-in*** — переход начинается медленно, к концу скорость увеличивается;
  + ***ease-out*** — переход начинается быстро, к концу скорость уменьшается;
  + ***ease-in-out*** — переход медленно начинается, к середине времени увеличивается и к концу опять уменьшается.

1. ***transition-delay***

Определяет задержку до начала перехода, указывается в s или в ms. Если значение не указывается, то по умолчанию оно равно 0 и переход создаётся сразу. Если в свойстве transition-property указано несколько свойств, задержку для каждого свойства можно указывать через запятую.

**CSS-трансформация. Свойство Transform**

С помощью трансформаций (свойство ***transform***) элемент можно увеличить в размере, вращать, наклонять, сдвигать, не затрагивая другие элементы на странице.  
Часто трансформации используются вместе с переходами. Переходы позволяют сделать процесс трансформации элемента плавным. Существуют 2D и 3D-типы трансформации. Трансформация происходит в двух осях X (по горизонтали) и Y (по вертикали).  
Для преобразования элемента используется свойство ***transform***, которое определяет тип трансформации элемента и свойство ***transform-origin***, которое определяет исходную точку начала трансформации.

1. ***transform-origin***

Значения по умолчанию: 50% 50% 0. Первое значение указывает координату x, перемещение элемента по горизонтали.  
Второе значение указывает координату y, перемещение элемента по вертикали. Эти значения вычисляются от верхнего левого угла элемента.  
Третье значение задает z-смещение и используется при 3D-преобразованиях. Значения можно указывать в единицах измерения длины элемента, в процентах или ключевыми словами ***center, left, right, top, bottom***. z-смещение задавать не обязательно.

1. ***translate(x, y)***

Изменение местоположения элемента на значение x слева направо, на значение y сверху вниз. В случае указания отрицательных значений, сдвиг будет направлен справа налево и снизу вверх.

* + ***translateX(x)*** — похожа на функцию *translate(x, y)*, однако, для неё указывается одно значение для сдвига элемента по оси x, влево или вправо.
  + ***translateY(y)*** — похожа на функцию *translate(x, y)*, однако, для неё указывается одно значение для сдвига элемента по оси y, вверх или вниз.
  + ***scale(x, y)*** — масштабирует элемент. Если указаны значения от 0 до 1, то элемент уменьшается. Если значения больше 1, то элемент увеличивается в масштабе. При отрицательных значениях отображает элемент зеркально. Следует отметить, что при увеличении элемента, качество отображаемого элемента может ухудшиться. Действует для элементов блочного типа *(display:block; display:inline-block;)*. Почему только для таких элементов? Так как масштабирование — это одновременное увеличение ширины и высоты элемента.
  + ***scaleY(y)*** — масштабирует элемент по вертикали. Если указаны значения от 0 до 1, то элемент уменьшается. Если больше 1, то увеличивается в масштабе. При отрицательных значениях отображает элемент зеркально. Следует отметить, что при увеличении элемента, качество может ухудшиться.
  + ***rotate()*** — вращает элемент вокруг исходной точки (за расположение исходной точки отвечает свойство *transform-origin*). Единицы измерения: градусы *(deg)*. При положительном значении — элемент поворачивается по часовой стрелке. При отрицательном — элемент поворачивается против часовой стрелки. При указании угла по ворота более 360 град, элемент будет вращаться на указанное значение, например, при transform: *rotate(3600deg)*, элемент совершить 10 оборотов вокруг исходной точки.
  + ***skew(x, y)*** — определяет наклон элемента вдоль осей X и Y. Если указать только одно значение, то наклон элемента произойдёт по оси X. Значения указываются в углах, градусах, радианах и оборотах. При положительных значениях наклон совершает по часовой стрелке, при отрицательных против часовой стрелки.
  + ***skewX(x)*** — определяет наклон элемента вдоль осей X. При положительных значениях наклон совершает влево, при отрицательных в вправо.
  + ***skewY(y)*** — определяет наклон элемента вдоль осей Y. При положительных значениях наклон совершает вверх, при отрицательных вниз.
  + ***matrix*** — объединяет ряд трансформаций *(scale, rotate, skew, traslate)* в одно объявление.  
    Синтаксис *matrix(a, c, b, d, tx, ty)*, где:
    - ***а*** — изменение масштаба по горизонтали scaleX;
    - ***c*** — наклон по вертикали skewY;
    - ***b*** — наклон по горизонтали skewX;
    - ***d*** — изменение масштаба по вертикали scaleY;
    - ***tx*** — смещение по горизонтали translateX;
    - ***ty*** — смещение по вертикали translateY.

