CSS: Position

НАШИ ПРАВИЛА

Включенная камера

Вопросы по поднятой руке

Не перебиваем друг друга

Все вопросы, не связанные с тематикой курса (орг-вопросы и т. д.), должны быть направлены куратору

Подготовьте свое рабочее окружение для возможной демонстрации экрана (закройте лишние соцсети и прочие приложения)

Повторим;)

Назовите основные способы подключения CSS

Основные свойства, используемые в box-model

Что включено в размеры элемента, если задано свойство box-sizing: border-box

ЦЕЛЬ

Изучить методы позиционирования элементов с помощью свойства position.
Разобрать свойство background

ПЛАН ЗАНЯТИЯ

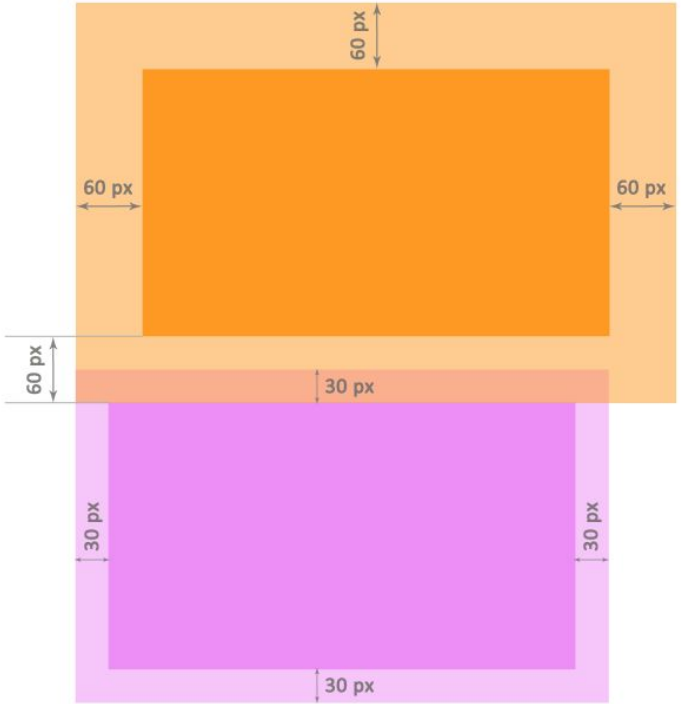
- Значения свойства position
- Свойство background

Схлопывание margin (Margin Collapse)

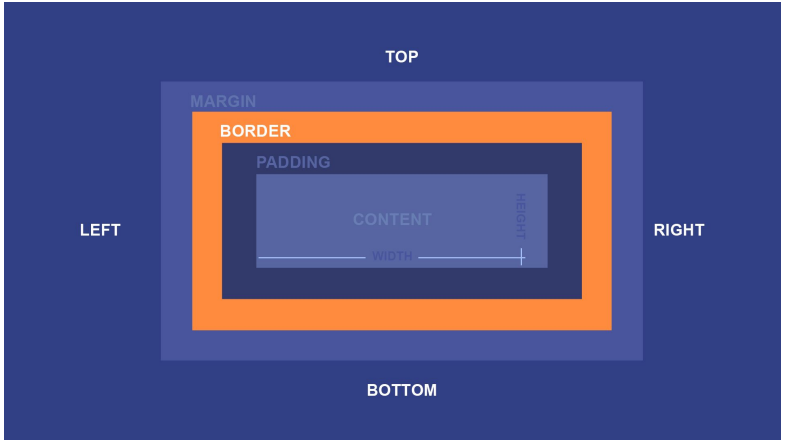
Представьте ситуацию: два блочных элемента находятся друг над другом и им заданы поля margin. Для верхнего блока установлено значение `margin`: 60px, а для нижнего – `margin`: 30px. Логично было бы предположить, что два граничащих поля двух элементов просто соприкоснутся и в итоге промежуток между блоками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

