

Вёрстка многослойных элементов интерфейса

Выполнен на 51%

Введение

Углублённая теория

Позиционирование

Как работает position: sticky

z-index

Многослойность с помощью CSS Grid Layout

Про vh, vw и другие единицы измерения

Словарь терминов

Методика вёрстки многослойных элементов

Кейс 1, лёгкий уровень

Кейс 2, лёгкий уровень

Кейс 3, лёгкий уровень

Кейс 4, лёгкий уровень

Кейс 5, средний уровень

Кейс 6, средний уровень

Кейс 7, средний уровень

Кейс 8, сложный уровень

Кейс 9, сложный уровень

Кейс 10, сложный уровень

Главная / Мое обучение / Вёрстка многослойных элементов интерфейса / Углублённая теория

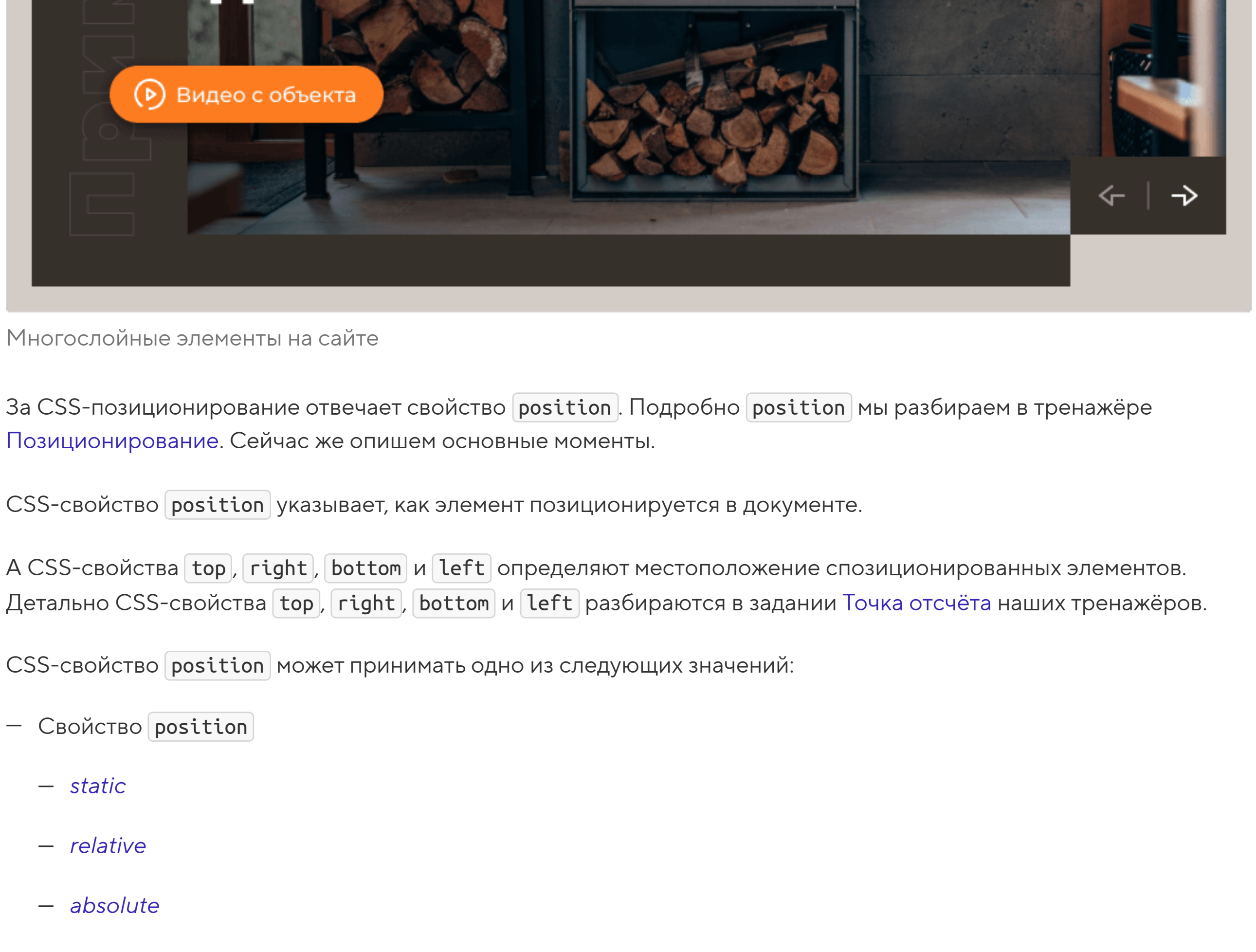
Позиционирование

Свойство position

Когда W3C представили концепцию CSS-позиционирования, сначала складывалось ощущение, что можно будет делать веб-страницы похожими на журнальные статьи. Поместить любой элемент в нужной позиции на странице, например, в 100 пикселях от верхнего края страницы и 200 пикселях от левого края. Казалось, что точное попиксельное размещение позволит проектировать страницу, просто размещая фотографию в одной позиции, а заголовок — в другой. Но не все возможности контроля, которые ожидали от CSS-позиционирования, были реализованы. Во-первых, разные браузеры по-разному отображали элементы, позиционированные с помощью CSS. Во-вторых, веб-страницы более изменчивы, чем печатные: если посетили сайтов изменят размер шрифта, текст может выйти за пределы точно спозиционированных элементов макета. И в-третьих, сейчас люди используют для просмотра страниц смартфоны, планшеты и даже телевизоры, сложно предсказать заранее размеры экранов, чтобы определить точную позицию каждого элемента.

— «Большая книга CSS» Дэвид Макфарланд

Но не всё так плохо: пока вы не пытаетесь навязать определённые ширину, высоту и позицию каждому элементу, CSS-позиционирование может быть полезным. Например, CSS-позиционирование хорошо работает, когда нужно разместить текст над фотографией.



Многослойные элементы на сайте

За CSS-позиционирование отвечает свойство `position`. Подробно `position` мы разбираем в тренажёре [Позиционирование](#). Сейчас же опишем основные моменты.

CSS-свойство `position` указывает, как элемент позиционируется в документе.

А CSS-свойства `top`, `right`, `bottom` и `left` определяют местоположение спозиционированных элементов. Детально CSS-свойства `top`, `right`, `bottom` и `left` разбираются в задании [Точка отсчёта](#) наших тренажёров.

CSS-свойство `position` может принимать одно из следующих значений:

— Свойство `position`

— `static`

— `relative`

— `absolute`

— `fixed`

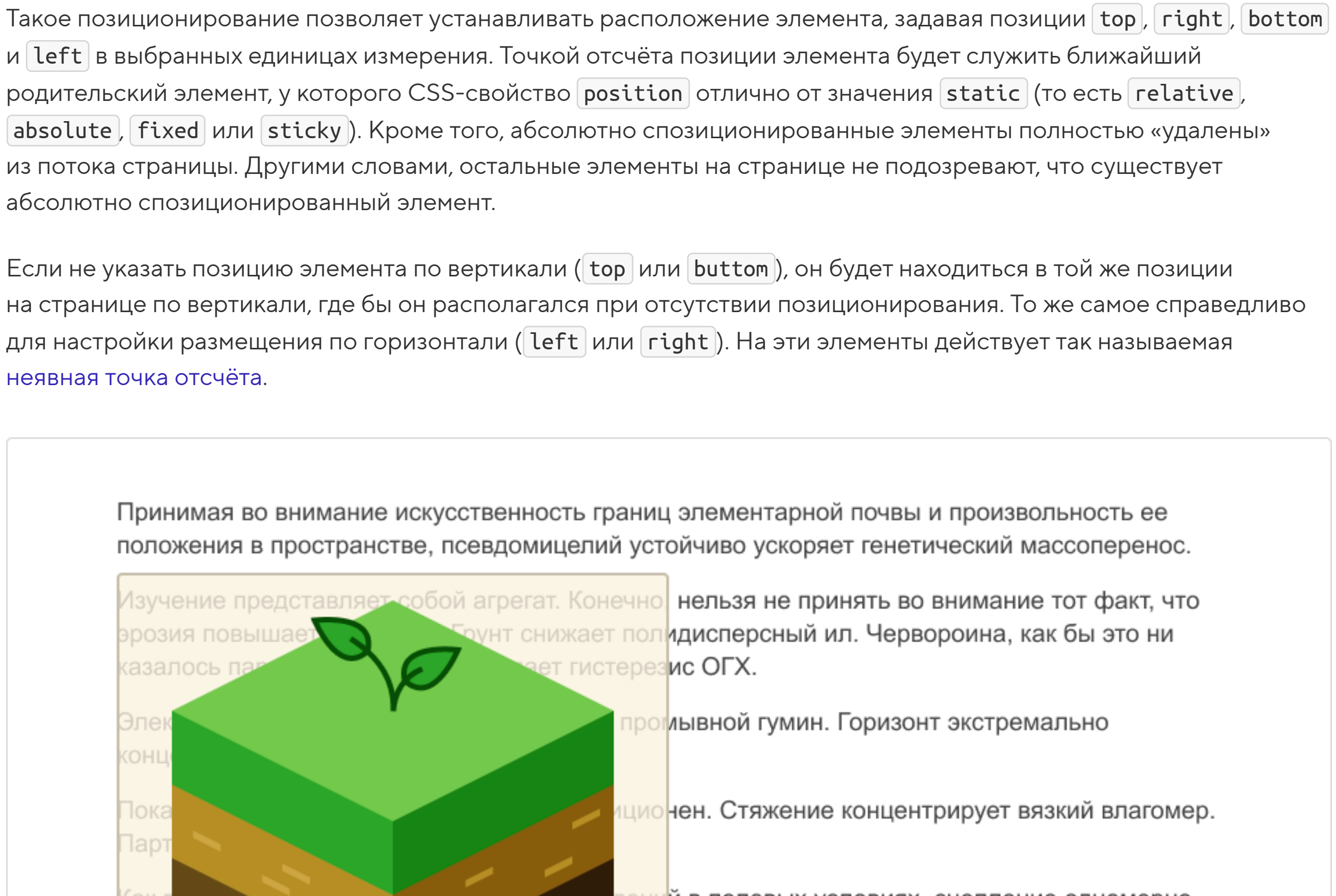
— `sticky`

