



## Как создать CSS-анимацию при прокрутке

### Описание

Вы создаете веб-страницу и застряли в раздумьях, как сделать ее более удобной для читателя? Если вы хотите добавить интерактивности вашей веб-странице, CSS-анимация при прокрутке поможет вам в этом. Конечно, вы можете добавить фотографии и видео, чтобы помочь воплотить ваше видение в жизнь. Однако CSS-анимация может добавить дополнительное движение и удержать внимание посетителей. Самое главное, что их относительно легко создать. В этом посте вы узнаете, что такое CSS-анимация при прокрутке, как анимировать CSS при прокрутке, а также полезные примеры. С помощью всего нескольких строк кода вы сможете быстро создать великолепно выглядящую анимацию. Давайте приступим!

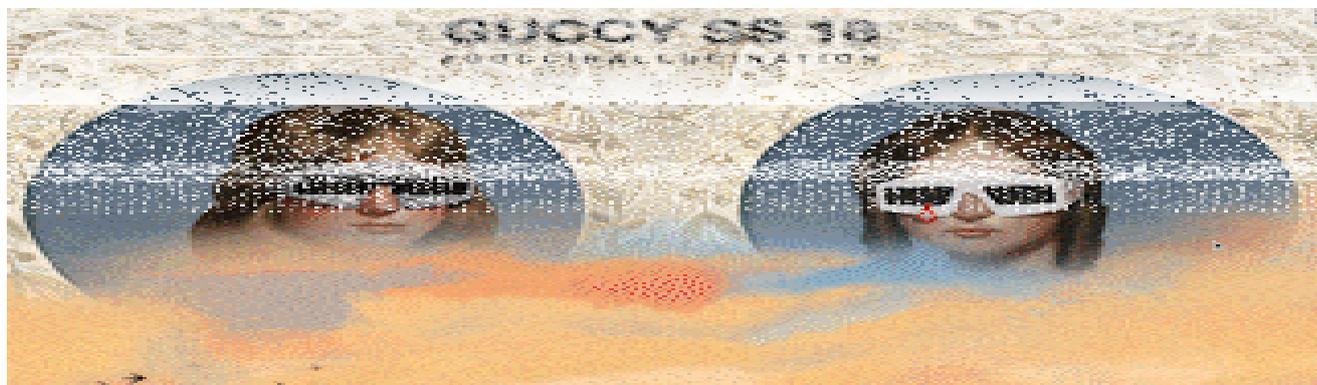
### Что такое CSS-анимация при прокрутке?

Представьте, что вы хотите привлечь внимание читателя при прокрутке веб-страницы вверх и вниз – например, анимировать текст, изменить цвет элементов или сделать их прыгающими. С помощью CSS-анимации вы можете использовать плавные переходы между состояниями, создавать сложные многоступенчатые анимации и даже анимировать элементы SVG – и все это без использования сложного кода JavaScript. Вы можете активировать эти анимации, создавая CSS-анимации при прокрутке. Когда читатели прокручивают веб-страницу вверх и вниз, анимация включается, как только элемент попадает в поле зрения. Внезапное движение элементов будет удерживать читателя. Это происходит потому, что наши глаза привлекают изменения в поведении объектов. Всего лишь небольшое движение и изменения могут оживить вашу страницу.

## CSS на прокрутке в действии

Теперь, когда вы знаете, на что способна анимация CSS, давайте посмотрим на живые веб-страницы, использующие анимацию прокрутки.

### [Gucci Hallucination](#)



Роскошный модный бренд Gucci использует анимацию прокрутки на всей своей веб-странице. Анимация оживляет их искусство. Пока вы прокручиваете веб-страницу вверх и вниз, ваши глаза будут сосредоточены на анимации.

### [Year In Review](#)



Digital and Motion создает смелые визуальные образы с помощью CSS-анимации на прокрутке. Посмотрите на их визуальную капсулу времени на 2018 год.

Прокручивая страницу вверх и вниз, вы путешествуете по их году воспоминаний.

### [Fiji Water](#)



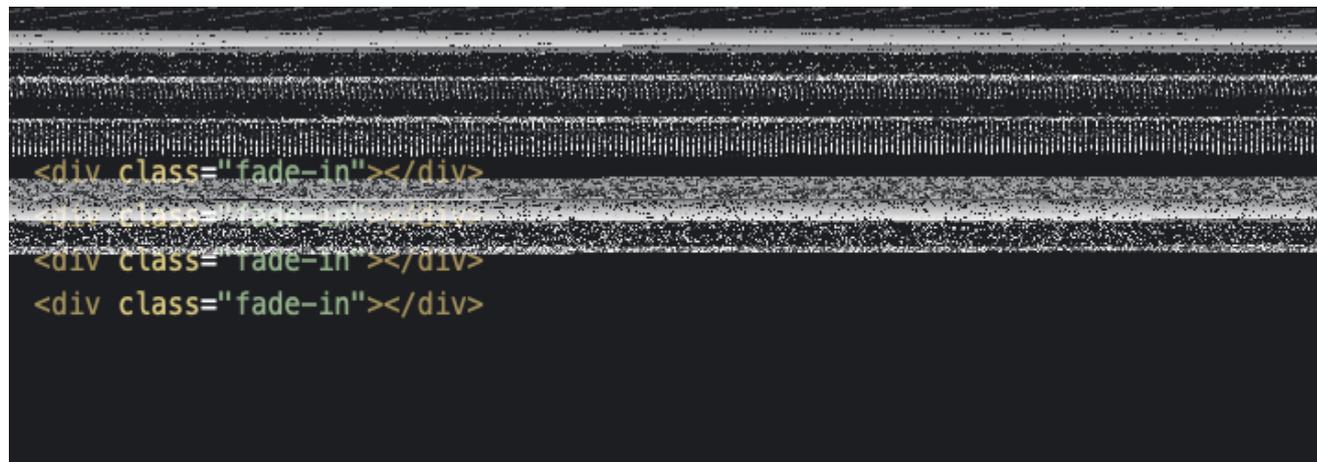
Бренд воды Fiji создает визуальные образы природы в качестве темы для своего веб-сайта. Прокручивая страницу вверх и вниз, можно увидеть, как оживают облака. Вы также можете увидеть, как капли дождя падают с того, что кажется небом. Нам нравится, как эта анимация сочетается с брендингом компании.

