

Модель отображения

Часть II

Сегодня в программе

1. Позиционирование элементов
2. Наложение элементов
3. Плавающие элементы

Позиционирование элементов

Свойство **position** устанавливает способ позиционирования элемента относительно окна браузера или других объектов на веб-странице

Значения

```
div {  
    position: static; /* Значение по умолчанию  
    position: relative;  
    position: absolute;  
    position: fixed;  
    position: sticky;  
}
```

`position: relative;`

Положение бокса вычисляется
относительно позиции, которую он занимал
бы в нормальном потоке

Смещение задается свойствами:
top, right, bottom, left

Относительное позиционирование

```
div {  
    position: relative;  
    top: 20px;  
    left: 20px;  
}
```

Относительное позиционирование

top

right

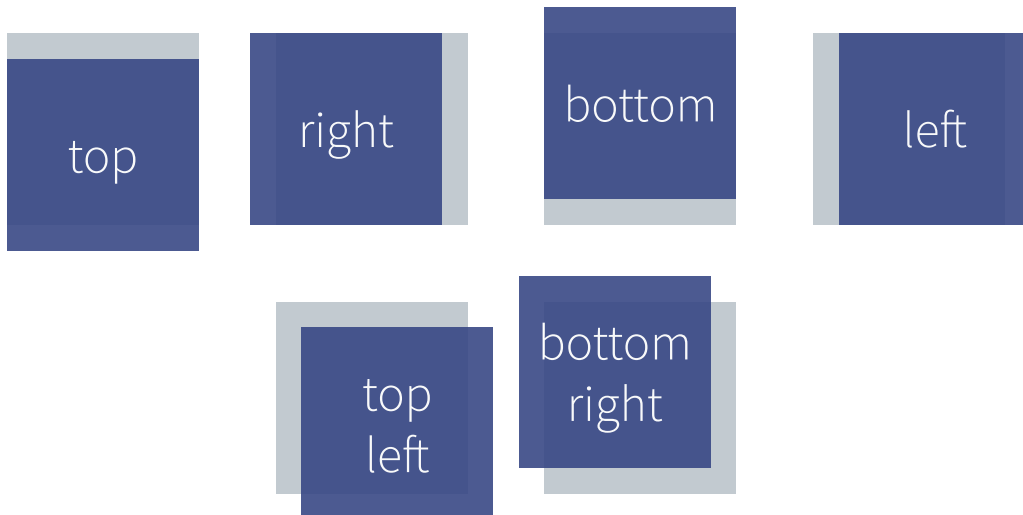
bottom

left

top
left

bottom
right

Относительное позиционирование



Относительное позиционирование

```
.box1 {  
  top: 20px;  
  bottom: 20px;  
}
```

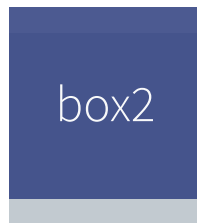
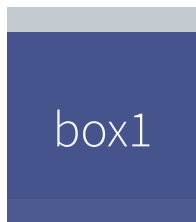
```
.box2 {  
  top: -20px;  
  bottom: -20px;  
}
```



Относительное позиционирование

```
.box1 {  
  top: 20px;  
  bottom: 20px;  
}
```

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



Относительное позиционирование

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

```
.parent {  
  dir: rtl; /* ltr */  
}
```

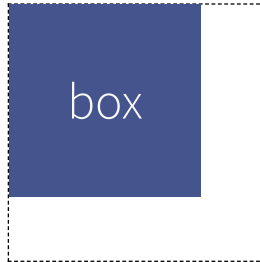
left to
right

يمين إلى
يسار

Относительное позиционирование

```
.parent {  
  width: 200px;  
  height: 200px;  
}
```

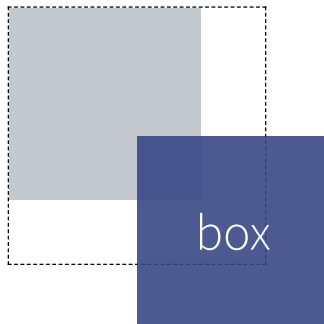
```
.box {  
  top: 50%;  
  left: 50%;  
}
```



Относительное позиционирование

```
.parent {  
  width: 200px;  
  height: 200px;  
}
```

```
.box {  
  top: 50%;  
  left: 50%;  
}
```



Как считаются размеры?

`width: auto;` Все пространство

`width: 50%;` От ширины родителя

`height: auto;` По содержимому

`height: 50%;` От высоты родителя, если его высота отлична от auto

`margin: 50%;` От ширины родителя

`padding: 50%;` От ширины родителя

Содержащий блок

Прямоугольник, относительно которого считается позиция и размеры элемента

Содержащий блок

1. Для корневых элементов - прямоугольник размерами с вьюпорт
2. Для элементов с `position: relative/static` - границы контента ближайшего родителя блочного уровня
3. Относительно спозиционированный блок создает новый содержащий блок

`position: absolute;`

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

`position: absolute;`

Содержащим блоком является

ближайший родитель с

`position ≠ static`

`position: absolute;`

Все абсолютно позиционированные
элементы имеют свойство

`display` в значении `block`

`position: absolute;`

У абсолютно спозиционированных
элементов **не схлапываются** внешние
отступы

Абсолютное позиционирование

```
<div class="parent">  
  parent  
  <div class="box">box</div>  
</div>
```



parent

box

Абсолютное позиционирование

```
<div class="parent">  
  parent  
  <div class="box">box</div>  
</div>
```

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

parent

box

Абсолютное позиционирование

```
<div class="parent">  
  parent  
  <div class="box">box</div>  
</div>
```

```
.box {  
  position: absolute;  
  top: 0;  
  left: 20px;  
}  
  
.parent {  
  position: relative;  
}
```


Абсолютное позиционирование

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

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

box1

box2

Абсолютное позиционирование

```
.box {  
  position: absolute;  
  top: 10px;  
  right: 10px;  
  bottom: 10px;  
  left: 10px;  
}
```

boxent

Абсолютное позиционирование

```
.box {  
  position: absolute;  
  top: 10px;  
  right: 10px;  
  bottom: 10px;  
  left: 10px;  
  
  height: 100px;  
}
```

boxent

Абсолютное позиционирование

```
.box {  
  position: absolute;  
  top: 10px;  
  right: 10px;  
  bottom: 10px;  
  left: 10px;  
  
  width: 100px;  
}
```

boxent

Абсолютное позиционирование

```
.box {  
  position: absolute;  
  top: 10px;  
  right: 10px;  
  bottom: 10px;  
  left: 10px;  
  
  margin: 15px;  
}
```

parent
box

Абсолютное позиционирование

Значения можно задавать в %. Тогда они будут рассчитаны относительно размеров содержащего блока

Абсолютное позиционирование

```
.box {  
  position: absolute;  
  top: 50%;  
  left: 50%;  
}
```

parent

box

Абсолютное позиционирование

```
.box {  
  position: absolute;  
  right: 0;  
  bottom: 100%;  
}
```

box

parent

Как считаются размеры?

`width: auto;` По содержимому или координатам
`height: auto;`

`width: 50%;` От ширины/высоты содержащего блока
`height: 50%;`

`margin: 50%;` От ширины содержащего блока

`padding: 50%;` От ширины содержащего блока

```
position: fixed;
```

Содержащий блок у fixed элементов — это всегда вьюпорт. **Всегда**

Фиксированное позиционирование

```
<div class="parent">  
  parent  
  <div class="box">fixed</div>  
</div>
```

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

Пример

`position: sticky;`

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

«Липкое» позиционирование

```
<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

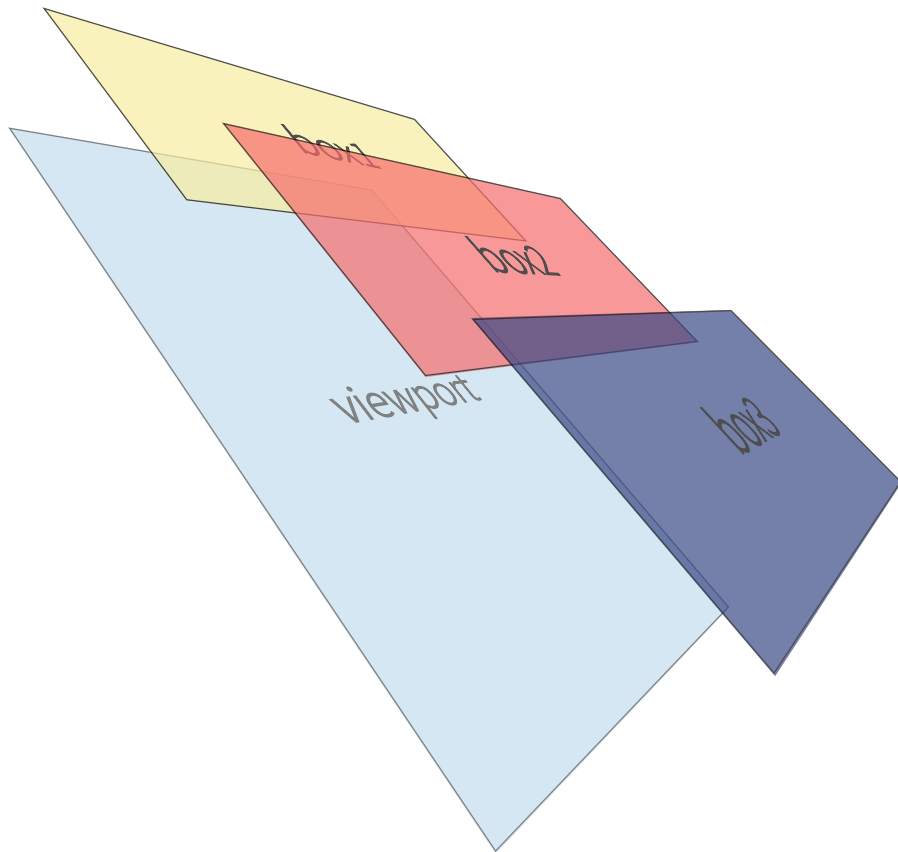
Наложение элементов

Если элементы находятся в равных условиях, то элемент, который находится ниже в потоке, будет **выше** по z-оси

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

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

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

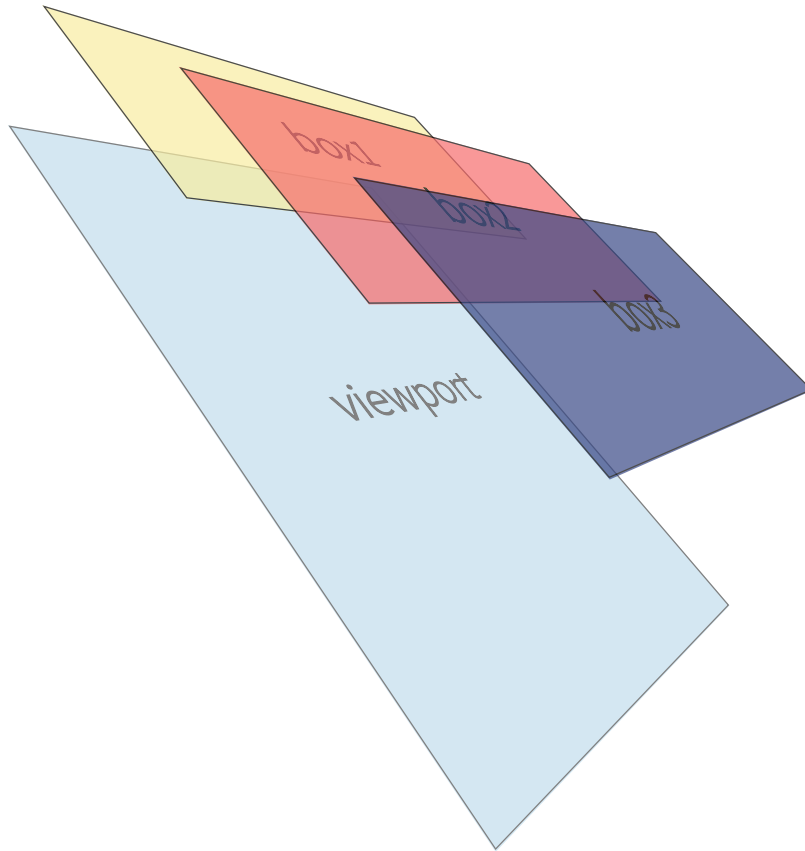


```
<div class="box1">box1</div>
```

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

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

```
.box2, .box3 {  
    margin-top: -50px;  
}
```



z-index

Благодаря этому свойству каждый элемент может находиться как ниже, так и выше других элементов

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

Контекст наложения

Элементы с общим родителем,
перемещающиеся на передний или задний
план вместе

Условия создания контекста наложения

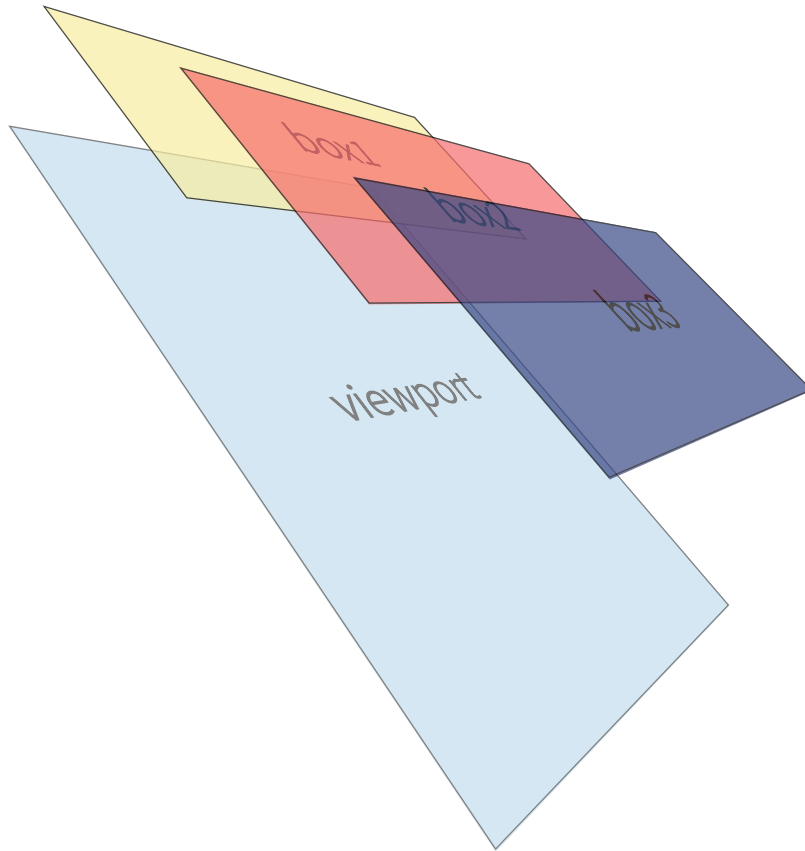
1. Если элемент — корневой элемент документа (html)
2. Если элемент имеет свойства `position` \neq `static` и `z-index` \neq `auto`
3. Если элемент имеет свойство `opacity` < 1

Нельзя расположить элемент из одного контекста наложения между элементами другого контекста

Порядок наложения

7. Позиционированные элементы (и их потомки) с $z\text{-index} > 0$
6. Позиционированные элементы (и их потомки) с $z\text{-index} = 0$ или `auto`, $opacity < 1$
5. Элементы `inline` уровня
4. Плавающие элементы
3. Элементы блочного уровня в нормальном потоке
2. Позиционированные элементы (и их потомки) с $z\text{-index} < 0$
1. `Background` и `border` элемента

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

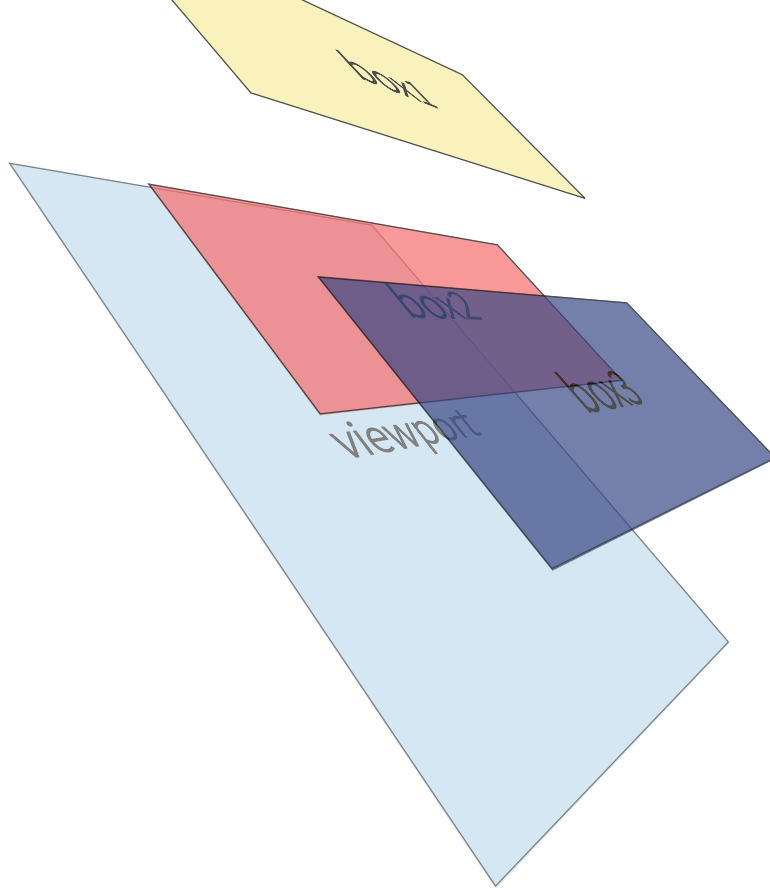


