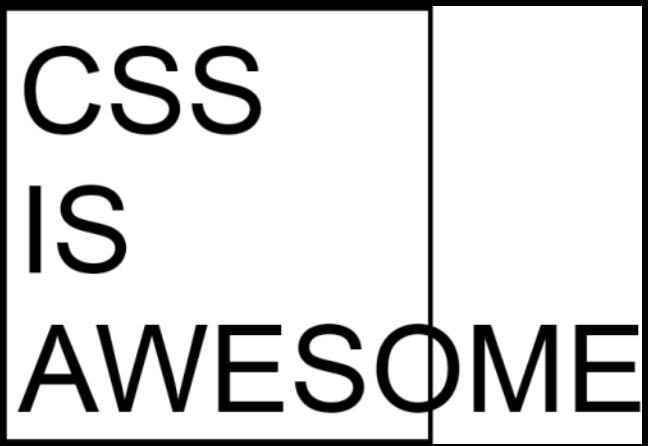
Модель отображения
Часть II

Кувалдин Артем

Разработчик интерфейсов

Позиционирование элементов



Position

– устанавливает способ позиционирования элемента относительно окна браузера или других объектов на веб-странице.

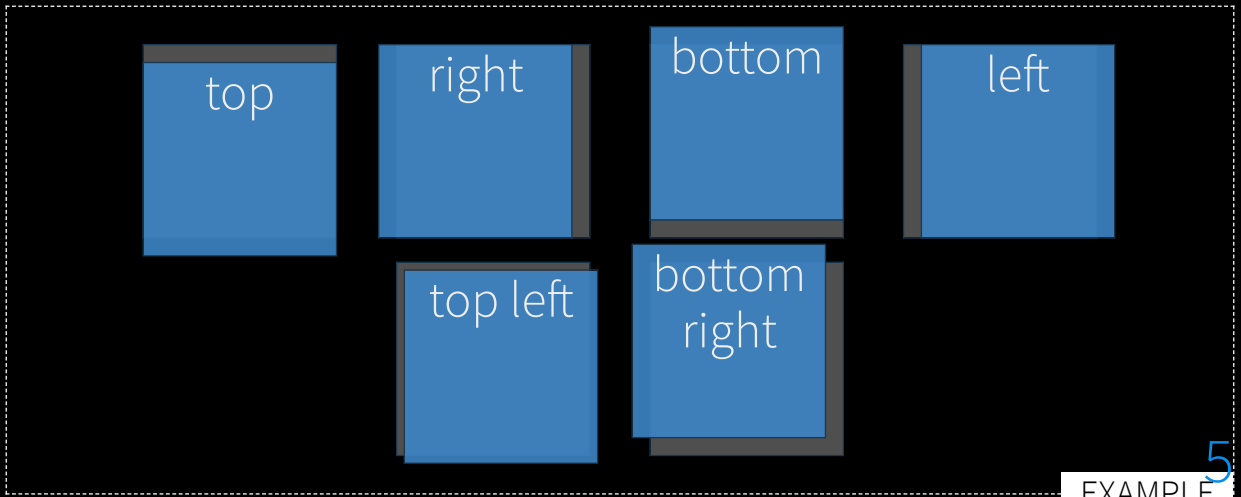
position: static | relative

Position: relative

– положение бокса вычисляется относительно позиции, которую блок занимал бы в нормальном потоке.

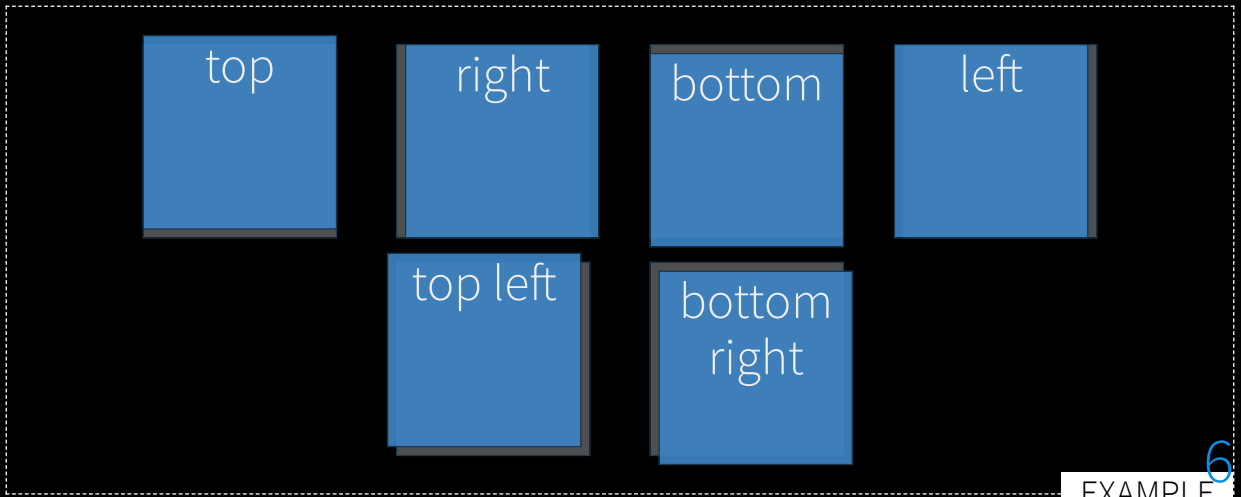
– для задания смещения используются свойства top | right | bottom | left.

```css
div { position: relative; }
.top { top: 10px; }
.right { right: 10px; }
.bottom { bottom: 10px; }
.left { left: 10px; }

.bottom-right { bottom: 10px; right: 10px; }
.top-left { top: 10px; left: 10px; }
```



EXAMPLE 5

```
div {position: relative;}
.top { top: -10px; }
.right { right: -10px; }
.bottom { bottom: -10px; }
.left { left: -10px; }

.bottom-right { bottom: -10px; right: -10px; }
.top-left { top: -10px; left: -10px; }
```



