

# Front-end

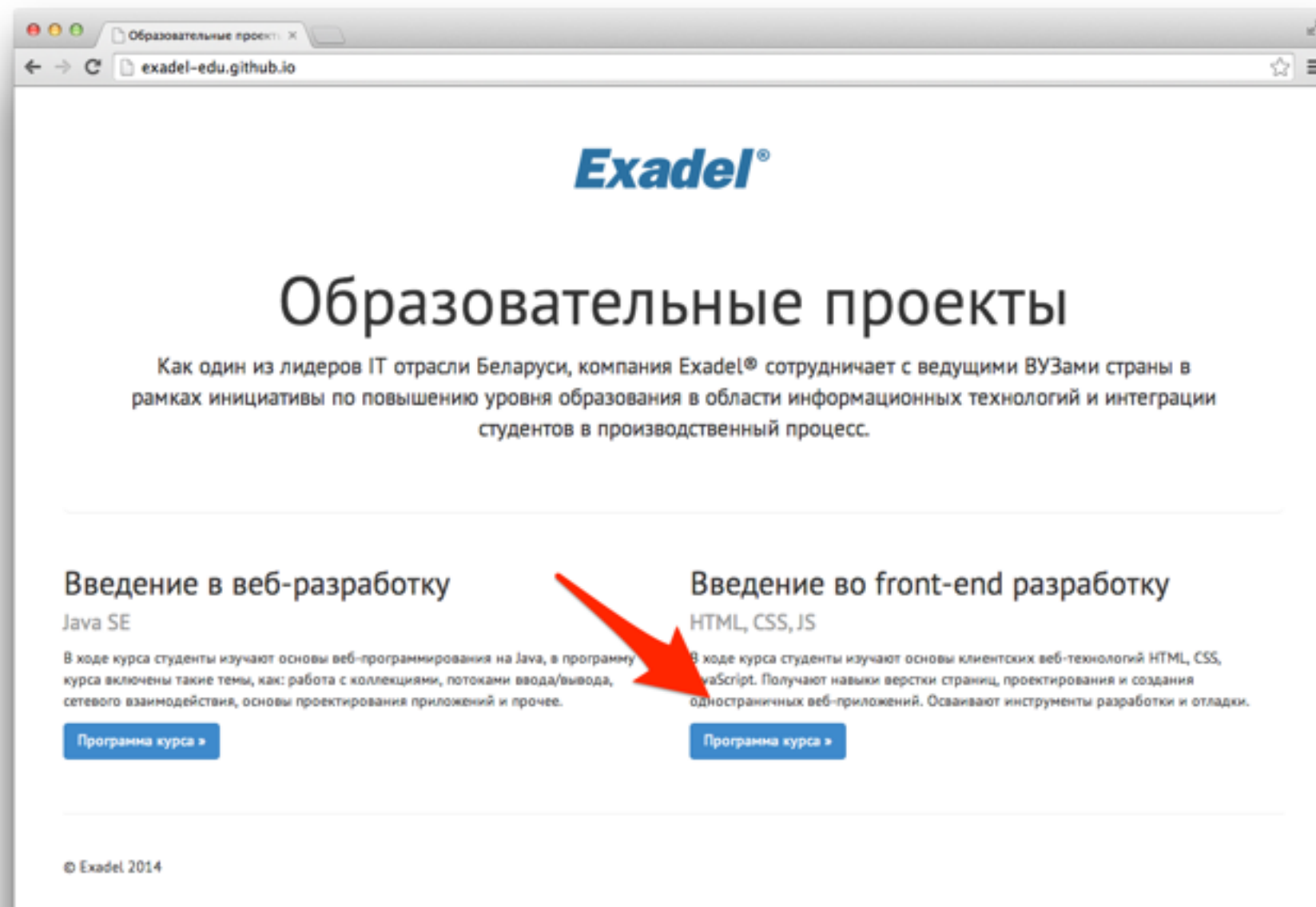
---

dive-in

# CSS

***Exadel®***

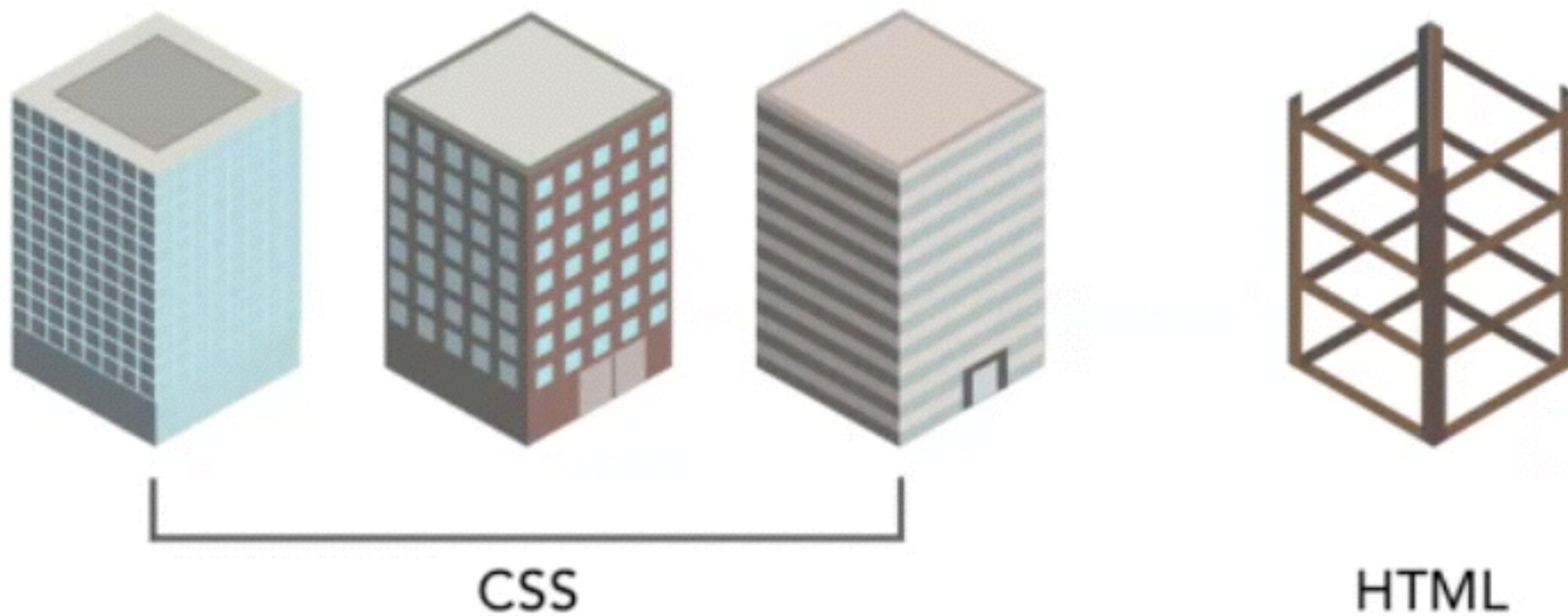
# Где раздобыть учебные материалы?



<http://exadel-edu.github.io>

# Что такое CSS

## Cascading Style Sheets



# Способы включения в документ

---

Ссылкой на внешний документ

```
<link rel="stylesheet" href="style.css">
```

Тэг <style>

```
<style>  
    /* CSS код */  
</style>
```

Inline

```
<div style="color:red;"></div>
```

# Синтаксис

---

```
div {  
    color    : red;  
    border   : 1px solid black;  
}
```

# Синтаксис

---

CSS правило

```
div {  
    color    : red;  
    border   : 1px solid black;  
}
```

# Синтаксис

---

селектор

**div**

{

color : red;

border : 1px solid black;

}

# Синтаксис

---

```
div {  
    декларация  
    color : red;  
    border : 1px solid black;  
}
```



# Синтаксис

---

```
div {  
    свойство color значение : red;  
    border : 1px solid black;  
}
```

# Селекторы | базовые

---

\*

element

.className

element[attr]

#id

\*

div

.super-duper

img[alt]

#superDuper

# Селекторы | комбинация

---

— div span

, div, span

> #super > span

+ #super + div

# Селекторы | псевдоклассы

---

`:first-child`

`:visited`

`:hover`

`p:first-child`

`a:visited`

`a:hover`

# Вес селекторов

---

Свойство, на основе которого браузер принимает решение в какой последовательности применить стили, если рассматриваемый элемент попадает сразу под несколько правил.

```
div span { color:red; }  
span.piu { color:blue; }
```

```
<div>  
  <span class="piu">  
    some text  
  </span>  
</div>
```

# Вес селекторов

---

Свойство, на основе которого браузер принимает решение в какой последовательности применить стили, если рассматриваемый элемент попадает сразу под несколько правил.

```
div span { color:red; }  
span.piu { color:blue; }
```

```
<div>  
  <span class="piu">  
    some text  
  </span>  
</div>
```

# Числовое представление | вес селекторов

Каждый селектор можно привести к набору из восьми десятичных значений, который однозначно определяет его вес.

0, 0, 0, 0      0, 0, 1, 2



- ТЭГИ
- классы, псевдоклассы, селекторы атрибутов
- id
- inline стили

# Примеры | вес селекторов

<code>div p .piu</code>	<code>0,0,0,0</code>	<code>0,0,1,2</code>
-------------------------	----------------------	----------------------

<code>p.piu.wiu.disco</code>	<code>0,0,0,0</code>	<code>0,0,3,1</code>
------------------------------	----------------------	----------------------

<code>div#piu</code>	<code>0,0,0,0</code>	<code>0,1,0,1</code>
----------------------	----------------------	----------------------

<code>div {   color: red !important; }</code>	<code>0,0,0,1</code>	<code>0,0,0,0</code>
---	----------------------	----------------------



# Единицы измерения

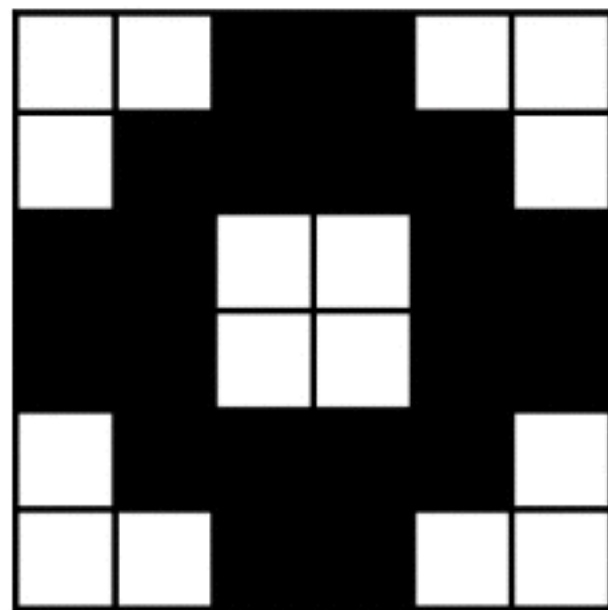
---

Речь о единицах измерения, применяемых для указания размеров чего-либо.

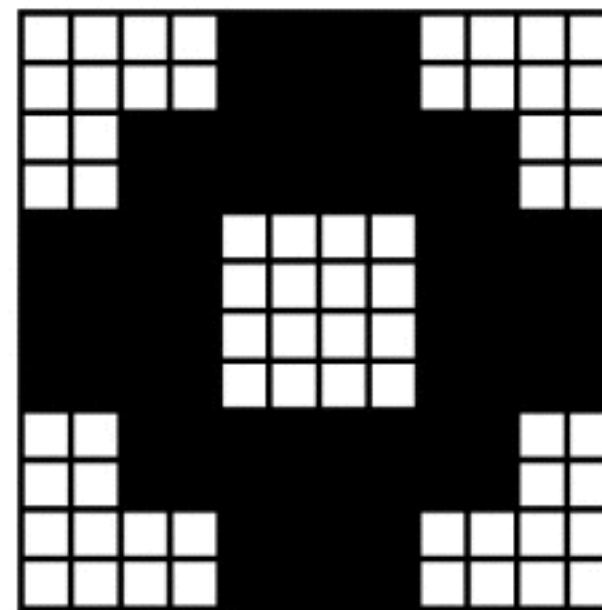
- абсолютные (in, cm, mm, pt)
- относительные (px, em, rem, ex, pc, ..)

Вот это поворот!

Пиксель — относительная величина.



1024 x 768



1440 x 900

# ЕМ | единицы измерения

---

Единица измерения, привязанная к размеру шрифта.

По умолчанию **1em = 16px**

Размеры чего-либо, заданные в em будут вычисляться **относительно** размера шрифта элемента.

```
div {  
    font-size      : 2 em;  
    margin-bottom  : 1 em;  
}
```

# REM (root em) | единицы измерения

---

Как **em**, только не ломатая.

