

Тема 9. Тег **DIV** и **SPAN**.  
**CSS**. Продолжение

# СОДЕРЖАНИЕ

---

1. Теги DIV и SPAN
2. Свойства и значения CSS
3. Вёрстка сайта с использованием тега DIV
4. Новые теги для верстки в HTML5
5. Примеры



# ОБЩАЯ ИНФОРМАЦИЯ О ТЕГАХ «**DIV**» и «**SPAN**»

---

- Теги `<div>` и `<span>`, обычно не используются, в одиночку, имея особенные эффекты если они используются в комбинации с другими HTML-тегами и стилями CSS
  - Тег **DIV** создает разделы блоков на странице, формой и графикой, которых можно управлять по отдельности
  - Тег **DIV** имеет атрибут **align** (для выравнивания по горизонтали), который не поддерживается HTML5
  - Тег **SPAN** создает возможность изменения части контекста (in-line); может быть использован в качестве "класса" с CSS
  - Оба тега могут использовать глобальный атрибут **style** или **id** или **class**
- 



# ТЕГ „DIV”

---

- В последнее время, один из самых используемых HTML-тегов
- Поддерживается всеми браузерами
- Используется для группировки HTML-элементов
- В тег DIV- можно включить любые HTML элементы: таблицы, списки, формы, параграфы текста или другие DIV-ы
- Кадр в котором включен DIV может иметь собственный фон (background), ширину, высоту, края с различными линиями и т.д.
- Синтаксис: **`<div>содержимое</div>`**



# ПРИМЕР ИСПОЛЬЗОВАНИЯ „DIV”

---

```
<html>
<head><title>Pagina cu sectiuni</title>
<link href="stil_exemplu.css" rel="stylesheet" type="text/css" />
</head>
<body>
  <div id="cap"><h1>IMAGINI</h1></div>
  <hr />
  <div id="corp">
    
    
    
    
  </div>
  <hr />
  <div id="coada"><h1>Priviti cu placere</h1></div>
</body>
</html>
```

---

# ФАЙЛ СО СТИЛЯМИ

```
#cap{
    background-color:black;
    color:white;
    text-align:center;
    padding:5px;
    font-space:5px;
    font-size:30px;
    text-shadow: 5px 5px 7px #ffcccc;}
#corp{ height:46%; width:100%; padding:5px; float: left; }
#coada{
    background-color:black;
    color:white;
    text-align:center;
    padding:5px;
    font-variant:small-caps;
    text-shadow: 3px 3px 5px #ffcccc;}
img{ border: 2px solid #aa0000; height: 200px; width: 235px;}
```

# РЕЗУЛТАТ ПРИМЕРА

## IMAGINI



PRIVITI CU PLACERE



# ТЕГ „SPAN”

---

- Тег `<span>` может добавить графические стили определенной части какого-то элемента (in-line)
- Слова которые необходимо выделить включаются между `<span>` и `</span>`
- Основная форма: `<span>содержание</span>`
- В файле со стилями определяются необходимые стили
- Пример:  
`span{color:#ff0000;}`

Применяются к следующим кодам:

...

```
<div id="coada">
```

```
<h1>Priviti cu placere</h1>
```

```
</div>...
```



PRIVITI CU PLACERE



# ПОРЯДОК ОТОБРАЖЕНИЯ ЭЛЕМЕНТОВ

---

- Порядок отображения элементов на странице называется *поток документа – нормальный поток документа*
- **Блочные элементы** отображаются как прямоугольные области, идущие друг за другом сверху вниз
- **Строчные элементы** располагаются сверху вниз и слева направо и при необходимости переносятся на новую строку



# СВОЙСТВА ДЛЯ РАЗМЕЩЕНИЯ HTML - ЭЛЕМЕНТОВ

---

- Свойства CSS используемые для размещения позволяют контролировать местоположение элементов на странице
  - Существует возможность размещения какого-то элемента позади другого, или указать как вывести какой-либо элемент, который имеет большое содержание
  - Элементы могут быть расположены с использованием свойств: **top**, **bottom**, **left** и **right**
  - Эти свойства не будут функциональными если свойство **position** не будет установлена первой
  - Имеет значение метод определения местоположения.
- ▶ Существуют **четыре** различных **способа** **позиционирования** HTML-элементов

# I – СТАТИЧЕСКОЕ МЕСТОПОЛОЖЕНИЕ

---

- **position: static**
- Все HTML-элементы расположены статически по умолчанию
- Статически расположенный элемент, постоянно располагается с учётом нормального потока на странице
- Статически расположенные элементы не будут затронуты свойствами **top**, **bottom**, **left** и **right**



# II – ФИКСИРОВАННОЕ МЕСТОПОЛОЖЕНИЕ

---

- **position: fixed**
  - Элемент с фиксированной позицией будет расположен в отношении с окном браузера
  - Данный элемент не будет передвигаться даже если окно прокручивается
  - Зафиксированные элементы не учитывают нормального потока вывода на странице – не имеет значение какие другие элементы существуют на странице
  - Элементы с фиксированной позицией могут накладываться на другие элементы
- 



# ПРИМЕР „position: fixed”

---

```
<html>
<head><title>Pozitii elemente</title>
<style>
div.fix {
    width:250px;
    position: fixed;
    top: 20px;
    right:20px;
p{color:brown; font-size:20px;}
</style></head>
<body>

<div class="fix"><p>Folosindu-se de vazul remarcabil ...spre pisici.</p>
</div>
</body></html>
```

---



# РЕЗУЛЬТАТ ПРИМЕРА



le din față ale pisicuteilor au 5 pernute.

