

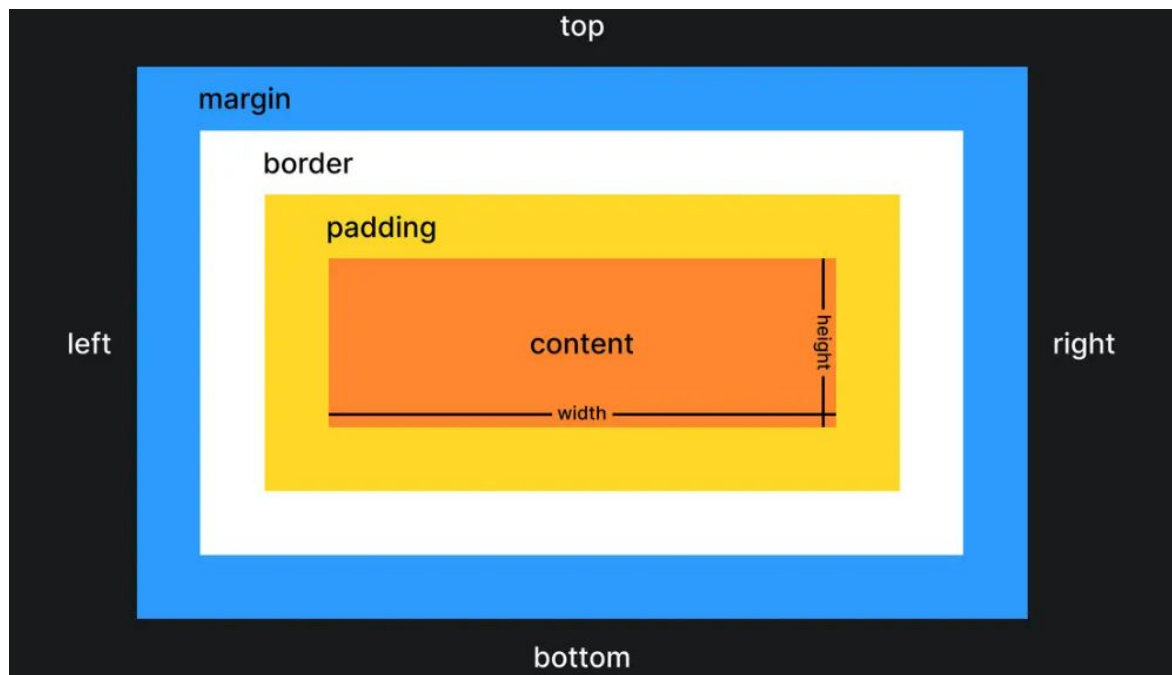
The image features decorative geometric patterns in the top-left and bottom-right corners. These patterns consist of overlapping dark blue and light blue shapes, some with gold-colored outlines and others with a dotted texture. The main content area is a plain white background.

HTML и CSS

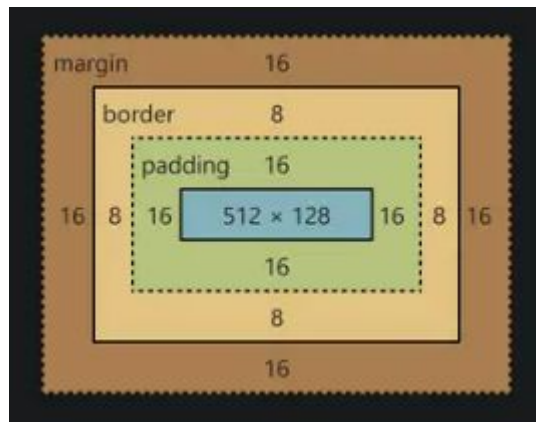
Блочная модель

Блочная модель

Блочная модель, она же box model — это алгоритм расчёта размеров каждого отдельного элемента на странице, которым браузеры пользуются при отрисовке.

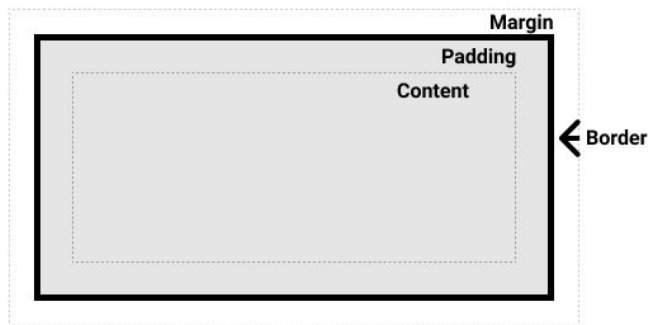


Ровно такую же схему, но в других цветах можно увидеть в инструментах разработчика любого из браузеров. Например, так выглядит блочная модель элемента в Chrome:



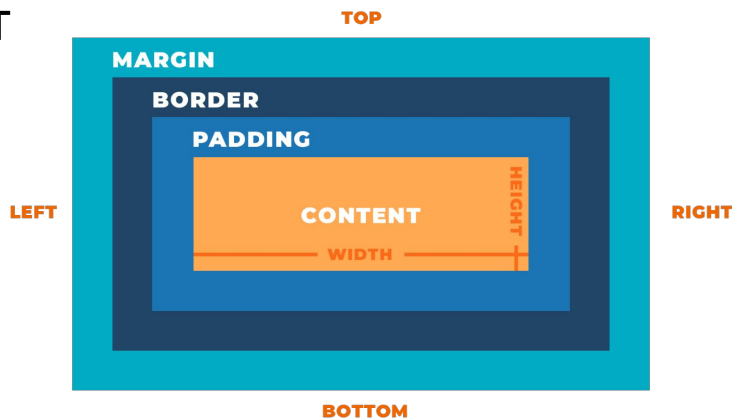
Блочная модель состоит из нескольких CSS-свойств, влияющих на размеры элемента:

- **width** — ширина элемента;
- **height** — высота элемента;
- **padding** — внутренние отступы от контента до краёв элемента;
- **border** — рамка, идущая по краю элемента;
- **margin** — внешние отступы вокруг элемента

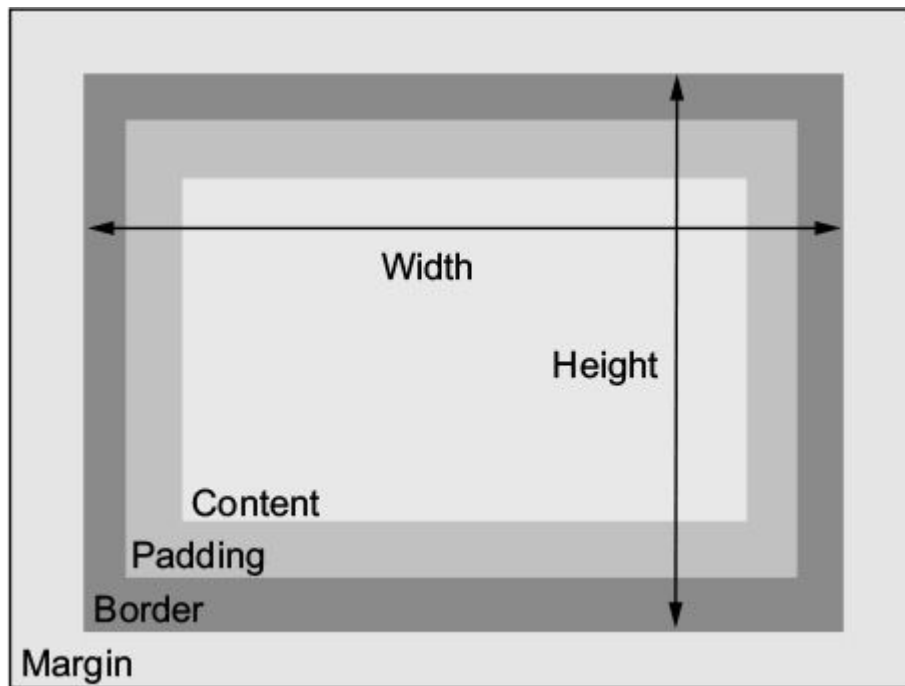


Свойство **box-sizing** в CSS определяет, как браузер рассчитывает общую ширину и высоту элемента, учитывая его содержимое, отступы (padding), границы (border) и внешние отступы (margin).

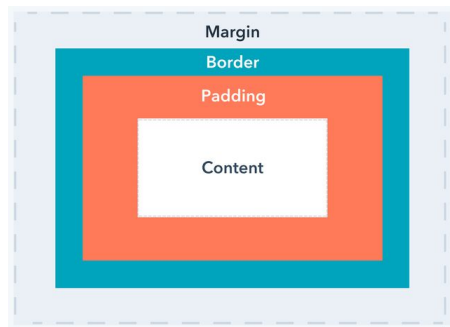
По умолчанию значение равно: **box-content** - ширина и высота, которые будут заданы для элемент, будут включать только контент



Когда свойство **box-sizing** установлено в значение **border-box**, размеры элемента включают в себя содержимое, отступы и границы, но не включают внешние отступы. Это позволяет более прозрачно управлять размерами элементов



Внешние и внутренние отступы

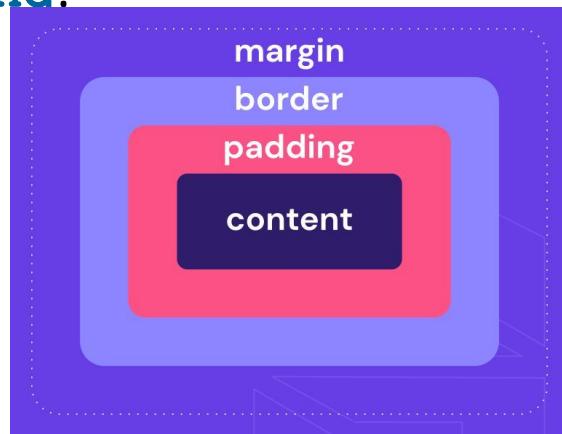


Внутренний отступ

Внутренний отступ - пустое пространство между содержимым элемента и его рамкой (если она установлена). Для добавления и управления шириной внутренних отступов со всех четырех сторон элемента используется свойство `padding`.

`padding: 10px 15px 22px 18px;`

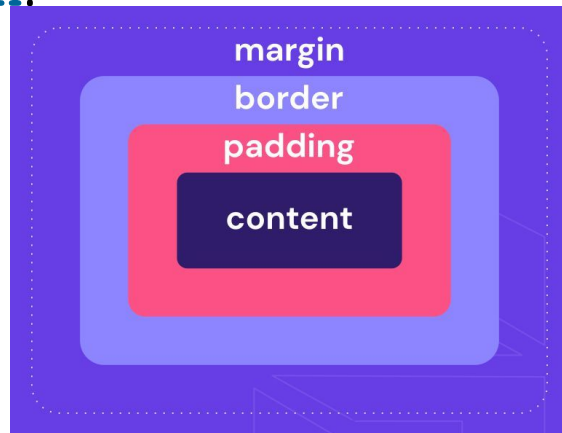
top right bottom left



Внешний отступ

Внешний отступ - пустое пространство, отделяющее элемент от других элементов или краев окна браузера. Для добавления и управления шириной внешних отступов со всех четырех сторон элемента используется свойство **margin**.

top **right** **bottom** **left**
`margin: 10px 15px 22px 18px;`



Внутренние и внешние отступы

Внутренние отступы могут регулироваться отдельными свойствами:

- padding-top
- padding-right
- padding-bottom
- padding-left

Внешние отступы могут регулироваться отдельными свойствами:

- margin-top
- margin-right
- margin-bottom
- margin-left.



Общие свойства:

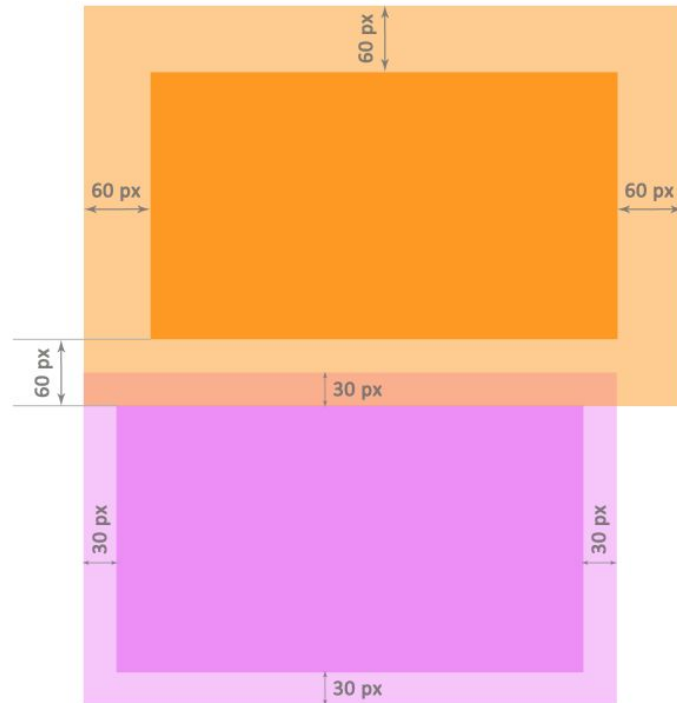
- padding
- margin

Схлопывание margin (Margin Collapse)

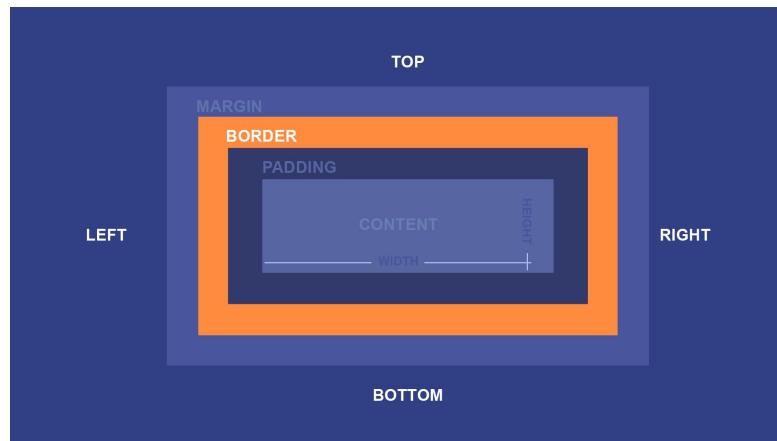
Представьте ситуацию: два блочных элемента находятся друг над другом и им заданы поля margin. Для верхнего блока установлено значение **margin**: 60px, а для нижнего – **margin**: 30px. Логично было бы предположить, что два граничащих поля двух элементов просто соприкоснутся и в итоге промежуток между блоками будет равен 90 пикселям.

Однако дела обстоят по-другому. На самом деле в такой ситуации проявляется эффект, который называют схлопыванием, когда из двух примыкающих полей элементов выбирается наибольший по размеру. В нашем примере итоговый промежуток между элементами будет равен 60 пикселям.

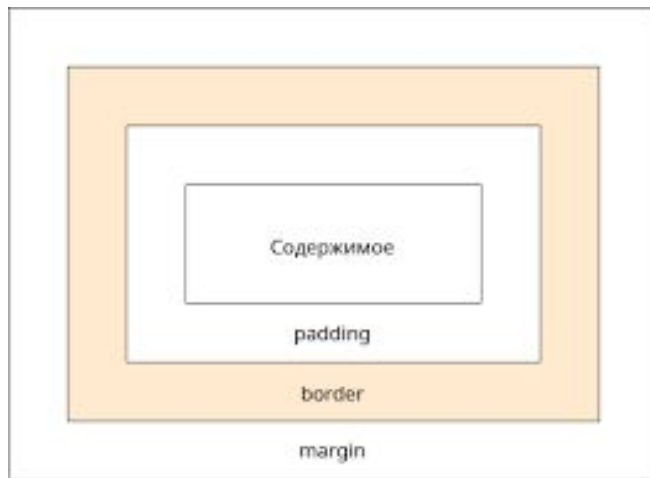
Схлопывание **margin** работает только для верхних и нижних полей



Параметры рамки. Цвет и стиль рамки



CSS-рамка элемента представляет собой одну или несколько линий, окружающих содержимое элемента и его поля.



Стиль рамки - border-style

По умолчанию рамки всегда отрисовываются поверх фона элемента, фон распространяется до внешнего края элемента. Стиль рамки определяет ее отображение, без этого свойства рамки не будут видны вообще. Для элемента можно задавать рамку для всех сторон одновременно с помощью свойства **border-style** или для каждой стороны отдельно с помощью уточняющих свойств **border-top-style** и т.д. Данное свойство не наследуется.



Стиль рамки - border-style

border-style

(border-top-style, border-right-style, border-bottom-style, border-left-style)

Значения:

`none`

Значение по умолчанию, означает отсутствие рамки. Также убирает рамку элемента из группы элементов с установленным значением данного свойства.

`hidden`

Эквивалентно `none`.

`dotted`

`dotted`

`dashed`

`dashed`

`solid`

`solid`

`double`

`double`

Стиль рамки - border-style

groove

groove

ridge

ridge

inset

inset

outset

outset

{1, 4}

Одновременное перечисление четырех разных стилей для рамок элемента, только для свойства `border-style` :

```
{border-style: solid dotted none dotted;}
```

initial

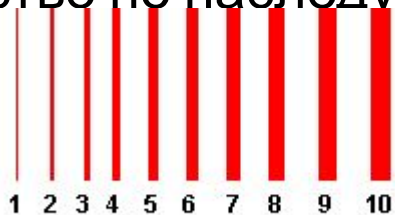
Устанавливает значение свойства в значение по умолчанию.

inherit

Наследует значение свойства от родительского элемента.

Ширина рамки - border-width

Ширина рамки задается с помощью единиц измерения длины или ключевых слов. Если для свойства **border-style** задано значение none, и для рамки элемента установлена какая-то ширина, то в данном случае ширина рамки приравнивается к нулю. Свойство не наследуется.



Ширина рамки - border-width

border-width

(border-top-width, border-right-width, border-bottom-width, border-left-width)

Значения:

`thin / medium`

`/ thick`

Ключевые слова, устанавливают ширину рамки относительно друг друга. Первое значение уже, чем второе, второе — тоньше третьего. Значение по умолчанию — `medium`

`width (px, em)`

```
{border-width: 5px;}
```

`{1, 4}`

Возможность одновременного задания четырех разных ширин для рамок элемента, только для свойства `border-width` :

```
{border-width: 5px 10px 15px 3px;}
```

`initial`

Устанавливает значение свойства в значение по умолчанию.

`inherit`

Наследует значение свойства от родительского элемента.

Цвет рамки - border-color

Свойство задаёт цвет рамок всех сторон одновременно. С помощью уточняющих свойств можно установить свой цвет для рамки каждой стороны элемента. Если для рамки цвет не задан, то он будет таким же, как и цвет текста элемента. Если в элементе нет текста, то цвет рамки будет таким же, как и цвет текста родительского элемента. Свойство не наследуется.



Цвет рамки - border-color

border-color

(border-top-color, border-right-color, border-bottom-color, border-left-color)

Значения:

transparent

Устанавливает прозрачный цвет для рамки. При этом ширина рамки остается. Можно использовать для смены цвета рамки при наведении курсора мыши на элемент, чтобы избежать смещение элемента.

цвет

Цвет рамок задается при помощи значений свойства `color`.

```
{border-color: #cacd58;}
```

{1, 4}

Одновременное перечисление четырех разных цветов для рамок элемента, только для свойства `border-color`:

```
{border-color: #cacd58 #5faf8a #b9cea5 #aab238;}
```

initial

Устанавливает значение свойства в значение по умолчанию.

inherit

Наследует значение свойства от родительского элемента.

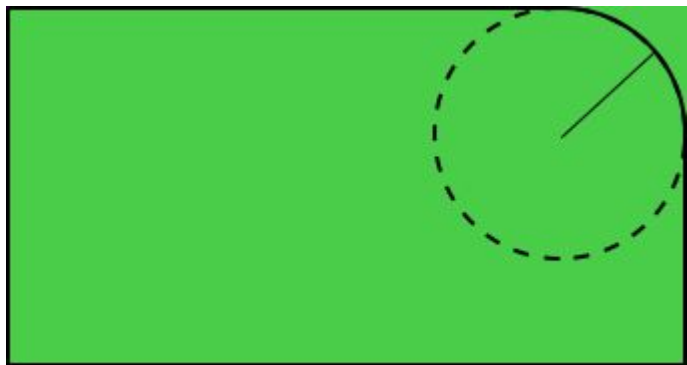
Задание рамки одним свойством

Свойство `border` позволяет объединить в себе следующие свойства: `border-width`, `border-style`, `border-color`, например:

```
div {  
  
    width: 200px;  
  
    height: 300px;  
  
    border: 1px solid grey;  
  
}
```

Примечание: Если какое-то из значений не указано, его место займет значение по умолчанию.

Скругление границ



border-radius

в CSS это свойство, которое используется для задания скругления углов для границы элемента. Оно позволяет создавать элементы с закругленными углами, делая интерфейс более эстетичным и приятным для восприятия.

```
border-radius: <размер>;
```

Значение **<размер>** указывает радиус скругления угла в пикселях (px), процентах (%) или других единицах измерения.

border-radius

- Указание одного значения для всех углов

```
.element {  
    border-radius: 10px;  
}
```

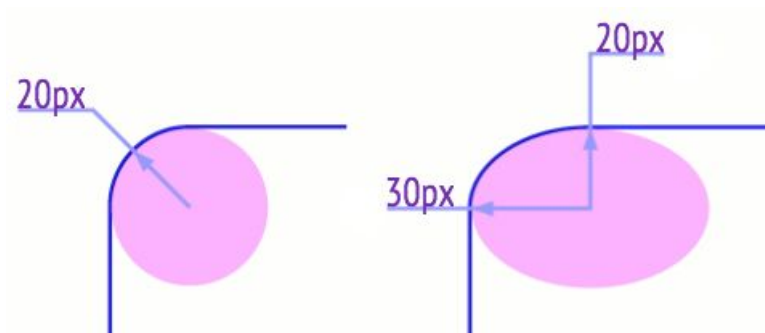
- Указание значений для каждого угла по отдельности (по часовой стрелке, начиная с верхнего левого угла):

```
.element {  
    border-radius: 10px 20px 30px 40px;  
}
```

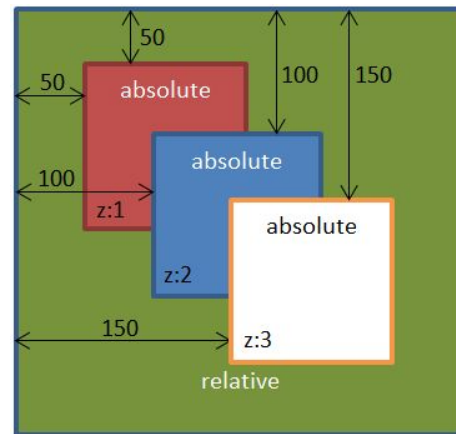

border-radius

- Указание значений для горизонтальных и вертикальных углов:

```
.element {  
  border-radius: 10px 20px 10px 20px / 20px 10px 20px 10px;  
}
```



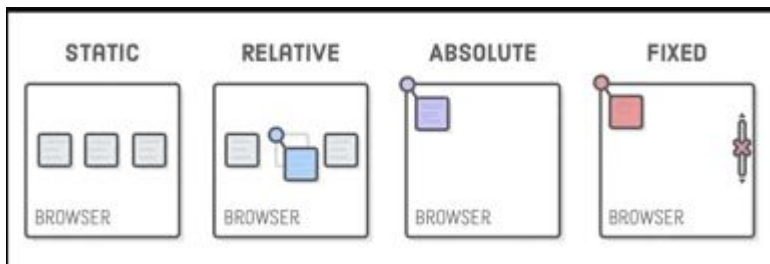
Позиционирование элементов



СВОЙСТВО position

В CSS есть свойства, с помощью которых элементы можно «вырвать» из основного потока документа и задать им другое положение вне базового потока элементов.

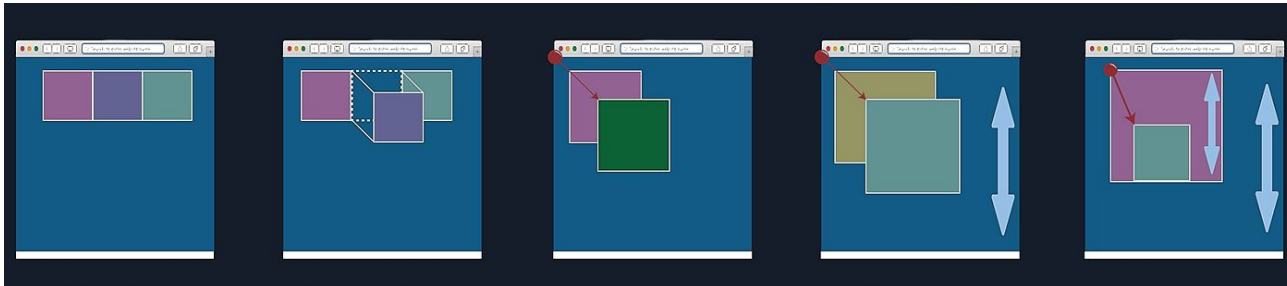
Свойство **position** — это одно из свойств с помощью которого можно изменить базовое поведение элементов в потоке. Другими словами, данное свойство позволяет «выдернуть» любой элемент из потока документа и разместить его в другом месте относительно окна браузера или других элементов на веб-странице.



СВОЙСТВО position

Свойство position имеет 5 значений:

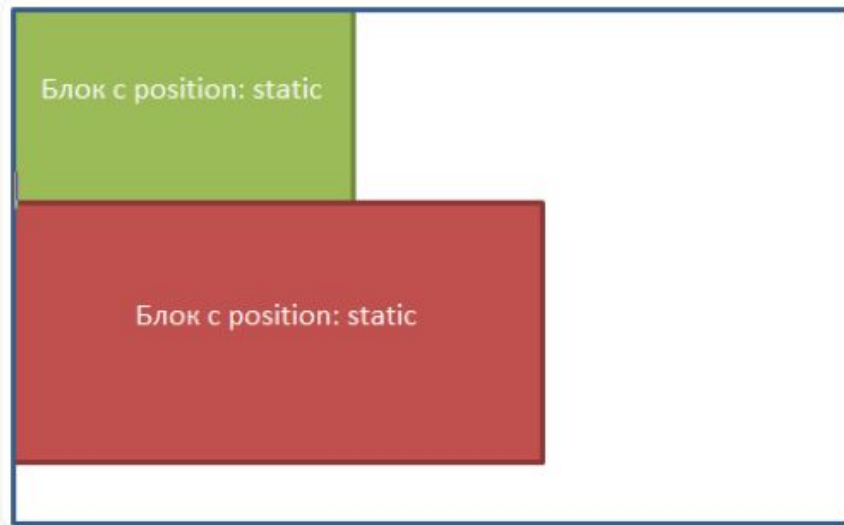
- **static** (статичное позиционирование);
- **relative** (относительное);
- **absolute** (абсолютное);
- **fixed** (фиксированное);
- **sticky** (липкое).



СВОЙСТВО position: static

Свойство **position** со значением **static** элементам назначается по умолчанию. Это значение означает что элемент является не позиционированным, т.е. отображается как обычно (в потоке).

Установка CSS свойств для задания положения элемента **left**, **top**, **right** и **bottom** никакого влияния на него не оказывают, т.к. его местонахождение определяется потоком документа.



СВОЙСТВО position: relative

Установка относительного позиционирования элементу осуществляется посредством задания ему CSS свойства **position: relative**.

Относительно позиционированный элемент ведёт себя как элемент в потоке за исключением того, что его текущее положение можно при помощи определённых CSS свойств сместить. К этим CSS свойствам относятся **left**, **top**, **right** и **bottom**



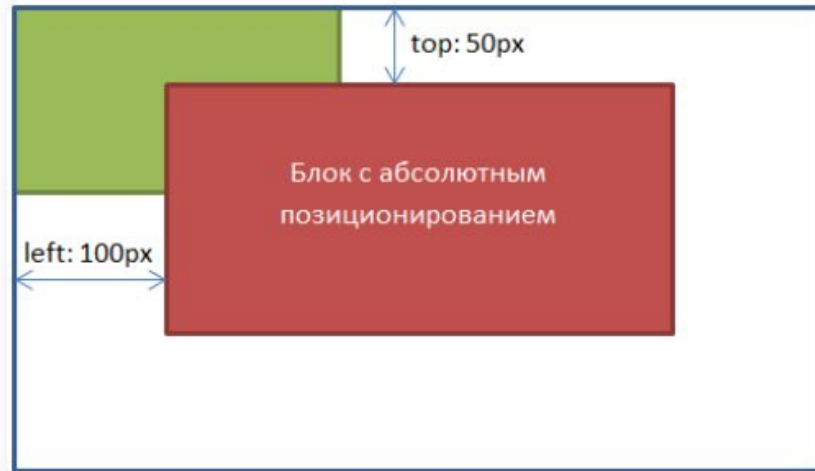
СВОЙСТВО position: absolute

Установка абсолютного позиционирования элементу осуществляется посредством задания ему `position: absolute`.

Этот тип позиционирования позволяет разместить элемент именно там, где вы хотите.

Позиционирование выполняется относительно ближайшего позиционированного предка.

Под позиционированным элементом понимается элемент с `position`, равным любому значению кроме `static`



СВОЙСТВО position: fixed

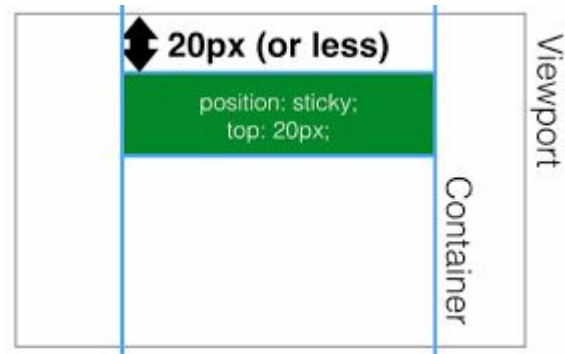
Фиксированное позиционирование похоже на абсолютное, но в отличие от него оно всегда привязывается к краям окна браузера (viewport), и остаётся в таком положении даже при скроллинге страницы.

Фиксированное позиционирование применяется для закрепления на странице навигационных меню, кнопки «вверх», панелей с социальными кнопками и многого другого.

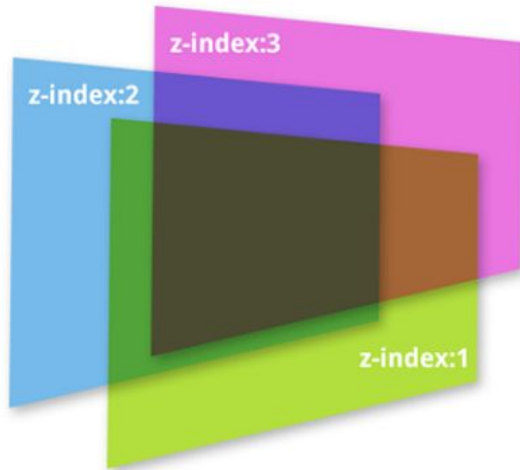


СВОЙСТВО position: sticky

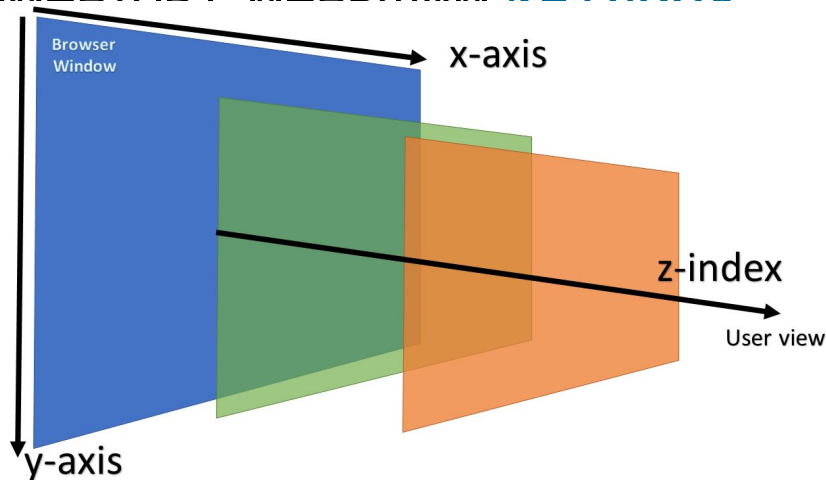
Липкие элементы (**position: sticky;**) очень похожи на фиксированные (**position: fixed;**), поскольку они сохраняют свое положение на экране, даже когда пользователь прокручивается страницу вверх или вниз. Разница в том, что липкий элемент остается ограниченным родительским контейнером, в котором он находится.



z-index



CSS-свойство **z-index** определяет положение позиционированного элемента и его дочерних элементов или флекс-элементов по оси z. Перекрывающие элементы с большим значением **z-index** будут накладываться поверх элементов с меньшим **z-index**.



Синтаксис

`z-index: число | auto | inherit`

В качестве значения используются целые числа (положительные, отрицательные и ноль). Чем больше значение, тем выше находится элемент по сравнению с теми элементами, у которых оно меньше. При равном значении `z-index`, на переднем плане находится тот элемент, который в коде HTML описан ниже.

Кроме числовых значений применяется `auto` — порядок элементов в этом случае строится автоматически, исходя из их положения в коде HTML и принадлежности к родителю, поскольку дочерние элементы имеют тот же номер, что их родительский элемент

! Важно понимать - z-index учитывается только на явно позиционированных элементах

