CSS - анимации

transform, transition, animation

transform (трансформация)

- Функции трансформации: свойство transform
- Точка трансформации: свойство transform-origin
- Область преобразования: свойство transform-box

Так же есть 3D трансформация для рендеринга полноценного трехмерного элемента в рамках HTML (крайне редко используется)

- Ресурсозатратная верстка (тратится много человеко часов)
- Существуют готовые решения
- Существуют библиотеки для реализации задач связанных с 3D

Функции трансформации: свойство transform

```
transform: none;
transform: matrix(1.0, 2.0, 3.0, 4.0, 5.0, 6.0);
transform: rotate(45deg);
transform: translate(12px, 50%);
transform: translateX(2em);
transform: translateY(3in);
transform: scale(2, 0.5);
transform: scaleX(2);
transform: scaleY(0.5);
transform: skew(30deg, 20deg);
transform: skewX(30deg);
transform: skewY(1.07rad);
transform: translateX(10px) rotate(10deg) translateY(5px);
transform: inherit;
transform: initial;
```

Свойство transform задает вид преобразования элемента. Свойство описывается с помощью функций трансформации, которые смещают элемент относительно его текущего положения на странице или изменяют его первоначальные размеры и форму.

Свойство не наследуется.

matrix() — любое число

translate(), translateX(), translateY() — единицы длины (положительные и отрицательные), %

scale(), scaleX(), scaleY() — любое число

rotate() — угол (deg, grad, rad или turn)

skew(), skewX(), skewY() — угол (deg, grad, rad)

Сдвиг элемента

- translateX(n) - Сдвигает элемент относительно его обычного положения по оси X.
- translateY(n) - Сдвигает элемент относительно его обычного положения по оси Y.
- translate(x,y) - Сдвигает элемент на новое место, перемещая относительно обычного положения вправо и вниз, используя координаты X и Y, не затрагивая при этом соседние элементы. Если нужно сдвинуть элемент влево или вверх, то нужно использовать отрицательные значения.

Масштабируемость элемента

- scaleX(n) - Функция масштабирует элемент по ширине, делая его шире или уже. Если значение больше единицы, элемент становится шире, если значение находится между единицей и нулем, элемент становится уже. Отрицательные значения отображают элемент зеркально по горизонтали.
- scaleY(n) - Функция масштабирует элемент по высоте, делая его выше или ниже. Если значение больше единицы, элемент становится выше, если значение находится между единицей и нулем — ниже. Отрицательные значения отображают элемент зеркально по вертикали.
- scale(x,y) - Масштабирует элементы, делая их больше или меньше. Значения от 0 до 1 уменьшают элемент. Первое значение масштабирует элемент по ширине, второе — по высоте. Отрицательные значения отображают элемент зеркально.

Поворот элементов

- rotate(угол) - Поворачивает элементы на заданное количество градусов, отрицательные значения от -1deg до -360deg поворачивают элемент против часовой стрелки, положительные — по часовой стрелке. Значение rotate(720deg) поворачивает элемент на два полных оборота.
- rotateX(угол) - Функция задает поворот по часовой стрелке под заданным углом относительно оси X. Функция rotateX(180deg) эквивалентна rotate3d(1,0,0,180deg).
- rotateY(угол) - Функция задает поворот по часовой стрелке под заданным углом относительно оси Y. Функция rotateY(180deg) эквивалентна rotate3d(0,1,0,180deg).
- rotateZ(угол) - Функция задает поворот по часовой стрелке под заданным углом относительно оси Z. Функция rotateZ(180deg) эквивалентна rotate3d(0,0,1,180deg)
- rotate3d(x,y,z,угол) - Функция вращает элемент по часовой стрелке относительно трех осей. Элемент поворачивается под углом, задаваемым последним параметром относительно вектора направления [x,y,z]. Отрицательные значения поворачивают элемент против часовой стрелки.

Деформирование элементов

- skewX(угол) - Деформирует стороны элемента относительно оси X.
- skewY(угол) - Деформирует стороны элемента относительно оси Y.
- skew(x-угол,y-угол) - Используется для деформирования (искажения) сторон элемента относительно координатных осей. Если указано одно значение, второе будет определено браузером автоматически.