Folosindu-se de vazul remarcabil, de auzul ascutit si de celelalte inzestrari incredibile de pradator fara de seama, o (aparent banala) pisica poate vana peste 1.000 de specii pentru a supravietui. In ciuda acestui fapt, felinele din specia *Felis silvestris catus*, asa cum le-au numit oamenii de stiinta, au ales sa traiasca alaturi de oameni de peste 9.500 de ani, iar in prezent, la nivel mondial, sunt cele mai populare animale de companie din lume. Iata 30 de lucruri mai putin cunoscute despre pisici.



## ...после прокрутки



În față ale pisicuțelor au 5 pernute,  
pe lăbuțele din spate - numai 4.

Folosindu-se de vazul remarcabil, de auzul ascuțit și de celelalte înzestrări incredibile de pradator fără de seamă, o (aparent banală) pisică poate vana peste 1.000 de specii pentru a supraviețui. În ciuda acestui fapt, felinele din specia *Felis silvestris catus*, așa cum le-au numit oamenii de știință, au ales să trăiască alături de oameni de peste 9.500 de ani, iar în prezent, la nivel mondial, sunt cele mai populare animale de companie din lume. Iată 30 de lucruri mai puțin cunoscute despre pisici.

# ПРИМЕР НАКЛАДКИ

---

```
div.fix {width:450px;position: fixed;  
        top: 20px; right:20px;    }
```

[Другой пример](#)



Folosindu-se de vazul remarcabil, de auzul ascutit si de celelalte inzestrari incredibile de pradator fara de srama, o (aparent banala) pisica poate vana peste 1.000 de specii pentru a supravietui. In ciuda acestui fapt, felinele din specia *Felis silvestris catus*, asa cum le-au numit oamenii de stiinta, au ales sa traiasca alaturi de oameni de peste 9.500 de ani, iar in prezent, la nivel mondial, sunt cele mai populare animale de companie din lume. Iata 30 de lucruri mai putin cunoscute despre pisici.

# III – ОТНОСИТЕЛЬНОЕ МЕСТОПОЛОЖЕНИЕ

---

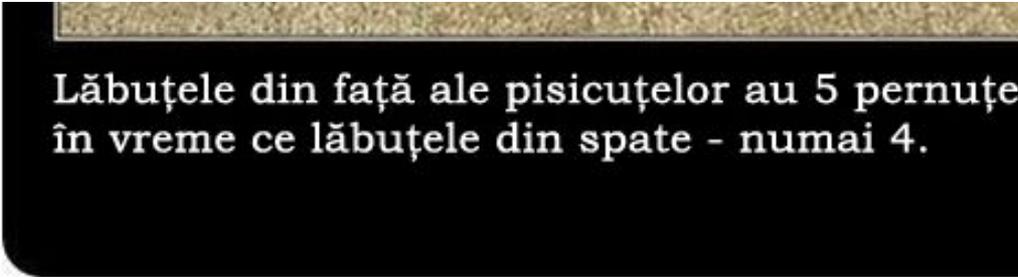
- **position: relative**
- Элемент расположенный относительно, позиционируется относительно его нормального положения
- Содержимое относительно-позиционируемых элементов может быть перемещено и может перекрывать другие элементы, но пространство, отведенное для элемента в нормальном потоке, сохраняется
- Элементы, расположенные относительно, часто используются в качестве блочных контейнеров для абсолютно позиционируемых элементов



# ПРИМЕР ОТНОСИТЕЛЬНОГО МЕСТОПОЛОЖЕНИЯ

---

div.fix {width:450px; **position: relative;** left: -20px;}



Lăbușele din față ale pisicușelor au 5 pernușe în vreme ce lăbușele din spate - numai 4.

folosindu-se de vâzul remarcabil, de auzul ascuțit și de celelalte înzestrări incredibile de pradator fără de răzămă, o (aparent banală) pisică poate vâna peste 1.000 de specii pentru a supraviețui. În ciuda acestui fapt, călinele din specia *Felis silvestris catus*, așa cum le-au numit oamenii de știință, au ales să trăiască alături de oameni de peste 9.500 de ani, iar în prezent, la nivel mondial, sunt cele mai populare animale de companie în lume. Iată 30 de lucruri mai puțin cunoscute despre pisici.



# ПРИМЕР ОТНОСИТЕЛЬНОГО МЕСТОПОЛОЖЕНИЯ

---

div.fix {width:450px; **position: relative;** left: 50px; top:-50px;}



Lăbuțele din față ale pisicuțelor au 5 pernute,  
în vreme ce lăbuțele din spate - numai 4.

