

Web-технологии и web-дизайн

HTML

Язык описания
Web-страниц

CSS

Язык описания
представлений



Беркунский Е.Ю., кафедра ИУСТ, НУК
eugeny.berkunsky@gmail.com
<http://www.berkut.mk.ua>

HTML Язык описания Web-страниц

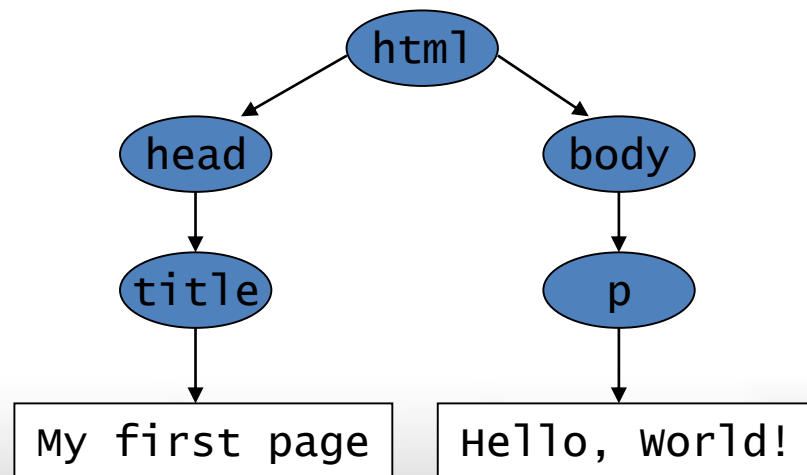


Что такое HTML?

- это не язык программирования, а «язык разметки» (HyperText Markup Language);
- определяет содержание и структуру страницы, но не внешний вид;
- элементы языка имеют структуру дерева (вложенные элементы);
- пробелы либо игнорируются, либо заменяются одним пробелом;
- узлы дерева представляют собой либо текст (содержание), либо «структурные элементы», маркируемые «тегами» и имеющие «атрибуты».

Текст на языке HTML и его структура

```
<html>  
  <head>  
    <title>My first page</title>  
  </head>  
  <body>  
    <p>Hello, world!</p>  
  </body>  
</html>
```



Элемент, маркированный тегом, имеет следующую структуру:

```
<element attr1="value1" attr2="value2"...>  
    внутреннее содержание  
</element>
```

```
<a href="page2.html">Next page</a>
```

Если тег не имеет *внутреннего содержания*, то он может иметь упрощенную структуру:

```
<element attr1="value1" attr2="value2"... />
```

```

```

```
<hr/>
```

Структура страницы

Правильная страница имеет две части – заголовок (информация о странице) и тело (содержание страницы)

```
<html>  
  <head>  
    информация о странице  
  </head>  
  <body>  
    внутреннее содержание  
  </body>  
</html>
```

Пример:

```
<html>  
  <head>  
    <title>My first web page</title>  
  </head>  
  <body>  
    <h1>welcome to my first web page!</h1>  
    <p>This is a paragraph of text.</p>  
  </body>  
</html>
```

ХНТМЛ – более современный HTML

HTML + XML (eXtended Markup Language) = ХНТМЛ

Браузеры отображают пришедшую к ним по запросу страницу. Страница может быть написана как на «старом» HTML, так и на более новых XML и ХНТМЛ. Мы будем использовать «современный» ХНТМЛ. Почему?

- Более строгий и хорошо структурированный язык
- Лучшая «переносимость» между различными браузерами
- Лучше использовать более современные средства, чтобы не пришлось в будущем переделывать ваши замечательные страницы
- ХНТМЛ допускает вставки на других «языках разметки» XML, SVG, MathML, MusicML и т.д.



