

Курсы HTML и CSS для начинающих





Анимация страницы “интерактивность сайта”

Занятие 5





С какими элементами может взаимодействовать пользователь

CSS анимация позволяет анимацию большинства HTML элементов без использования JavaScript или Flash.



Анимация CSS -

Анимация позволяет элементу постепенно переходить от одного стиля к другому. Вы можете изменить любое количество свойств CSS, сколько угодно раз.

3 преимущества анимации CSS:



- 1 Простота использования для простых анимаций; вы можете создать анимацию, не зная JavaScript.
- 2 Анимации будут хорошо работать даже при умеренных нагрузках системы.
- 3 Позволяет браузеру контролировать последовательность анимации, тем самым оптимизируя производительность и эффективность браузера.



Анимация CSS



Чтобы создать **CSS-анимацию** вы должны добавить в стиль элемента, который хотите анимировать, свойство `animation`.



UX [юикс] (User Experience) -

— дословно в переводе с английского «опыт пользователя». Это проектирование эффективного взаимодействия человека с сайтом, приложением, программой.

Два принципа хорошего UX дизайна:

1. Discoverability — понятность, наглядность
2. Feedback — обратная реакция, отклик

Правила **хорошего** дизайна - UX



- Избавьтесь от беспорядка
- Работа одной рукой
- Сделайте навигацию очевидной
- Текстовое содержание должно быть разборчивым
- Подсказки и обратная связь
- Сделайте элементы интерфейса четко видимыми
- Минимальный ввод данных
- Один опыт для всех платформ





Наложение эффектов -

Эффекты трансформации в CSS3 задаются с помощью функции **transform**

Наиболее интересно использовать эффекты с псевдоклассом **:hover**, которые дают нужный эффект при наведении мыши.

Логические операторы, применяемые в медиа-запросах



Псевдокласс **:hover**, который добавляется к селектору A, определяет состояние ссылки при наведённом на неё курсоре мыши.

Добавляйте эти эффекты с атрибутов **:hover**

- 1 Яркость filter: brightness(2);
- 2 Насыщенность filter: saturate(150%);
- 3 Оттенок освещения: filter: hue-rotate(250deg);
- 4 Контрастность filter: contrast(160%);
- 5 Прозрачность: filter: opacity(80%);



Логические операторы, применяемые в медиа-запросах



Псевдокласс **:hover**, который добавляется к селектору А, определяет состояние ссылки при наведённом на неё курсоре мыши.

Добавляйте эти эффекты с атрибутов **:hover**

- 6 Сепия: filter: sepia(0.6);
- 7 Оттенки серого: filter: grayscale(0.6);
- 8 Тень: filter: drop-shadow(5px 5px 5px #333);
- 9 Размытие: filter: blur(2px);
- 10 Инверсия: filter: invert(100%);