```
<div class="box1">box1</div>
```

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

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

```
.box1 {  
    position: relative;  
}
```



Порядок наложения

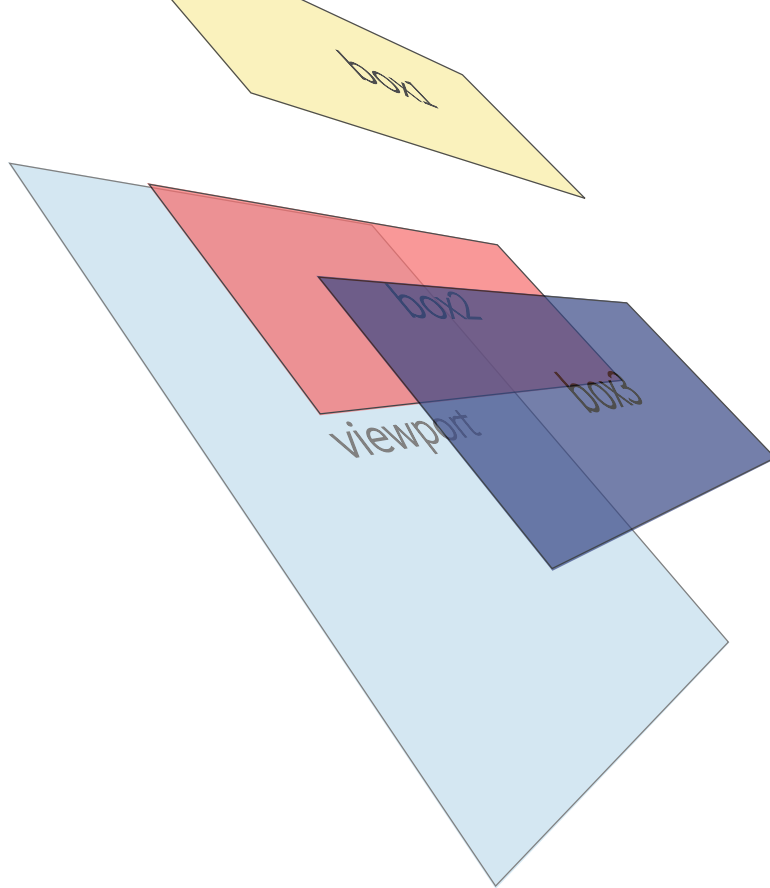
7. Позиционированные элементы (и их потомки) с $z\text{-index} > 0$
6. Позиционированные элементы (и их потомки) с $z\text{-index} = 0$ или `auto`, `opacity < 1`
5. Элементы `inline` уровня
4. Плавающие элементы
3. Элементы блочного уровня в нормальном потоке
2. Позиционированные элементы (и их потомки) с $z\text{-index} < 0$
1. `Background` и `border` элемента

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

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

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

```
.box1 {  
    opacity: .99;  
}
```

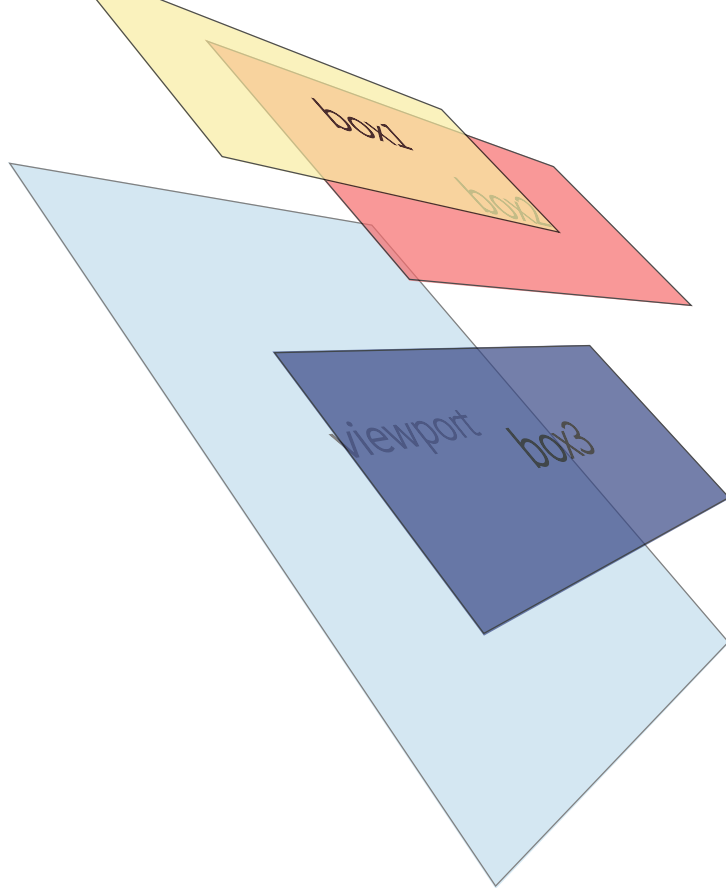


Порядок наложения

7. Позиционированные элементы (и их потомки) с $z\text{-index} > 0$
6. Позиционированные элементы (и их потомки) с $z\text{-index} = 0$ или `auto`, $opacity < 1$
5. Элементы `inline` уровня
4. Плавающие элементы
3. Элементы блочного уровня в нормальном потоке
2. Позиционированные элементы (и их потомки) с $z\text{-index} < 0$
1. `Background` и `border` элемента

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

```
.box1 {  
    position: relative;  
    z-index: 0;  
}  
.box2 {  
    position: relative;  
    z-index: 1;  
}
```

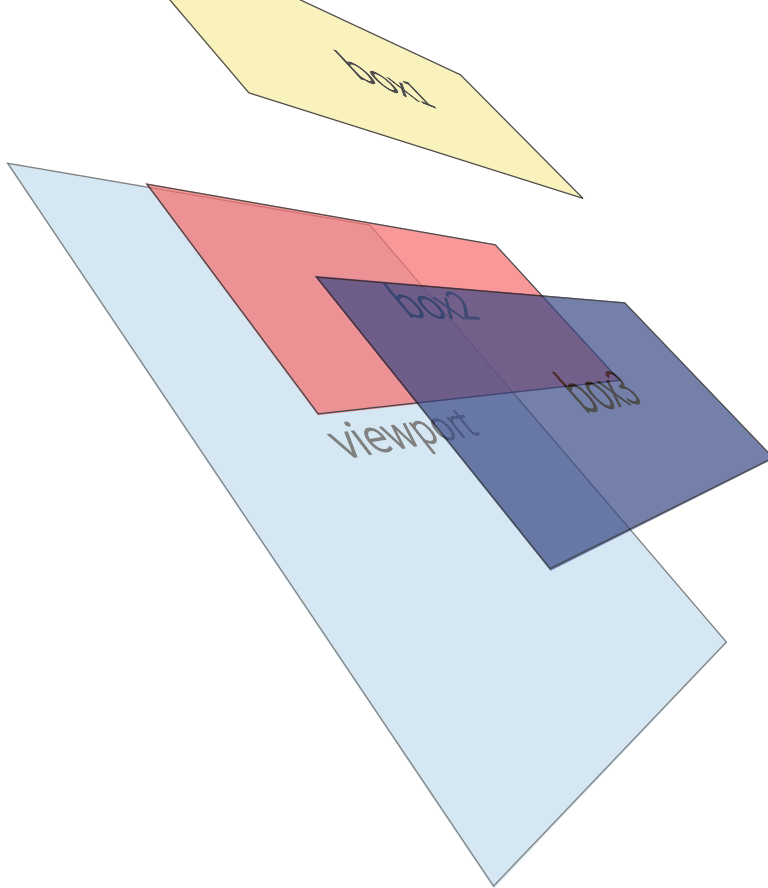


Порядок наложения

7. Позиционированные элементы (и их потомки) с $z\text{-index} > 0$
6. Позиционированные элементы (и их потомки) с $z\text{-index} = 0$ или `auto`, $\text{opacity} < 1$
5. Элементы `inline` уровня
4. Плавающие элементы
3. Элементы блочного уровня в нормальном потоке
2. Позиционированные элементы (и их потомки) с $z\text{-index} < 0$
1. `Background` и `border` элемента

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

```
.box1 {  
    position: relative;  
    z-index: 0;  
}  
.box2 {  
    position: relative;  
    z-index: -1;  
}
```

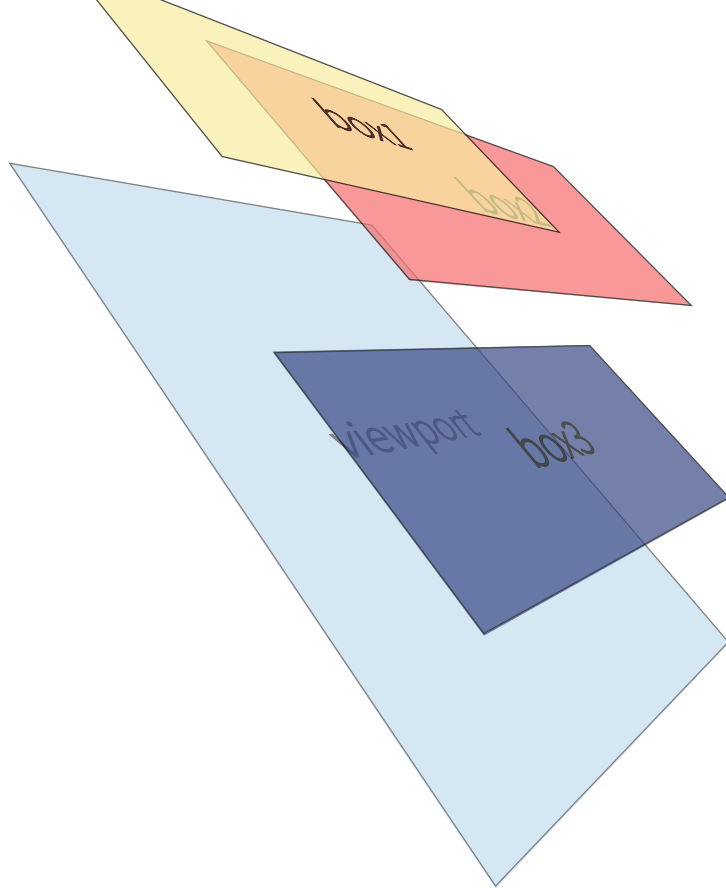


Порядок наложения

7. Позиционированные элементы (и их потомки) с $z\text{-index} > 0$
6. Позиционированные элементы (и их потомки) с $z\text{-index} = 0$ или `auto`, $opacity < 1$
5. Элементы `inline` уровня
4. Плавающие элементы
3. Элементы блочного уровня в нормальном потоке
2. Позиционированные элементы (и их потомки) с $z\text{-index} < 0$
1. `Background` и `border` элемента

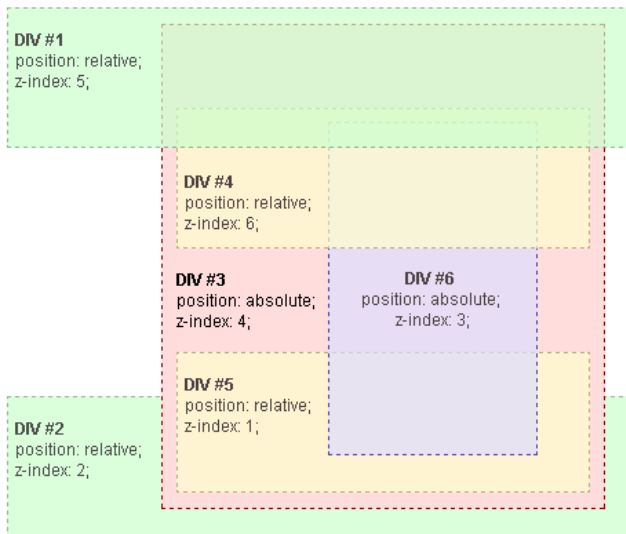
```
<div class="wrap">  
  <div class="box1">box1</div>  
</div>  
<div class="box2">box2</div>  
<div class="box3">box3</div>
```

```
.wrap {  
  opacity: .99;  
}  
.box1 {  
  position: relative;  
  z-index: 9999;  
}  
.box2 {  
  position: relative;  
  z-index: 0;  
}
```

Порядок наложения

7. Позиционированные элементы (и их потомки) с $z\text{-index} > 0$
6. Позиционированные элементы (и их потомки) с $z\text{-index} = 0$ или `auto`,
`opacity < 1`
5. Элементы `inline` уровня
4. Плавающие элементы
3. Элементы блочного уровня в нормальном потоке
2. Позиционированные элементы (и их потомки) с $z\text{-index} < 0$
1. `Background` и `border` элемента





z-index hell

Группировка z-index

1. 0000 - 1999: Основной контент
2. 2000 - 2999: Выпадающие списки
3. 3000 - 3999: Навигация
4. 4000 - 4999: Шапка / Подвал
5. ...
6. 8000 - 8999: Модальные окна
7. 9000 - 9999: Уведомления

Плавающие элементы

Свойство float

```
.box {  
    float: left; /* right */  
}
```

Элемент вынимается из потока и сдвигается влево (для left) или вправо (для right) до того как коснется либо границы родителя, либо другого элемента с float

СВОЙСТВО float

```
.box {  
    float: left; /* right */  
}
```

inline элементы «знают» о float и обтекают элемент по сторонам

Свойство float

```
.box {  
    float: left; /* right */  
}
```

Ширина float блока определяется по
содержимому

Свойство float

```
.box {  
  float: left; /* right */  
}
```

Создают новый блочный контекст
форматирования

Свойство float

```
.box {  
    float: left; /* right */  
}
```

Отступы float элементов не схлапываются с отступами соседей

inline элементы обтекают float'ы

```
<div class="box">  
  <div class="left">Left</div>
```

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

```
  <div class="right">Right</div>  
</div>
```

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

inline элементы обтекают float'ы

Left

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

Right

Где 4-ый будет?

1

2

3

Где 4-ый будет?

1

2

3

4

Где 4-ый будет?

1

2

3

4

Где 4-ый будет?

1

2

3

4

Где 4-ый будет?

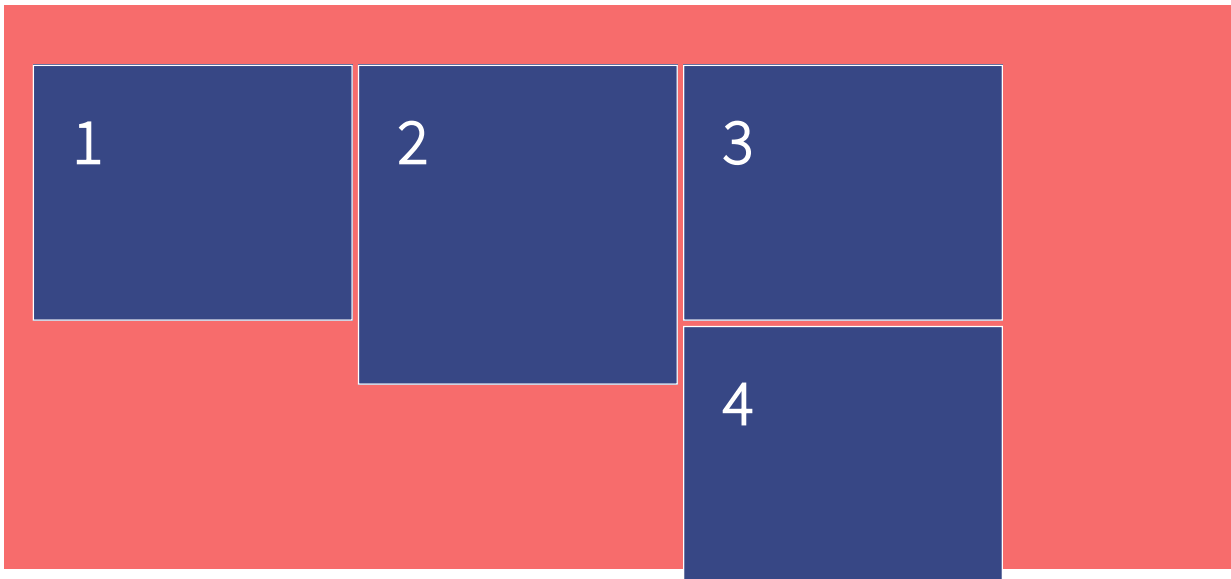
1

2

3

4

Где 4-ый будет?



Где 4-ый будет?

1

2

3

4

Где 4-ый будет?

1

2

3

4

Где 4-ый будет?

1

2

3

4

Где 4-ый будет?

1

2

3

4

Свойство clear

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

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


```
.box2 {  
  clear: left;  
}
```



```
.box2 {  
  clear: left;  
}
```



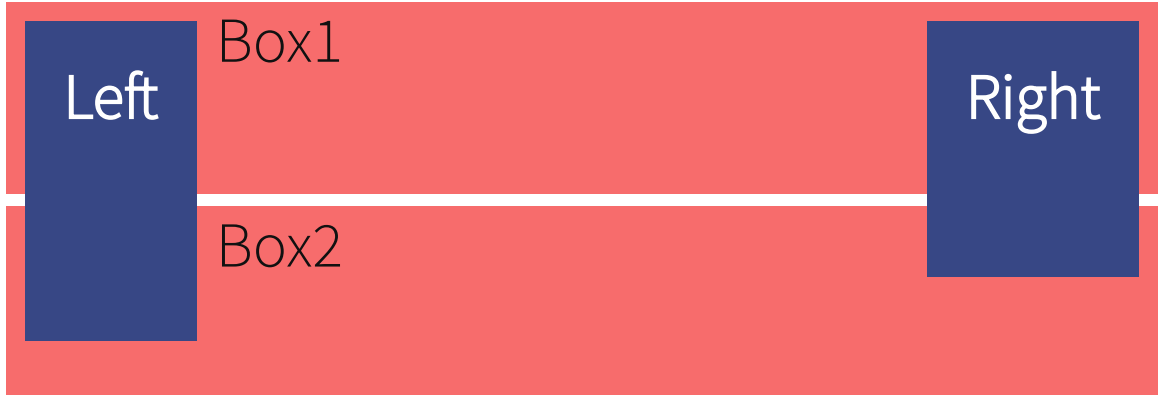
```
.box2 {  
  clear: right;  
}
```



```
.box2 {  
  clear: right;  
}
```



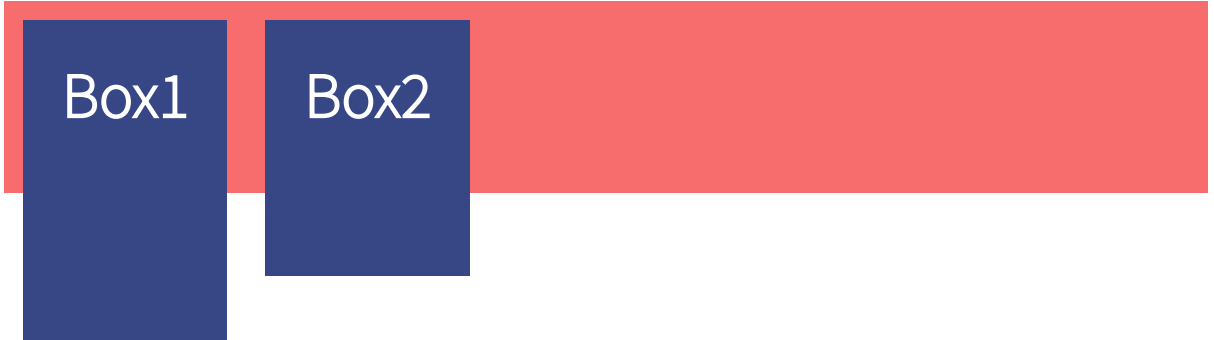
```
.box2 {  
  clear: both;  
}
```



```
.box2 {  
  clear: both;  
}
```



clearfix

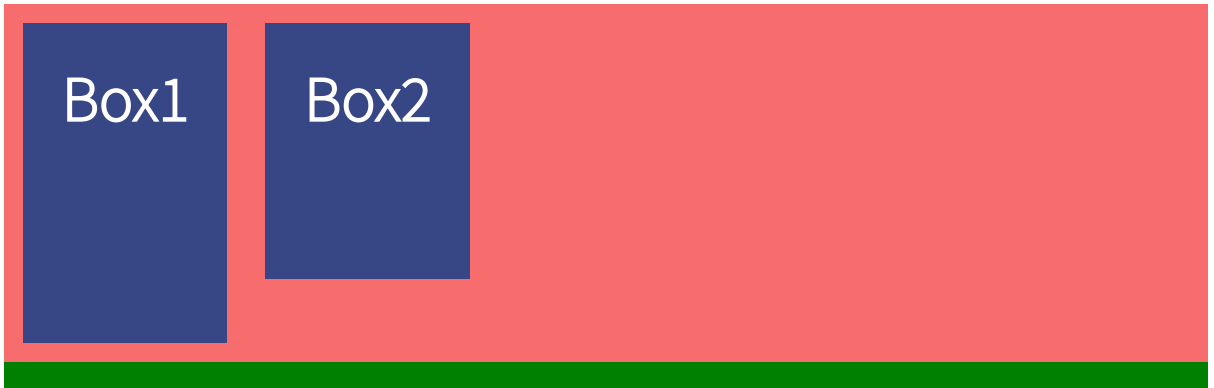


```
.clearfix::after {  
    content: '';  
    display: table;  
    clear: both;  
}
```


clearfix



clearfix



Блочный контекст форматирования

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

Способы создания нового блочного контекста форматирования

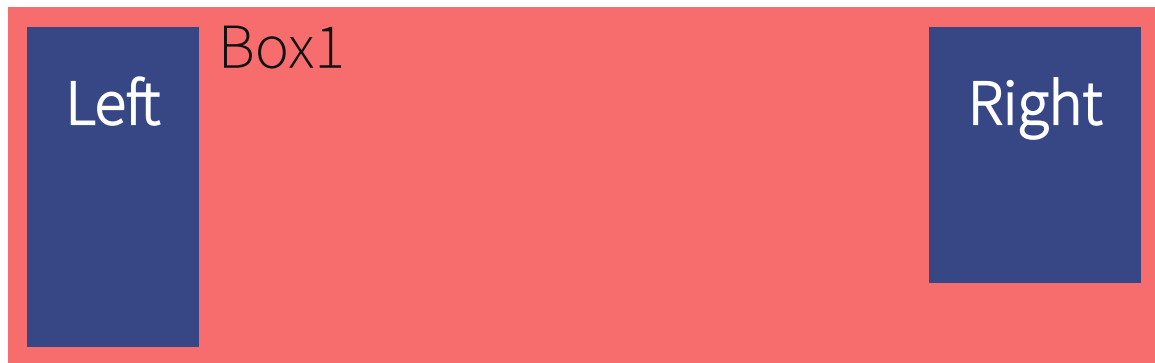
1. float элементы
2. position: absolute | fixed
3. display: inline, table... (не block)
4. блочные элементы с overflow: hidden | auto

Block formatting contexts

```
.box1 {  
    overflow: hidden;  
}
```



```
.box1 {  
    overflow: hidden;  
}
```



```
.box2 {  
    overflow: hidden;  
}
```



```
.box2 {  
    overflow: hidden;  
}
```



Трёхколоночный макет

```
<div class="left">Left</div>  
<div class="right">Right</div>  
<div class="center">Center</div>
```

Left

Right

Center

Трёхколоночный макет

```
<div class="left">Left</div>  
<div class="right">Right</div>  
<div class="center">Center</div>
```

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



Left Right

Center

Трёхколоночный макет

```
<div class="left">Left</div>  
<div class="right">Right</div>  
  
<div class="center">Center</div>
```

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



Трёхколоночный макет

```
<div class="left">Left</div>  
<div class="right">Right</div>  
<div class="center">Center</div>
```

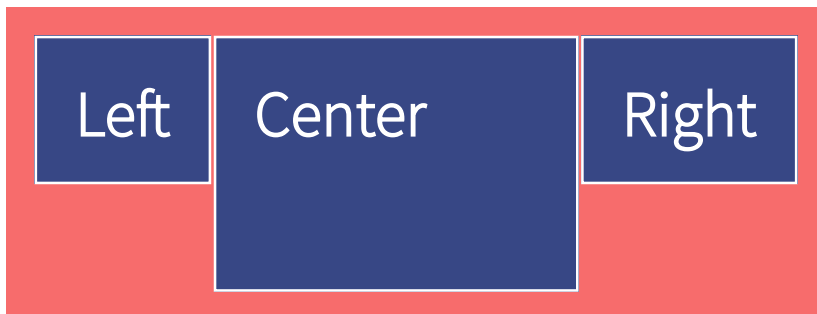
```
.left { float: left; }  
.right { float: right; }  
.center { overflow: hidden; }
```



Трёхколоночный 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%; }
```



Ссылки

Про содержащий блок

Про блочный контекст форматирования

Про z-index

Про float