# % | единицы измерения

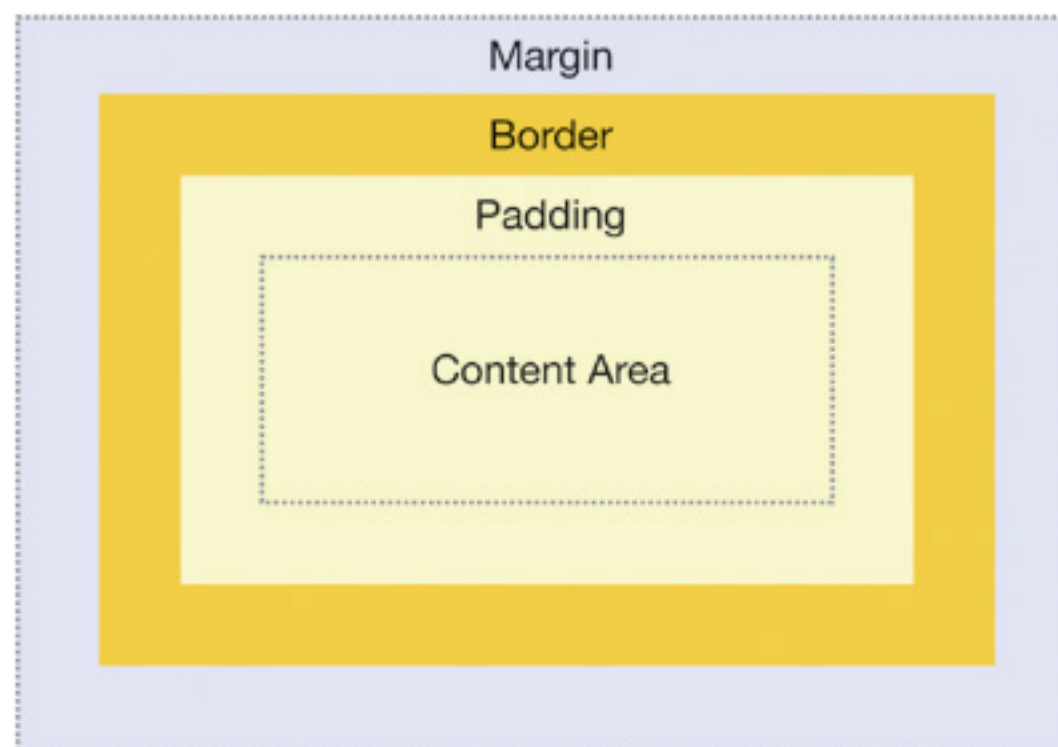
---

Величина равная заданной доле от соответствующего свойства контейнера.

При изменении свойств родительского элемента будут меняться и значения у его потомков.

# Блочная модель

Термин, используемый для описания правил, по которым вычисляются геометрические свойства всех элементов дерева отображения на этапах компоновки и отрисовки.



# Блочная модель

---

Для того, чтобы скомпоновать блок, нужно определить 4 его характеристики:

- ТИП
- схему позиционирования
- размеры
- внешнюю информацию (размер экрана)

# Тип элемента | блочная модель

---

- блочный
- строчный



# Схема позиционирования | блочная модель

---

## **Стандартная**

объект размещается в документе согласно своему положению в дереве отображения и в дереве DOM, а также своему типу и размерам окна

## **Плавающая**

объект сначала компануется по стандартной схеме, затем смещается в крайнее правое или крайнее левое положение

## **Абсолютная**

положение объекта в дереве отображения отличается от его положения в дереве DOM

# z-index | блочная модель

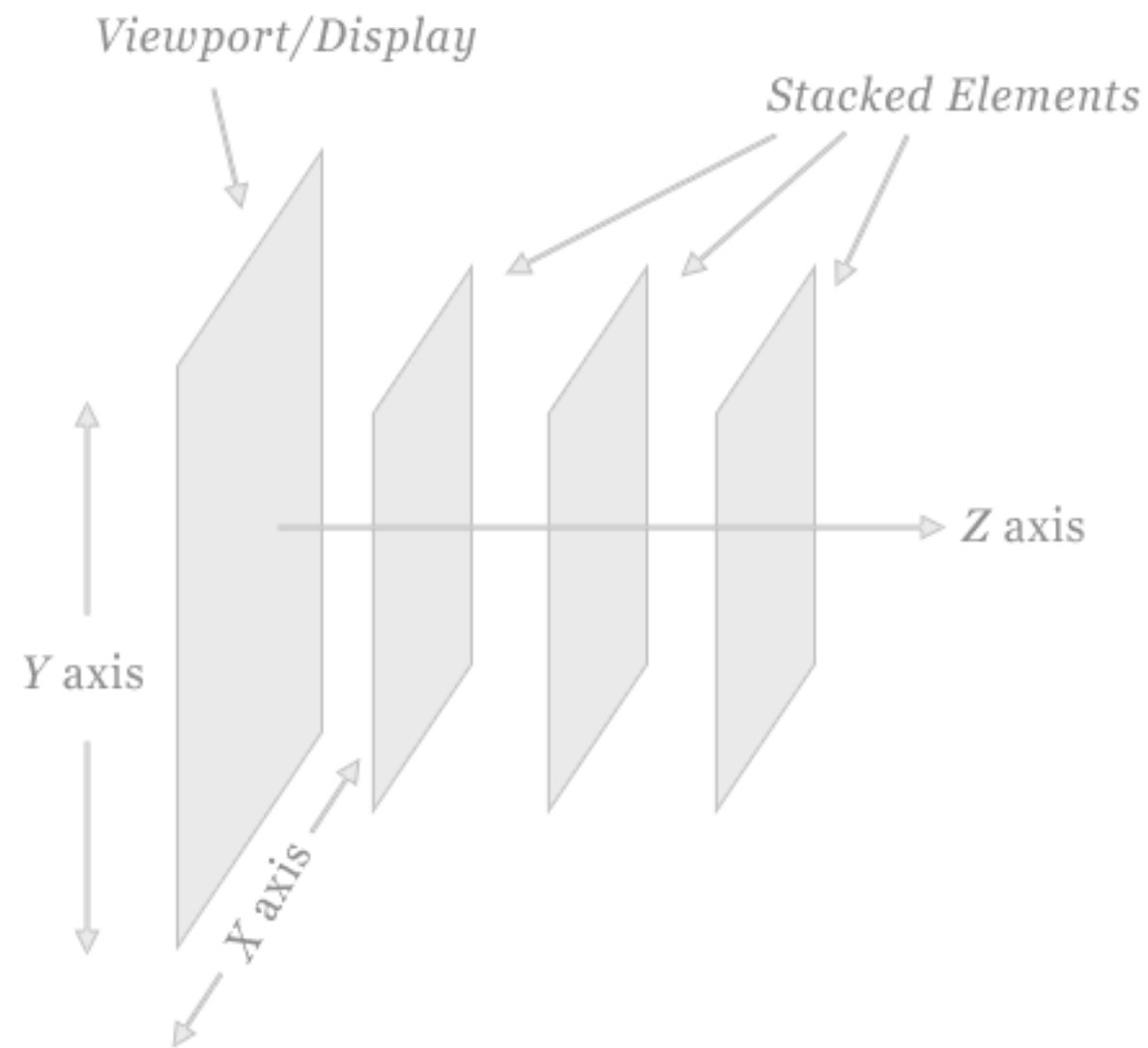
---

Данное CSS-свойство характеризует «координату» элемента в третьем измерении документа.

Применимо к элементам, для которых:

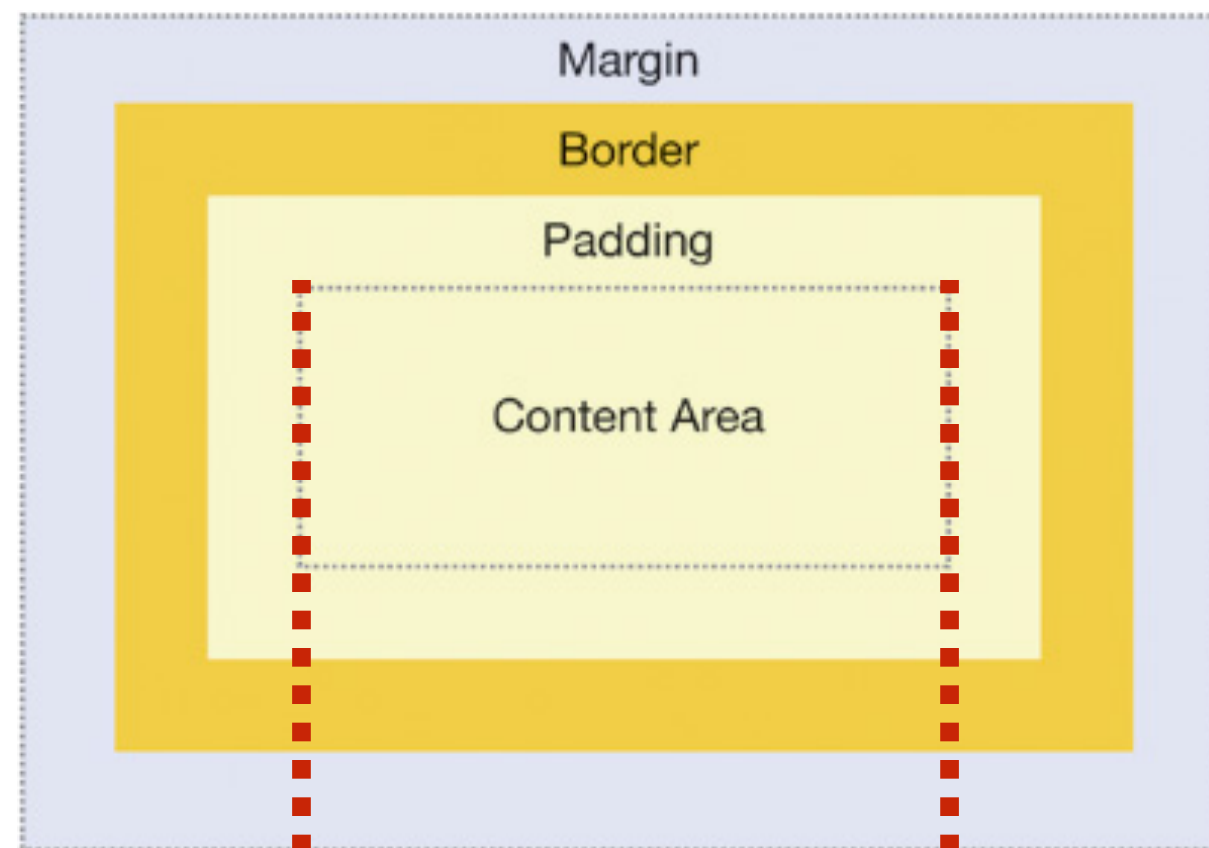
- свойство `position` отлично от `static`
- свойство `float` отлично от `none`