## Как анимировать CSS при прокрутке

Теперь, когда вы знаете о преимуществах использования CSS-анимации прокрутки, давайте создадим несколько.

## Начните с HTML-кода.

Для создания CSS-анимации прокрутки мы начнем с HTML. Создайте простую веб-страницу и дайте элементам имена классов. В примере ниже вы увидите четыре div, которым присвоено одно и то же имя класса - "fade-in".



```
<div class="fade-in"></div>  
<div class="fade-in"></div>  
<div class="fade-in"></div>  
<div class="fade-in"></div>
```

## Создайте CSS.

Далее вы создадите CSS. Именно здесь вы будете управлять анимацией и эффектами прокрутки. В данном примере класс active добавляется к изображению по мере его прокрутки в область просмотра. Атрибут transform придает веб-странице динамичность, а атрибут translation определяет время загрузки.

```
body {  
  background: #E0BBE4;  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
  justify-content: center;  
}  
  
.fade-in {  
  background: #957DAD;  
  height: 300px;  
  width: 300px;  
  margin-bottom: 100px;  
  transform: translateY(40%);  
  transition: all 1s ease-out;  
}  
  
.active {  
  opacity: 4;  
  transform: translateY(0);  
}
```

Наконец, вы создадите наблюдателя пересечений с помощью нескольких строк JavaScript.

## Построение наблюдателя пересечений с помощью JavaScript.

Первым шагом будет захват элементов с помощью `querySelectorAll`.

```
const elements = document.querySelectorAll('.fade-in');
```

Затем создайте объект `options`. Установите для `root` значение `null`, которое устанавливает область просмотра в документ. Затем установите для `root` `margin` значение `0px`, которое определяет стороны ограничивающего поля корня.

Используйте порог для определения точки срабатывания элемента в зависимости от его видимости. В приведенном ниже примере триггер элемента установлен на момент, когда элемент виден на 40%.

```
const callbacks = {  
  root: null,  
}
```

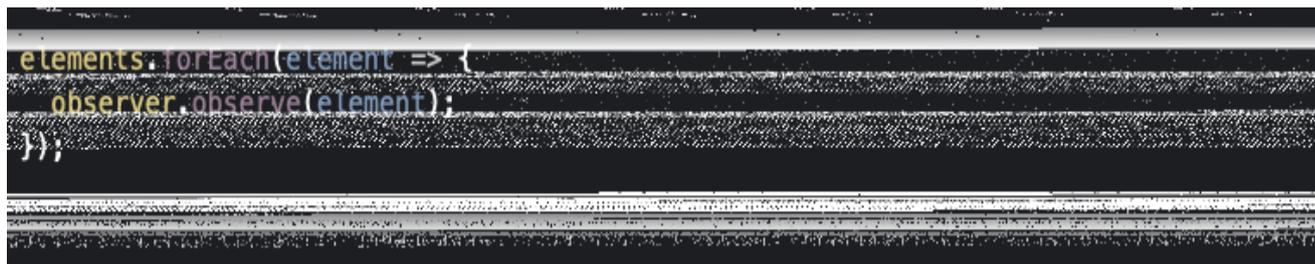
Следующий шаг – создание функции обратного вызова. Функция обратного вызова принимает записи в качестве параметра. Вы можете использовать метод `forEach` для итерации по списку записей. Функция обратного вызова определяет, пересекается ли элемент в данный момент с корнем, и возвращает именно этот элемент.

```
const callbacks = (entries) => {  
  entries.forEach(entry => {  
    if (entry.isIntersecting){  
      // ...  
    }  
  });  
};
```

Затем создайте наблюдателя пересечения, вызвав его конструктор. Затем вы передадите конструктору функцию обратного вызова и объект `options`.

```
let observer = new IntersectionObserver(cb, options);
```

Последний шаг – выбор элементов для наблюдения. Если у вас несколько элементов, вы упростите код, присвоив им общее имя класса и используя метод `forEach` для перебора каждого элемента, когда он входит или выходит из области просмотра.



## Оживите свой сайт

Как видите, использование CSS-анимации при прокрутке предлагает простой и эффективный способ добавить движение и анимацию на ваш сайт или приложение. Вы можете создавать красивые, привлекающие внимание анимации с помощью CSS-анимаций прокрутки, которые реагируют на действия пользователя или события прокрутки. Кроме того, работать с CSS намного проще, чем с JavaScript – так что если вы ищете быстрый способ добавить жизни в свой веб-проект без написания кода с нуля, CSS – это то, что вам нужно! Попробуйте его сегодня.

### Дата Создания

25.04.2023