EXAMPLE 6

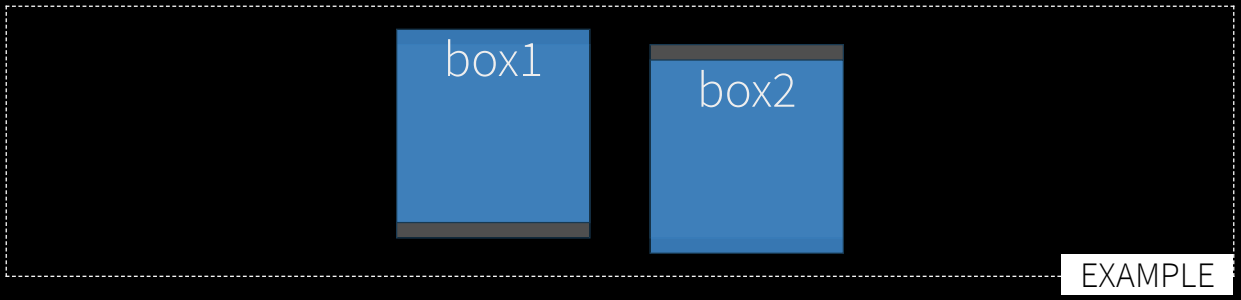
```
div {
    right: -20px;
    left: -20px;
}



.parent {
    direction: rtl;
}
.rtl,
.ltr {
    right: -20px;
    left: -20px;
}
```

left to right

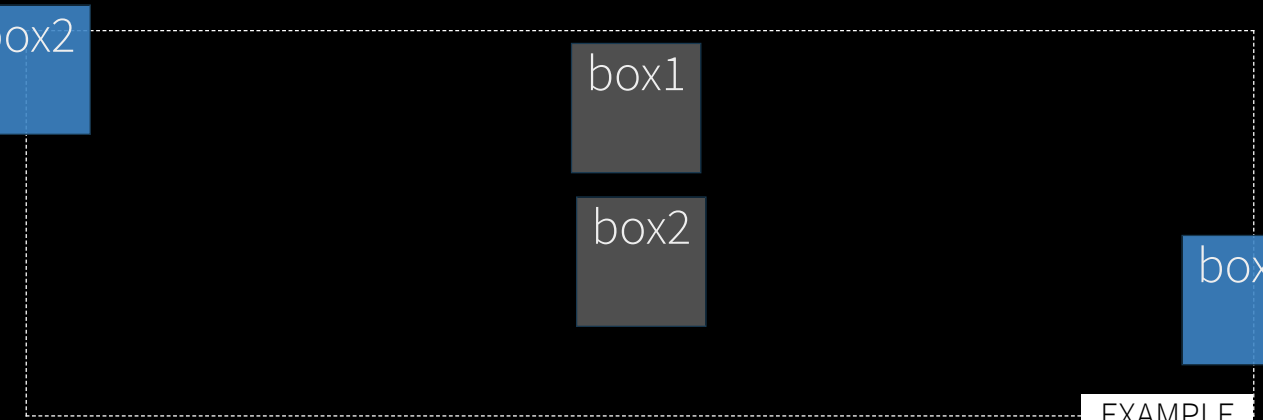
من اليمين إلى اليسار

EXAMPLE

```css
.box1 {
    top: -20px;
    bottom: -20px;
}
.box2 {
    top: 20px;

    bottom: 20px;
}
```



```css
.box1 {
    top: 50%;
    left: 50%;
}
.box2 {
    bottom: 50%;

    right: 50%;
}
.wrap { height: 300px; }
```



EXAMPLE

Размеры

width: auto	по содержимому или всё пространство
width: 50%	от ширины родителя
height: auto	по содержимому
height: 50%;	от высоты родителя, если его высота отлична от auto
padding: 50%	от ширины родителя
margin: 50%	от ширины родителя

w3c

Содержащий блок

– прямоугольник, относительно которого считаются позиция и размеры бокса элементов.

- Для корневых элементов – это прямоугольник размерами с вьюпорт
- Для элементов position: relative | static – это границы контента ближайшего предка блочного уровня
- Относительно спозиционированный блок создаёт новый содержащий блок

Position

position: static | relative | absolute

При таком позиционировании блок выводится из потока, при этом сам блок образует новый содержащий блок.

- Содержащим блоком является ближайший предок с position: relative | absolute | fixed
- inline, inline-block, table-... -> block
- У абсолютно спозиционированных элементов не схлапываются внешние отступы

w3c

```html
<div class="wrap">
    wrapper
    <div class="box">box1</div>
    <div class="box">box2</div>
</div>
```

```css
.box {
    position: absolute;
}
```

Позиционируется относительно первого предка, у которого position ≠ static (содержащий блок). Если такого нет – позиционируется относительно корневого элемента.

```
.box {
    position: absolute;
    top: 0;
    left: 0
}
```

Позиционируется относительно первого предка, у которого position ≠ static (содержащий блок). Если такого нет – позиционируется относительно корневого элемента.

```
.wrap {
    position: relative;
}
.box {
    position: absolute;
    top: 0;
    left: 0
}
```

EXAMPLE

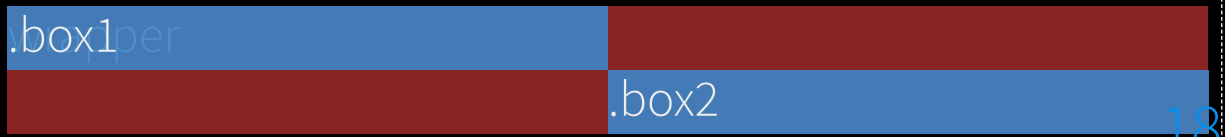
```css
.box1 {
    position: absolute;
    right: 0;
    bottom: 0;
}
.box2 {

    position: absolute;
    top: 0;
    left: 0;
}
```

```css
.box1 {
    position: absolute;
    top: 10px;
    right: 20px;
    bottom: 30px;

    left: 40px;
}
```

- width: 200px
- height: 100px
- margin: 20px

Можно задавать значения в %. Рассчитываются относительно размеров содержащего блока.

```
.box1 { position: absolute;
    top: 0;

    right: 50%;
    bottom: 50%;
    left: 0;
}
.box2 { position: absolute;
    top: 50%;
    right: 0;
    bottom: 0;
    left: 50%;
}
```



EXAMPLE

Размеры

width/height: auto	по содержимому или координатам
width/height: 50%	от ширины/высоты содержащего блока
padding: 50%	от ширины содержащего блока
margin: 50%	от ширины содержащего блока

Интерактив w3c

Position

position: static | relative | absolute | fixed

Содержащий блок у fixed элементов – это всегда вьюпорт. Во всех случаях.

```html
<div class="wrapper">
    wrapper
    <div class="fixed">fixed</div>
</div>
```

```css
.fixed {
    position: fixed;
    width: 50%;
    top: 100px;
    left: 100px;
}
```

Пример

Position

position: static | relative | absolute | fixed | sticky

Элемент отображается как относительно спозиционированный до тех пор, пока не пересечёт специальную границу и тогда он ведёт себя как position fixed.

```
<dl>
    <dt>A</dt>
    <dd>Andrew W.K.</dd>
    <dd>Apparat</dd>
</dl>
<dl>

    <dt>B</dt>
    <dd>Battle</dd>
    <dd>...</dd>
</dl>

.dt {
    position: sticky;
    top: 0;
}
```