Сокращенная запись matrix - матрица параметров

matrix(a, c, b, d, x, y) - Смещает элементы и задает способ их трансформации, позволяя объединить несколько функций 2D-трансформаций в одной. В качестве трансформации допустимы поворот, масштабирование, наклон и изменение положения.

- Значение a изменяет масштаб по горизонтали. Значение от 0 до 1 уменьшает элемент, больше 1 — увеличивает.
- Значение c деформирует (сдвигает) стороны элемента по оси Y, положительное значение — вверх, отрицательное — вниз.
- Значение b деформирует (сдвигает) стороны элемента по оси X, положительное значение — влево, отрицательное — вправо.
- Значение d изменяет масштаб по вертикали. Значение меньше 1 уменьшает элемент, больше 1 — увеличивает.
- Значение x смещает элемент по оси X, положительное — вправо, отрицательное — влево.
- Значение y смещает элемент по оси Y, положительное значение — вниз, отрицательное — вверх.

Точка трансформации: свойство transform-origin

```
transform-origin: 2px;
transform-origin: bottom;
transform-origin: 3cm 2px;
transform-origin: left 2px;
transform-origin: right top;
transform-origin: inherit;
transform-origin: initial;
```

Свойство transform-origin позволяет сместить центр трансформации, относительно которого происходит изменение положения/размера/формы элемента. Значение по умолчанию — center, или 50% 50%. Задается только для трансформированных элементов.

Свойство не наследуется.

Область преобразования: свойство transform-box

transform-box

Значения:

`content-box`	В качестве опорного блока выступает область содержимого элемента. Для элемента `<table>` эта область включает также заголовок таблицы и рамку.
`border-box`	В качестве опорного блока выступает область рамки элемента. Для элемента `<table>` эта область включает также заголовок таблицы и рамку.
`fill-box`	В качестве опорного блока выступает ограничивающая рамка, содержащая только геометрическую форму элемента.
`stroke-box`	В качестве опорного блока выступает ограничивающая рамка, содержащая геометрическую форму и обводку элемента.
`view-box`	В качестве опорного блока используется область просмотра на SVG-холсте, которая определяет прямоугольную область, в которую отображается содержимое SVG.
`initial`	Устанавливает значение свойства в значение по умолчанию.
`inherit`	Наследует значение свойства от родительского элемента.

Все преобразования, определяемые свойством transform и transform-origin, относятся к положению и размерам опорного блока элемента. Опорный блок элемента это виртуальный прямоугольник вокруг элемента, который формирует систему координат для отрисовки.

В некоторых браузерах опорный блок принимает центр SVG-холста в качестве точки преобразования. Чтобы решить эту проблему, можно задать для элемента transform-box.

Опорный блок добавляет дополнительное смещение к исходной точке, заданной свойством transform-origin.

Свойство не наследуется.

transition (переходы)

- transition-property
- transition-duration
- transition-timing-function
- transition-delay
- transition - краткая запись

CSS3-переходы позволяют анимировать исходное значение CSS-свойства на новое значение с течением времени, управляя скоростью смены значений свойств. Большинство свойств меняют свои значения за 16 миллисекунд, поэтому рекомендуемое время стандартного перехода — 200ms.

Название свойства transition-property

Содержит название CSS-свойств, к которым будет применен эффект перехода. Значение свойства может содержать как одно свойство, так и список свойств через запятую. При создании перехода можно использовать как начальное, так и конечное состояние элемента. Свойство не наследуется.

Значения:

`none`	Отсутствие свойства для перехода.
`all`	Значение по умолчанию. Применяет эффект перехода ко всем свойствам элемента.
свойство	Определяет список CSS-свойств, перечисленных через запятую, участвующих в переходе.
`initial`	Устанавливает значение свойства в значение по умолчанию.
`inherit`	Наследует значение свойства от родительского элемента.