Структура документа в формате XHTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
  <head>  
    інформація о страниці  
  </head>  
  <body>  
    внутреннее содержание  
  </body>  
</html>
```

Основное отличие во внутреннем содержании документов HTML и XHTML: если браузер видит ошибку в HTML-документе, то он обязан постараться понять, что имел в виду автор документа. Если ошибка обнаружена в XHTML документе, то браузер просто сообщает об ошибке.

Кроме того, форматирование элементов XHTML-документа должно быть оформлено с помощью CSS-стилей. Многие теги, такие как `font` и атрибуты, такие как `bgcolor` и `align` не поддерживаются в XHTML.



- Неправильно :

- Правильно :

- ```
<hr/> (или <hr> </hr>)

<i>some text</i>
<body><p>Hello, world!</p></body>

&

```



# Блочные и строчные элементы

Блочные элементы содержат фрагменты текста, которые всегда отображаются в отдельных блоках. Браузеры визуальнo отделяют блочные элементы друг от друга.

Примеры: `<div>`, `<p>`, `<ul>`, `<tr>`

Строчные элементы могут располагаться друг за другом в пределах одной строки

Примеры: `<img>`, `<a>`, `<b>`, `<span>`

Комментарии вставляются в HTML текст так же, как и обычные элементы:  
`<!-- Это моя самая прикольная страница -->`



# Некоторые элементы HTML и их атрибуты

*Абзац (параграф) – блочный элемент*

`<p>`Видимое здесь содержимое  
отображается в браузере в виде одного абзаца.  
Повторяющиеся пробелы и переходы со строки на  
строку игнорируются. `</p>`

Видимое здесь содержимое отображается в браузере в виде одного абзаца.  
Повторяющиеся пробелы и переходы со строки на строку игнорируются.

Если необходимо сделать переход на следующую строку внутри абзаца,  
то это достигается с помощью элемента `<br/>`

`<p>`Это первая строка параграфа, `<br/>`а это уже вторая `</p>`  
`<p>`А вот это уже следующий абзац. `</p>`

Это первая строка параграфа,  
а это уже вторая  
А вот это уже следующий абзац.

# Некоторые элементы HTML и их атрибуты

*Заголовки – блочные элементы*

`<h1>Национальный Университет Кораблестроения</h1>`

`<h2>Факультет компьютерных наук</h2>`

`<h3>Расписание занятий</h3>`

**Национальный Университет Кораблестроения**

**Факультет компьютерных наук**

**Расписание занятий**

Всего может быть до шести уровней заголовков (от `<h1>` до `<h6>`)

*Горизонтальная черта (разделитель) – блочный элемент*

`<p>Национальный Университет Кораблестроения</p><hr/>`

`<p>Факультет компьютерных наук</p>`

Национальный Университет Кораблестроения

---

Факультет компьютерных наук

# Гиперссылка

*Гиперссылка – строчный элемент*

`<p>Пользуйтесь поиском`

`<a href="http://www.google.com">Google</a>`

`– самым распространенным поисковиком в мире!`

`</p>`

Пользуйтесь поиском [Google](http://www.google.com) - самым распространенным поисковиком в мире!

*Будьте аккуратны с вложенностью элементов!*

`<p><a href="Page2.html">Это в первом абзаце</p>`

`<p>А это уже во втором!</a></p>`

Здесь две ошибки: блочный элемент внутри строчного и неправильная вложенность элементов. Тем не менее, HTML-браузер может корректно отобразить эти элементы!

# Картинки

*Вставка изображений – строчный элемент*

```
<p></p>
```



*Дополнительные атрибуты*

```

```

```

```

Изображение может служить ссылкой так же, как и текст:

```



```

*Список (нумерованный или нenumерованный) – блочный элемент, Содержащий внутри себя блочные элементы – члены списка*

```

 Первая строка списка
 Вторая строка списка
 Третья строка списка

```

- Первая строка списка
- Вторая строка списка
- Третья строка списка

*Атрибуты списка могут задавать вид маркеров (не поддерживается в «строгом» XHTML), например*

```
<ul type="square">
 Первая строка списка

```

- Первая строка списка

# Нумерованные списки

## Нумерованный список

```

 Вымыть посуду
 Постирать носки
 Купить жене цветы

```

1. Вымыть посуду
2. Постирать носки
3. Купить жене цветы

*Атрибуты списка могут задавать тип и способ нумерации  
(не поддерживается в «строгом» XHTML)*

```
<ol start="3" type="a">
 Сделать домашнее задание
 Купить билеты на концерт

```

- c. Сделать домашнее задание
- d. Купить билеты на концерт

# Вложенные списки

*Один список может быть вложен в другой*

```

 Документы:

 паспорт
 билеты
 Гостиница - бронь

 Предметы личной гигиены:

