

ПОЗИЦИОНИРОВАНИЕ ЭЛЕМЕНТОВ

Статическое позиционирование по умолчанию. Абсолютное и относительное позиционирование. Z-index. Фиксированное положение элемента. «Липкое» позиционирование.

Итак, элементы на странице выстраиваются определенным образом, даже если мы не задавали какие-то особые настройки с помощью стилей. Отвечало за поведение элементов свойство **display** и разные варианты его значений. Такое позиционирование «по умолчанию» называют **статическим**.

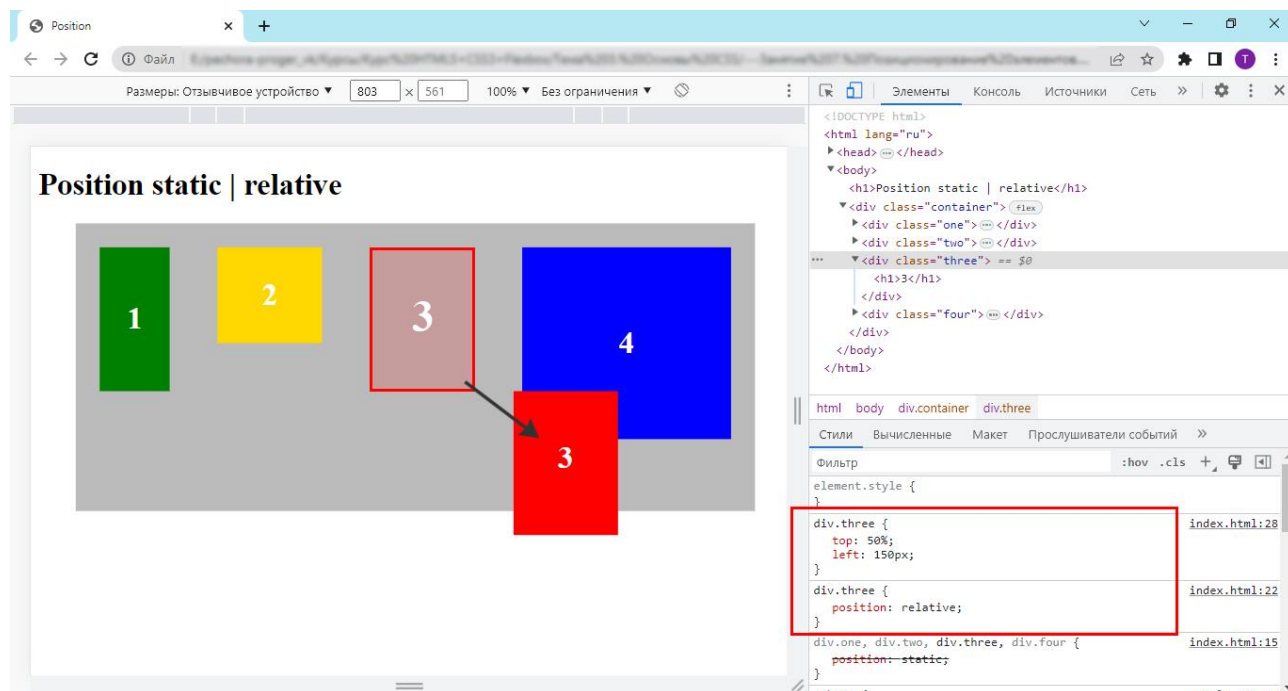
Но, чтобы изменить положение элемента на странице, существует еще одно замечательное свойство – **position**. Изменяя и настраивая его, сможем манипулировать положением элемента и даже сделаем его несколько динамичным.

При этом вашей первоочередной задачей при верстке страницы станет выстроить её структуру так, чтобы она была понятной, четкой и адекватной даже без стилей.

Разнообразные свойства позиционирования должны облегчить нам работу, улучшить макет, который и со стандартными настройками «очень даже ничего». Это гарантирует, что ваш контент будет удобочитаемым, если пользователь использует браузер или устройство с ограниченными возможностями, например, программу чтения с экрана, которая считывает содержимое страницы, не реагируя на стили.

Относительное позиционирование - position: relative;

Относительное позиционирование очень похоже на статическое, за исключением того, что после того, как позиционированный элемент занял свое место в обычном потоке, вы можете изменить его конечное положение, в том числе сделать так, чтобы он перекрывал другие элементы на странице.



Итак, элемент позиционируется в соответствии с нормальным потоком документа, а затем смещается относительно самого себя на основе значений **top**, **right**, **bottom** и **left**. Такое смещение не влияет на положение всех остальных элементов.