Folosindu-se de vazul remarcabil, de auzul ascuțit și de celelalte înzestrări incredibile de pradator fără de seamă, o (aparent banală) pisică poate vana peste 1.000 de specii pentru a supraviețui. În ciuda acestui fapt, felinele din specia *Felis silvestris catus*, așa cum le-au numit oamenii de știință, au ales să trăiască alături de oameni de peste 9.500 de ani, iar în prezent, la nivel mondial, sunt cele mai populare animale de companie din lume. Iată 30 de lucruri mai puțin cunoscute despre pisici.



# IV – АБСОЛЮТНОЕ МЕСТОПОЛОЖЕНИЕ

---

- **position: absolute**
- Абсолютно-расположенный элемент позиционируется относительно первого (родительского) основного элемента, имеющий другой тип позиционирования, чем статическое
- Если нет такого элемента, блок будет позиционироваться по отношению к тегу <HTML>
- Элементы, абсолютно-расположенные, удаляются из нормального потока. Документ и другие элементы ведут себя по отношению к абсолютно- позиционированному элементу как к не существующему
- Элементы, абсолютно-расположенные, могут перекрыть другие элементы



# ПРИМЕР АБСОЛЮТНОГО РАСПОЛОЖЕНИЯ

```
div.fix {  
width:450px;  
position: absolute;  
left: 80px;  
top: 120px;}
```



# АБСОЛЮТНОЕ МЕСТОПОЛОЖЕНИЕ

---

- По умолчанию «абсолютные» элементы позиционируются в системе координат, привязанной к окну браузера. Однако, систему координат можно изменить
- Если какой-нибудь из родителей «абсолютного» элемента имеет относительное позиционирование, то элемент будет позиционироваться относительно этого родителя. Если среди родителей несколько относительно позиционированных элементов, то «абсолютный» элемент будет позиционироваться относительно ближайшего из них



# СТИЛИ, блок позиционированный абсолютно

```
body { margin: 10px; padding: 20px; color: white; border: 1px dotted #999; }  
.block-1 { min-height: 100px; padding: 10px; background: #3A78A1; }  
.block-2 { min-height: 100px; padding: 10px; background: #27AE60; }  
.block-absolute { position: absolute;  
    top: 10px;  
    right: 10px;  
    padding: 10px;  
    font-family: "Courier New", monospace;  
    font-size: 12px;  
    background: #E74C3C;  
    opacity: 0.8; }
```

Rezultat:

Блок 1

Блок 2

```
position: absolute;  
right: 10px;  
top: 10px;
```

# БЛОК ПОЗИЦИОНИРОВАННЫЙ АБСОЛЮТНО, относительно “body”

---

Если добавить: *position: relative;* для body:

```
body {
```

```
    margin: 10px;
```

```
    padding: 20px;
```

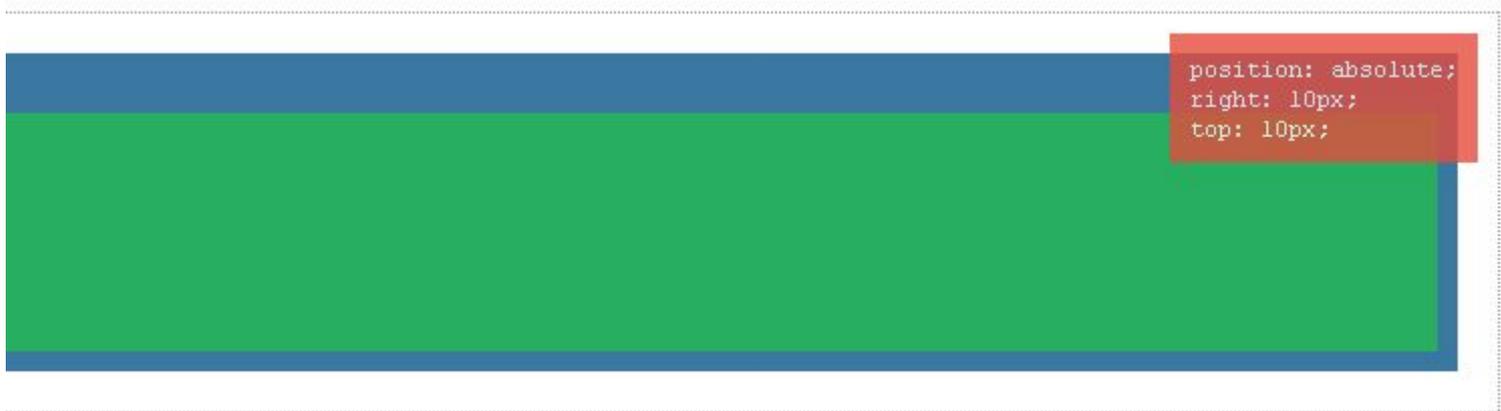
```
    color: white;
```

```
    border: 1px dotted #999;
```

```
    position: relative;
```

```
}
```

Результат:



# БЛОК ПОЗИЦИОНИРОВАННЫЙ АБСОЛЮТНО, относительно другого относительно позиционированного блока

---

□ Если добавить: `position: relative;` другому блоку:

```
.block-1 {  
    min-height: 100px;  
    padding: 10px;  
    background: #3A78A1;  
    position: relative;  
}
```