 Зубная щетка
 паста
 Мыло


```

1. Документы:
  - Паспорт
  - Билеты
  - Гостиница - бронь
2. Предметы личной гигиены:
  - Зубная щетка
  - Паста
  - Мыло



# Цитирование

*Цитирование может быть коротким (строчным) и длинным (блочным)*

<p>В своем выступлении перед рабочими Леонид Ильич Брежнев, в частности, сказал:

<blockquote>Мы еще теснее сплотимся вокруг родной Коммунистической Партии и стройными рядами будем двигаться вперед к полному торжеству коммунизма!</blockquote>

</p>

В своем выступлении перед рабочими Леонид Ильич Брежнев, в частности, сказал:

Мы еще теснее сплотимся вокруг родной Коммунистической Партии и стройными рядами будем двигаться вперед к полному торжеству коммунизма!

<p>Каркнул ворон: <q>Никогда!</q></p>

Каркнул ворон: “Никогда!”

# Выделение фрагментов текста

*Выделение производится с помощью тегов `<em>`, `<strong>`, `<kbd>`, `<code>`*

`<p>`Изучая `<code>HTML</code>` следует обратить `<strong>`особое внимание`</strong>` на различие между `<em>`нумерованными`</em>` списками `<kbd>&lt;ol&gt;``</kbd>` и `<em>`нenumерованными`</em>` списками `<kbd>&lt;ul&gt;``</kbd>`.`</p>`

Изучая HTML следует обратить **особое внимание** на различие между нумерованными списками `<ol>` и нenumерованными списками `<ul>`.

*Фрагменты кода можно выделять с помощью блочного тега `<pre>`*

```
<pre>
 public static void main(String[] args) {
 System.out.println("Hello, world!");
 }
</pre>
```

Этот фрагмент выводится в точности так же, как он напечатан, включая все пробелы и переводы строк.

# Элементы заголовка HTML

```
<title>Это моя страница</title>
```

Отображается в заголовке страницы в браузере.

```
<meta name="description"
```

```
 content="Эта страница содержит полезные сведения"/>
```

Описывает содержимое страницы.

```
<meta name="keywords" content="мир, труд, май"/>
```

Описывает ключевые слова (часто используется поисковыми машинами).

```
<meta name="author" content="Eugeny Berkunsky"/>
```

Указание на автора.

```
<meta name="generator" content="Microsoft FrontPage 8.0"/>
```

Указание на программу, которая сгенерировала этот текст.

# Элементы заголовка HTML

```
<meta http-equiv="Content-Language" content="ru">
```

Определяет основной язык, на котором написана страница.

```
<meta http-equiv="Content-Type"
 content="text/html; charset=windows-1251">
```

Описывает формат и кодировку страницы.

```
<meta http-equiv="refresh"
 content="5;http://www.google.com/">
```

Описывает частоту перезагрузки страницы (в секундах) и, возможно, делает “redirection” на другую страницу.

```
<link type="image/jpeg" rel="shortcut icon"
 href="favicon.jpg">
```

Описывает иконку, ассоциирующуюся со страницей (устаревший вариант: поместить в корневой каталог сайта файл с именем favicon.ico).

# MIME-ТИПЫ

Internet media type (MIME – Multipurpose Internet Mail/Media Extension)

<code>application/javascript</code>	Текст на JavaScript (обычно .js)
<code>application/octet-stream</code>	Двоичная программа (обычно .exe)
<code>image/jpeg</code>	Картинка в формате JPEG
<code>image/gif</code>	Картинка в формате GIF
<code>image/vnd.microsoft.icon</code>	Иконка в формате Microsoft
<code>audio/mpeg</code>	Звуковой файл (обычно .mp3)
<code>text/plain</code>	Просто текст (например, .txt)
<code>text/html</code>	Текст на языке HTML
<code>text/css</code>	Каскадная таблица стилей (.css)
<code>video/mp4</code>	Кино в формате MP4

Подробнее см. <http://ru.wikipedia.org/wiki/MIME>

# Мнемоники

Специальные символы отображаются в текстах в виде специальных слов (entities). Вот примеры некоторых мнемоник:

< >	&lt; &gt;
™ ©	&trade; &copy;
π δ Δ	&pi; &delta; &Delta;
И	&1048;
¢ £ ¥	&cent; &pound; &yen;
" &	&quot; &amp;

Полный список см.

[http://www.w3schools.com/tags/ref\\_entities.asp](http://www.w3schools.com/tags/ref_entities.asp)

## CSS

### Язык описания представлений



# Что такое CSS?

- это язык описания внешнего представления для содержимого, описанного в HTML-страницах;
- определяет внешний вид текста – шрифты, размеры, цвет;
- определяет расположение элементов друг относительно друга;
- описание внешнего представления может быть физически отделено от описания содержания.

В прежнем стандарте HTML допускалось использование описания внешнего представления с помощью атрибутов и отдельных элементов.

