JAVARUSH

Путь к карьере Frontend Fullstack разработчика

Модуль 1. WEB CORE

Уровень 12. Трансформации. Переходы и анимация. Часть 2

JAVARUSH

Основы CSS-анимации

Первые анимации реализовывались при помощи Flash и JavaScript. Позже многие инструменты были внедрены в CSS:

- animation-name: Имя анимации, которое связывает её с ключевыми кадрами, определенными в @keyframes.
- animation-duration: Длительность анимации (например, 2s).
- animation-iteration-count: Количество повторений анимации (например, infinite для бесконечного повторения).
- animation-direction: Направление воспроизведения анимации (normal, reverse, alternate, alternate-reverse).
- animation-timing-function: Функция плавности анимации (ease, linear, ease-in, ease-out, ease-in-out и другие).
- animation-delay: Задержка перед началом анимации (например, 1s).
- animation-play-state: Состояние анимации (running или paused).
- animation-fill-mode: Определяет, какие стили применять до и после анимации (none, forwards, backwards, both).
- animation: Шорткат для всех вышеперечисленных свойств.

JAVARUSH

@keyframes: Магия ключевых кадров

Директива @keyframes используется для определения последовательности стилей, которые будут меняться во время анимации.

После ключевого слова @keyframes мы должны написать **имя анимации**.

Внутри @keyframes указываются процентные значения (от 0% до 100%) или ключевые слова (from и to), определяющие моменты времени, в которые будут применяться определенные стили.

Если кадров больше двух, то можно использовать проценты. Браузер расшифровывает ключевое слово from как 0%, а ключевое слово to как 100%.

```
@keyframes animation-name {
    from {
        /* начальные стили */
    }

    to {
        /* конечные стили */
    }
```

JAVARUSH

@keyframes: Магия ключевых кадров

Представьте, что у нас есть красный круг, который мы хотим превратить в синий квадрат. Для этого нам нужно изменить его ширину, высоту и цвет фона.

Как это работает:

- @keyframes: Ключевое слово, указывающее на начало определения анимации.
- circle-to-square: Уникальное имя анимации, которое будет использоваться для её применения к элементам.
- from и to: Ключевые слова, обозначающие начальный (0%) и конечный (100%) кадры анимации.
- Внутри каждого кадра указываются CSS-свойства и их значения, которые будут применяться к элементу в этот момент времени.

```
@keyframes circle-to-square {
    from { /* Начальное состояние */
        width: 50px;
        height: 50px;
        background-color: #ff0000; /* Красный  */
    }
    to { /* Конечное состояние */
        width: 200px;
        height: 200px;
        background-color: #2E9AFF; /* Синий */
    }
}
```

JAVARUSH

Промежуточные кадры

Вы можете добавить промежуточные кадры, используя процентные значения:

```
@keyframes circle-to-square {
   from { /* 0% */
       /* ... */
   }
   50% { /* Промежуточный кадр */
       width: 50px;
       height: 200px;
       background-color: #7F6EDB; /* Фиолетовый */
   }
   to { /* 100% */
       /* ... */
   }
}
```

JAVARUSH

Даём имя анимации

Свойство animation-name связывает элемент с определённой анимацией, заданной в ключевых кадрах (@keyframes).

```
.child-one {
   animation-name: circle-to-square; /* Применяем анимацию с именем "circle-to-square" */
}
```

Значение none:

- Отменяет анимацию для элемента.
- Полезно для сброса анимации, например, при наведении курсора.

```
.element {
   animation: some-animation;
}

.element:hover {
   animation: none; /* Отменяем анимацию при наведении */
}
```

Для работы анимации нужно также указать её длительность (animation-duration).

JAVARUSH

Длительность анимации

Свойство animation-duration определяет, сколько времени займет один полный цикл вашей анимации.

Единицы измерения:

- Секунды (s)
- Миллисекунды (ms)

```
.child-one {
   animation-name: circle-to-square;
   animation-duration: 5s; /* Анимация будет длиться 5 секунд */
}
```

JAVARUSH

Сколько раз повторять анимацию?

Свойство animation-iteration-count позволяет вам контролировать, сколько раз анимация будет воспроизводиться. Вы можете указать конкретное число повторений или использовать ключевое слово infinite для бесконечного повтора.

По умолчанию анимация воспроизводится только один раз.

```
.child-one {
   animation-name: circle-to-square;
   animation-duration: 5s;
   animation-iteration-count: infinite; /* Анимация будет повторяться бесконечно */
}
```

JAVARUSH

Управление направлением анимации

Свойство animation-direction позволяет вам контролировать направление воспроизведения анимации. Это особенно полезно при создании циклических анимаций или эффектов, которые должны повторяться в разных направлениях.

- normal — значение по умолчанию, анимация воспроизводится от начала до конца, после чего возвращается к начальному кадру.
- reverse — анимация проигрывается в обратном порядке, от последнего ключевого кадра до первого, после чего возвращается к последнему кадру.
- alternate — каждый нечётный повтор (первый, третий, пятый) анимации воспроизводится в прямом порядке, а каждый чётный повтор (второй, четвёртый, шестой) анимации воспроизводится в обратном порядке.
- alternate-reverse — аналогично значению alternate, но чётные и нечётные повторы меняются местами.

```
.child-one {
   animation-name: circle-to-square;
   animation-duration: 5s;
   animation-iteration-count: infinite;
   animation-direction: alternate; /* Анимация будет повторяться
бесконечно, меняя направление каждый цикл */
}
```

JAVARUSH

Плавность и динамика анимации

Свойство animation-timing-function определяет, как будет меняться скорость анимации между ключевыми кадрами.

Стандартные функции:

- linear: Анимация проигрывается с постоянной скоростью, без ускорения или замедления.
- ease (по умолчанию): Анимация начинается медленно, затем ускоряется и снова замедляется к концу.
- ease-in: Анимация начинается медленно и постепенно ускоряется.
- ease-out: Анимация начинается быстро и постепенно замедляется к концу.
- ease-in-out: Анимация начинается и заканчивается медленно, ускоряясь в середине.

Пример Gif- https://media.geeksforgeeks.org/wp-content/cdn-uploads/20190826143402/transition.gif
Шпаргалка по функциям плавности https://easings.net/

Кастомная функция: cubic-bezier(x1, y1, x2, y2)

- Позволяет создавать собственные кривые ускорения с помощью кривой Безье.
- Оси: X - временная шкала анимации, Y - прогресс анимации.
- Значения x1 и x2 должны быть в диапазоне от 0 до 1.
- Значения y1 и y2 могут выходить за пределы 0 и 1, создавая эффект "пружины" или "отскока".

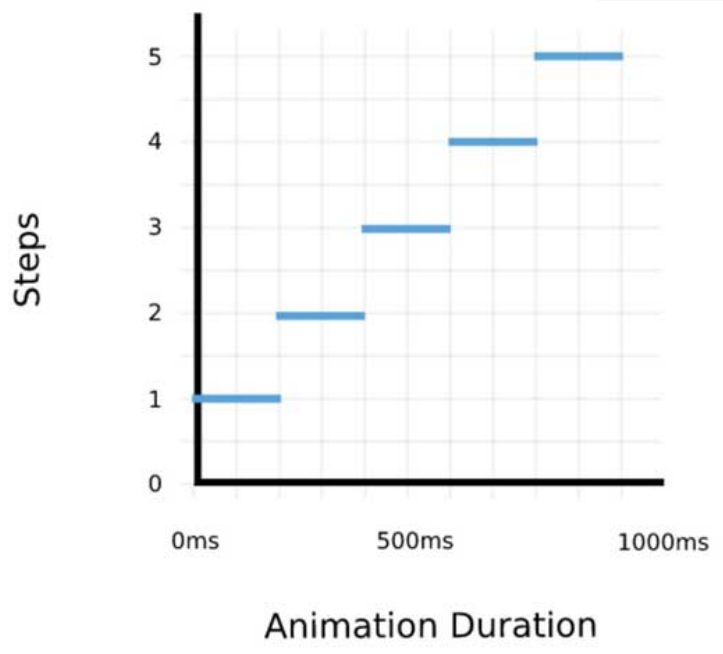
Один из популярных инструментов — cubic-bezier.com