Продолжительность перехода transition-duration

Задаёт промежуток времени, в течение которого должен осуществляться переход. Если разные свойства имеют разные значения для перехода, они указываются через запятую. Если продолжительность перехода не указана, то анимация при смене значений свойств происходить не будет. Свойство не наследуется.

Значения:

время	Время перехода указывается в секундах или миллисекундах, например, `2s` или `5ms`.
`initial`	Устанавливает значение свойства в значение по умолчанию.
`inherit`	Наследует значение свойства от родительского элемента.

Функция перехода transition-timing-function

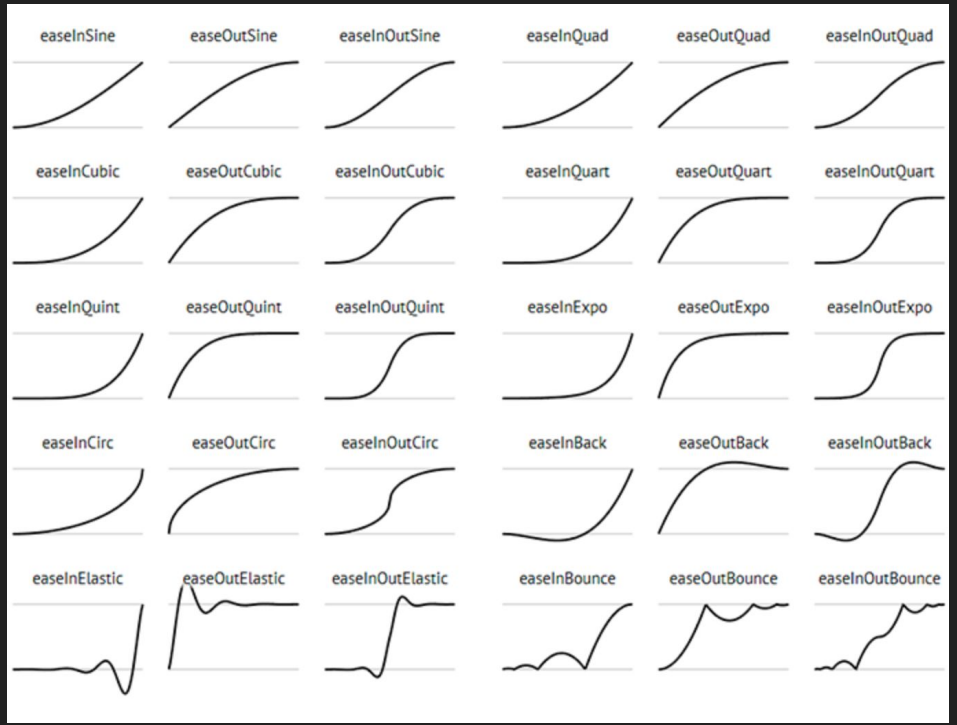
Значения:

`ease`	Функция по умолчанию, переход начинается медленно, разгоняется быстро и замедляется в конце. Соответствует `cubic-bezier(0.25,0.1,0.25,1)`.
`linear`	Переход происходит равномерно на протяжении всего времени, без колебаний в скорости. Соответствует `cubic-bezier(0,0,1,1)`.
`ease-in`	Переход начинается медленно, а затем плавно ускоряется в конце. Соответствует `cubic-bezier(0.42,0,1,1)`.
`ease-out`	Переход начинается быстро и плавно замедляется в конце. Соответствует `cubic-bezier(0,0,0.58,1)`.
`ease-in-out`	Переход медленно начинается и медленно заканчивается. Соответствует `cubic-bezier(0.42,0,0.58,1)`.
`cubic-bezier(x1, y1, x2, y2)`	Позволяет вручную установить значения от 0 до 1 для кривой ускорения. На этом сайте вы сможете построить любую траекторию перехода.
`initial`	Устанавливает значение свойства в значение по умолчанию.
`inherit`	Наследует значение свойства от родительского элемента.

Свойство задает временную функцию, которая описывает скорость перехода объекта от одного значения к другому. Если вы определяете более одного перехода для элемент, например, цвет фона элемента и его положение, вы можете использовать разные функции для каждого свойства. Свойство не наследуется.