```
<p>добро пожаловать в НУК им.адм.Макарова. Вы
получите самое полное, <i>лучшее, <u>ЛУЧШЕЕ</u></i>
образование в Николаеве с МИНИМАЛЬНЫМ
начальным багажом знаний!</p>
```

Добро пожаловать в НУК им.адм.Макарова. Вы получите **самое полное, лучшее, КАЧЕСТВЕННОЕ** образование в Николаеве с **МИНИМАЛЬНЫМ** начальным багажом знаний!

Однако, подобный способ описания представления  
**НЕ ПОДДЕРЖИВАЕТСЯ** в «строгом» XHTML!



# Базовая структура элементов стиля

Элементы стиля описываются в виде

*атрибут: значение;*

и погружаются либо в атрибут `style` непосредственно в элементе HTML, либо собираются в заголовке документа (или на отдельной CSS-странице).

```
<h2 style="color: blue; text-align: center;">курс лекций.</h2>
```

---

```
<head>
 <style type="text/css">
 h2 { color: blue; text-align: center; }
 </style>
 <style class="red-class" { color: red; text-align: center; }
</head>
```

```
<body>
 <h2>курс лекций.</h2>
 <h2 class="red-class">практические занятия.</h2>
```

# Пример

В НУК вы получите **самое полное, лучшее, качественное** образование в Николаеве с **МИНИМАЛЬНЫМ** начальным багажом знаний!

```
<p>
 В НУК вы получите
 самое полное,
 <i>лучшее,
 <u>качественное</u>
 </i>
 образование в Николаеве с
 минимальным
 начальным багажом знаний!

</p>
```

```
<p style="font-family: Arial;">В НУК вы получите
 самое полное,
 лучшее,
 качественное

 образование в Николаеве с
 минимальным
 начальным багажом знаний!
</p>
```

# Привязывание страницы стилей к документу

```
p { color: white; background-color: black; }
h1 { font-size: large; font-weight: bold; }
h2 { font-weight: 500; color: blue; }
```

*mystyles.css*

```
<html>
 <head>
 <link rel="stylesheet" type="text/css" href="mystyles.css"/>
 </head>
 <body>
 содержание страницы HTML-документа
 </body>
</html>
```

*example.html*

Ссылка может быть как на «локальную» страницу стилей, созданную специально для этого документа, так и на «глобальную», хранящуюся в сети Интернет.

```
<head>
 <link rel="stylesheet" type="text/css"
 href="http://www.google.com/uds/css/gsearch.css" />
</head>
```

*example1.html*

# Некоторые атрибуты и варианты значений

Атрибуты шрифта (font) и текста (text).

```
font-family: "lucida console", "courier new", sans-serif;
font-size: small;
font-size: larger;
font-size: 10px;
font-size: 80%;
font-weight: bold;
font-weight: 400;
font-style: italic;
```

```
font: sans-serif bold x-large;
```

```
text-align: center;
text-align: right;
text-transform: uppercase;
text-indent: 2cm;
text-decoration: underline;
text-decoration: blink;
```

# Атрибуты цвета