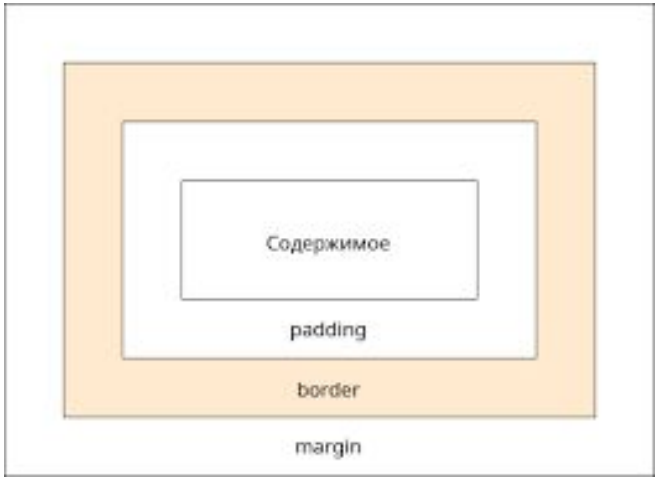
Схлопывание `margin` работает только для верхних и нижних полей



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:

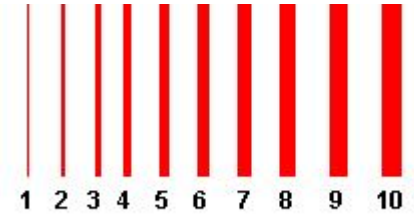
`none`	Значение по умолчанию, означает отсутствие рамки. Также убирает рамку элемента из группы элементов с установленным значением данного свойства.
`hidden`	Эквивалентно `none`.
`dotted`	dotted
`dashed`	dashed
`solid`	solid
`double`	double

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.

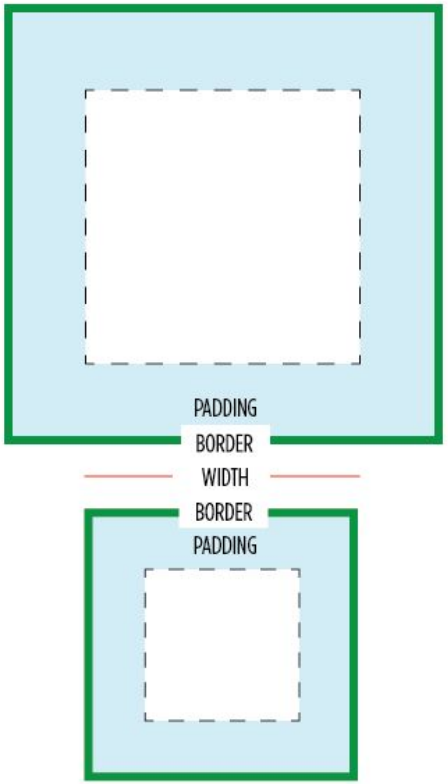
Задание рамки одним свойством

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

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

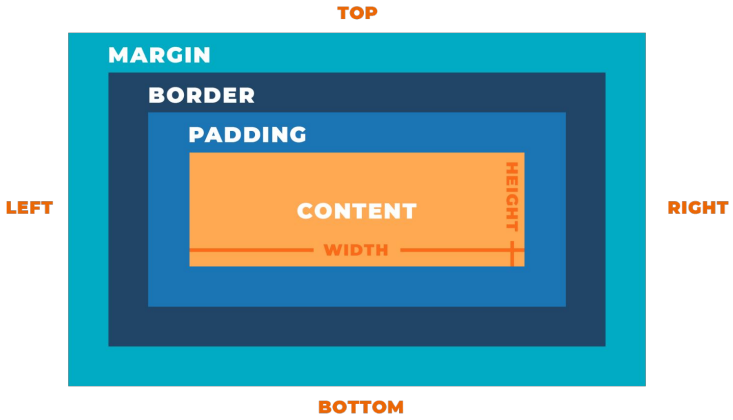
Примечание: Если какое-то из значений не указано, его место займет значение по умолчанию.

box-sizing

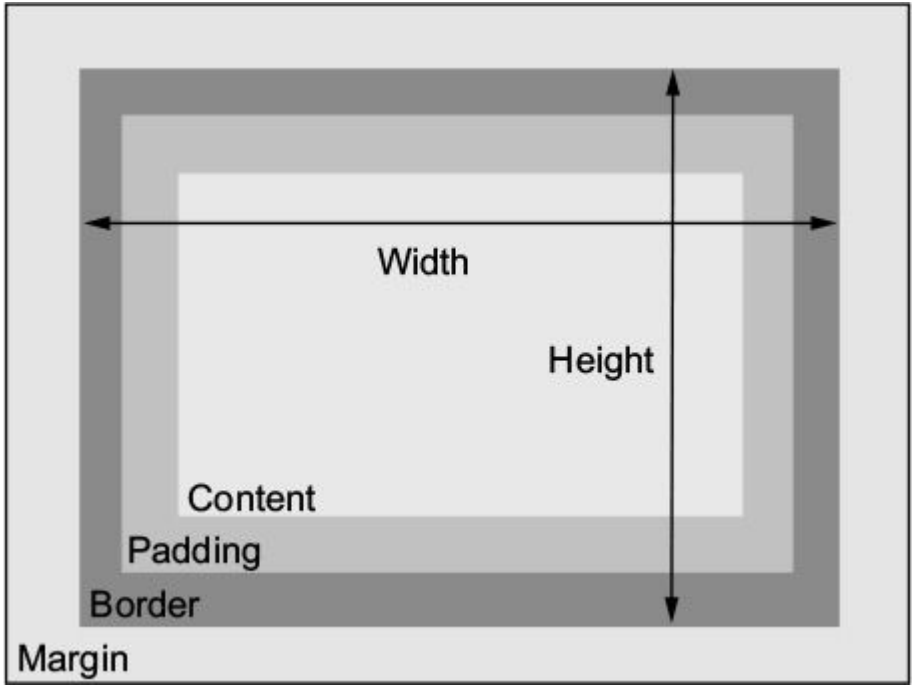


Свойство box-sizing в CSS определяет, как браузер рассчитывает общую ширину и высоту элемента, учитывая его содержимое, отступы (padding), границы (border) и внешние отступы (margin).

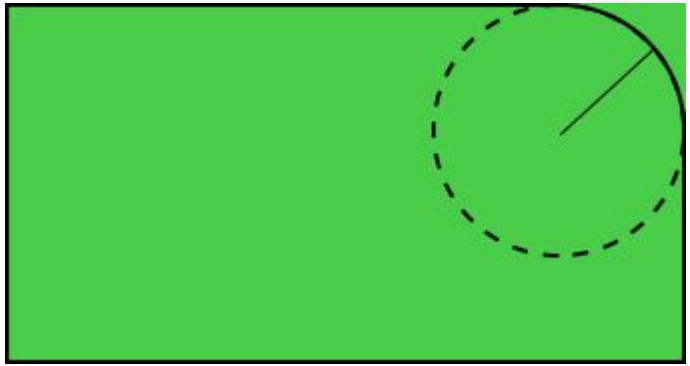
По умолчание значение равно: box-content - ширина и высота, которые будут заданы для элемент, будут включать только контент



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



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;
}
```

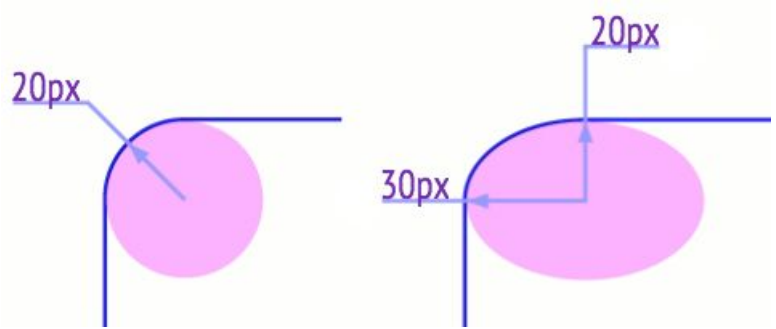
- Указание значений для каждого угла по отдельности (по часовой стрелке, начиная с верхнего левого угла):

```
.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;
}
```



Параметры фона

```
background-size: no-repeat;
background-position: center;
background-size: cover;
width: 400px;
```

Фоновый цвет

Свойство `background-color` определяет цвет фона для любого элемента в CSS. Свойство принимает любой допустимый цвет.

```
.box {
  background-color: #BCBCBC;
}
.box {
  background-color: grey;
}
.box {
  background-color: rgba(188,188,188);
}
```

Фоновое изображение

Свойство **`background-image`** позволяет отображать изображение в качестве фона элемента. В приведённом ниже примере у нас есть два блока — в одном фоновое изображение больше, чем размеры блока, а в другом - маленькое изображение звезды.

Этот пример демонстрирует две особенности фоновых изображений. По умолчанию большое изображение не масштабируется до размера блока, поэтому мы видим только его небольшой угол, в то время как маленькое изображение повторяется, чтобы заполнить весь блок. В нашем случае фактически было использовано изображение одной маленькой звезды.

Фоновое изображение

HTML

```
  <div class="first_box">
</div>
  <div class="second_box">
</div>
```

CSS

```
.first_box {

  background-image:
url(balloons.jpg);

}

.second_box {

  background-image: url(star.png);

}
```

Примечание: Если кроме фонового изображения вы добавили фоновый цвет, то изображение будет отображаться над цветом.

Свойство background-repeat

Свойство **`background-repeat`** используется для управления повторениями фонового изображения. Доступные значения:

- `no-repeat` — останавливает повторение фонового изображения во всех направлениях.
- `repeat-x` — повторение фонового изображения по горизонтали.
- `repeat-y` — повторение фонового изображения по вертикали.
- `repeat` — повторение фонового изображения в обоих направлениях. Установлено по умолчанию.

Свойство background-repeat

Используем свойство `background-repeat` для работы с изображением звёздочкой из предыдущего примера

CSS

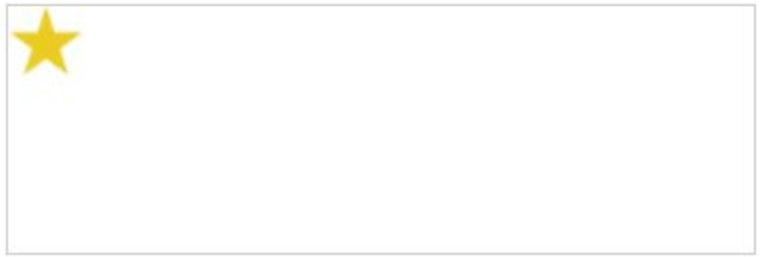
```
.second_box {

  background-image:
url(star.png);

  background-repeat: no-repeat;

}
```

Итог



Изменение размеров фонового изображения

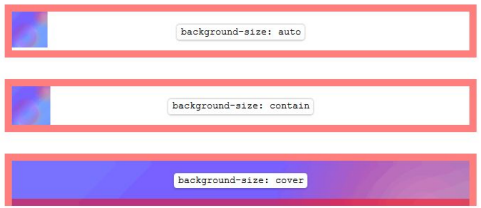
Для более корректного отображения больших картинок для фона мы можем использовать использовать свойство `background-size,` которое может принимать значения длины или в процентах, чтобы размер изображения соответствовал размеру фона.

```
background-size: 30%;
```

Изменение размеров фонового изображения

Для свойства `background-size` можно использовать не только показатели ширины, но и ключевые слова:

- `cover` — браузер сделает изображение достаточно большим, чтобы оно полностью заполнило блок, сохраняя при этом соотношение сторон. В этом случае часть изображения, скорее всего, окажется за пределами блока.
- `contain` — браузер сделает изображение нужного размера, чтобы поместиться в блоке. В этом случае могут появиться пробелы с обеих сторон или сверху и снизу изображения, если соотношение сторон изображения отличается от соотношения сторон блока.



Позиционирование фонового изображения

Свойство `background-position` позволяет вам изменять позицию, в которой фоновое изображение появляется в блоке. При этом используется система координат, в которой левый верхний угол блока равен (0,0), а сам блок располагается вдоль горизонтальной (x) и вертикальной (y) осей.

Примечание: По умолчанию значение `background-position` равно (0,0).

```
background-position: top;
```

```
background-position: left;
```

```
background-position: center;
```

```
background-position: 25% 75%;
```

Несколько фоновых изображений

Возможно создавать несколько фоновых изображений — просто разделив значения свойства `background-image` запятыми.

Когда вы сделаете это, произойдёт наложение фоновых изображений друг на друга. Фоновые изображения будут наложены слоями, где каждое новое фоновое изображение, перечисленное в коде, будет накладываться поверх ранее указанного изображения.

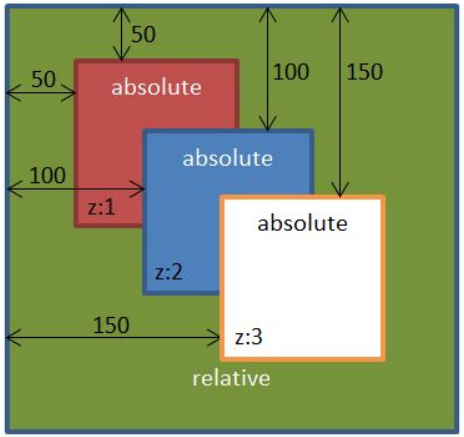
```
background-image: url(image1.png), url(image2.png), url(image3.png),
url(image1.png);
```

Сокращённое свойство background

Свойство background объединяет в себе все свойства, которые используются для определения фона страницы.

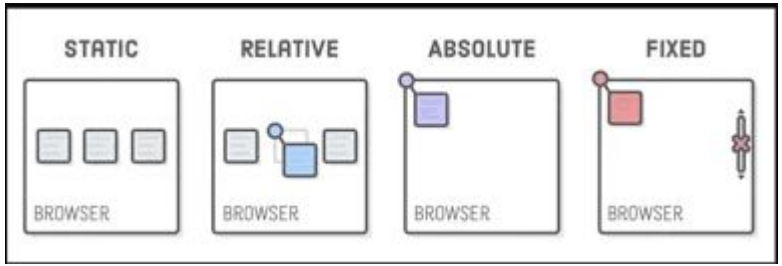
Свойства, которые можно задать (по порядку): `background-color`, `background-image`, `background-repeat`, `background-position`.