Шаги

Временная функция **steps(количество шагов[, start/end])** позволяет разделить анимацию на шаги.

Варианты:

- step-start: Изменения происходят в начале каждого шага анимации.
- step-end: Изменения происходят в конце каждого шага анимации (по умолчанию).
- steps(количество_шагов, положение_шага):
  - количество_шагов: Целое число, определяющее количество шагов в анимации.
  - положение_шага: Опциональный параметр, указывающий, когда происходят изменения:
    - jump-start / start: В начале каждого шага.
    - jump-end / end: В конце каждого шага.
    - jump-none: Все шаги происходят внутри интервала анимации, без изменений в начале и конце.
    - jump-both: Изменения происходят и в начале, и в конце каждого шага.




```
.element {
   animation-timing-function: steps(5, end); /*
Анимация из 5 шагов, изменения в конце каждого шага */
}
```

Примеры — https://danielcwilson.com/blog/2019/02/step-and-jump/

JAVARUSH

Управление стартом анимации

Свойство animation-delay позволяет вам задать задержку перед началом воспроизведения анимации. Это может быть полезно для создания последовательных анимаций или эффектов, которые должны запускаться с определённым интервалом.

- **Положительное число:** Задержка перед началом анимации (например, 2s - задержка в 2 секунды).
- **Отрицательное число:** Анимация начинается не с первого кадра, а с того, который соответствует указанному времени (например, -2.5s - анимация начнётся с середины, если её длительность 5 секунд).

```
.child-two {
   animation-name: circle-to-square;
   animation-duration: 5s;
   animation-iteration-count: infinite;
   animation-direction: alternate-reverse;
   animation-timing-function: ease-in;
   animation-delay: -2.5s; /* Анимация начнётся с середины */
}
```

JAVARUSH

Контролируем воспроизведение анимации

Свойство animation-play-state дает вам возможность управлять анимацией, ставя ее на паузу и возобновляя воспроизведение.

- running: Анимация проигрывается (значение по умолчанию).
- paused: Анимация приостановлена. При повторном запуске она продолжится с того места, где была остановлена.

```
.child:hover {
   animation-play-state: paused; /* При наведении курсора анимация останавливается */
}
```

JAVARUSH

Супер-шорткат

Свойство animation - это настоящий чемпион по экономии времени!
Оно позволяет вам задать все свойства анимации в одной строке, делая ваш CSS-код более компактным и читаемым.

```
.child-two {
   animation: circle-to-square 2s infinite alternate-reverse ease-in 1s;
}
```

Важно:

- Порядок значений не важен, браузер сам разберётся, какое значение к какому свойству относится.
- Первое значение времени интерпретируется как animation-duration, второе - как animation-delay.
- Необязательно указывать все значения, можно задать только самые необходимые.

JAVARUSH

Несколько анимаций

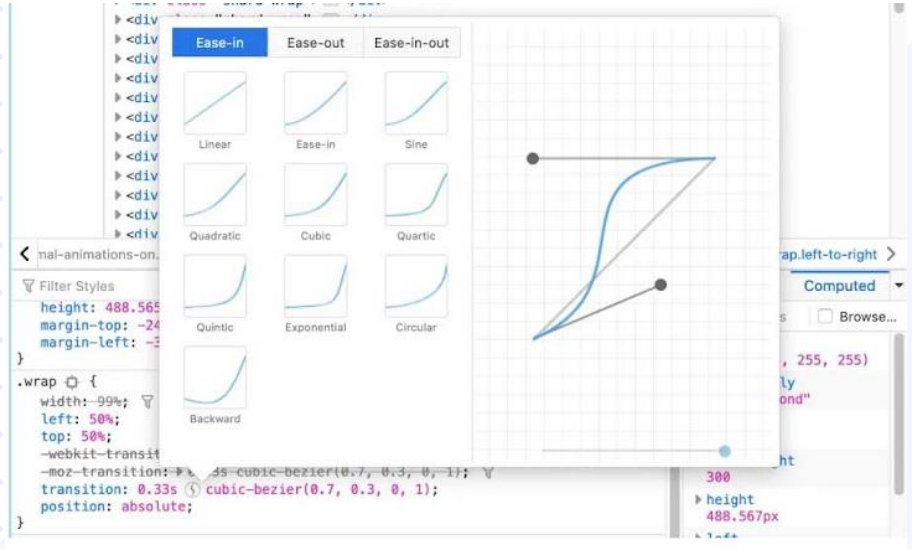
CSS позволяет применять к одному элементу сразу несколько анимаций, что открывает огромные возможности для создания сложных и интересных визуальных эффектов.

```
/* Анимация изменения формы */
@keyframes circle-to-square {
   /* ... */
}

/* Анимация изменения цвета */
@keyframes color-change {
   /* ... */
}

.child {
   animation:
           circle-to-square 10s infinite alternate ease-out 1s, /* Первая
анимация с задержкой 1s */
           color-change 5s alternate linear infinite; /* Вторая анимация */
}
```

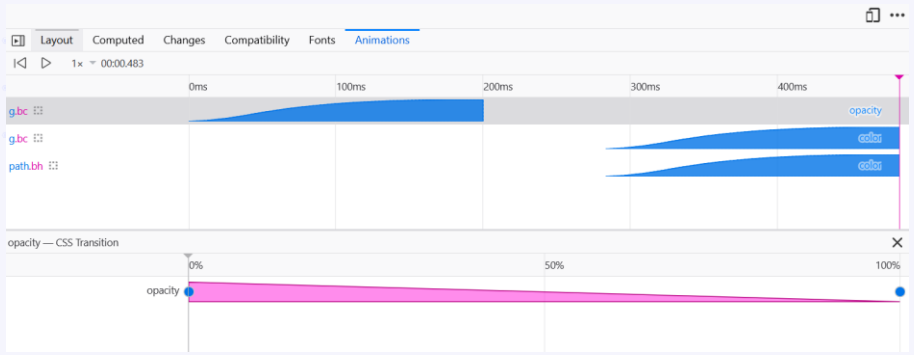
Инструменты браузера

Инструменты разработчика в браузерах предоставляют полезный инструментарий редактирования функций плавности.

Редактирования здесь доступны только функции кубических кривых Безье.



Chrome, Safari и Firefox предлагают отдельную вкладку в инструментах разработчика, посвященную анимациям.



Отображение easeInSine - https://easings.net/

JAVARUSH

Анимации для улучшения UX

Они могут направлять внимание пользователя, обеспечивать обратную связь, улучшать навигацию и делать интерфейсы более интуитивно понятными и приятными для использования.

Анимации при нажатии на элементы могут улучшить обратную связь для пользователей и сделать взаимодействие более увлекательным.

```
/*Анимация нажатия на кнопку*/
<style>
.button {
    padding: 10px 20px;
    background-color: #e74c3c;
    color: white;
    border: none;
    cursor: pointer;
    transition: transform 0.1s ease;
}

.button:active {
    transform: scale(0.95);
}
</style>
```

Cool CSS animation

JAVARUSH

Домашнее задание

Уровень 12. Трансформации. Переходы и анимация