

## Практическая работа №7. Анимация на чистом CSS

**Цель работы:** познакомиться с самым простым способом создания анимации — CSS-свойством **transition**, освоить принципы анимирования, и потренироваться в анимации логотипа.

### 1. CSS-анимации

CSS-анимации — самый простой способ анимировать объекты на странице. Он позволяет вращать, растягивать, уменьшать и увеличивать объект, перемещать его или выполнять другие действия.

CSS-анимации создают с помощью группы свойств **animation**. Они задают длительность анимации, количество повторений, зацикленность, стороны движения и другие характеристики. CSS-анимации можно создать и с помощью свойства **transition**, но оно работает только при загрузке страницы, добавлении классов или изменении состояния, к примеру, **hover**.

Также с помощью CSS можно анимировать векторную графику — SVG. Например, если в проекте есть SVG-иконки, можно заставить их контур постепенно появляться.

#### Задание 1.

1. Изучить статью об [анимации дерева](#). Внимательно прочитать, выполнить все шаги, указанные в описании.
2. В песочнице [codepen.io](#) создать свою собственную версию дерева. Можно изменять не только отдельные параметры, отвечающие за анимацию, но и параметры, определяющие форму элементов, цветовые сочетания и даже количество деревьев на полотне.
3. В качестве результата прикрепить ссылку на «пен».

### 2. SVG и CSS анимация для иконки

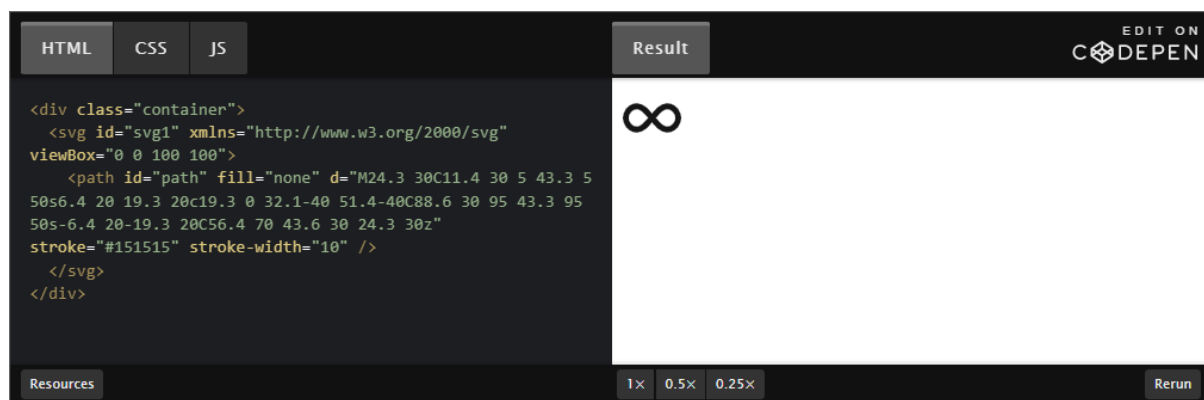
Посмотрите на эти [анимированные иконки](#). Как вы думаете, в чем их «магия»?

То, что они делают, это скользящая прямоугольная маска справа налево, которая покрывает первую иконку и открывает вторую иконку. По ссылке можно найти упрощенную версию [разметки](#), использующей чистый CSS, который делает ее более понятной.

По следующей [ссылке](#) вы можете скачать так называемые «лоадеры» — иконки, которые задерживают внимание пользователя на экране во время загрузки контента. Давайте анимируем [«Running Infinity»](#)

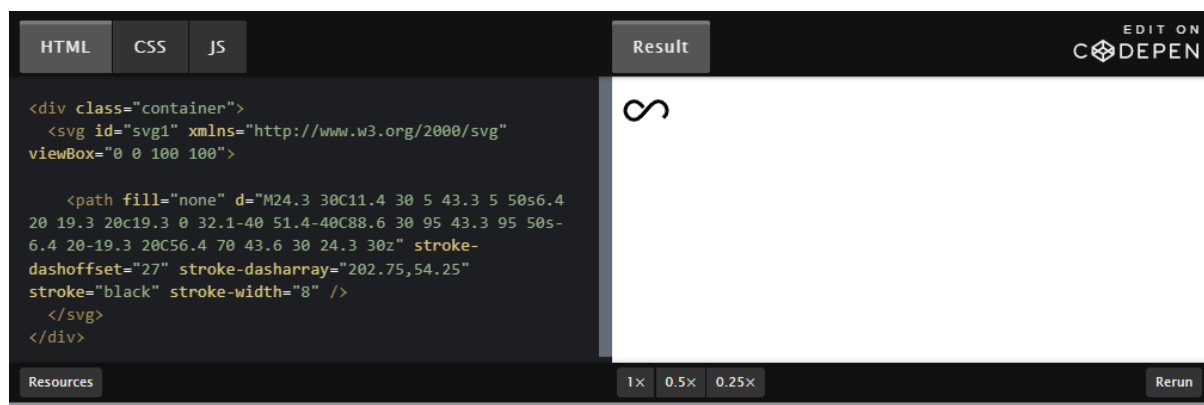
## Создание анимации иконки по шагам:

**Шаг#1.** В векторном редакторе рисуем восьмерку и вычисляем полную длину контура - `getTotalLength()` (смотри [Pen](#))



**Шаг#2.** Вырезаем часть контура с помощью `stroke-dasharray="202.75,54.25"`, где 54.25 – пробел

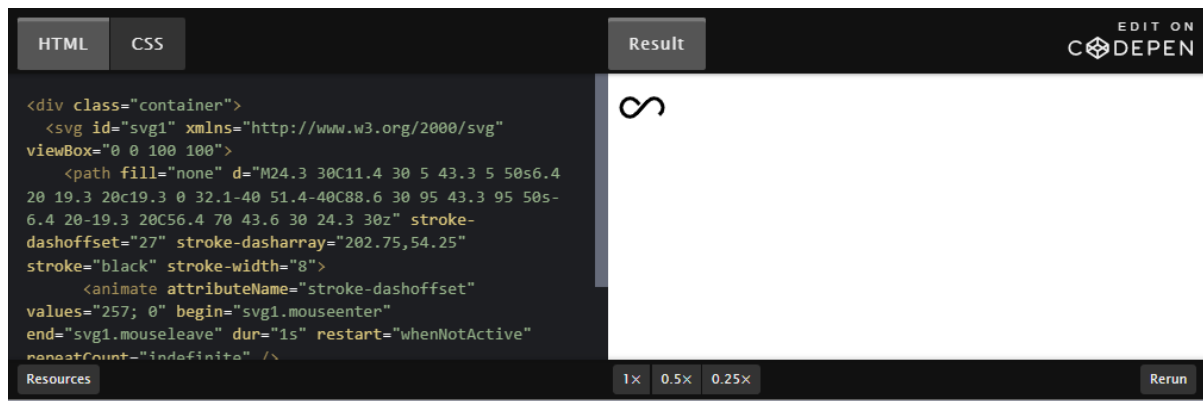
**Обратите внимание:** если сложить цифры длин черты и пробела получится максимальная длина контура 257px (смотри [Pen](#))



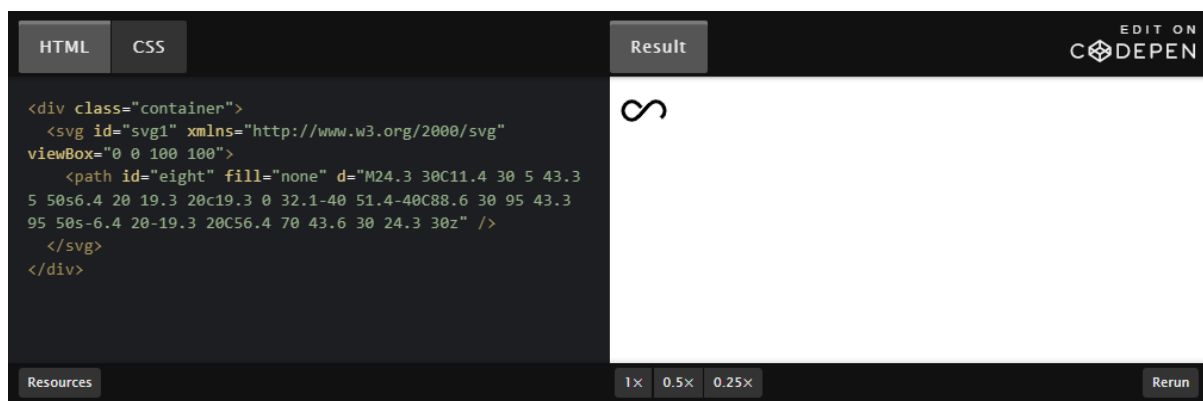
**Шаг#3.** Добавляем анимацию вращения контура с помощью `stroke-dashoffset` (смотри [Pen](#))



**Шаг#4.** Вращение иконки при наведении :hover. Добавляем в условие запуска begin="svg1.mouseenter" end="svg1.mouseleave" (смотри [Pen](#))



**Шаг#5.** При разработке варианта анимации SVG были вычислены необходимые значения параметров атрибутов stroke-dasharray:202.75,54.25; и stroke-dashoffset. Остается их перенести в CSS правила (смотри [Pen](#)):



**Задание 2.** Выбрать свой вариант [анимированного изображения](#) и реализовать этот пример в codepen.

**Задание 3.** Изучить пример с построением SVG диаграммы (<https://htmlacademy.ru/blog/boost/tutorial/svg-chart> ). Подобрать собственный контент для диаграммы. Создать свою версию диаграммы в codepen по примеру, описанному в статье.

## Вспомогательные материалы

1. [Тutorial. Простая анимация на чистом CSS — Блог HTML Academy](#)
2. [Как проектировать, создавать и анимировать SVG — Блог HTML Academy](#)
3. [15 примеров SVG-анимации для веб-дизайнеров – Институт Навыков](#)
4. <https://svgontheweb.com/ru/>