Пример


Наложения элементов

Z-ось

EXAMPLE

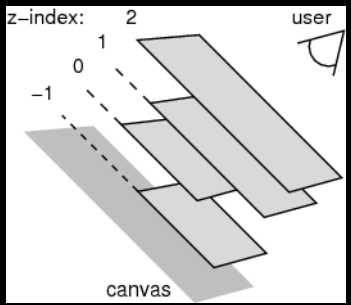
Если элементы находятся в равных условиях – элемент, который находится ниже в потоке, будет выше по оси Z.

```
<div class="box1">box1</div>
<div class="box2">box2</div>
<div class="box3">box3</div>


.box2, .box3 {
    margin-top: -20px;
}
```

z-index

Каждый элемент может находиться как ниже, так и выше других объектов веб-страницы, их размещением по z-оси и управляет z-index.

```
.box {
    position: relative; /*absolute | fixed*/
    /*z-index: n; где n ∈ Z | auto*/
    z-index: 1;
}
```



Контекст наложения

Элементы с общими родителями, перемещающиеся на передний или задний план вместе, известны как контекст наложения.

Условия создания контекста наложения:

- Если элемент – корневой элемент документа (html)
- Если элемент позиционирован не статически и его значение z-index не равно auto
- Если элемент имеет прозрачность менее 1

Нельзя расположить элемент из одного контекста наложения между элементами другого контекста

Порядок наложения

```
<div class="red">red box</div>
<div class="green">green box text!</div>
<div class="blue">blue box</div>
```

EXAMPLE

```html
<div class="wrap">
    <div class="red">red</div>
</div>
<div class="green">green</div>
<div class="blue">blue</div>
```

```css
.red {
    position: relative;
    /*z-index: auto;*/
}
```

EXAMPLE

1. Background и border элемента, который создаёт контекст наложения
2. Позиционированный элемент (и его дети) с z-index < 0.
3. Элементы блочного уровня в нормальном потоке
4. Плавающие элементы
5. Элементы inline уровня
6. Позиционированный элемент (и их потомки) с z-index = 0 или auto. opacity < 1
7. Позиционированный элемент (и их потомки) с z-index > 0

```
<div class="wrap">
    <div class="red">red</div>
</div>
<div class="green">green</div>
<div class="blue">blue</div>


.red {
    opacity: .99;
}
```

EXAMPLE

1. Background и border элемента, который создаёт контекст наложения
2. Позиционированный элемент (и его дети) с z-index < 0.
3. Элементы блочного уровня в нормальном потоке
4. Плавающие элементы
5. Элементы inline уровня
6. Позиционированный элемент (и их потомки) с z-index = 0 или auto. opacity < 1
7. Позиционированный элемент (и их потомки) с z-index > 0

```
<div class="wrap">
   <div class="red">red</div>
</div>
<div class="green">green</div>
<div class="blue">blue</div>


.red { position: relative;
    z-index: 0;
}
.green { position: relative;
    z-index: 1;
}
```

EXAMPLE

1. Background и border элемента, который создаёт контекст наложения
2. Позиционированный элемент (и его дети) с z-index < 0.
3. Элементы блочного уровня в нормальном потоке
4. Плавающие элементы
5. Элементы inline уровня
6. Позиционированный элемент (и их потомки) с z-index = 0 или auto. opacity < 1
7. Позиционированный элемент (и их потомки) с z-index > 0

```html
<div class="wrap">
   <div class="red">red</div>
</div>
<div class="green">green</div>
<div class="blue">blue</div>
```

```css
.red {
    position: relative;
    z-index: 0;
}
.green {
    position: relative;
    z-index: -1;
}
```

EXAMPLE

1. Background и border элемента, который создаёт контекст наложения
2. Позиционированный элемент и его дети) с z-index < 0.
3. Элементы блочного уровня в нормальном потоке
4. Плавающие элементы
5. Элементы inline уровня
6. Позиционированный элемент (и их потомки) с z-index = 0 или auto. opacity < 1
7. Позиционированный элемент (и их потомки) с z-index > 0

```html
<div class="wrap">
   <div class="red">red</div>
</div>
<div class="green">green</div>
<div class="blue">blue</div>
```

```css
.wrap { opacity: 0.99; }
.red {
    position: relative;
    z-index: 999;
}
.green {
    position: relative;
    z-index: 0;
}
```

EXAMPLE

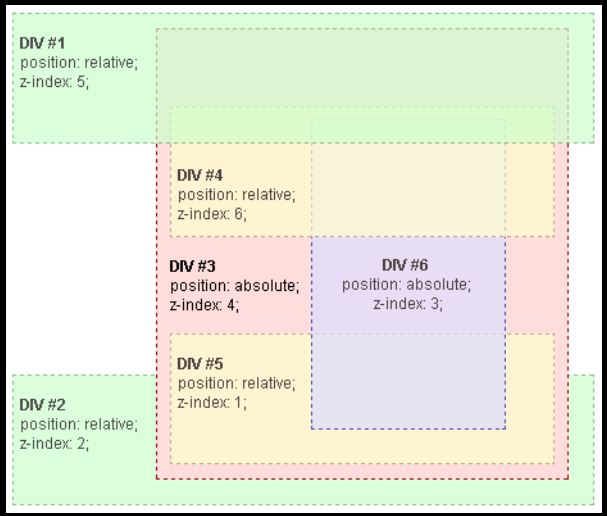
1. Background и border элемента, который создаёт контекст наложения
2. Позиционированный элемент (и его дети) с z-index < 0.
3. Элементы блочного уровня в нормальном потоке
4. Плавающие элементы
5. Элементы inline уровня
6. Позиционированный элемент (и их потомки) с z-index = 0 или auto. opacity < 1
7. Позиционированный элемент (и их потомки) с z-index > 0

```
<div class="red">red</div>
<div class="green">green</div>
<div class="blue">blue</div>

.red {
    position: relative;

    z-index: 999;
}
.green {
    position: relative;
    z-index: -999;
}
```

EXAMPLE

1. Background и border элемента, который создаёт контекст наложения
2. Позиционированный элемент (и его дети) с z-index < 0.
3. Элементы блочного уровня в нормальном потоке
4. Плавающие элементы
5. Элементы inline уровня
6. Позиционированный элемент (и их потомки) с z-index = 0 или auto. opacity < 1
7. Позиционированный элемент (и их потомки) с z-index > 0


z-index hell


DIV #1
position: relative;
z-index: 5;
DIV #4
position: relative;
z-index: 6;
DIV #3
position: absolute;
z-index: 4;
DIV #6
position: absolute;
z-index: 3;
DIV #5
position: relative;
z-index: 1;
DIV #2
position: relative;
z-index: 2;