```
color: red;
color: rgb(25, 30, 120);
color: #c0c0c0;
```

```
background-color: yellow;
```

Допустимые словесные названия цветов:

aqua – голубой,  
black – черный,  
blue – синий,  
fuchsia – розовый,  
gray – серый,  
green – зеленый,  
lime – светло-зеленый,  
maroon – коричневый,

navy – темно-синий,  
olive – оливковый (желто-зеленый),  
purple – фиолетовый,  
red – красный,  
silver – серебряный (светло-серый),  
teal – «морской волны»,  
white – белый,  
yellow – желтый.

Вот как выглядят эти цвета: [colors.html](https://www.w3schools.com/colors/colors_names.asp)

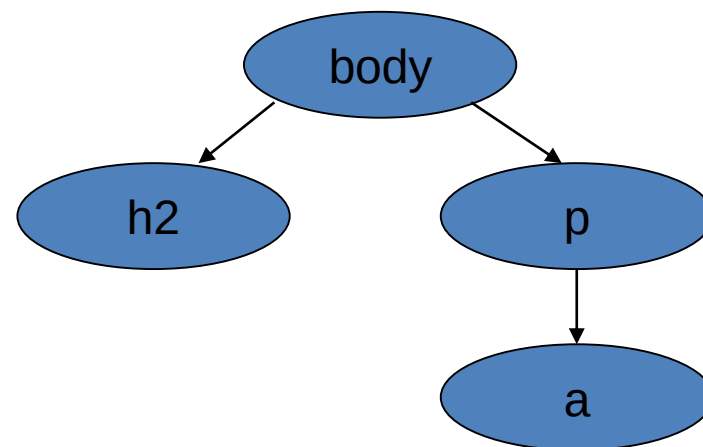
# Глобальное определение стиля в документе

```
<html>
 <head>
 <style type="text/css">
 body { color: blue; font-family: sans-serif;
 width: 400px; text-align: justify; }
 strong { color: red; font-weight: 600; }
 em { background-color: silver; }
 </style>
 </head>
 <body>
 <p>Италия издавна была одной из самых романтических и посещаемых
 стран мира. С годами она ничуть не утратила
 своей притягательной силы, так как в этой солнечной стране
 царит атмосфера, располагающая к веселью,
 общению и отдыху на любой вкус. Не только
 красивые и разнообразные ландшафты, семейные пляжи, но и
 многочисленные памятники истории и культуры очаровывают
 путешественников, прибывающих в Италию со всех концов земли.
 </p>
 </body>
</html>
```

# Взаимодействие стилей

```
body { font-family: sans-serif; background-color: yellow; }
p { color: red; background-color: aqua; }
a { text-decoration: overline underline; }
h2 { font-weight: bold; text-align: center; }
```

```
<body>
 <h2>Это заголовок</h2>
 <p>А это абзац со
 ссылкой
 внутри
</p>
</body>
```



Это заголовок \_\_\_\_\_

А это абзац со ссылкой внутри

# Использование классов

```
p { font-family: sans-serif; }
p.special { background-color: yellow; font-weight: bold; }
.standout { color: red; font-family: cursive; }
```

```
<body>
 <h2 class="standout">Это заголовок класса standout</h2>
 <p>Это просто абзац</p>
 <p class="special">Это абзац класса special</p>
 <p class="standout special">Это абзац классов special
 и standout</p>
</body>
```

[classes.html](#)

*Это заголовок класса standout*

Это просто абзац

**Это абзац класса special**

*Это абзац классов special и standout*



# Псевдо-классы

```
:link { color: #FF0000 } /* Еще не посещенная гиперссылка */
:visited { color: #00FF00 } /* Уже посещенная гиперссылка */
:hover { color: #FF00FF } /* Курсор мыши над ссылкой */
:active { color: #0000FF } /* Выделенная гиперссылка */
```

<body>

  <h2>Проверяем, как работают разные псевдо-классы</h2>

  <p>Ссылка на страницу

    <a href="classes.html" title="Страница с предыдущего слайда">  
      classes.html</a>.</p>

</body>

<pseudo-classes.html>

# Использование тегов div и span

```
div.style1 { font-family: sans-serif; }
div.style2 { font-family: Times; color: blue; }
.bold { font-weight: bold; }

<body>
 <div class="style1">
 <h2>Это заголовок класса style1</h2>
 <p>Это абзац класса style1</p>
 </div>
 <div class="style2">
 <h2>Это заголовок класса style2</h2>
 <p>Это абзац класса style2</p>
 </div>
</body>
```

**Это заголовок класса style1**

Это абзац класса style1

**Это заголовок класса style2**

Это абзац класса style2

# Приписывание стилей конкретному элементу