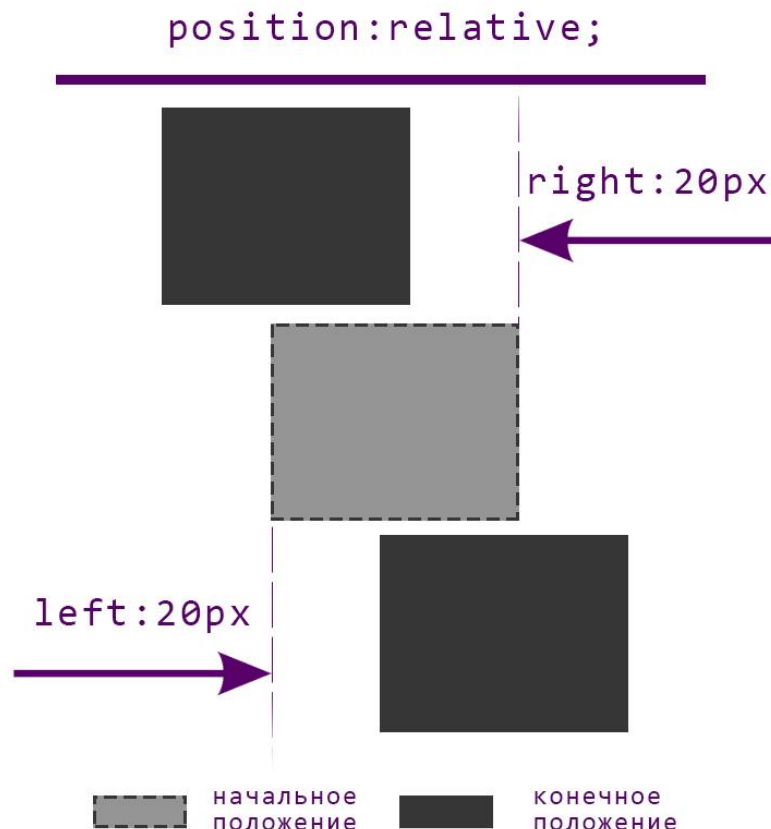
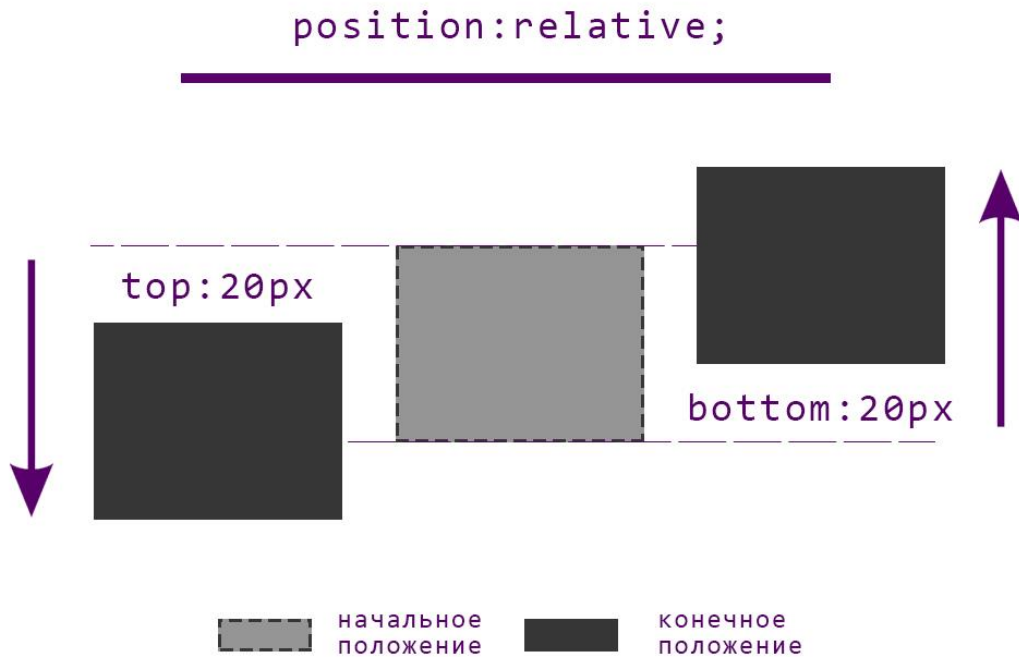
Это значит, что, несмотря на смещение, пространство, отведенное для элемента в макете страницы, остается таким же, как если бы свойство **position** оставалось **static**. А если значения **top**, **right**, **bottom** и **left** будут равны 0 или не будут заданы вовсе, то внешне расположение элемента не будет отличаться от **static**.

top, right, bottom и left

Эти свойства используются как дополнение для **position**, чтобы точно указать, куда переместить позиционируемый элемент.

Значения этих свойств могут принимать любые разумные единицы измерения: пиксели, проценты и т. д.

Одна важная особенность. Задавая значение, мы назначаем не отступ с указанной стороны, а **направление смещения** на какое-то значение.

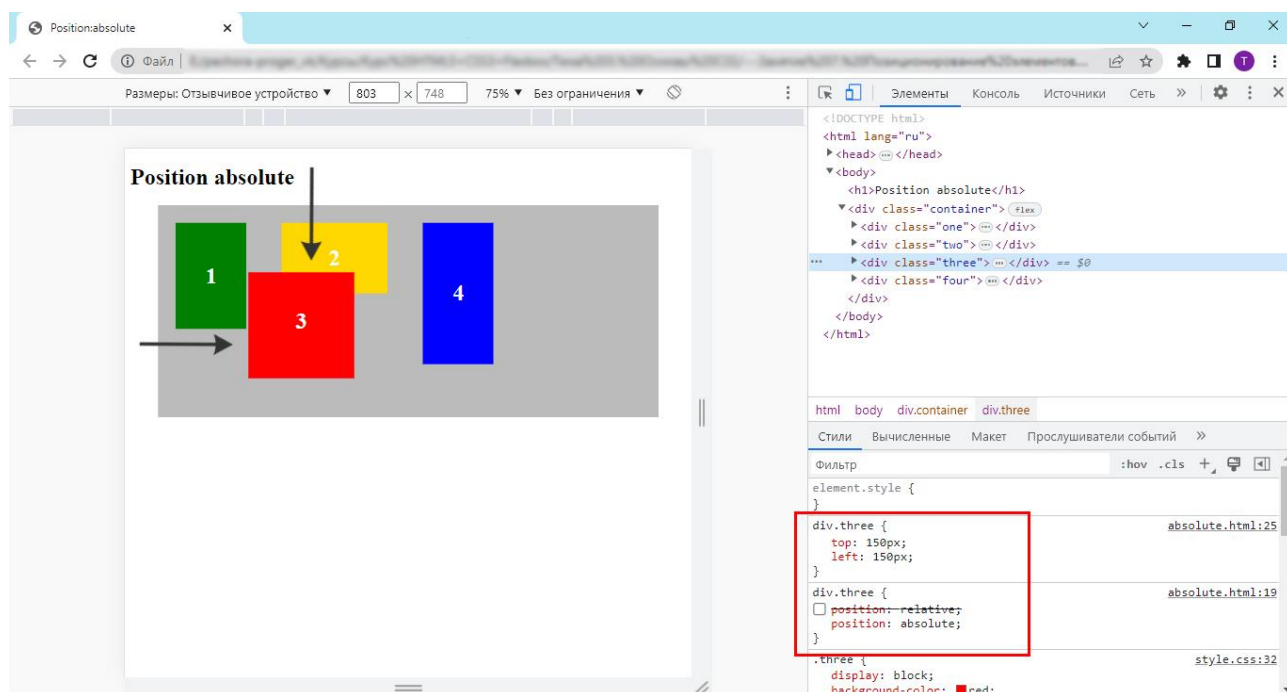


Абсолютное позиционирование - `position: absolute`;

Абсолютное позиционирование работает по-другому.

Элемент удаляется из обычного потока документа, и для него **не создается место** в макете страницы.

