**Position** — одно из свойств, позволяющих расположить элементы в порядке, отличном от классического потока документа.

Дополнительные свойства, которые определяют расположение элемента:

* + top,
  + right,
  + bottom,
  + left.

Однако дополнительные свойства не будут работать, если не установлено значение position. К значениям относятся:

* + static,
  + relative,
  + fixed,
  + absolute,
  + sticky.

|  |  |
| --- | --- |
| **Position: static;** | Элемент находится в обычном состоянии, расположен на своём месте в документе. Свойства top, right, bottom, left и z-index не применяются к такому элементу. Это значение по умолчанию. |

Элемент считается позиционированным, только если значение position отлично от static.

|  |  |
| --- | --- |
| **Position: relative;** | Элемент с этим значением позиционируется относительно своего нормального положения при помощи свойств top, right, bottom и left. |
| **Position: fixed;** | Элемент позиционируется относительно *viewport*, то есть его положение всегда остаётся неизменным при использовании прокрутки. |

Такое значение часто используется, чтобы зафиксировать верхнюю часть в начале страницы и оставить её доступной постоянно. Также в последнее время часто используют для всплывающего чата в нижней части экрана.

Элемент с position: fixed; не оставляет никакого зарезервированного пространства в том месте, где он определён в документе изначально.

|  |  |  |
| --- | --- | --- |
| <div class="red"></div>  <div class="yellow"></div>  <div class="blue"></div> | div {  **display**: **inline**-**block**;  **width**: **100px**;  **height**: **100px**;  }  **.red** {  **background-color**: red;  **position**: **relative**;  **top**: **150px**;  **left**: **20px**;  }  **.yellow** {  **background-color**: yellow;  }  **.blue** {  **background-color**: blue;  }  ИЛИ  div {  **display**: **inline**-**block**;  **width**: **100px**;  **height**: **100px**;  }  **.red** {  **background-color**: red;  **position**: **fixed**;  **top**: **50px**;  **left**: **50px**;  }  **.yellow** {  **background-color**: yellow;  }  **.blue** {  **background-color**: blue;  } |  |

|  |  |
| --- | --- |
| **Position: absolute;** | Элемент позиционируется относительно ближайшего предка (вместо позиционированного относительно окна просмотра, как fixed). Если элемент с абсолютным позиционированием не имеет позиционированных предков, он использует тело документа и перемещается вместе с прокруткой страницы. |

Родителю, относительно которого хочется спозиционировать элемент, часто добавляется свойство position: relative; без указания дополнительных свойств top, right, bottom и left. Таким образом, родительский элемент никак визуально не меняется, а необходимый элемент смещается относительно него.

|  |  |
| --- | --- |
| **Position: sticky;** | Элемент позиционируется на основе позиции прокрутки пользователя. Можно перевести как «липкое» позиционирование. |

Position: sticky; переключает элемент между *relative* и *fixed*-позиционированием. *Relative*используется, пока положение по умолчанию остаётся во*viewport.*Когда скролл проходит дальше, включается *fixed*-позиционирование.

Обратите внимание, что это свойство position **НЕ** поддерживается ни в одной версии браузера *Internet Explorer*. Подробнее [здесь](https://caniuse.com/#search=sticky).

**Слои и z-index**

Когда элементы позиционированы, они могут перекрывать другие элементы, и необходимы какие-то правила, регулирующие это перекрытие.

**Свойство z-index** определяет порядок (какой элемент должен быть расположен сверху или снизу других). Это свойство может иметь как положительные, так и отрицательные значения. Элемент с большим значением перекрывает элемент с меньшим значением.

Если два позиционированных элемента перекрываются без указания *z*-индекса, элемент, размещённый последним в коде *HTML*, будет показан сверху.

**Что же регулирует, с точки зрения браузера, z-index?**

У каждого элемента в *CSS* есть определённое место на экране, которое характеризуется координатами *X* и *Y*, но если бы существовали только эти две координаты, то мы бы не смогли накладывать один элемент поверх другого. Поэтому существует ещё одна координата — пространственная координата *Z*. Поэтому z-index так и называется: по сути, это номер слоя, координата по оси *Z*.

Позиционирование и указание свойства z-index создаёт **контекст наложения** (*stacking context*).