функция cubic Bézier (Кривая Безье)

easeInSine	`cubic-bezier(0.47, 0, 0.745, 0.715)`
easeOutSine	`cubic-bezier(0.39, 0.575, 0.565, 1)`
easeInOutSine	`cubic-bezier(0.445, 0.05, 0.55, 0.95)`
easeInQuad	`cubic-bezier(0.55, 0.085, 0.68, 0.53)`
easeOutQuad	`cubic-bezier(0.25, 0.46, 0.45, 0.94)`
easeInOutQuad	`cubic-bezier(0.455, 0.03, 0.515, 0.955)`
easeInCubic	`cubic-bezier(0.55, 0.055, 0.675, 0.19)`
easeOutCubic	`cubic-bezier(0.215, 0.61, 0.355, 1)`
easeInOutCubic	`cubic-bezier(0.645, 0.045, 0.355, 1)`
easeInQuart	`cubic-bezier(0.895, 0.03, 0.685, 0.22)`
easeOutQuart	`cubic-bezier(0.165, 0.84, 0.44, 1)`
easeInOutQuart	`cubic-bezier(0.77, 0, 0.175, 1)`
easeInQuint	`cubic-bezier(0.755, 0.05, 0.855, 0.06)`
easeOutQuint	`cubic-bezier(0.23, 1, 0.32, 1)`
easeInOutQuint	`cubic-bezier(0.86, 0, 0.07, 1)`
easeInExpo	`cubic-bezier(0.95, 0.05, 0.795, 0.035)`
easeOutExpo	`cubic-bezier(0.19, 1, 0.22, 1)`
easeInOutExpo	`cubic-bezier(1, 0, 0, 1)`
easeInCirc	`cubic-bezier(0.6, 0.04, 0.98, 0.335)`
easeOutCirc	`cubic-bezier(0.075, 0.82, 0.165, 1)`
easeInOutCirc	`cubic-bezier(0.785, 0.135, 0.15, 0.86)`
easeInBack	`cubic-bezier(0.6, -0.28, 0.735, 0.045)`
easeOutBack	`cubic-bezier(0.175, 0.885, 0.32, 1.275)`
easeInOutBack	`cubic-bezier(0.68, -0.55, 0.265, 1.55)`



Задержка перехода transition-delay

Необязательное свойство, позволяет сделать так, чтобы изменение свойства происходило не моментально, а с некоторой задержкой. Не наследуется.

Значения:	
время	Время задержки перехода указывается в секундах или миллисекундах.
`initial`	Устанавливает значение свойства в значение по умолчанию.
`inherit`	Наследует значение свойства от родительского элемента.

Краткая запись перехода transition

Все свойства, отвечающие за изменение внешнего вида элемента, можно объединить в одно свойство transition

transition: transition-property transition-duration transition-timing-function transition-delay;

Если воспользоваться значениями по умолчанию, то запись

```
div {transition: 1s;}
```

будет эквивалентна

```
div {transition: all 1s ease 0s;}
```

6. Плавный переход нескольких свойств

Для элемента можно задать несколько последовательных переходов, перечислив их через запятую. Каждый переход можно оформить своей временной функцией.

CSS

```
div {transition: background 0.3s ease, color 0.2s linear;}
```

или

CSS

```
div {
transition-property: height, width, background-color;
transition-duration: 3s;
transition-timing-function: ease-in, ease, linear;
}
```

animation (анимация)

CSS3-анимация придаёт сайтам динамичность. Она оживляет веб-страницы, улучшая взаимодействие с пользователем. В отличие от CSS3-переходов, создание анимации базируется на ключевых кадрах, которые позволяют автоматически воспроизводить и повторять эффекты на протяжении заданного времени, а также останавливать анимацию внутри цикла.