Позиционирование элементов



Свойство position

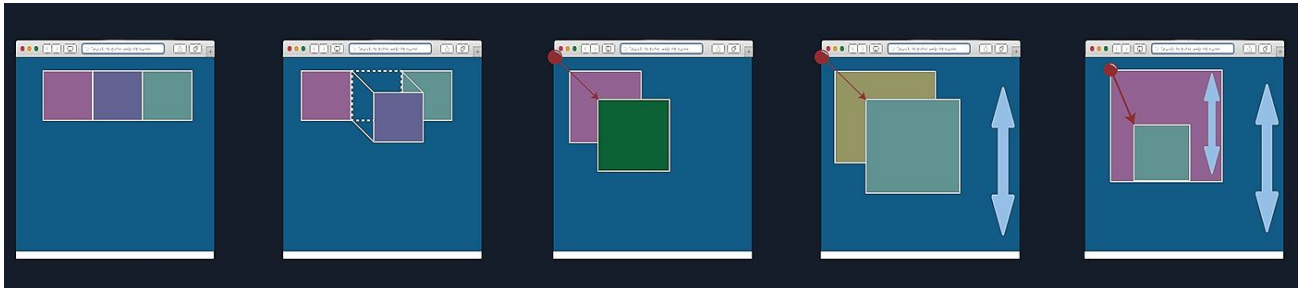
Свойство **position** — это одно из свойств с помощью которого можно изменить базовое поведение элементов в потоке. Другими словами, данное свойство позволяет «выдернуть» любой элемент из потока документа и разместить его в другом месте относительно окна браузера или других элементов на веб-странице.



Свойство position

Свойство position имеет 5 значений:

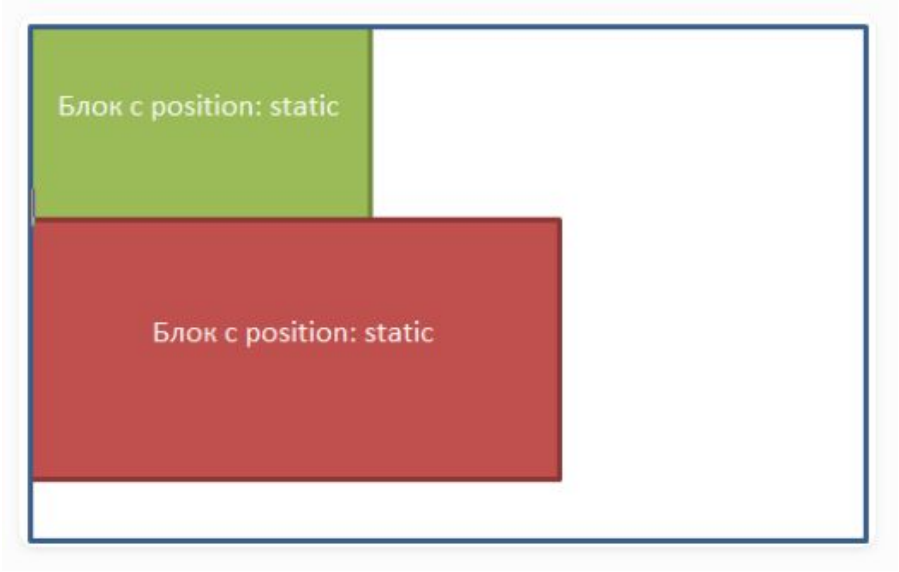
- `static` (статичное позиционирование);
- `relative` (относительное);
- `absolute` (абсолютное);
- `fixed` (фиксированное);
- `sticky` (липкое).



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`.

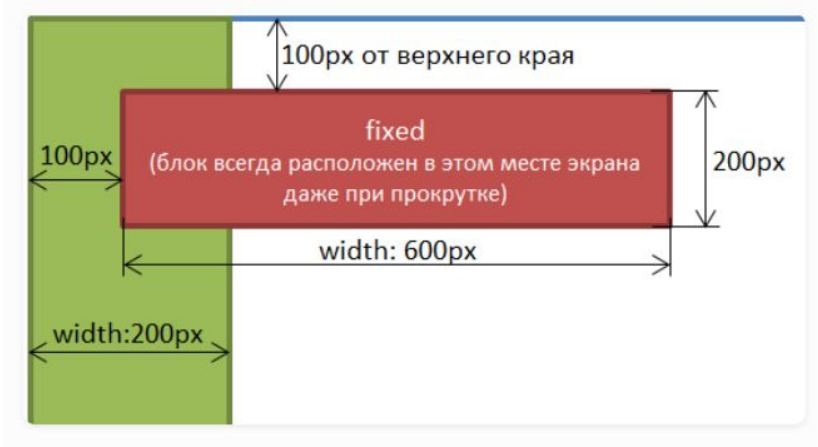
Позиционирование выполняется относительно ближайшего позиционированного предка.

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

Свойство position: fixed

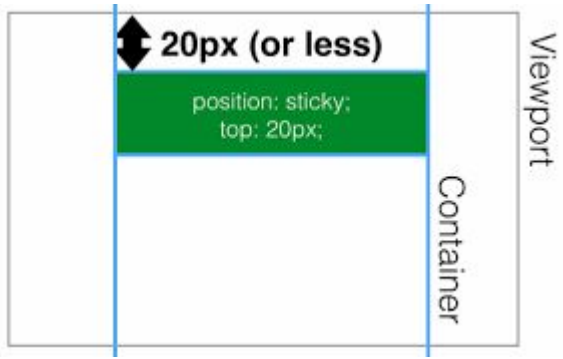