# ПРИМЕНЕНИЕ

---

- На практике абсолютное позиционирование часто применяют в декоративных целях, когда нужно аккуратно разместить какие-либо иконки, управляющие элементы или другие «рюшечки»



# ПЕРЕКРЫВАЮЩИЕСЯ ЭЛЕМЕНТЫ

---

- Когда элементы расположены вне нормального потока (position: absolute, relative, fixed) они могут перекрывать другие элементы
- Свойство «**z-index**» указывает порядок выявления элемента (элемент который должен быть размещен в передней части, или за другими элементами)
- Элемент может иметь положительный или отрицательный порядок в стеке
- Элемент с наибольшим порядком в стеке всегда будет расположен перед элементом с более низким порядком
- Свойство «**z-index**» применимо только для элементов расположенных абсолютно, относительно или фиксировано



# ПРИМЕР ПЕРЕКРЫВАНИЯ ЭЛЕМЕНТОВ

```
div.fix {width:450px; position: absolute; left: 230px;  
top: 200px; z-index: -1;}
```



scutit si de  
ra de  
peste 1.000  
ni fapt,  
cum le-au  
alaturi de  
la nivel  
companie  
te despre

# ПРИМЕР ПЕРЕКРЫВАНИЯ ЭЛЕМЕНТОВ

```
div.fix {width:450px; position: absolute; left: 230px;  
top: 200px; z-index: 1;}
```



Folosindu-se de vazul remarcabil, de auzul ascutit si de celelalte inzestrari incredibile de pradator fara de seama, o (aparent banala) pisica poate vana peste 1.000 de specii pentru a supravietui. In ciuda acestui fapt, felinele din specia *Felis silvestris catus*, asa cum le-au numit oamenii de stiinta, au ales sa traiasca alaturi de oameni de peste 9.500 de ani, iar in prezent, la nivel mondial, sunt cele mai populare animale de companie din lume. Iata 30 de lucruri mai putin cunoscute despre pisici.

buțele din față ale pisicutei au 5 pernute,



# ПРИМЕНЕНИЕ

---

- С помощью CSS-свойства *z-index* можно управлять тем, как перекрываются блоки. Значением этого свойства может быть целое число. Чем больше *z-index*, тем выше располагается блок
- Свойство *z-index* работает для элементов, у которых *position* задано как *absolute*, *fixed* и *relative*
- Таким образом, «относительный» элемент может перекрывать «абсолютный»



# СВОЙСТВО „FLOAT”

---

- CSS свойство „float” определяет, как можно сдвинуть элемент (плавающий) влево или вправо, то что позволит другим элементам его окружить
- Свойство „float” часто используется когда вставляются изображения, но так же применяется, и при верстке web-страницы
- Элементы „плавают” горизонтально, то что означает что любой элемент можно «сдвинуть» в левую сторону или в правую, но никак нельзя сдвинуть вверх или вниз
- Элементы следующие за «плавающим» элементом окружают его
- Элементы перед «плавающим» элементом не будут никак затронуты



# НЕКОТОРЫЕ ЗНАЧЕНИЯ СВОЙСТВА „FLOAT”

---

## Значение

## Описание

none

Элемент не будет плавать. Это значение по умолчанию для свойства.

left

Элемент передвинется влево

right

Элемент передвинется вправо



# ПРИМЕР СО СВОЙСТВОМ „FLOAT”

---

```
img{ border: 2px solid #aa0000;  
      height: 110px;  
      width: 155px;  
      float: left;}
```

Применяется к следующему элементу:

```
<div id="corp">  
      
      
      
    
```

---

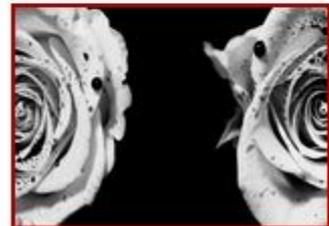
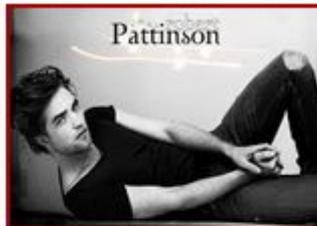
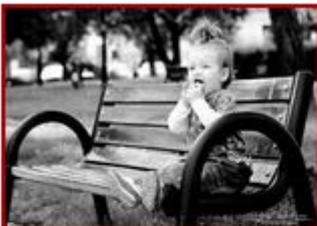
▶ *</div>*

# РЕЗУЛЬТАТ ПРИМЕРА

---

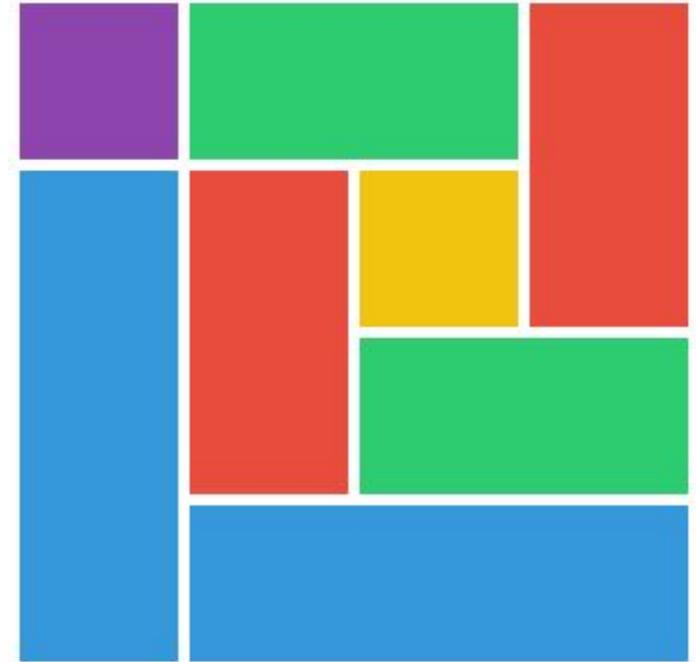


Или когда добавляем еще несколько изображений



# Другой пример применения свойства „float”

```
<body>  
  <div class=„container”>  
    <div class=„block5”></div>  
    <div class=„block7”></div>  
    <div class=„block3”></div>  
    <div class=„block1”></div>  
    <div class=„block6”></div>  
    <div class=„block4”></div>  
    <div class=„block8”></div>  
    <div class=„block2”></div>  
  </div>  
</body>
```



# Стили для блоков

---

```
html, body{margin:0; padding:0; }
```

```
body{ width:400px; height:300px; }
```

```
.container{ width:300px; margin:0 auto; padding-top:5px; }
```

```
.container div{ margin-right:5px; margin-bottom:5px; }
```

```
.block1{ float:left; width:70px; height:220px; background:#3498db; }
```

```
.block2{ float:right; width:220px; height:70px; background:#3498db; }
```

```
.block3{ height:70px; width:70px; background:#8e44ad; }
```

```
.block4{ float:left; width:70px; height:145px; background:#e74c3c; }
```

```
.block5{ float:right; width:70px; height:145px; background:#e74c3c; }
```

```
.block6{ float:right; width:70px; height:70px; background:#f1c40f; }
```

```
.block7{ float:right; width:145px; height:70px; background:#2ecc71; }
```

```
.block8{ float:left; width:145px; height:70px; background:#2ecc71; }
```

---



# СВОЙСТВО *CLEAR*

---

- Свойство *clear* запрещает обтекание элемента другими элементами. Его значения:
  - **left** — запрещено обтекание слева
  - **right** — запрещено обтекание справа
  - **both** — запрещено обтекание с обеих сторон
  - **none** — обтекание разрешено
- Если после флоатного элемента расположен элемент с запрещённым обтеканием, то последний опускается под флоатный
- Свойство *clear* учит блочные элементы «видеть» «плавающие» элементы



# ПРИМЕР ПРИМЕНЕНИЯ „CLEAR”

---

```
.fara_clear { float: left;
              width: 32px;
              height: 32px;
              margin: 10px;
              border: 3px solid #73AD21; }
.text1 {      border: 1px solid red; }
.cu_clear {   float: left;
              width: 32px;
              height: 32px;
              margin: 10px;
              border: 3px solid #73AD21; }
.text2 {      border: 1px solid red;
              clear: left; }
```



# РЕЗУЛЬТАТ

---

## Daca nu se foloseste "clear"...



Imaginea posedă proprietatea "float" - și atunci acest text, chiar dacă a fost definit în documentul HTML după elementul cu "float", va înconjura div-ul cu imaginea - va pluti în jurul imaginii.

## Daca se foloseste "clear"



În acest caz se va folosi "clear" pentru elementul care urmează după cel cu "float" - și atunci se observă că textul nu va mai înconjura imaginea.



# СВОЙТВА КРАЙОВ ЭЛЕМЕНТА

---

- Края элемента (или внешние поля) означают пространство вокруг элемента (кроме границ)
- Край может быть определен для каждой части в отдельности
- Пример: ***margin-top: 50px; margin-bottom: 70px; margin-right: 10px; margin-left: 50px;***
- Или можно использовать сокращенную форму:
- Пример: ***margin: 25px 50px 75px 100px;***  
(сверху - 25, справа – 50, в низу – 75, слева - 100)
- ***margin: 25px 50px 75px;***  
(сверху – 25, справа – 50, снизу – 75)
- ***margin: 25px 50px;*** или ***margin: 25px;***

# ВЫРАВНИВАНИЕ БЛОКА ЭЛЕМЕНТОВ

---

- Блок элементов, группированных при помощи DIV-а, могут быть выравнены по центру установив значение „**auto**” для свойства „**margin**” (внимательное использование в IE8-)
- Пример:

```
div.imagine{  
margin-left: auto;  
margin-right: auto;  
width: 50%;    }
```



# РЕЗУЛТАТ ПРИМЕРА

---



Lăbuțele din față ale pisicuțelor au 5 pernuțe,  
în vreme ce lăbuțele din spate - numai 4.