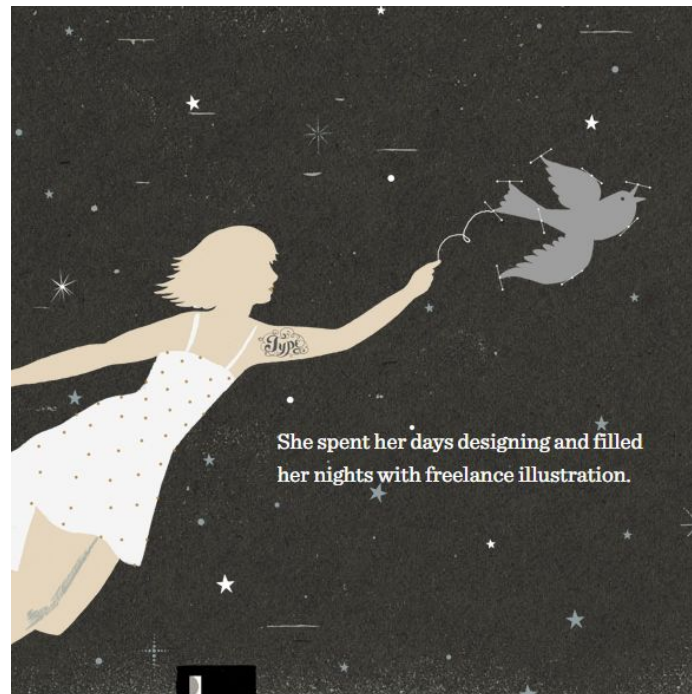


Эффект параллакса



Обязательным свойством для простейшей версии этого эффекта является **background-attachment: fixed**, фиксирующее расположение фонового изображения внутри окна просмотра (viewport).

[Пример](#)



Простая анимация: движение, вращение



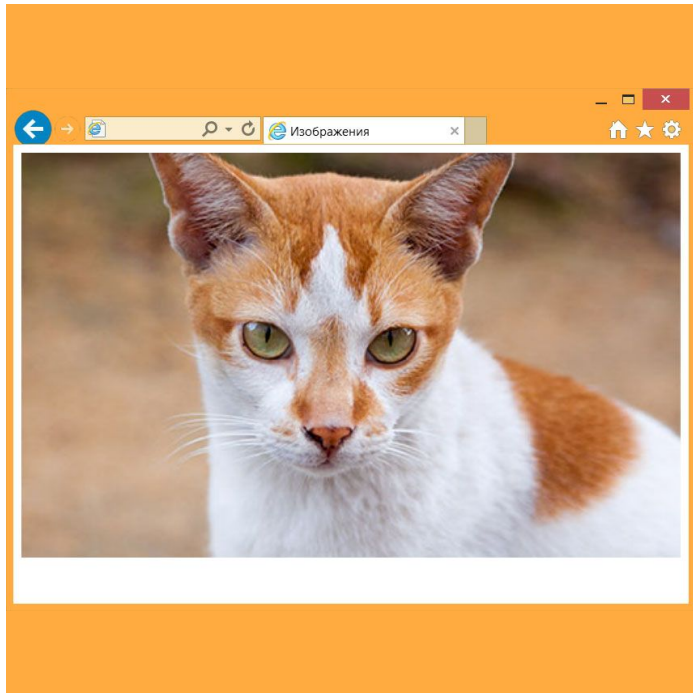
МАСШТАБИРОВАНИЕ

`transform: scale ()`

В функции `scale` используется либо один, либо два параметра (масштабирование по оси X и Y).

Для более плавного изменения масштаба используется параметр **`transition-duration`**, задающий время в секундах или миллисекундах, длительность анимации перехода до её завершения.

Характер анимации масштабирования задаётся параметром **`transition-timing-function`**

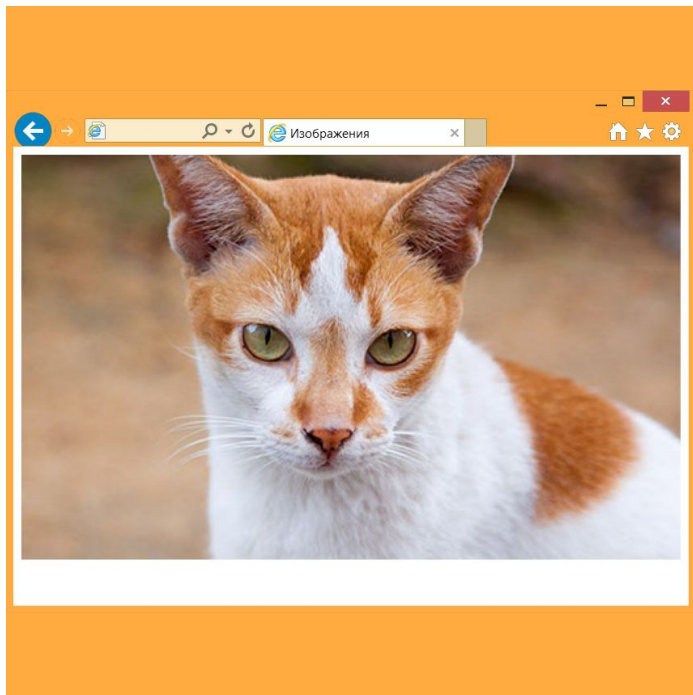


Простая анимация: движение, вращение



Значения параметра transition-timing-function:

- ease-out - анимация начинается быстро, к концу замедляется
- ease — масштабирование начинается медленно, затем ускоряется и к концу движения опять замедляется
- ease-in – масштабирование медленно начинается, к концу ускоряется
- ease-in-out — масштабирование начинается и заканчивается медленно



Простая анимация: движение, вращение

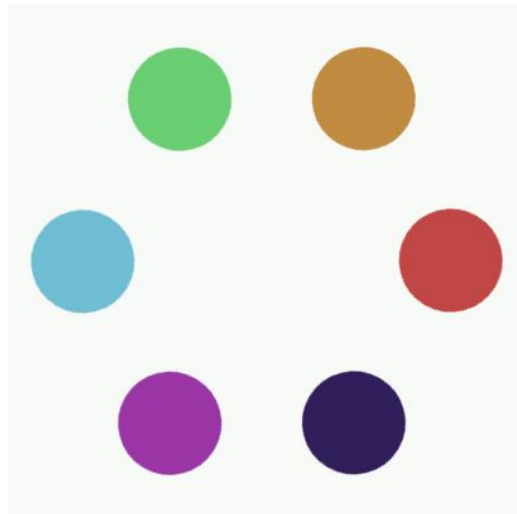


ВРАЩЕНИЕ

`transform: rotate(360deg);`
Задает вращение на 360 градусов.

СМЕЩЕНИЕ ЭЛЕМЕНТА

`transform: translate`



Простая анимация: движение, вращение



ИСКАЖЕНИЕ ЭЛЕМЕНТА

transform: skew (сдвиг)

Функция **skew** сдвигает элемент на определенный угол.

Свойство **transition-delay** задает время ожидания перед запуском эффекта перехода. Значение 1s запускает анимацию через секунду.

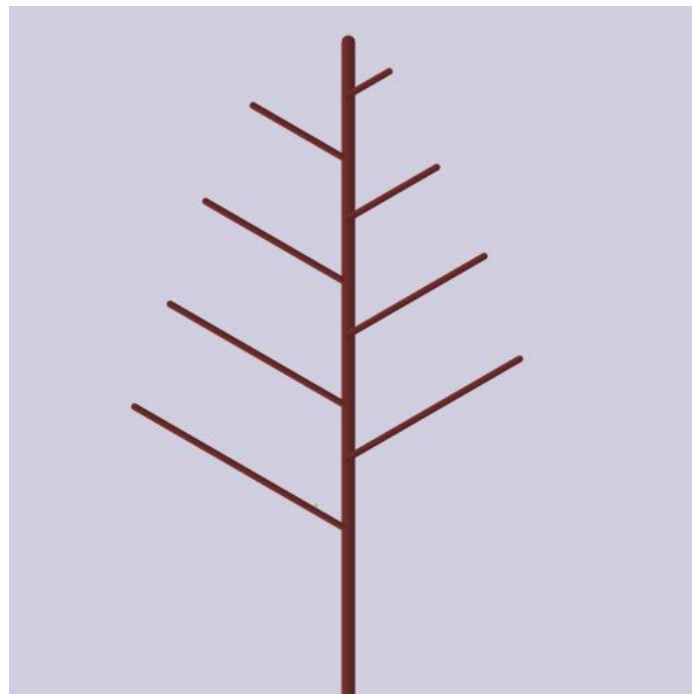
Свойство **transform-origin** задает координаты точки (в процента, пикселях), относительно которой будет происходить трансформация элемента.



Разработка более сложной анимации



Создание анимации базируется на ключевых кадрах, которые позволяют автоматически воспроизводить и повторять эффекты на протяжении заданного времени, а также останавливать анимацию внутри цикла.