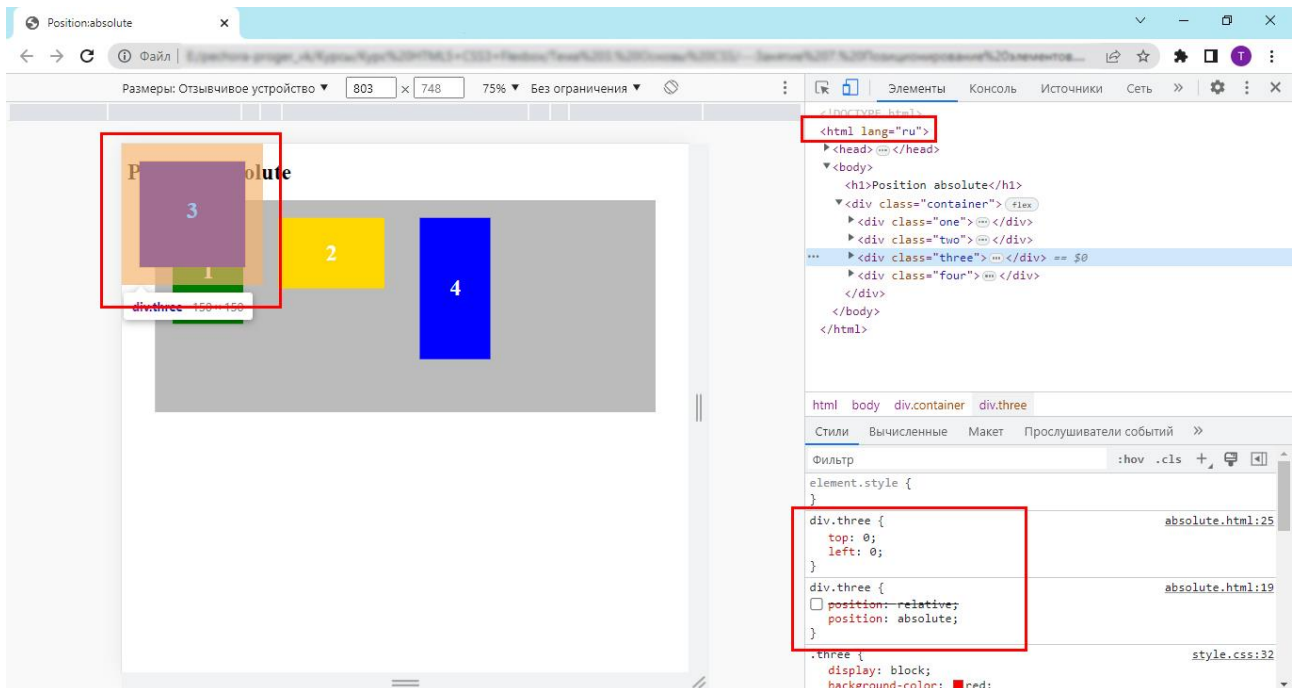
Вместо этого он находится на своем собственном слое, отдельном от всего остального содержимого страницы.



Это значит, что мы можем создавать изолированные элементы пользовательского интерфейса, которые не мешают расположению других элементов на странице. Например, всплывающие информационные окна, выпадающие меню, прокручиваемые панели и многое другое.

Для такого позиционирования важно понятие **контекст позиционирования**, то есть, относительно какого элемента позиционируется абсолютно позиционированный элемент. Это делается путем установки позиционирования отличного от **static** для одного из **предков** элемента: для одного из элементов, внутри которого он вложен (вы не можете позиционировать его относительно элемента, внутри которого он не вложен).

Если позиционированный предок не существует, он позиционируется относительно **начального содержащего блока**, который является содержащим блоком **корневого элемента** документа.



Но его конечное положение определяется также значениями `top`, `right`, `bottom` и `left`.

Важно! Поля абсолютно позиционированных блоков не схлопываются с другими полями.

Z-index

Если абсолютно позиционирован только один блок, все достаточно просто, он имеет преимущество над непозиционированными элементами и располагается над ними (как будто ближе к нам). А что делать, если таких элементов несколько?

Игры с абсолютным позиционированием могут привести к тому, что наши элементы будут располагаться на странице в хаотичном порядке и рано или поздно начнут друг другу мешать.

Поэтому необходимо использовать свойство, которое называется **z-index**.

В теме Flexbox-ы мы уже говорили о расположении элементов по главной и побочной осям – такой аналог знакомым из математики осям X и Y.

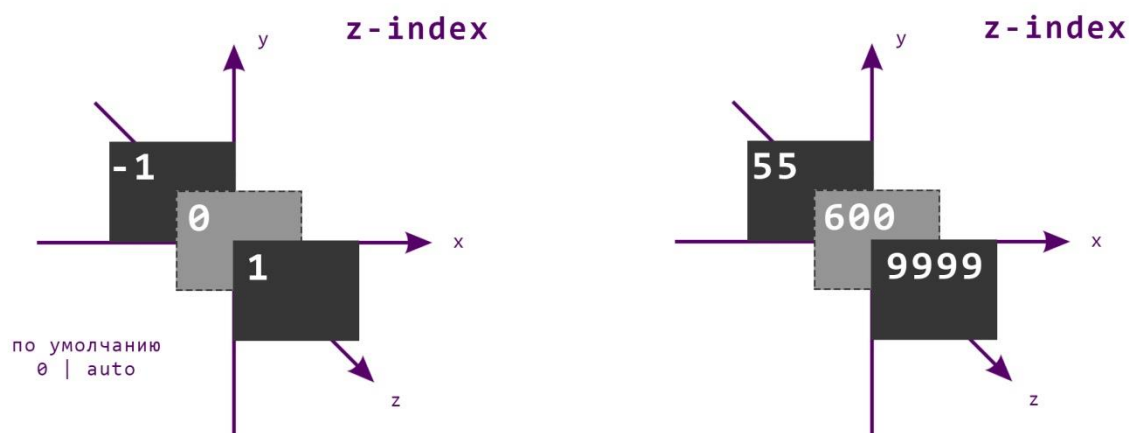
Так вот, z-index – это отсылка к оси Z.