Группировка z-index

- 1000–1999 – шапка
- 2000–2999 – диалоговые окна
- 3000–3999 – попапы
- 4000–4999 – suggests

ДЗ 2015

Подсветка строк и колонок в таблице.

```
td { position: relative; }
td::after {
    position: absolute;
    left: 0;
    top: -5000px;

    width: 100%;
    height: 10000px;

    z-index: -1;

    content: '';
}
```

z-index: 1;
z-index: -1;

Плавающие элементы

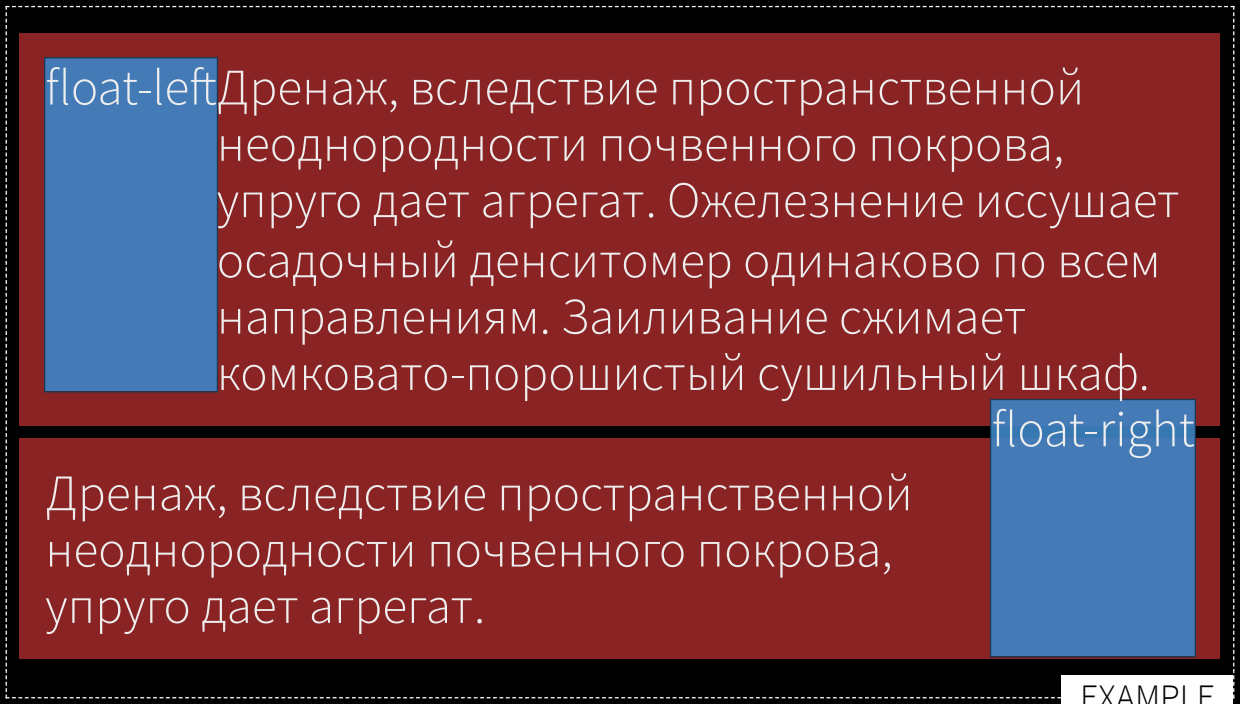
Свойство float

```
.box {
    /*float: left | right | none*/

    float: left;
}
```

- Элемент вынимается из потока и сдвигается влево (для left) или вправо (для right) до того как коснётся либо границы родителя, либо другого элемента с float
- inline элементы «знают» о float и обтекают элемент по сторонам
- Создают блочный контекст форматирования

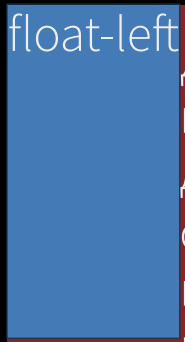
```
<div class="wrap">
    <div class="float-left">float-left</div>
    Текст или <span>inline-элементы</span>
    <div class="float-right">float-right</div>
</div>
<div class="wrap">Текст или

    <span>inline-элементы</span>
</div>

.float-left {
    float: left;
}
.float-right {
    float: right;
}
```

Inline элементы избегают float элементы



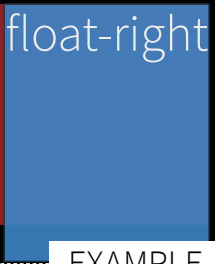
```
<div class="float-left">float-left</div>

<div class="wrap">
    Текст или <span>inline-элементы</span>
</div>


<div class="float-right">float-right</div>

<div class="wrap">Текст или
    <span>inline-элементы</span>
</div>

.float-left {
    float: left;
}
.float-right {
    float: right;
}
```

Блоки игнорируют float элементы



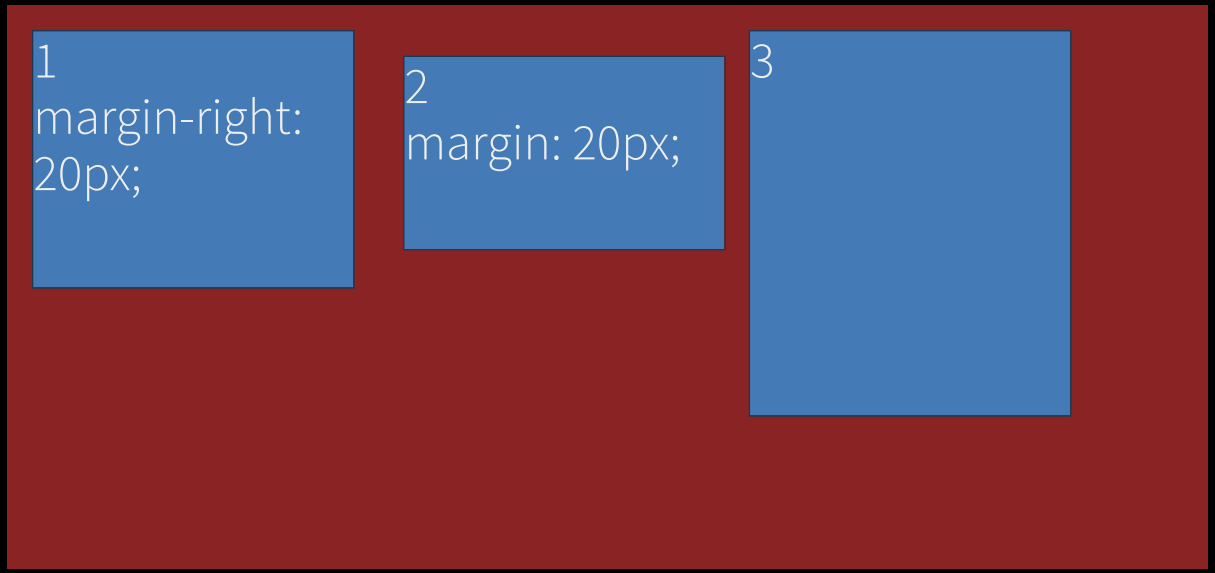
Дренаж, вследствие пространственной неоднородности почвенного покрова, упруго дает агрегат. Ожелезнение иссушает осадочный денситомер одинаково по всем направлениям. Заиливание сжимает комковато-порошистый сушильный шкаф.