- Ключевые кадры @keyframes
- Название анимации: свойство animation-name
- Продолжительность анимации: свойство animation-duration
- Временная функция: свойство animation-timing-function
- Повтор анимации: свойство animation-iteration-count
- Направление анимации: свойство animation-direction
- Проигрывание анимации: свойство animation-play-state
- Задержка анимации: свойство animation-delay
- свойство animation-fill-mode
- Краткая запись анимации: свойство animation

Ключевые кадры

Создание анимации начинается с установки ключевых кадров правила @keyframes. Кадры определяют, какие свойства на каком шаге будут анимированы. Каждый кадр может включать один или более блоков объявления из одного или более пар свойств и значений.

Ключевые кадры создаются с помощью ключевых слов from и to (эквивалентны значениям 0% и 100%) или с помощью процентных пунктов, которых можно задавать сколько угодно. Также можно комбинировать ключевые слова и процентные пункты

```
@keyframes имя анимации { список правил }
```

Название анимации: свойство animation-name

Свойство animation-name определяет список применяемых к элементу анимаций. Каждое имя используется для выбора ключевого кадра в правиле, которое предоставляет значения свойств для анимации. Если имя не соответствует ни одному ключевому кадру в правиле, нет свойств для анимации, отсутствует имя анимации, анимация не будет выполняться.

Если несколько анимаций пытаются изменить одно и то же свойство, то выполнится анимация, ближайшая к концу списка имен.

Имя анимации чувствительно к регистру, не допускается использование ключевого слова none. Рекомендуется использовать название, отражающее суть анимации, при этом можно использовать одно или несколько слов, перечисленных через дефис - или символ нижнего подчеркивания _.

Свойство не наследуется.

Значения:

`none`	Означает отсутствие анимации. Также используется, чтобы отменить анимацию элемента из группы элементов, для которых задана анимация. Значение по умолчанию.
имя анимации	Имя анимации, которое связывает правило `@keyframes` с селектором.
`initial`	Устанавливает значение свойства в значение по умолчанию.
`inherit`	Наследует значение свойства от родительского элемента.

Продолжительность анимации: свойство animation-duration

Свойство animation-duration определяет продолжительность одного цикла анимации. Задаётся в секундах s или миллисекундах ms. Если для элемента задано более одной анимации, то можно установить разное время для каждой, перечислив значения через запятую.

Свойство не наследуется.

Значения:

время	Указывает время, которое анимация занимает для завершения одного цикла. Отрицательные значения недействительны. Если время равно `0s`, ключевые кадры анимации не действуют, но сама анимация происходит мгновенно. Значение по умолчанию `0s`.

Временная функция: свойство animation-timing-function

Свойство animation-timing-function описывает, как будет развиваться анимация между каждой парой ключевых кадров. Во время задержки анимации временные функции не применяются.

Свойство не наследуется.

Значения:

`linear`	Линейная функция, анимация происходит равномерно на протяжении всего времени, без колебаний в скорости.

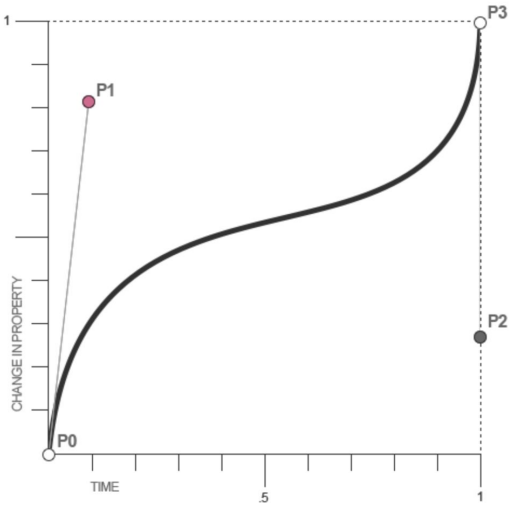
функции Безье

`ease`	Функция по умолчанию, анимация начинается медленно, разгоняется быстро и замедляется в конце. Соответствует `cubic-bezier(0.25,0.1,0.25,1)`.
`ease-in`	Анимация начинается медленно, а затем плавно ускоряется в конце. Соответствует `cubic-bezier(0.42,0,1,1)`.
`ease-out`	Анимация начинается быстро и плавно замедляется в конце. Соответствует `cubic-bezier(0,0,0.58,1)`.
`ease-in-out`	Анимация медленно начинается и медленно заканчивается. Соответствует `cubic-bezier(0.42,0,0.58,1)`.