# ВЫРАВНИВАНИЕ БЛОКА ЭЛЕМЕНТОВ

---

- Другой подход для выравнивания блока элементов это использование свойства «*position*» со значением „*absolute*” (по центру)

*div.imagine{*

*position: absolute;*

*left: 25%;*

*width: 50%; }*

- Другой метод – использование свойства „*float*” (выравнивание по правой стороне)

*div.imagine{*

*float: right;*

*width: 50%; }*

---



# ИЗОБРАЖЕНИЯ-SPRITE

---

- Спрайт-изображение это то изображение которое состоит из множества маленьких изображений
- Иногда в сайте необходимо использовать несколько маленьких изображений, которые можно «извлечь» из одного изображения
- Используя спрайт-изображения – уменьшается количество запросов на сервер
- Для доступа к мини-изображению необходимо высчитать ее координаты



# ПРИМЕР ИСПОЛЬЗОВАНИЯ СПРАЙТОВ

---

**INTRARE**

Log-in 

Parola 

RETINE-MA [AM UITAT PAROLA...](#)

**INTRA**



# Для этой части страницы „Контакты”

- Я использовала для украшения ниже изложенное изображение

## Datele noastre de contact



R. Moldova, or. Chisinau,  
str. Mateevici, 60



+(373) 22 574555



[eu@mail.ru](mailto:eu@mail.ru)





# ДЛЯ ПОСЛЕДНЕГО ПРИМЕРА

---

## □ HTML коды:

```
<div class="retele_so">  
    <div class="retefa fb"></div>  
    <div class="retefa google"></div>  
    <div class="retefa twiter"></div>  
    <div class="retefa tube"></div>  
</div>
```

## □ CSS коды:

```
.retele_so .retefa { width: 50px; height: 50px;  
margin: 0px 5px; float: left; display: inline-block;  
background: url("social_sprite.jpg") no-repeat 0px 0px; }  
.retele_so .fb { background-position: -35px -145px; }  
.retele_so .google { background-position: -215px -145px; }  
.retele_so .twiter { background-position: -215px -207px; }  
.retele_so .tube { background-position: -515px -145px; }
```

# ОБЪЯСНЕНИЯ

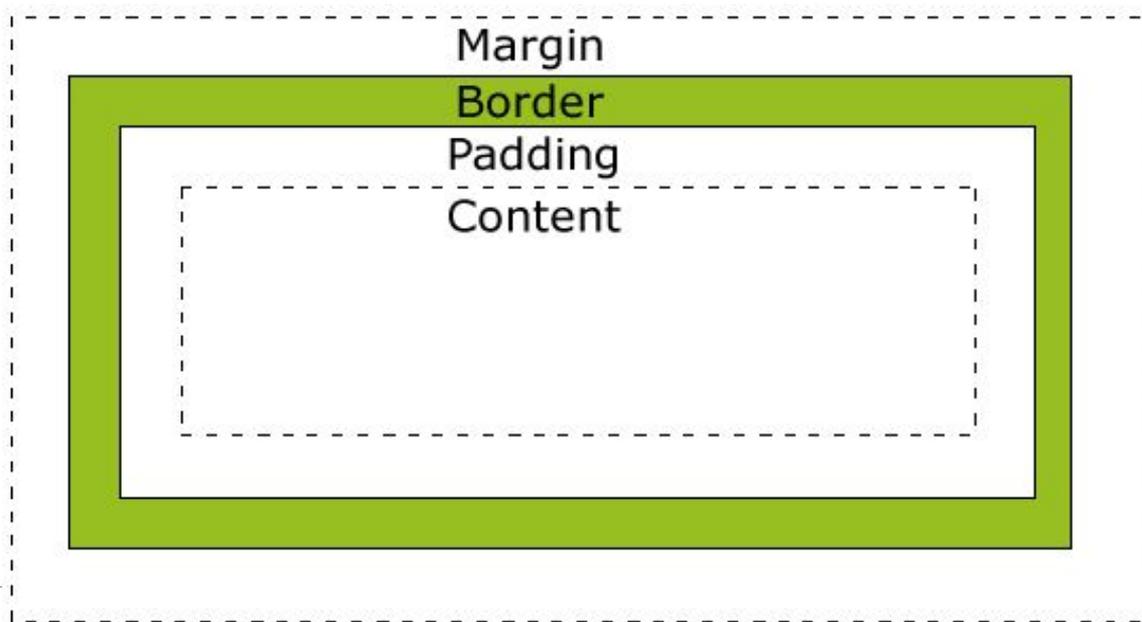
- Для каждого **div**-а определяются 2 класса
- Определяется высота и ширина блока **div** в который определим фон (тот же класс, **retea**, для всех 4 **div**-ов)
- Для каждого **div**-а (соответственно для каждого класса) определяются координаты верхней-левой точки мини-изображения
- Например для первого **div**-а, с классом **fb**, -35px – вниз и -145px – на право



# МОДЕЛЬ „КОРОБКИ” В CSS

---

- Все HTML-элементы можно считать коробками (box)
- В CSS, понятие «модель коробки» используется тогда когда необходимо произвести дизайн страницы
- Данная модель представляет собой коробки выставленные вокруг HTML-элементов с контентом и состоит из: внешних полей, границ, внутренних полей и содержимого