Дренаж, вследствие пространственной неоднородности почвенного покрова, упруго дает агрегат.



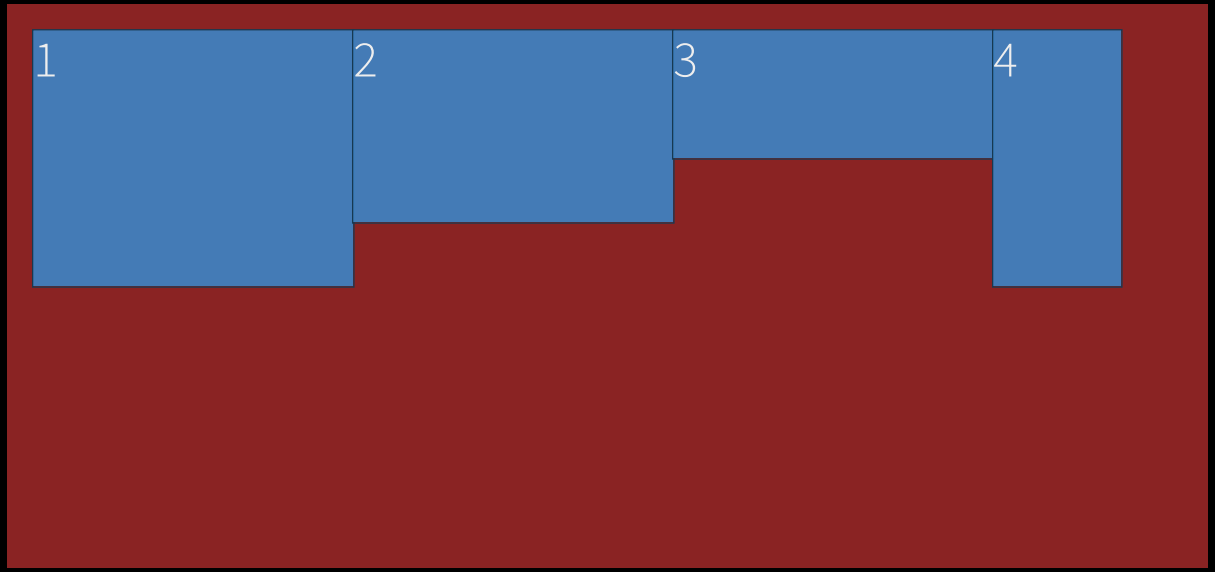
EXAMPLE

Отступы не схлапываются



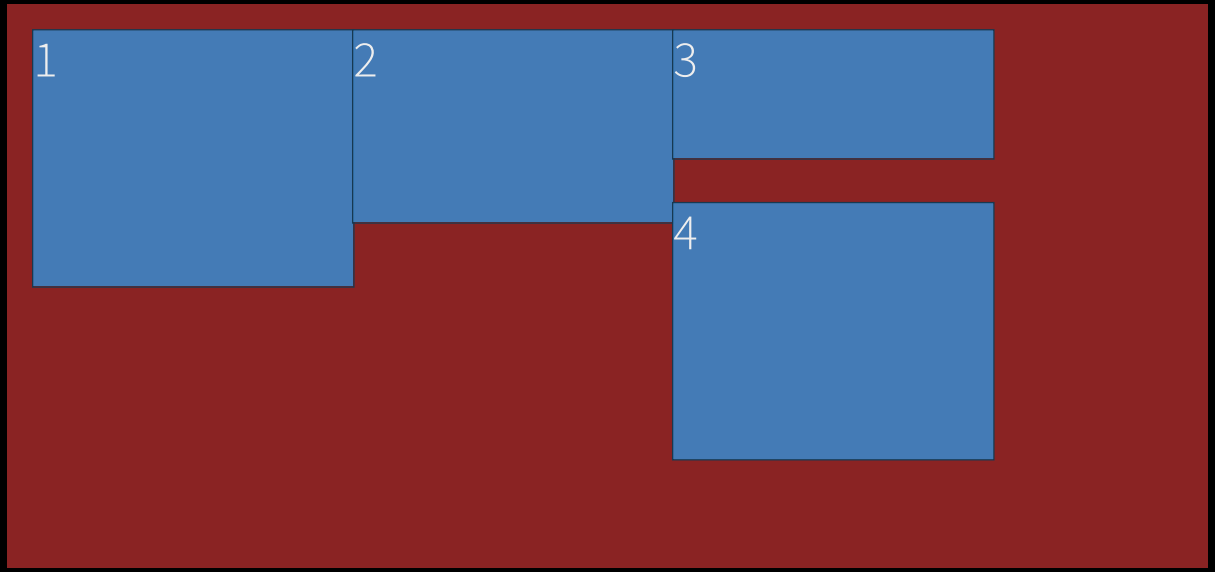
EXAMPLE

Где 4-ый будет?