пошаговые функции

`step-start`	Задаёт пошаговую анимацию, разбивая анимацию на отрезки, изменения происходят в начале каждого шага. Вычисляется в `steps(1, start)`.
`step-end`	Пошаговая анимация, изменения происходят в конце каждого шага. Вычисляется в `steps(1, end)`.

`cubic-bezier(x1, y1, x2, y2)` — Позволяет установить значения вручную.



P0 и P3 всегда находятся в одном месте, P0 — в точке (0,0), а P3 — в точке (1,1). По оси X показано время анимации, а по оси Y — изменяемое свойство. Кривая будет растягиваться по-разному в зависимости от того, какие значения вы дадите точкам P1 и P2 (-∞,∞). Основываясь на приведенном выше графике, вы можете визуализировать, как он быстро замедляется в начале, замедляется в середине и набирает скорость в конце.

Повтор анимации: свойство animation-iteration-count

Свойство animation-iteration-count указывает, сколько раз проигрывается цикл анимации. Начальное значение 1 означает, что анимация будет воспроизводиться от начала до конца один раз. Это свойство часто используется в сочетании со значением alternate свойства animation-direction, которое заставляет анимацию воспроизводиться в обратном порядке в альтернативных циклах.

Свойство не наследуется.

Значения:

`infinite`	Анимация проигрывается бесконечно.
число	Анимация будет повторяться указанное количество раз. Если число не является целым числом, анимация закончится в середине последнего цикла. Отрицательные числа недействительны. Значение `0` вызывает мгновенное срабатывание анимации.

Направление анимации: свойство animation-direction

Свойство animation-direction определяет, должна ли анимация воспроизводиться в обратном порядке в некоторых или во всех циклах. Когда анимация воспроизводится в обратном порядке, временные функции также меняются местами. Например, при воспроизведении в обратном порядке функция ease-in будет вести себя как ease-out.

Свойство не наследуется.

Значения:

`normal`	Все повторы анимации воспроизводятся так, как указано. Значение по умолчанию.
`reverse`	Все повторы анимации воспроизводятся в обратном направлении от того, как они были определены.
`alternate`	Каждый нечетный повтор цикла анимации воспроизводятся в нормальном направлении, каждый четный повтор воспроизводится в обратном направлении.
`alternate-reverse`	Каждый нечетный повтор цикла анимации воспроизводятся в обратном направлении, каждый четный повтор воспроизводится в нормальном направлении.
`initial`	Устанавливает значение свойства в значение по умолчанию.
`inherit`	Наследует значение свойства от родительского элемента.

Проигрывание анимации: свойство animation-play-state

Свойство animation-play-state определяет, будет ли анимация запущена или приостановлена. Остановка анимации внутри цикла возможна через использование этого свойства в скрипте JavaScript. Также можно останавливать анимацию при наведении курсора мыши на объект — состояние :hover.

Свойство не наследуется.

Значения:

`running`	Анимация выполняется. Значение по умолчанию.
`paused`	Анимация приостанавливается. При перезапуске анимация начинается с того места, где она была остановлена, как если бы «часы», управляющие анимацией, остановились и снова запустились. Если анимация остановлена во время задержки, при повторном воспроизведении время задержки также возобновляется.
`initial`	Устанавливает значение свойства в значение по умолчанию.
`inherit`	Наследует значение свойства от родительского элемента.

Задержка анимации: свойство animation-delay

Свойство animation-delay определяет, когда анимация начнется. Задается в секундах s или миллисекундах ms.

Свойство не наследуется.

Значения:

время	Время определяет длительность задержки между началом анимации (когда анимация применяется к элементу через свойства) и когда она начинает выполняться. Отрицательные значения разрешены, такая задержка начинает анимацию с определенного момента внутри её цикла, т.е. со времени, указанного в задержке. Это позволяет применять анимацию к нескольким элементам со сдвигом фазы, изменяя лишь время задержки. Чтобы анимация началась с середины, нужно задать отрицательную задержку, равную половине времени, установленном в `animation-duration`. Значение по умолчанию `0s`.