```
#myElement { font-family: sans-serif; color: blue; }
```

```
<body>
 <h2 id="myElement">Это мой заголовок</h2>
</body>
```

**Это мой заголовок**

# Каскадирование стилей

Чем определяется стиль конкретного элемента?

1. Стилем, определенным браузером «по умолчанию»
2. Стилем, указанным в отдельной CSS-странице, привязанной к HTML-документу элементом `<link>` в заголовке:  
`<link rel="stylesheet" type="text/css" href="styles.css"/>`
3. Стилем, указанным в заголовке HTML-документа с помощью элемента `<style>`:  
`<style> body { background-color: yellow; } </style>`
4. Стилем, указанным в самом элементе с помощью атрибута `style`:  
`<p style="margin: 0.5in;">Отступ в полдюйма от края</p>`

Чем «ближе» определение стиля к элементу, тем приоритетнее он будет в случае конфликта параметров стиля.

```
selector1 selector2 { styles }
```

Стиль применяется к элементам, определенным селектором `selector2`, только если этот элемент находится внутри элемента, определенного селектором `selector1`.

```
selector1 > selector2 { styles }
```

Стиль применяется к элементам, определенным селектором `selector2`, только если этот элемент находится *непосредственно* внутри элемента, определенного селектором `selector1`.



# Дополнительные возможности - пример

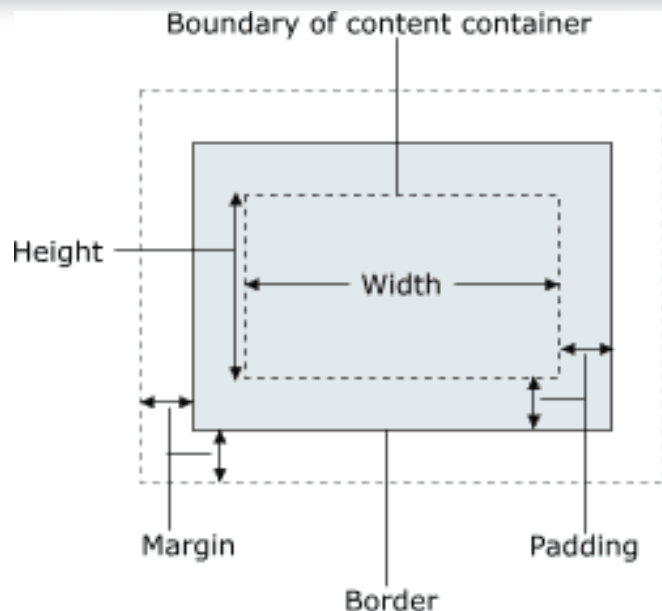
```
li strong { text-decoration: underline; }
```

```
<p>Почему люди пользуются поиском Google?
</p><u1>Он очень быстрый
Он дает нужные результаты
```

Почему люди пользуются **поиском Google?**

- Он **очень** быстрый
- Он дает нужные результаты

# Размещение фрагментов с помощью CSS



Блочные элементы имеют:

- внутреннее содержание заданной ширины и высоты (width, height);
- прослойку (padding);
- границу (border);
- поля (margin).

По умолчанию блоки располагаются вертикально, при этом поля соседних блоков перекрываются (общее поле двух соседних блоков равно по высоте максимальному из двух полей элементов).

B1

B2



# Пример размещения блоков

```
<style type="text/css">
 p { font-family: sans-serif;
 font-size: 16pt;
 border: 2px solid red; }
 p.class1 { width: 400px;
 background-color: yellow;
 padding: 0.5cm;
 margin: 0.5cm; }
 p.class2 { width: 500px;
 background-color:
 green; padding: 0.3cm;
 margin: 1cm; }

</style>

<body>
 <p class="class1">Первый параграф</p>
 <p class="class1">Второй параграф</p>
 <p class="class2">Третий параграф</p>
 <p class="class2">Четвертый параграф</p>
 <p class="class1">Пятый параграф</p>
</body>
```



# Характеристики границы (border)

```
{ border: width style color; }
```

width – ширина – 10px; 25%; 0.5in; thin; thick; medium;  
style – стиль – none; solid; dotted; dashed; double;  
color – цвет – red; rgb(240,240,240); #C02510;

Можно задавать отдельные характеристики границы

```
{ border-width: ...;
 border-style: ...;
 border-color: ...; }
```

Можно задавать характеристики границы по сторонам блока

```
{ border-top: ...;
 border-bottom: ...;
 border-left: ...;
 border-right: ...; }
```

или и то и другое вместе