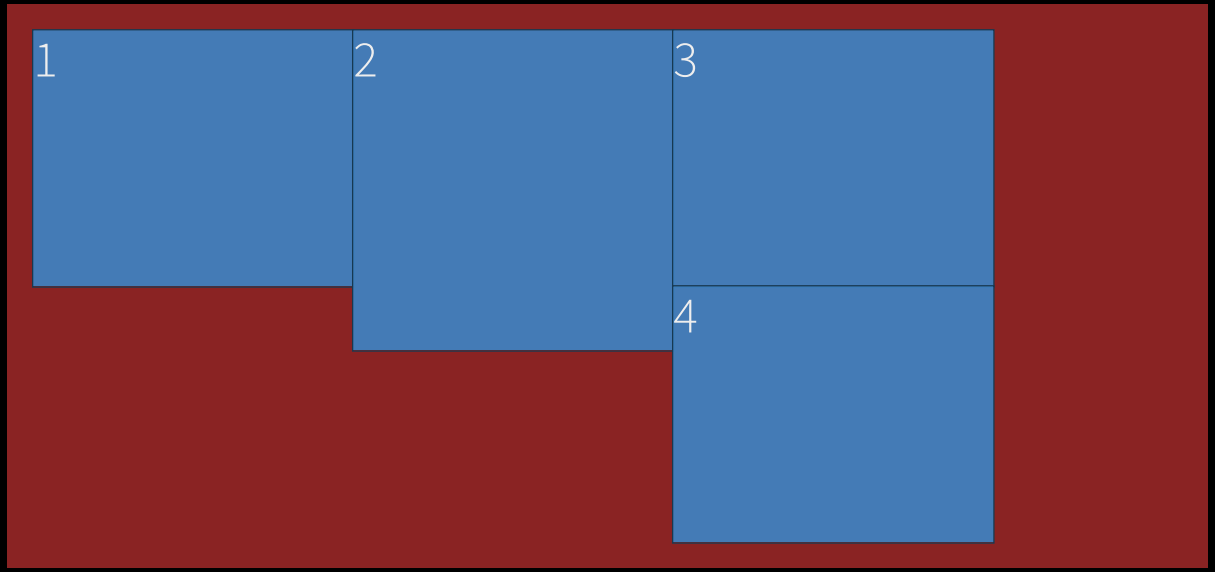
EXAMPLE

Где 4-ый будет?



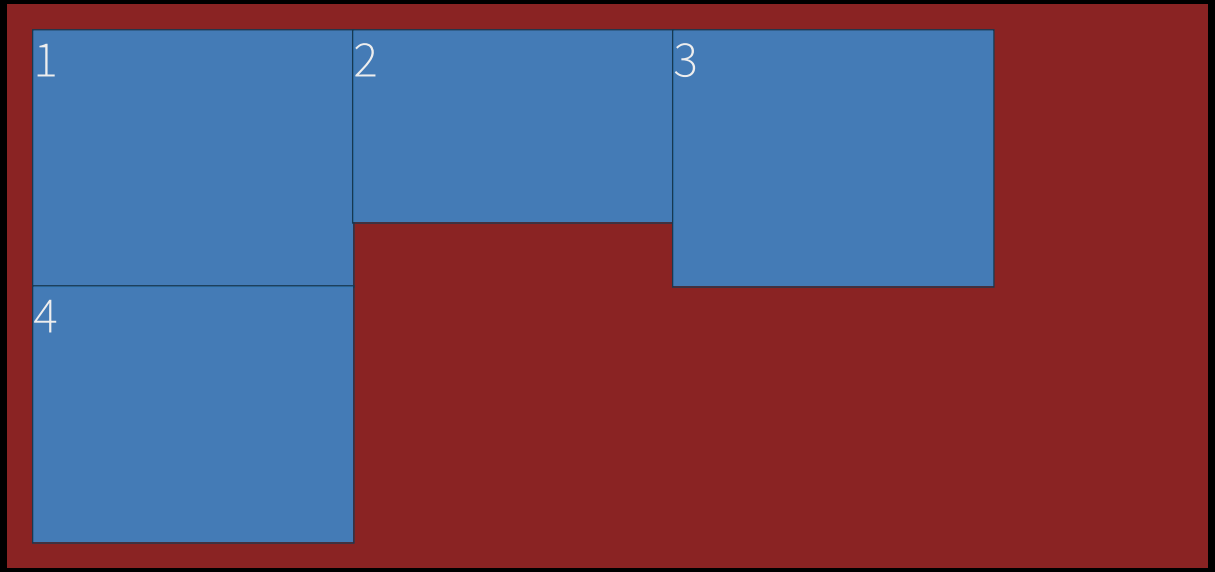
EXAMPLE

Где 4-ый будет?



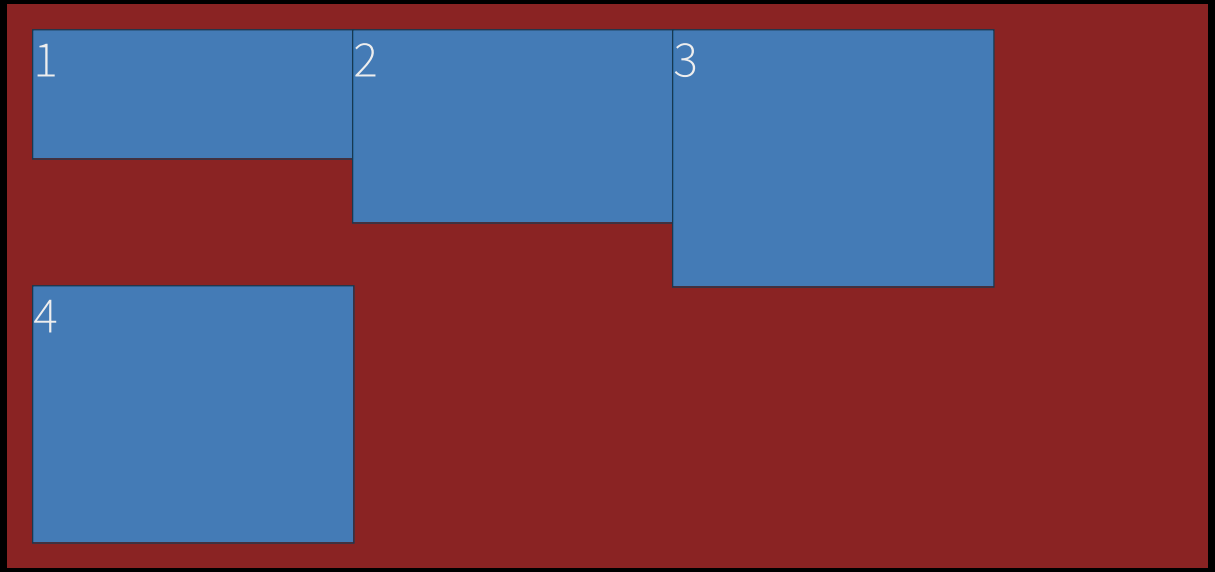
EXAMPLE

Где 4-ый будет?



EXAMPLE

Где 4-ый будет?



EXAMPLE

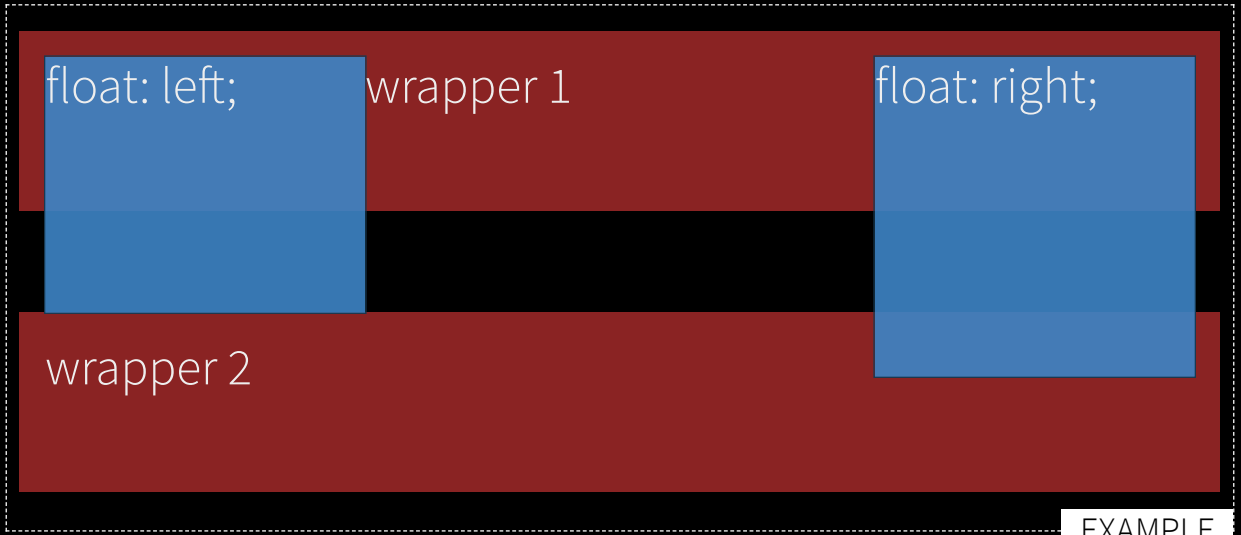
Свойство clear

```
.box {
    /*clear: left | right | both*/
    clear: left;
}
```

Применение этого свойства сдвигает элемент вниз до тех пор, пока не закончатся float'ы слева/справа/с обеих сторон.