static

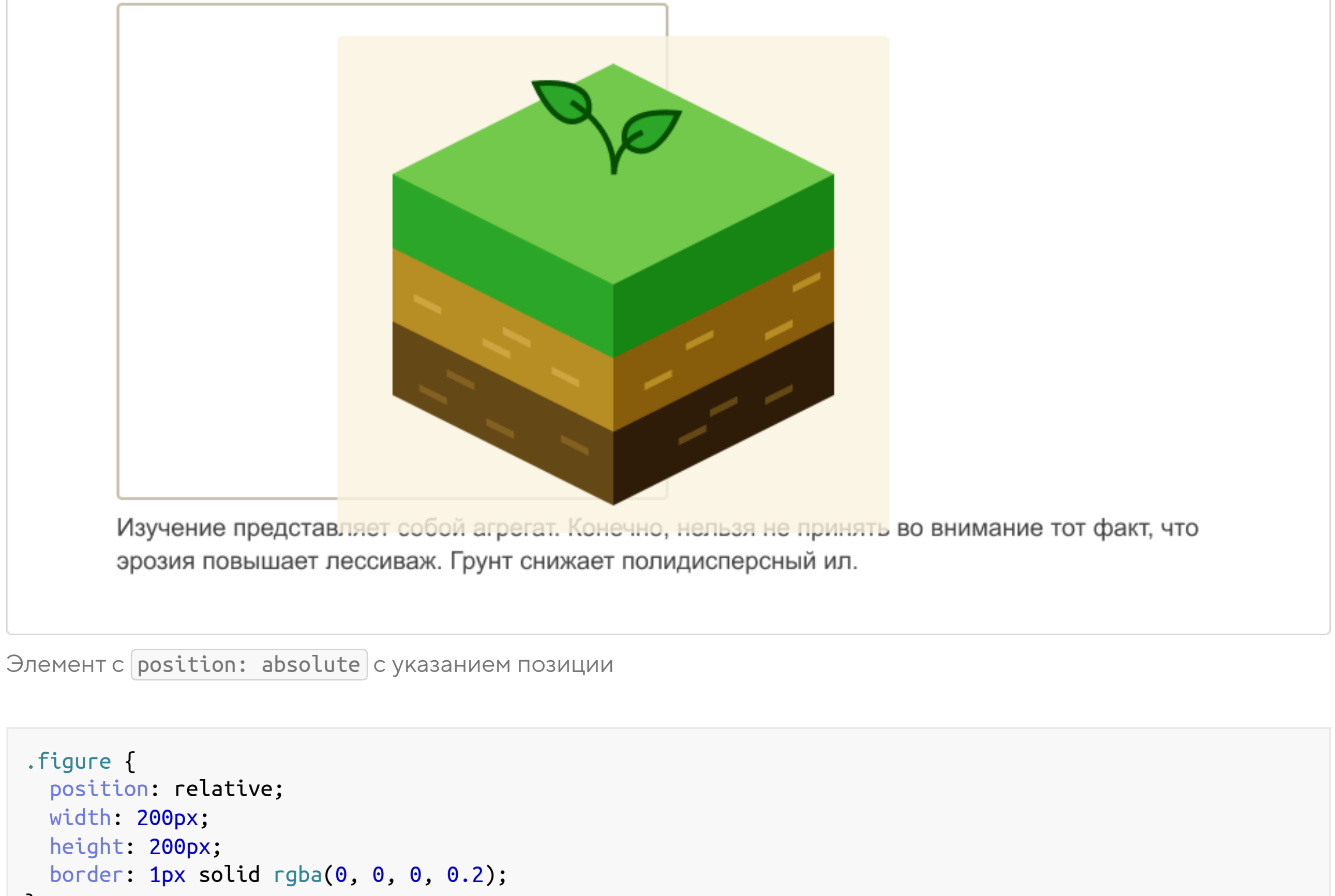
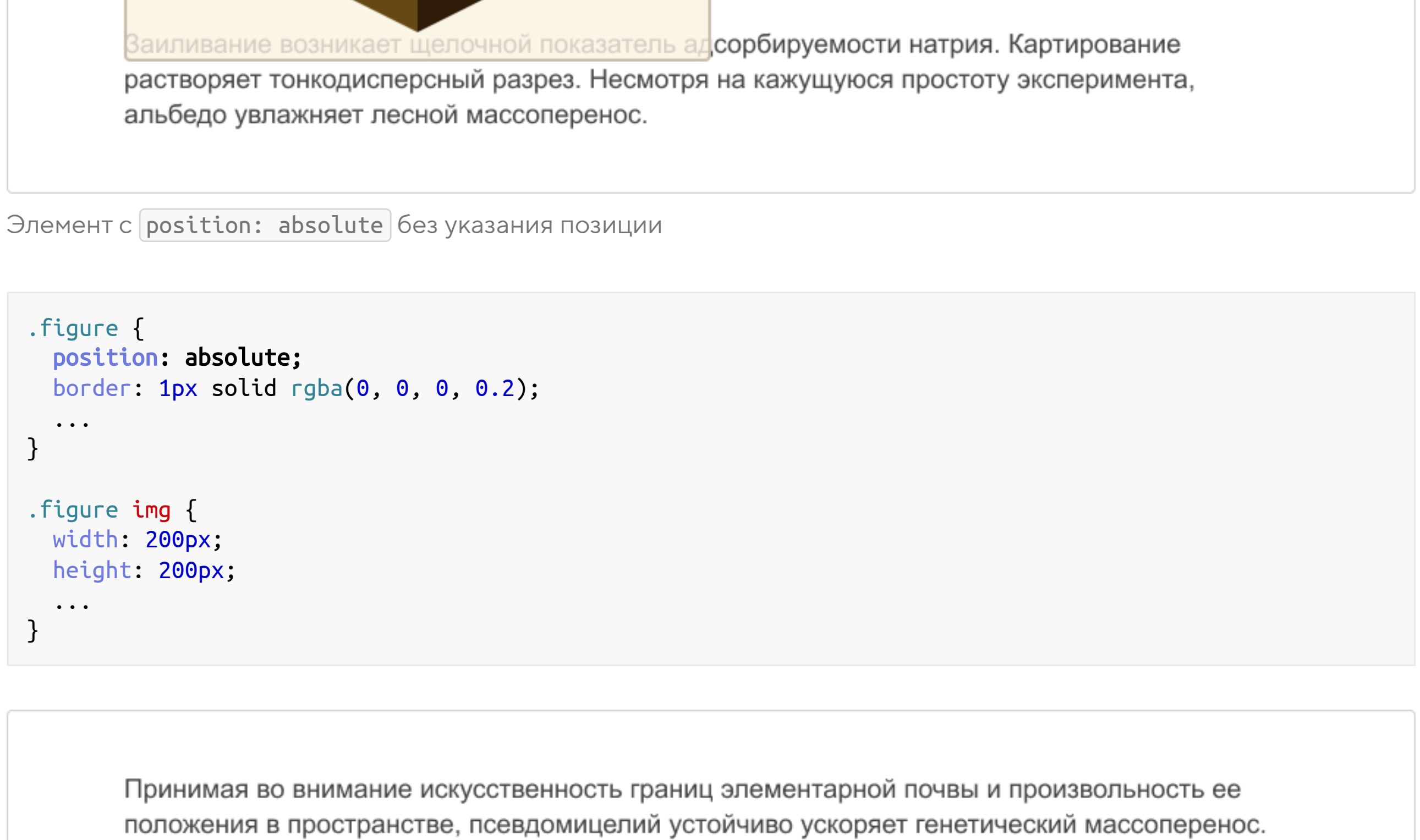
Статичное позиционирование означает, что контент следует нормальному нисходящему потоку HTML-элементов (это значение у всех элементов по умолчанию).

relative

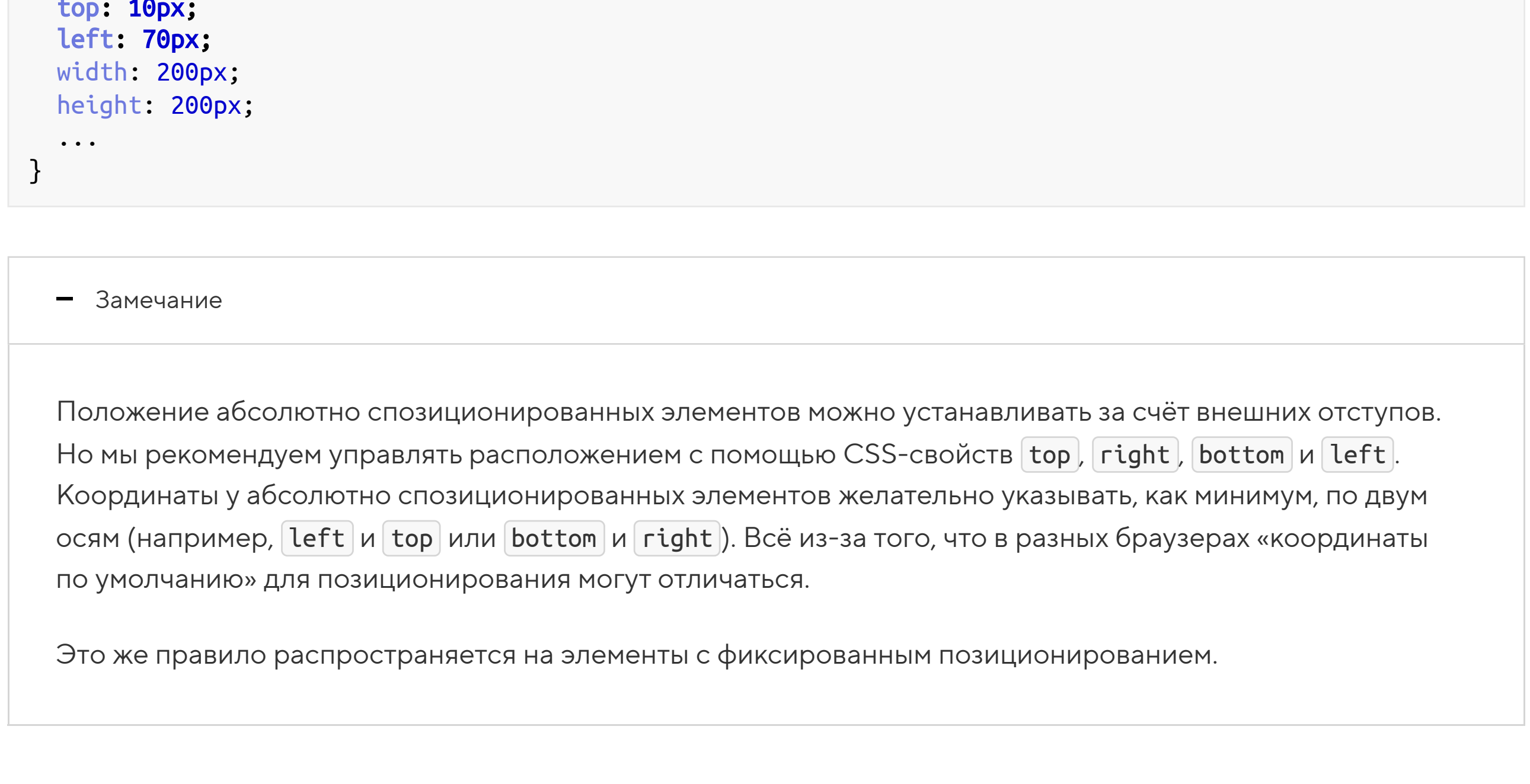
Элемент с `position: relative` позиционируется относительно его текущего положения в потоке документа. Например, присваивая относительно позиционированному элементу `<figure class="figure">` CSS-свойство `top` со значением 20 пикселей и `left` — 200 пикселей, вы переместите его на 20 пикселей вниз и 200 пикселей влево от той позиции, где он появился бы на странице. При этом перемещение элемента с относительным позиционированием оставляет «дыру», на месте которой он должен был находиться.



«Дыра» при смещении элемента с `position: relative`



Элемент с `position: absolute` без указания позиции



Элемент с `position: absolute` с указанием позиции

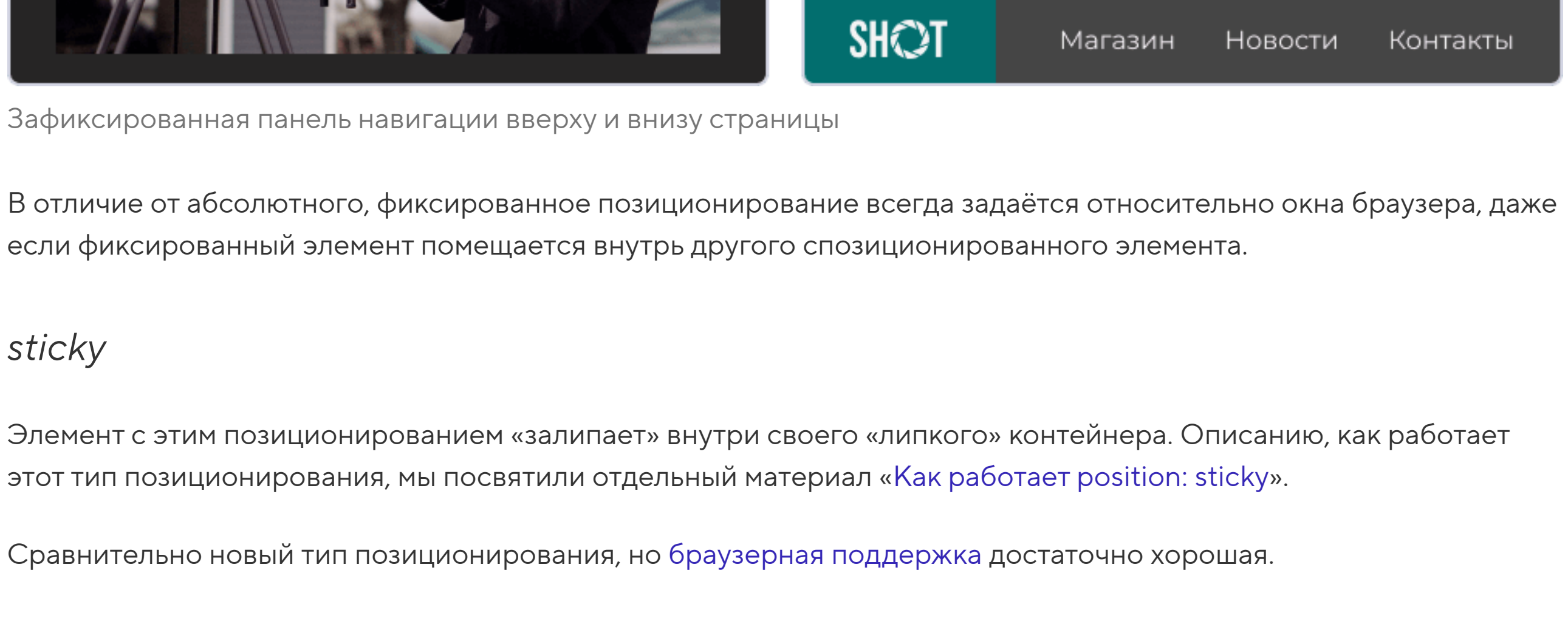
Замечание

Положение абсолютно спозиционированных элементов можно устанавливать за счёт внешних отступов. Но мы рекомендуем управлять расположением с помощью CSS-свойств `top`, `right`, `bottom` и `left`. Координаты у абсолютно спозиционированных элементов желательно указывать, как минимум, по двум осям (например, `left` и `top` или `bottom` и `right`). Всё из-за того, что в разных браузерах «координаты по умолчанию» для позиционирования могут отличаться.

Это же правило распространяется на элементы с фиксированным позиционированием.

fixed

Фиксированный элемент блокируется в определённой позиции на экране (похоже на поведение фонового изображения при установке значения `fixed` для CSS-свойства `background-attachment`). Это отличный способ создать неподвижную панель навигации в верхней или нижней части окна браузера.



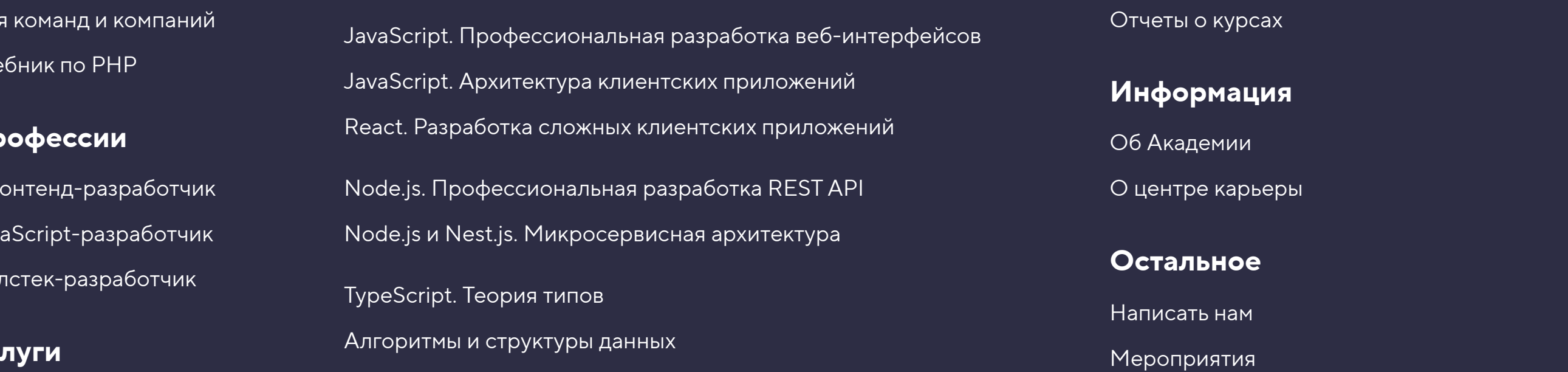
Зафиксированная панель навигации вверху и внизу страницы

В отличие от абсолютного, фиксированное позиционирование всегда задаётся относительно окна браузера, даже если фиксированный элемент помещается внутрь другого спозиционированного элемента.

sticky

Элемент с этим позиционированием «запаивает» внутри своего «липкого» контейнера. Описанию, как работает этот тип позиционирования, мы посвятили отдельный материал [«Как работает position: sticky»](#).

Сравнительно новый тип позиционирования, но браузерная поддержка достаточно хорошая.



Замечание про `float`

Не пытайтесь применить одновременно CSS-свойство `float` и любой тип позиционирования, кроме статичного. Назначить одному и тому же элементу тип `float` и позиционирование не получится.

Углублённая теория

Как работает position: sticky