**Пример: с помощью *transform: translateX()*, блок выезжает на 700px**

**transform: scale(1.1);**

**transform: rotate(90deg);**

**transform: skewX(-30deg);**

**CSS-анимация. Свойство Animation**

Создание анимации базируется на ключевых кадрах, которые позволяют автоматически воспроизводить и повторять эффекты на протяжении заданного времени, а также останавливать анимацию внутри цикла.  
CSS3-анимация может применяться практически для всех html-элементов, а также для псевдоэлементов ***:before и :after.***

Кадры создаются с помощью специального правила @keyframes. Сами кадры объявляются ключевыми словами ***from — «от»*** или начальный, первый кадр. И ***to — «до»*** или конечный кадр, второй и последний кадр. Если количество кадров больше, чем два, то к кадру обращаются, как к отрезкам, выраженным в процентах пройденного пути. Например, три кадра — **0%** — начальный первый кадр, **50%**— второй кадр, **100%** — конечный третий кадр.

**Синтаксис**

*h1{  
animation: name 5s linear alternate infinite;  
}  
  
@keyframes name{  
from{  
условие анимации  
}  
to{  
условие анимации  
}  
}  
где, name — имя анимации, создаётся произвольно согласно смыслу анимации.*

CSS свойство ***animation*** — это краткая запись для ***animation-name, animation-duration, animation-timingfunction, animation-delay, animation-iteration-count, animation-direction, animation-fill-mode, animation-play-state***.

1. ***animation-name***

Имя анимации, создается произвольно;

1. ***animation-duration***

Длительность анимации за один цикл;

1. ***animation-timingfunction***

Временная функция, определяет, как происходит анимация в течение одного цикла.

**Значение для собственной функции:**

* + ***сubic-bezier***

**Стандартные функции:**

* + ***ease*** — переход начинается быстро, к концу скорость увеличивается;
  + ***linear*** — скорость линейная, постоянная;
  + ***ease-in*** — переход начинается медленно, к концу скорость увеличивается;
  + ***ease-out*** — переход начинается быстро, к концу скорость уменьшается;
  + ***ease-in-out*** — переход медленно начинается, к середине времени увеличивается и к концу опять уменьшается.

1. ***animation-delay***

Определяет время задержки перед началом анимации;

1. ***animation-iteration-count***

Определяет сколько раз будет проигрываться цикл анимации, перед тем как остановиться. Значение ***infinite*** определяет бесконечное число циклов.

1. ***animation-direction***

Определяет направление анимации. Существует несколько стандартных значений:

* + ***normal*** — значение по умолчанию, анимация начинается в начальном кадре проходит цикл до конечного кадра и моментально возвращается в начальный кадр;
  + ***reverse*** — анимация проигрывается наоборот. Начинается в конченом кадре, проигрывается до начального и сбрасывается в начальном кадре, моментально возвращаясь в конечный кадр;
  + ***alternate*** — анимация меняет направление в каждом цикле. От начального кадра идёт вперёд до конченого в одном цикле, в следующем возвращается из конечного кадра в начальный, количество за от числа итераций ***animation-iteration-count***;

1. ***animation-fill-mode***

Определяет необходимо ли применять стили анимации к элементу до и после окончания анимации. Имеет стандартные значения:

* + ***none*** — стили анимации не будут применены до и после окончания анимации;
  + ***forwards*** — после окончания анимации, элемент сохранит стили анимации конечного ключевого кадра, которые зависит от ***animation-direction и animation-iteration-count***;
  + ***alternate*** — анимация меняет направление в каждом цикле. От начального кадра идёт вперёд до конченого в одном цикле, в следующем возвращается из конечного кадра в начальный, количество за от числа итераций ***animation-iteration-count***;
  + ***backwards*** — элемент сохранит стили анимации начального кадра, на протяжении всей задержки ***(animation-delay)*** до начала анимации, которая определяется значением ***animation-direction***;
  + ***both*** — определяет одновременного применение значений ***forwards и backwards***.

1. ***animation-play-state***

Определяет состояния паузы и проигрыша анимации. Используется в скриптах,анимацию можно остановить и возобновить с помощью скриптов и проигрываться она начнет с момента остановки, не сначала. Имеет два значения:

* + ***runing*** — анимация проигрывается;
  + ***paused*** — анимация приостановлена.

**Пример:**