```
{ border-bottom-width: ...;
 border-left-style: ...;
 border-top-color: ...; }
```

# Поля и заполнение

```
{ margin: width; padding: width; }
```

width – ширина – 10px; 5%; 0.5in;

Можно задавать ширину полей и заполнения отдельно по сторонам

```
{ margin-left: ...;
 padding-bottom: ...;
 padding-top: ...; }
```

Пример:

```
<h1 style="margin-top: 0.2cm; padding: 0.1cm;">Заголовок</h1>
```

# Размещение блоков на странице

```
{ width: 70%; margin-left: auto; margin-right: auto; }
```

Заданный таким образом, как показано выше, стиль позволяет разместить блок по центру страницы (соответственно слева или справа, если задано только одно поле).

Размещение текста (или других строчных элементов) *внутри* блока задается иначе:

```
{ text-align: left; } (или center или right)
```

```
<h1 style="width: 50%;
 margin-left: auto;
 background-color: yellow;">Заголовок</h1>
<h2 style="width: 70%;
 margin-left: auto; margin-right: auto;
 background-color: yellow;
 text-align: right;">Еще один заголовок</h2>
```

Замечание: IE6 может не распознавать правильно указания `margin: auto;`

# Дополнительно о размерах блоков

Вместо указания точной ширины/высоты блока можно задавать *максимальные* или *минимальные* размеры:

```
{ min-width: 100px; max-height: 5cm; }
```

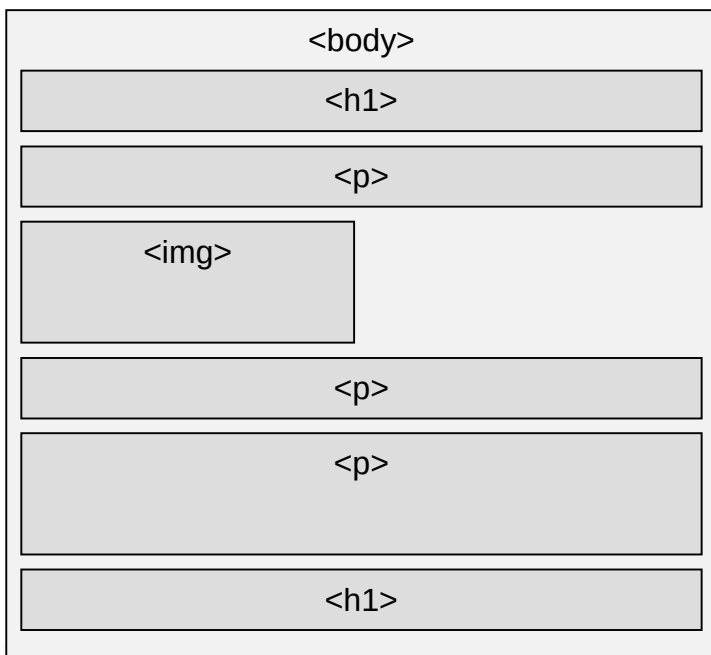
В случае, если содержимое блока превышает минимальный размер – он автоматически увеличивается;

В случае, если содержимое блока превышает максимальный размер – содержимое автоматически обрезается;



# Позиционирование блоков

Блоки можно позиционировать, убирая их из «автоматического» размещения по вертикали и указывая свой вариант размещения. Это можно сделать двумя способами, один из которых – указание блока как «плавающего».



Содержимое блоков «обтекает» плавающий блок таким образом, что не происходит наложения содержимого. Чтобы дополнительно визуально отделить плавающий блок, можно использовать задание в этом блоке полей (margin).

# Позиционирование блоков (продолжение)

Задание «плавающего» блока:

```
img.to-right { float: right; width: 200px; }
img.to-left { float: left; width: 200px; }
```

[selenicereus.html](http://selenicereus.html)

<body>

```

```

```
<p>На фотографии справа цветок совсем не ослепительно белый,
а кремовый. Но это не естественный его цвет, просто освещение
ночью искусственное, вот и дает оно желтый оттенок
на фотографии.</p>
```

```

```

```
<p>Фото слева: Вид на этот замечательный цветок сбоку, размер –
28 см (29 июня 2003 года, 2 часа ночи). