

# 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;)

■ Основные свойства,  
используемые в box-model

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

■ Что такое margin collapse

# ЦЕЛЬ

**Изучить методы позиционирования элементов с помощью свойства position.**

# ПЛАН ЗАНЯТИЯ

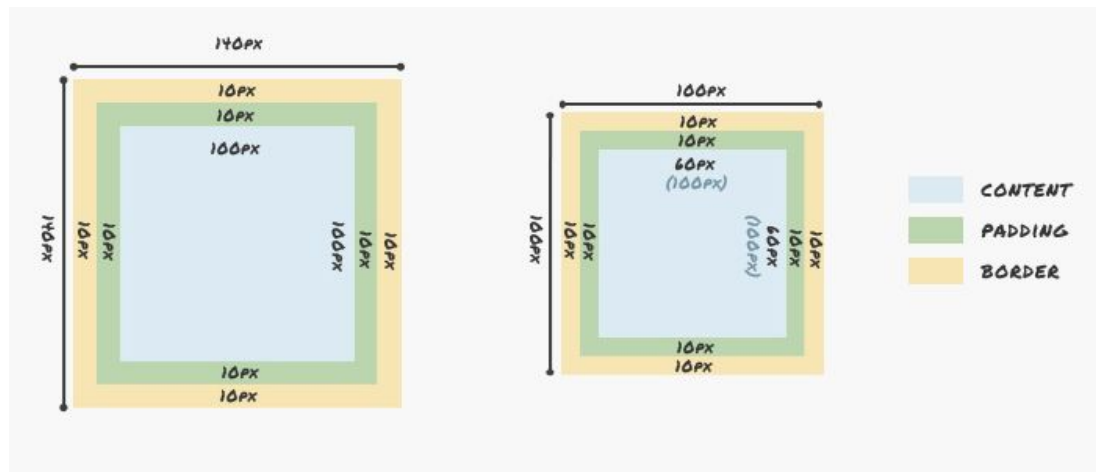
- Единицы измерения
- Значения свойства position

# Размеры



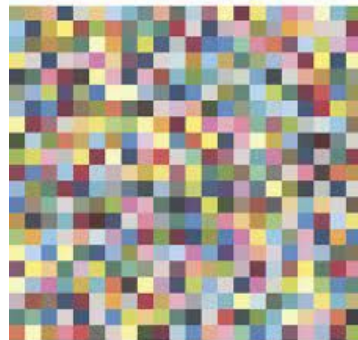
## Основные способы задания размеров в CSS:

1. Пиксели (px):
2. Проценты (%)
3. EM и REM
4. Viewport Units (vw, vh)



**Пиксели** - это наиболее распространенная единица измерения в CSS и представляют собой точки на экране. Задавая размеры в пикселях, вы указываете конкретное количество пикселей в ширине, высоте или других размерах элемента.

```
div {  
  width: 200px;  
  height: 100px;  
}
```





Проценты могут быть использованы для установки размеров элементов относительно размеров их родительских контейнеров.

```
.container {  
  width: 80%;  
}
```



**Эм (em) и относительные эм (rem)** - это единицы измерения, которые зависят от текущего размера шрифта.

**em** измеряет размер относительно размера текущего шрифта элемента, в то время как **rem** измеряет размер относительно размера шрифта корневого элемента (обычно `<html>`).

```
p {  
  font-size: 1.2em;  
}
```

```
.container {  
  padding: 1rem;  
}
```

## Viewport Units (vw, vh):

Viewport units - это единицы измерения, которые зависят от размеров видимой области браузера (viewport).

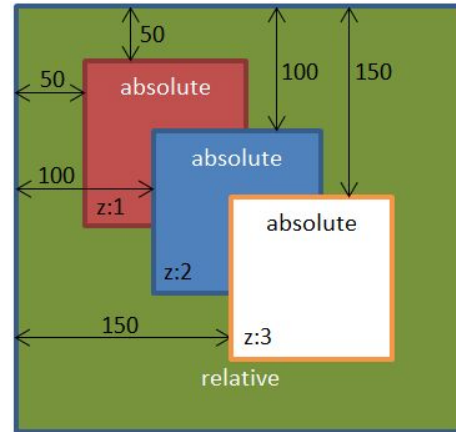
Например, **1vw** представляет 1% ширины видимой области, а **1vh** представляет 1% высоты видимой области.

Это полезно для создания адаптивных макетов, которые реагируют на размер экрана пользователя.

```
.full-width {  
  width: 100vw; /* Элемент будет занимать всю ширину видимой области браузера */  
}
```

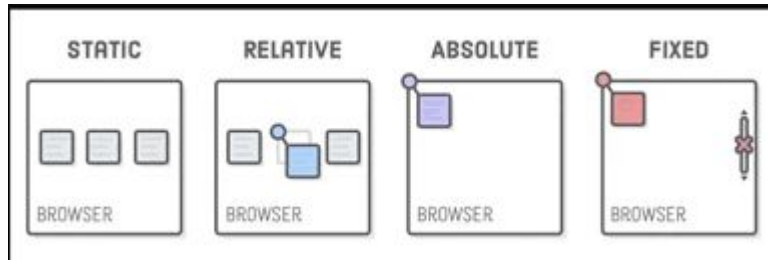
```
.half-height {  
  height: 50vh; /* Элемент будет занимать половину высоты видимой области */  
}
```

# Позиционирование элементов



## Свойство position

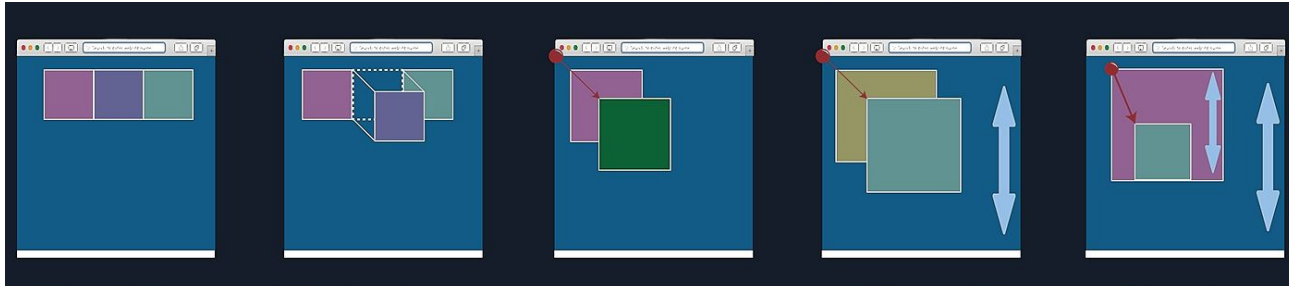
Свойство **position** — это одно из свойств с помощью которого можно изменить базовое поведение элементов в потоке. Другими словами, данное свойство позволяет «выдернуть» любой элемент из потока документа и разместить его в другом месте относительно окна браузера или других элементов на веб-странице.



# Свойство position

Свойство position имеет 5 значений:

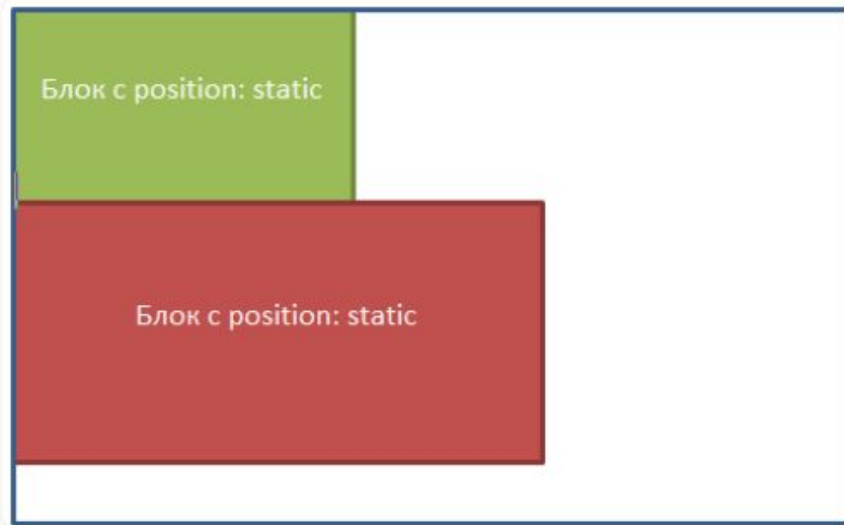
- **static** (статичное позиционирование);
- **relative** (относительное);
- **absolute** (абсолютное);
- **fixed** (фиксированное);
- **sticky** (липкое).



## 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).

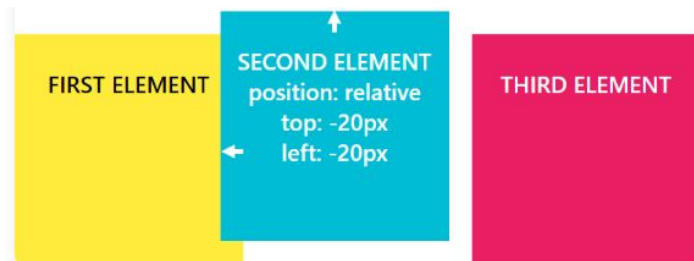
Установка CSS свойств для задания положения элемента `left`, `top`, `right` и `bottom` никакого влияния на него не оказывают, т.к. его местонахождение определяется потоком документа.



## 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

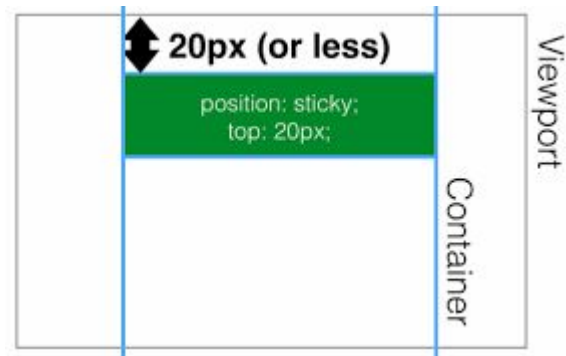
Фиксированное позиционирование похоже на абсолютное, но в отличие от него оно всегда привязывается к краям окна браузера (viewport), и остаётся в таком положении даже при скроллинге страницы.

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

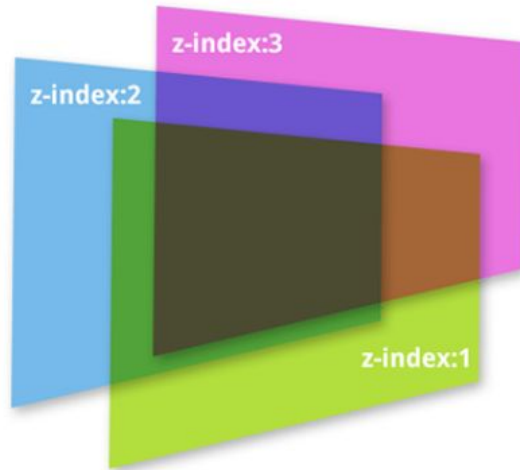


## Свойство position: sticky

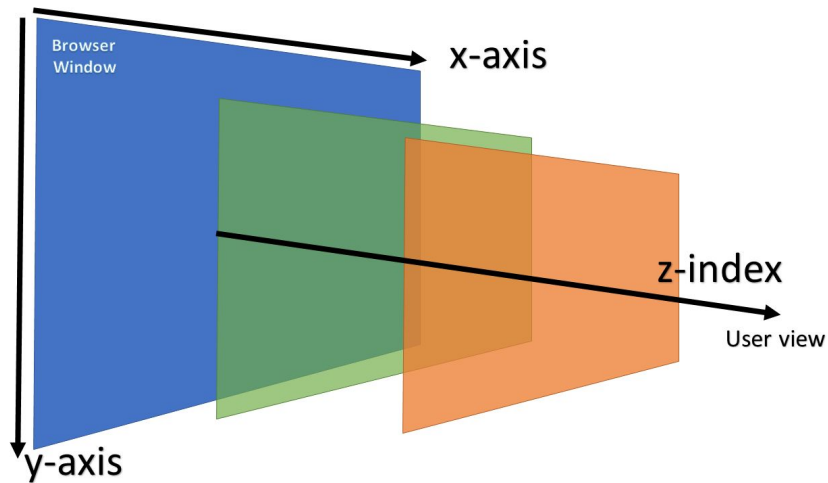
Липкие элементы (`position: sticky;`) очень похожи на фиксированные (`position: fixed;`), поскольку они сохраняют свое положение на экране, даже когда пользователь прокручивается страницу вверх или вниз. Разница в том, что липкий элемент остается ограниченным родительским контейнером, в котором он находится.



# z-index



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



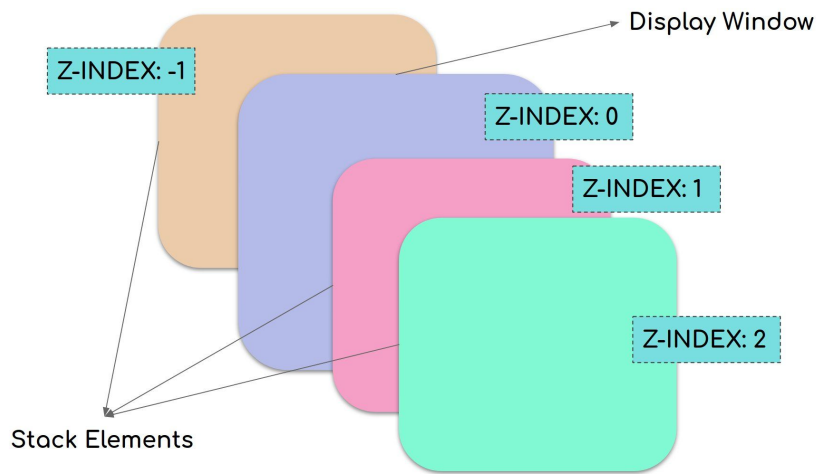
## Синтаксис

`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