Состояние элемента до и после воспроизведения анимации: свойство animation-fill-mode

animation-fill-mode - это CSS свойство, которое определяет, как стили элемента применяются до и после проигрывания анимации.

Возможные значения:

- none: Это значение по умолчанию. Элемент будет находиться в исходном состоянии до начала анимации, а после завершения анимации он вернется в это же исходное состояние.
- forwards: Элемент будет находиться в состоянии, в котором он находится после окончания анимации. То есть стили, примененные в конце анимации, будут применены и после ее завершения.
- backwards: Элемент будет находиться в состоянии, в котором он должен быть до начала анимации, включая стили, примененные к нему в начале анимации.
- both: Комбинация значений forwards и backwards. Элемент будет находиться в состоянии, в котором он должен быть до начала анимации до ее начала, и оставаться в состоянии, в котором он находится после завершения анимации.

Это свойство полезно, когда вы хотите сохранить состояние элемента после завершения анимации или применить начальные стили перед началом анимации.

Краткая запись анимации: свойство animation

Все параметры воспроизведения анимации можно объединить в одном свойстве — animation, перечислив их через пробел:

animation: animation-name animation-duration animation-timing-function animation-delay animation-iteration-count animation-direction;

Для воспроизведения анимации достаточно указать только два свойства — animation-name и animation-duration, остальные свойства примут значения по умолчанию. Порядок перечисления свойств не имеет значения, единственное, время выполнения анимации animation-duration обязательно должно стоять перед задержкой animation-delay.

Для одного элемента можно задавать несколько анимаций, перечислив их названия через запятую:

```
div {animation: shadow 1s ease-in-out 0.5s alternate, move 5s linear 2s;}
```

loader

Готовые loader-ы - ТЫК

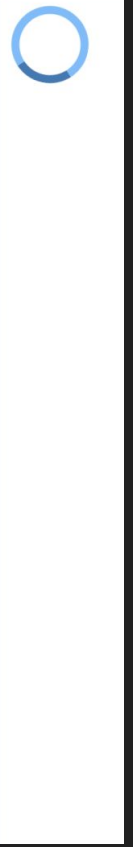
"Loader" в фронтенд-разработке обычно относится к анимированному индикатору загрузки, который отображается на веб-странице во время выполнения какой-либо асинхронной операции, такой как загрузка данных с сервера или выполнение сложных вычислений на клиентской стороне.

Он может представлять из себя анимированную иконку, крутящееся кольцо, прогресс-бар или любой другой элемент, который позволяет пользователю видеть, что что-то происходит на странице, и что он не застрял в ожидании.

Loaderы важны для улучшения пользовательского опыта, особенно в случаях, когда операции занимают некоторое время, и пользователь может подумать, что что-то пошло не так, если он не видит никаких индикаций прогресса.

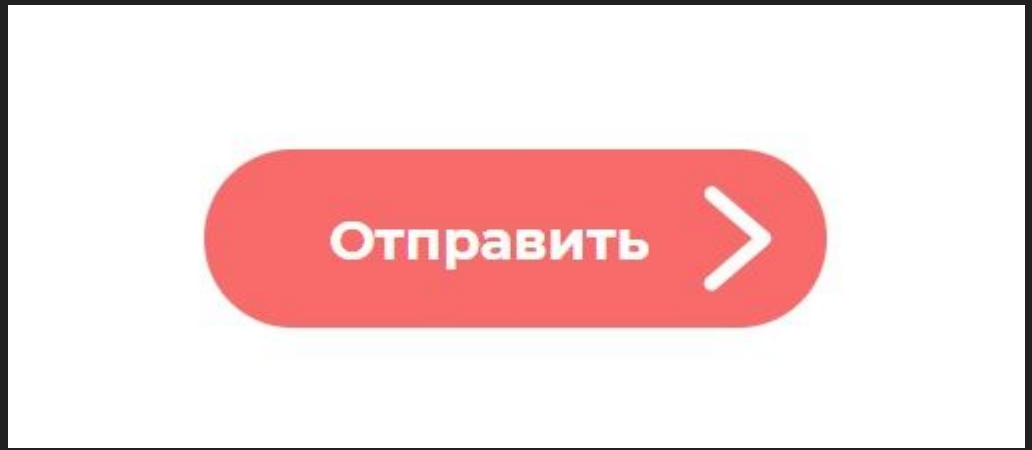
```
<!DOCTYPE html>
<html>
<head>
<style>
.loader {
  width: 48px;
  height: 48px;
  border: 5px solid #6ebeff;
  border-bottom-color: #337AB7;
  border-radius: 50%;
  display: inline-block;
  box-sizing: border-box;
  animation: rotation 1s linear infinite;
}

@keyframes rotation {
  0% {
    transform: rotate(0deg);
  }
  100% {
    transform: rotate(360deg);
  }
}
</style>
</head>
<body>
    <span class="loader"></span>
</body>
</html>
```