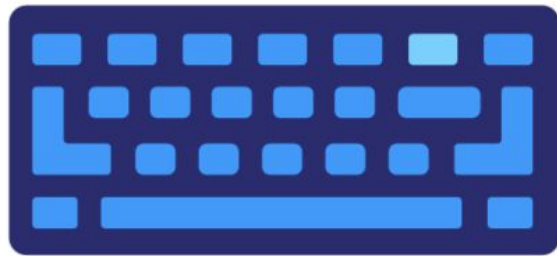
Правило @keyframes



При указании стилей CSS внутри **@keyframes** правило, анимация будет постепенно меняться от текущего стиля к новому стилю в определенное время.

@keyframes имя анимации { список правил }

Создание анимации начинается с установки ключевых кадров правила **@keyframes**.
Кадры определяют, какие свойства на каком шаге будут анимированы.

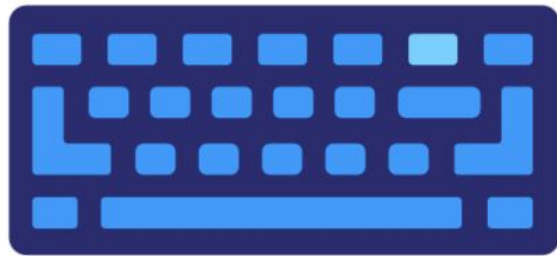


Правило @keyframes



Правило **@keyframes** содержит имя анимации элемента, которое связывает правило и блок объявления элемента.

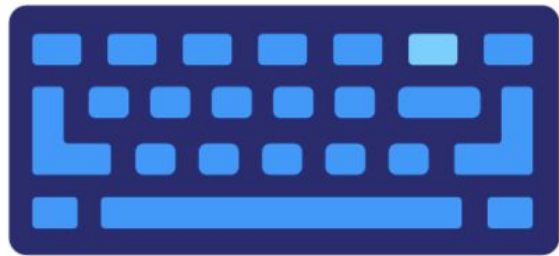
Ключевые кадры создаются с помощью ключевых слов **from** и **to** (эквивалентны значениям 0% и 100%) или с помощью процентных пунктов, которых можно задавать сколько угодно.



Правило @keyframes



- Не рекомендуется анимировать нечисловые значения
- Не следует создавать ключевые кадры для значений свойств, не имеющих средней точки, например, для значений свойства `color: pink` и `color: #ffffff`, `width: auto` и `width: 100px`



Правило @keyframes



Свойство:	Описание:
@keyframes	Указывает код анимации
animation	Сокращённое свойство для задания всех свойств анимации
animation-delay	Указывает задержку начала анимации
animation-direction	Указывает, следует ли воспроизвести анимацию вперёд, назад или в альтернативных циклах
animation-duration	Указывает, сколько времени должна занять анимация для завершения одного цикла

Правило @keyframes



Свойство:	Описание:
animation-fill-mode	Задаёт стиль элемента, если анимация не воспроизводится (до начала, после завершения или и то, и другое)
animation-iteration-count	Указывает, сколько раз должна воспроизводиться анимация
animation-name	Указывает имя анимации @keyframes
animation-play-state	Указывает, запущена ли анимация или приостановлена
animation-timing-function	Задаёт кривую скорости анимации

Временная функция и название анимации



Временная функция для ключевых кадров

Правило стиля ключевого кадра также может объявлять временную функцию, которая должна использоваться при перемещении анимации к следующему ключевому кадру.

Название анимации: свойство `animation-name`

Свойство `animation-name` определяет список применяемых к элементу анимаций.

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`

Временная функция: свойство animation-timing-function



Свойство animation-timing-function описывает, как будет развиваться анимация между каждой парой ключевых кадров.