# z-index | блочная модель



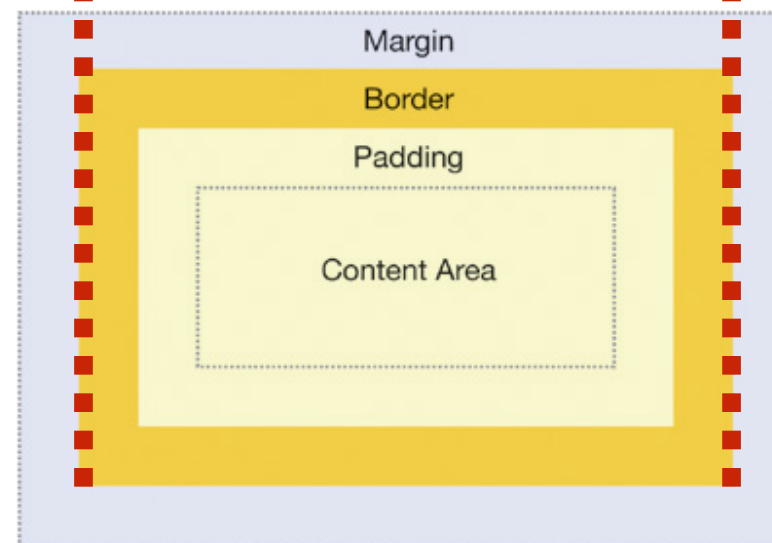
# Варианты блочной модели | блочная модель

W3C



width

IE



# Стили по-умолчанию

---

Не забываем их обнулять.

- reset CSS
- normalize CSS

# Media types

---

Позволяют выбрать тип устройства, для которого применить тот или иной стиль:

- all
- aural
- braille
- embossed
- handheld
- print
- projection
- screen
- tty
- tv

# Media types

---

@media

```
@media screen{  
    // тут правила  
}
```

@import

```
@import url("projection.css") projection;
```

link media attribute

```
<link rel="stylesheet"  
      href="print.css"  
      media="print">
```

# Media queries

---

СИНТАКСИС

`[not | only] screen [and] (expression)`



# Примеры | media queries

```
@media only screen and (-webkit-min-device-pixel-ratio: 1.5),  
       only screen and (min-resolution: 144dpi) { ... }
```

```
@media only screen and (min-device-width : 768px)  
       and (max-device-width : 1024px)  
       and (orientation : portrait) {  
    /* Styles or @import */  
}
```

```
@import url(small.css) (min-width : 400px);
```

```
<link rel="stylesheet"  
      href="css/ipad.css"  
      media="screen and (min-device-width: 768px)  
            and (max-device-width: 1024px)" />
```

# The End

---

учебные материалы и практическое задание

<http://exadel-edu.github.io>

**Exadel®**