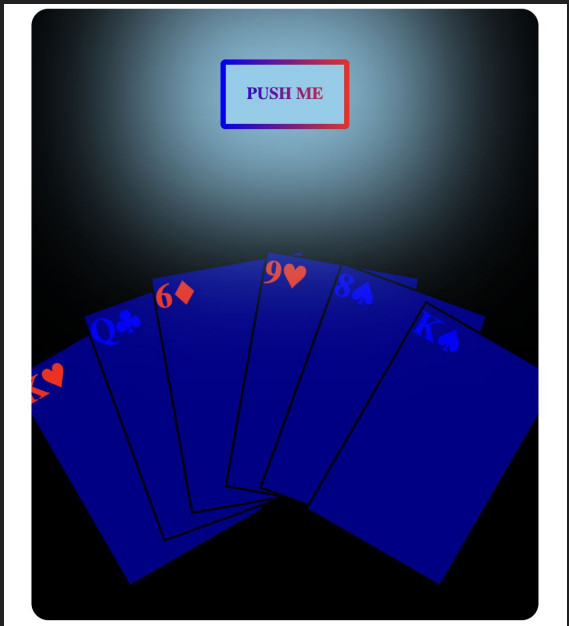
button

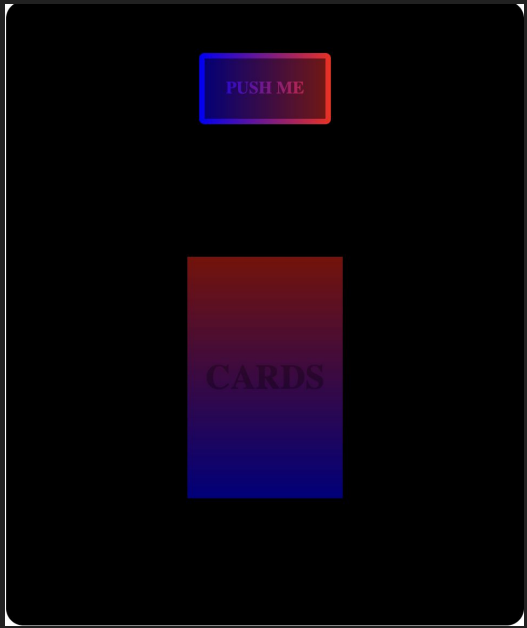
Пример красивой кнопки - ТЫК



бессмысленно но красиво - раскрывающиеся карты

ССЫЛКА НА КОД - ТЫК





Домашнее задание

- dropdown сделать выпадающий с анимацией
- expansion panel - сделать “гармошку” - раскрывающаяся панель, нажимаю на титульник и открывается описание с анимацией - ПРИМЕР ИЗ БИБЛИОТЕКИ BOOTSTRAP (не копировать оттуда код) для визуального восприятия
- сделать свой loader

Ресурсы

- трансформация теория - ТЫК
- трансформация теория (3D) - ТЫК
- переходы теория - ТЫК
- анимация теория - ТЫК
- трансформация документация - ТЫК
- переходы документация - ТЫК
- анимация документация - ТЫК