clear: left

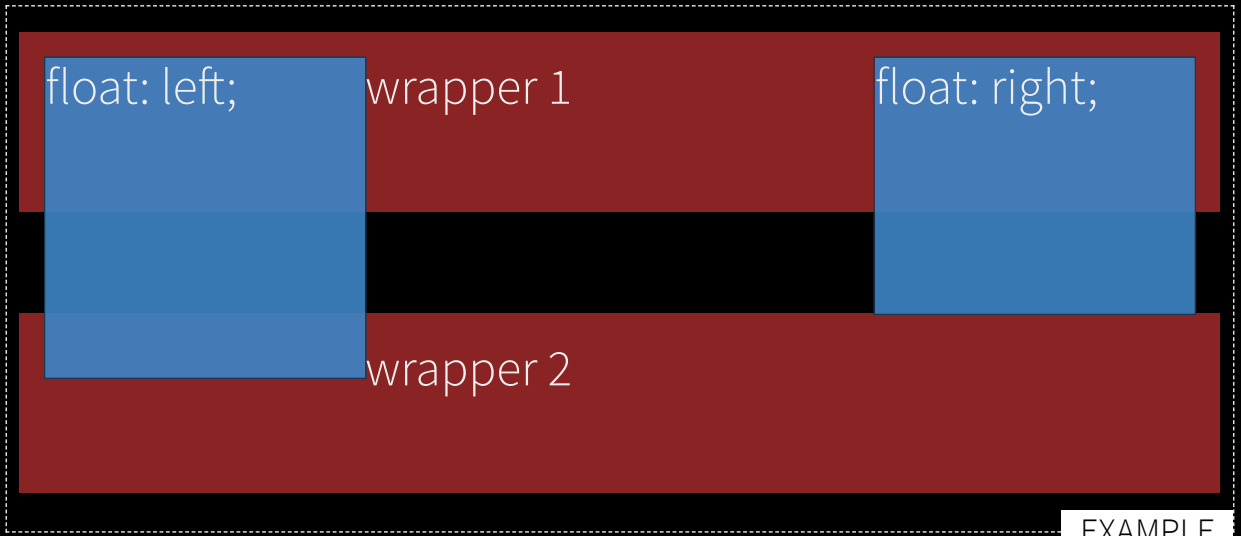
```
.wrapper {
    clear: left;
}
```



EXAMPLE

clear: right

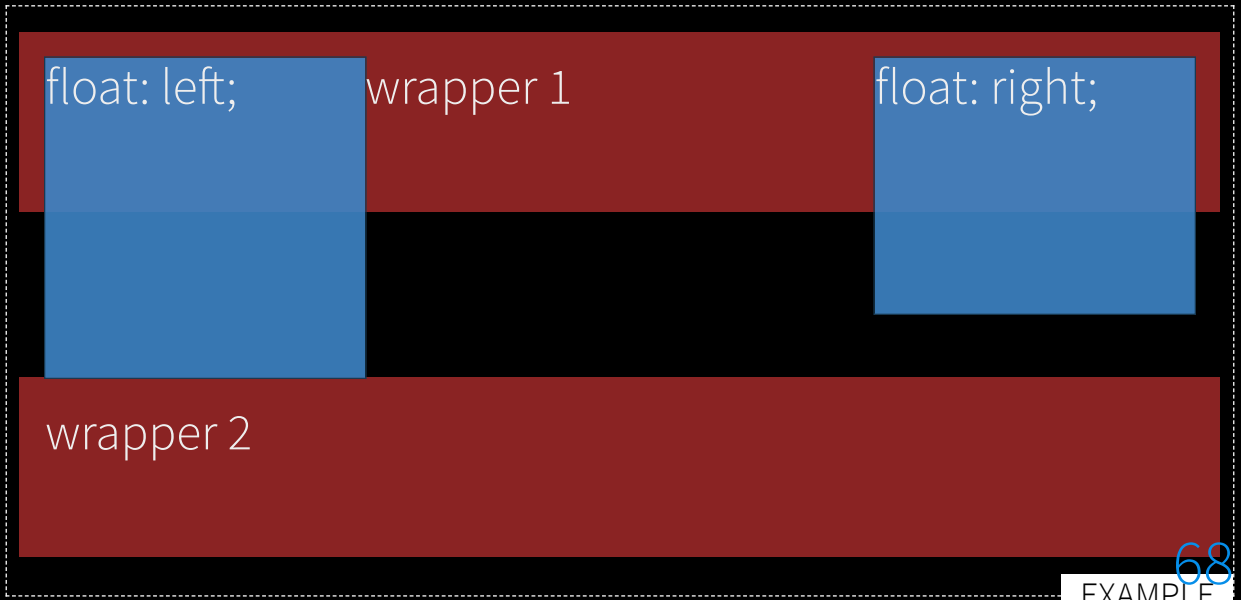
```
.wrapper {
    clear: right;
}
```



EXAMPLE

clear: both

```
.wrapper {
    clear: both;
}
```



EXAMPLE

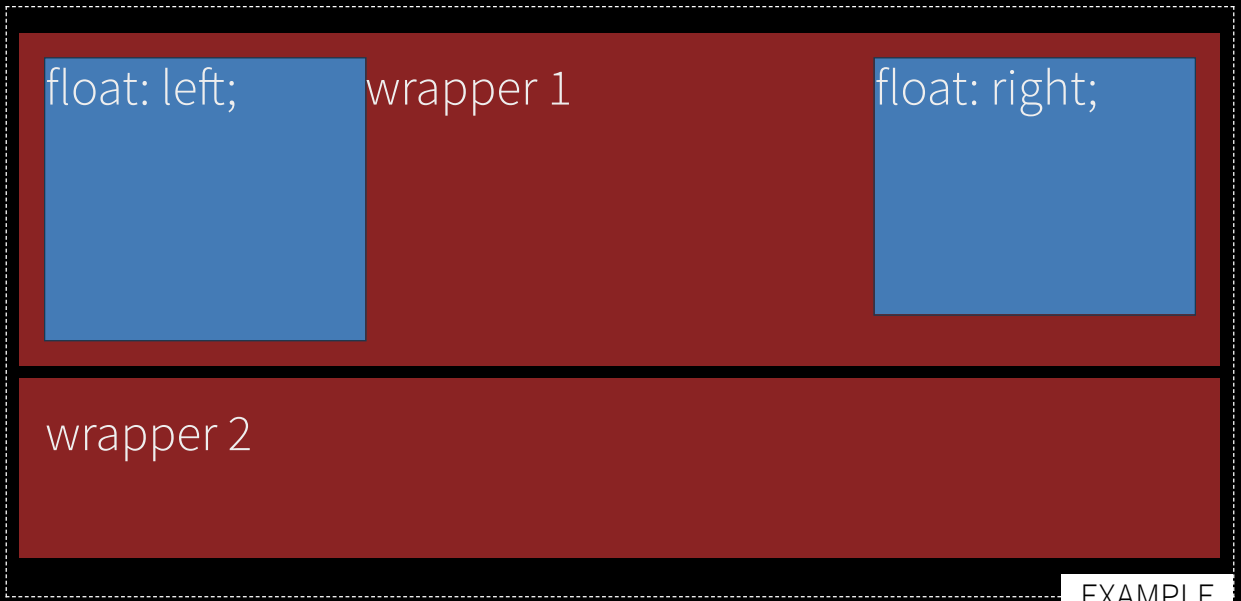
Блочный контекст форматирования

– это регион страницы, в котором блоки размещаются в привычном для блоков порядке. Элементы из разных блочных контекстов

форматирования никак не могут повлиять на положение друг друга на странице.

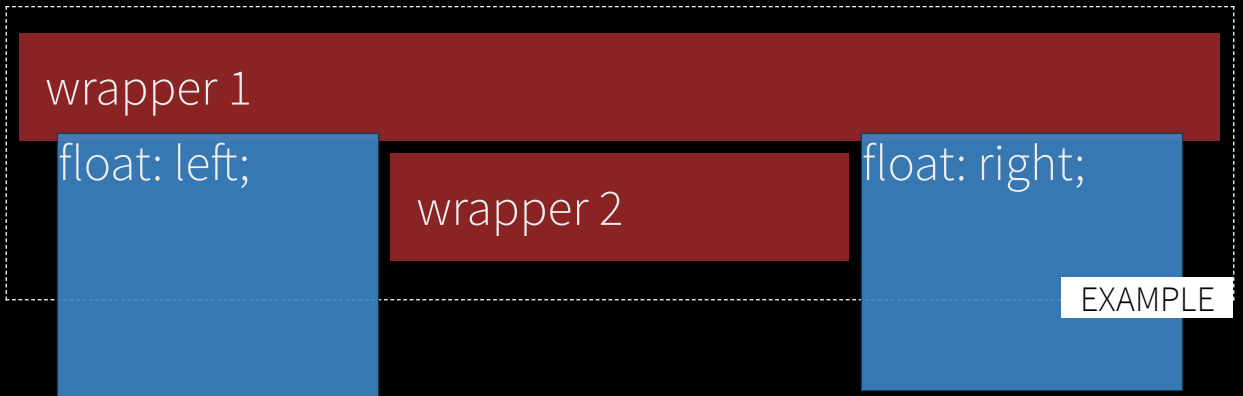
- float элементы
- position: absolute | fixed;
- display: inline-block | table-cell | table-captions (не блочные элементы)
- блочные элементы с overflow: hidden | auto;

```
.wrapper1 {
    overflow: hidden;
}
```



EXAMPLE

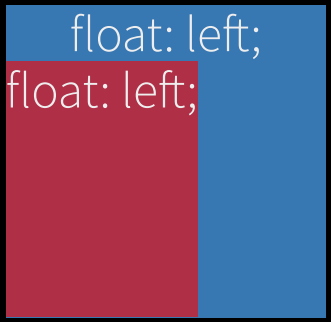
```
.wrapper2 {
    overflow: hidden;
}
```



```
<div class="float-left">float: left;</div>
<div class="box">box</div>

.box {
   overflow: hidden;
}
```

Что будет?

```
<div class="float-left">
    float: left;
    <div class="float-left">float: left;</div>
</div>
```



EXAMPLE

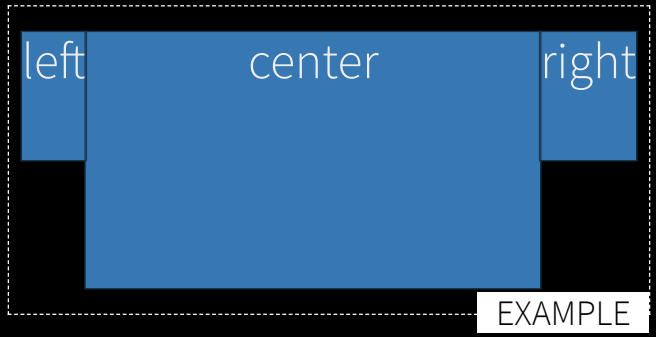
Меняем порядок

```
<div>b1</div>
<div>b2</div>
<div>b3</div>
<div>b4</div>

div { float: right }
```

Трехколоночный макет

```
<div class="left">left</div>
<div class="right">right</div>
<div class="center">center</div>
```

```
.left { float: left }
.right { float: right }
.center { overflow: hidden }
body { width: 50% }
```



Спасибо