Фиксированное позиционирование похоже на абсолютное, но в отличии от него оно всегда привязывается к краям окна браузера (viewport), и остаётся в таком положении даже при скроллинге страницы.

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

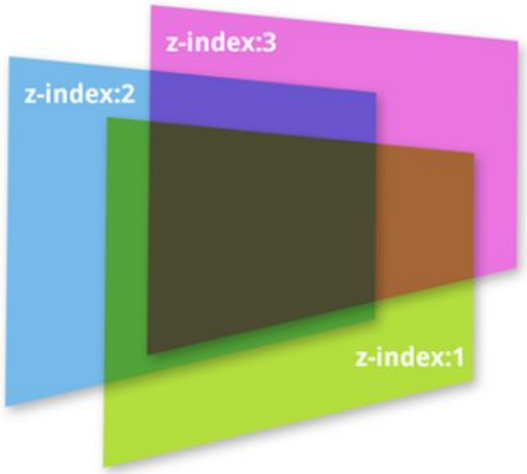


Свойство position: sticky

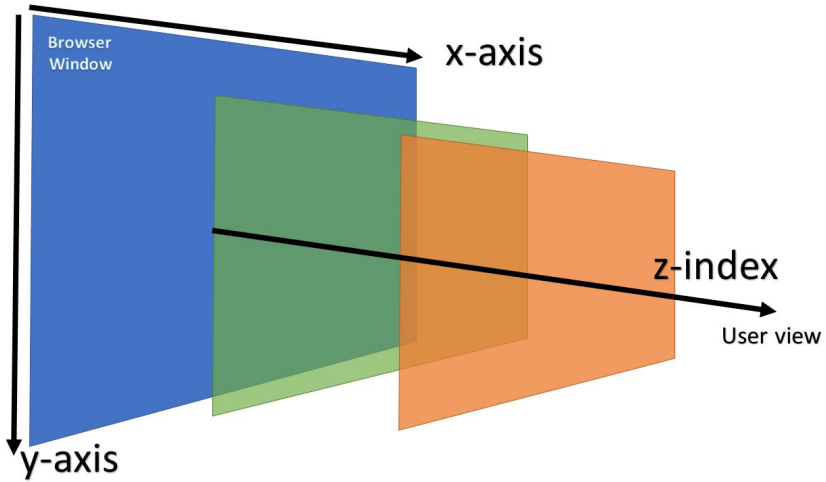
Липкие элементы (`position: sticky;`) очень похожи на фиксированные (`position: fixed`), поскольку они сохраняют свое положение на экране, даже когда пользователь прокручивается страницу вверх или вниз. Разница в том, что липкий элемент остается ограниченным родительским контейнером, в котором он находится.



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`.



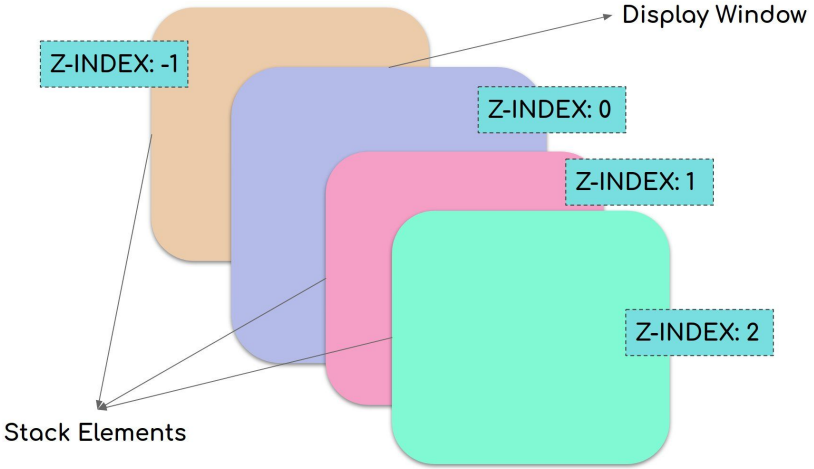
Синтаксис

```
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



Ваша новая IT-профессия – Ваш новый уровень жизни

Программирование с нуля в
немецкой школе AIT TR GmbH

AIT
Hi-tech team