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 <code>cubic-bezier(0.25,0.1,0.25,1)</code> .
ease-in	Анимация начинается медленно, а затем плавно ускоряется в конце. Соответствует <code>cubic-bezier(0.42,0,1,1)</code> .
ease-out	Анимация начинается быстро и плавно замедляется в конце. Соответствует <code>cubic-bezier(0,0,0.58,1)</code> .
ease-in-out	Анимация медленно начинается и медленно заканчивается. Соответствует <code>cubic-bezier(0.42,0,0.58,1)</code> .
cubic-bezier (x1, y1, x2, y2)	Позволяет вручную установить значения от 0 до 1. На этом сайте вы сможете построить любую траекторию скорости изменения анимации.

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).
steps - количество шагов - положение шага	Ступенчатая временная функция, которая принимает два параметра. Первый параметр указывает количество интервалов в функции. Со значением start анимация начинается в начале каждого шага, со значением end — в конце каждого шага с задержкой.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

Повтор анимации: свойство animation-iteration-count



Свойство animation-iteration-count указывает, сколько раз проигрывается цикл анимации.

Значения:

infinite

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

число

Анимация будет повторяться указанное количество раз.

Направление анимации: свойство animation-direction



Свойство animation-direction определяет, должна ли анимация воспроизводиться в обратном порядке в некоторых или во всех циклах.

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.

Направление анимации: свойство animation-direction



Свойство animation-direction определяет, должна ли анимация воспроизводиться в обратном порядке в некоторых или во всех циклах.

Значения:

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.

Значения:

running	Анимация выполняется. Значение по умолчанию.
paused	Анимация приостанавливается. При перезапуске анимация начинается с того места, где она была остановлена.
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.

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

Свойство `animation-fill-mode` определяет, какие значения применяются анимацией вне времени ее выполнения.

Краткая запись анимации: свойство `animation-delay`

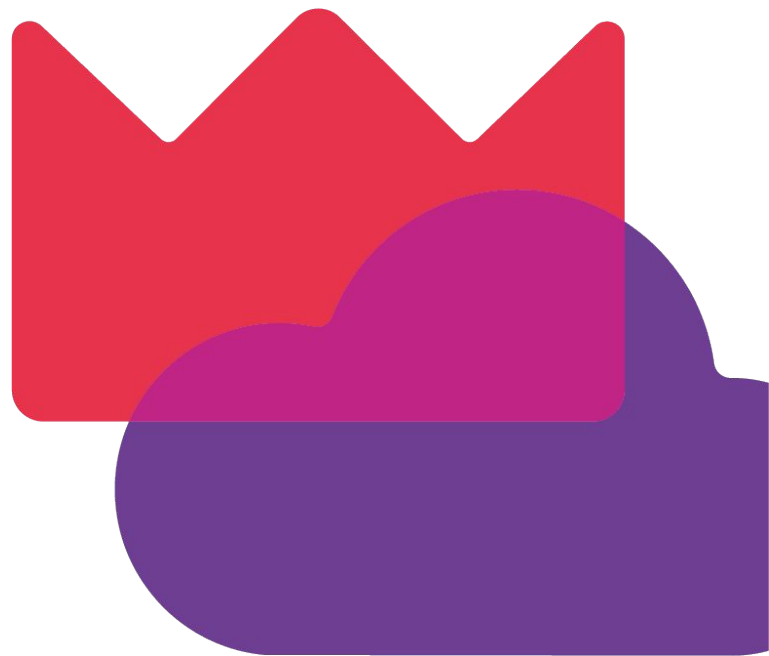


- Все параметры воспроизведения анимации можно объединить в одном свойстве — `animation`, перечислив их через пробел.
- Для одного элемента можно задавать несколько анимаций, перечислив их названия через запятую.



Домашнее задание:

- 1 Анимирование своей страницы-портфолио
- 2 Добавлении анимации при наведении мыши
- 3 Создание анимации со всеми критериями



Теперь вы знаете:

- ✓ Что из себя представляет css анимация
- ✓ Как создать красивую анимацию
- ✓ Основы дизайна для создания хорошего сайта
- ✓ Создание анимации с помощью ключей