Веб-страницы также имеют свою собственную ось Z: воображаемую линию, которая проходит от поверхности экрана к вашему лицу (или к тому, что вы хотите видеть перед экраном). Значения z-index влияют на расположение позиционированных элементов на этой оси; положительные значения перемещают их выше по стеку, отрицательные значения перемещают их вниз по стеку.

По умолчанию все позиционированные элементы имеют **z-index** значение **auto**, которое фактически равно 0.

Обратите внимание, что для z-index принимаются только безразмерные значения.

Невозможно указать, чтобы один элемент был на 17 пикселей ниже по оси Z. Элементы с более высоким значением z-index будут ближе к вам, чем элементы с более низким. Использование значений 2 или 3 даст тот же эффект, что и значения 300 или 40000.



Активно используя при распределении элементов на странице z-index со значениями подряд (1,2,3 и т.д.) будьте осторожнее. При добавлении в такую «иерархию» новых элементов ваша система может рухнуть и вам придется вручную переписывать значения свойства для всех ранее настроенных элементов.

Кстати, когда хотят чтобы какой-то элемент был абсолютно точно выше остальных, ему задают z-index с большим значением, например, 99999. А если z-index элементов совпадает, то их позиция будет зависеть от порядка в HTML коде.

Фиксированное позиционирование - position: fixed;

Еще один вариант расположения. Очень похож на абсолютное.

Элемент также удаляется из обычного потока документа, и для него не создается место в макете страницы. Позиционируется относительно начального содержащего блока, установленного областью просмотра, за исключением некоторых нестандартных случаев (в рамках данного курса рассматривать их не будем). Его окончательная позиция определяется значениями top, right, bottom и left.

Самое главное отличие такого позиционирования от абсолютного в том, что оно **фиксирует** элемент в определенном месте относительно видимой части окна просмотра. Исключением является случай, когда один из предков элемента является фиксированным блоком.

В печатных документах элемент размещается в одном и том же месте на каждой странице.

Это позволит создавать элементы пользовательского интерфейса, которые фиксируются на своем месте, например меню навигации, которые всегда видимо независимо от того, сколько прокручивается страница или кнопка «наверх».

Липкое позиционирование - **position: sticky;**

Такое позиционирование представляет собой гибрид относительного и фиксированного. Элемент ведет себя как относительно позиционированный, до тех пор, пока при прокрутке не пройдет определенный порог. После этого он станет фиксированным.

Такое липкое позиционирование можно использовать, например, чтобы заставить панель навигации прокручиваться вместе со страницей до определенной точки, а затем прикрепляться к верхней части страницы.

Интересным и распространенным применением **position: sticky** является создание индексной страницы с прокруткой, где разные заголовки прикрепляются к верхней части страницы по мере их прокрутки.

Элемент позиционируется в соответствии с нормальным потоком документа, а затем смещается относительно своего **ближайшего предка** и содержащего блока, включая элементы, связанные с таблицей, на основе значений **top**, **right**, **bottom** и **left**. Смещение не влияет на положение любых других элемен

sample

ЗАДАНИЕ 13 ЗАКРЕПЛЕННОЕ МЕНЮ.

макет страницы

Настро стили так, чтобы меню страницы оставалось

Код во избежание путаницы допишем в самом документе в блоке style.

1. Установим блоку **nav** липкое позиционирование и отступ сверху – 0px.

```
<style>
  nav {
    position: sticky;
    top: 0px;
  }
</style>
```

2. Добавим на страницу ссылку, которая позволит переместиться наверх страницы. Конечно, лучше реализовывать такие фишки с помощью скриптов (смотрятся эффектнее внешне), но механика работает и на чистом css/html.

Код ссылки

```
<a href="#" title="Вернуться к началу" class="top_btn">Наверх</a>
```

Код стиля

```
.top_btn {
  width: 100px;
  text-align: center;
  position: fixed;
  bottom: 100px;
  right: 25px;
}
```

Поэкспериментируйте с настройками, оцените эффект.