# ПРИМЕР

---

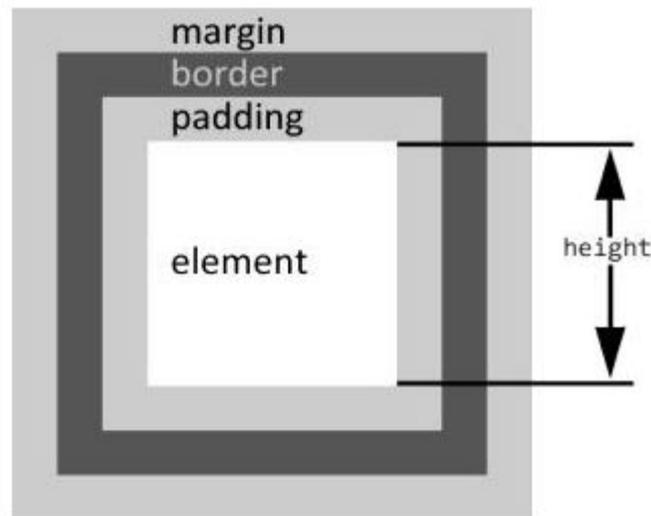
```
div.imagine{ width: 60%; margin: 25px; border: 25px solid  
black;  
border-radius: 25px; padding: 25px; margin-left: auto;  
margin-right: auto: }
```



# Дополнительно, прочтите

---

- "Блочные элементы"  
[<http://htmlbook.ru/content/blochnye-elementy>]
- "Встроенные элементы"  
[<http://htmlbook.ru/content/vstroennye-elementy>]



# ВЕРСКА САЙТОВ

---

- **Верстка** это процесс преобразования рисунка веб-страницы (например *.psd* файл) в HTML-документ, используя HTML и CSS
- При верстке веб страниц необходимо соблюдать минимум следующие правила:
  - Результат должен, максимально совпасть с рисунком дизайнера
  - Полученная страница должна адекватно выглядеть во всех браузерах – проверка кросс-платформенности (**cross platform**)
  - Написать HTML коды отдельно от CSS



# РЕКОМЕНДАЦИИ ДЛЯ ВЕРСТКИ СТРАНИЦ

---

- Консорциум **w3** рекомендует использование **div-**ов для вертки
- Дополнительно, рекомендуется, при определении ширины (`width`) и высоты (`height`) блока использовать свойства: *min-width* и *max-width*, *min-height* и *max-height*
- *Min-width* приоритетнее *max-width* и *width* (*max-width* приоритетнее *width*)
- Приоритет тот же и при определении высоты: свойство *min-height* приоритетнее чем свойство *max-height* и *height*



# ПРИМЕР

---

Header

Meniu

Coloana 1

Coloana 2

Coloana 3

Subsol



# ПРИМЕР

---

```
<body>
```

```
  <div class="container">
```

```
    <div class="header block">Header</div>
```

```
  <div class="menu block"> Meniu</div>
```

```
    <div class="column1 block">Coloana 1</div>
```

```
    <div class="column2 block">Coloana 2</div>
```

```
    <div class="column3 block">Coloana 3</div>
```

```
    <div class="footer block">Subsol</div>
```

```
  </div>
```

```
</body>
```

---



# ВНЕШНЯЯ ТАБЛИЦА СТИЛЕЙ

---

```
body{ font-family:'PT Sans', sans-serif; }
.container{ max-width:600px;
  margin:10px auto;  }
.block{ padding:10px 0px;
  margin-bottom:10px;
  color:#CC0000;  }
.header { background:rgba(230, 210, 250, 0.7);
  width: 100%;  }
.menu { background:rgba(230, 210, 250, 0.5);
  width: 100%;  }
.column1 { background:rgba(230, 210, 250, 0.4);
  float:left; width:180px;
  min-height:50px;
  margin-right:30px; }
.column2{ background:rgba(230, 210, 250, 0.4);
  float:left; width:180px;
  min-height:50px;  }
.column3{
  background:rgba(230, 210, 250, 0.4);
  float:right;
  width:180px;
  min-height:50px;  }
.footer{
  background:rgba(230, 210, 250, 0.7);
  width: 100%;
  clear: both;  }
```

---

▶ min-height:50px; }

# ДРУГОЙ ПРИМЕР (чуть поменяли стили)

---

- Код HTML остался тот же

Header

Meniu

Coloana 1

Coloana 2

Coloana 3

Subsol



# ВНЕШНЯЯ ТАБЛИЦА СТИЛЕЙ

---

...

```
.column1{ background:rgba(230, 210, 250, 0.4);  
          float:left; max-width:280px;  
          min-height:50px; }
```

```
.column2{ background:rgba(230, 210, 250, 0.4);  
          float:right; max-width:280px;  
          min-height:50px; }
```

```
.column3{ background:rgba(230, 210, 250, 0.4);  
          clear: both; margin-left: auto; margin-right: auto;  
          max-width:280px;  
          min-height:50px; }
```

```
.footer{ background:rgba(230, 210, 250, 0.7);  
         width: 100%; }
```



# АДАПТИВНЫЙ ДИЗАЙН (HTML Responsive Web Design)

---

- Веб сайт является *responsive* если его страницы адаптируются в зависимости от среды вывода (разрешение экрана – desktop, laptop, планшет, телефон), без больших изменений, с целью предоставления пользователю хорошего опыта (удобное передвижение по сайту). **Responsive design** предполагает:
  - Использование **адаптивных сеток** – они могут поменять ширину в зависимости от разрешения экрана. Размеры задаются при помощи относительных величин
  - Размеры изображений так же задаются относительными величинами
  - **Используются медиа-запросы (media queries)** – процесс загрузки разных CSS свойств в зависимости от разрешения устройства. Сайт распознает тип устройства вывода и его разрешение, и загружает соответствующие свойства
- Для упрощения внедрения адаптивного дизайна появились много framework-ов. Самые распространенные: Bootstrap, Foundation и Skeleton

# АДАПТИВНЫЙ ДИЗАЙН

---



Web Design Responsive



# W3.CSS

---

- Качественные эффекты достигаются и с помощью W3.CSS - frame-work используемый для адаптивного дизайна
- Используются только стили, без применения JS или JQuery
- W3.CSS это свободно доступный frame-work и не требует специальных лицензий
- Для его использования загрузите его и используйте
- Достиг версию 1.9 в январе 2016, появился в июне 2015)
- Описание: <http://www.w3schools.com/w3css>
- Примеры: [http://www.w3schools.com/w3css/demo\\_website.htm](http://www.w3schools.com/w3css/demo_website.htm)

# Пример с применением w3.css

```
<!DOCTYPE html>
```

```
<html><head>
```

```
  <title>W3.CSS</title>
```

```
  <meta name="viewport" content="redimensionarea contentului">
```

```
  <link rel="stylesheet" href=" ../lib/w3.css">
```

```
  <style>
```

```
    .align {    text-align: center; valign: middle; }
```

```
    img {      padding: 10px; float: left;    }
```

```
  </style></head>
```

```
<body>
```

```
  <div class="w3-container w3-brown"><h1 class="align">Florile verii</h1></div>
```

```
  
```

```
  
```

```
  <div class="w3-container"><p>Macii - cea mai frumoasa privesc a lunilor mai - iunie.</p>
```

```
</div>
```

```
  <div class="w3-container w3-brown">
```

```
  <h5 class="align">Toate drepturile rezervate</h5>
```

```
  <p class="align">E-mail: natalia-plesca@yandex.ru</p>
```

```
</div></body></html>
```

# РЕЗУЛТАТ



# ВЕРСТКА С ИСПОЛЬЗОВАНИЕМ HTML5

- В HTML5 появились специальные теги, которые используются для вёрстки сайта
- Дополнительно: [http://www.w3schools.com/html/html5\\_new\\_elements.asp](http://www.w3schools.com/html/html5_new_elements.asp)

<b>header</b>	Определяет заголовок документа или какого-то раздела
<b>nav</b>	Определяет контейнер для ссылок
<b>section</b>	Определяет раздел в документе
<b>article</b>	Определяет какую-то самостоятельную статью
<b>aside</b>	Определяет содержание рядом с основным содержанием или боковую панель
<b>footer</b>	Определяет нижний колонтитул или раздел



# ПРИМЕР ИСПОЛЬЗОВАНИЯ HTML5

---

```
<body>
```

```
  <main>
```

```
    <header>Header</header>
```

```
    <nav> Meniu</nav>
```

```
    <section><h1>Coloana 1</h1></section>
```

```
    <aside><h1>Coloana 2</h1></aside>
```

```
    <article><h1>Coloana 3</h1></article>
```

```
    <footer>Subsol</footer>
```

```
  </main>
```

```
</body>
```

---



# СТИЛИ

---

```
body{ font-family:'PT Sans', sans-serif; }
main{ max-width:600px; margin:10px auto; }
header, nav, section, aside, article, footer { padding:10px 0px;
    margin-bottom:10px; color:#CC0000; }
header { background:rgba(30, 210, 250, 0.7); width: 100%; }
nav { background:rgba(30, 210, 250, 0.5); width: 100%; }
section{ background:rgba(30, 210, 250, 0.4); float:left;
    max-width:280px; min-height:50px; }
aside{ background:rgba(30, 210, 250, 0.4); float:right;
    max-width:280px; min-height:50px; }
article{ background:rgba(30, 210, 250, 0.4); clear: both; margin-left: auto;
margin-right: auto; max-width:280px; min-height:50px; }
footer{ background:rgba(30, 210, 250, 0.7); width: 100%; }
```



# РЕЗУЛТАТ

Header

Meniu

**Coloana 1**

**Coloana 2**

**Coloana 3**

Subsol

Header

Meniu

**Coloana 1**

**Coloana 2**

**Coloana 3**

Subsol

<https://validator.w3.org/nu/#file>

**Document checking completed. No errors or warnings to show.**





- 
- ▣ **3** понятия выученные сегодня
  - ▣ **2** вопроса которые возникли сегодня
  - ▣ **1** предложение для следующей пары

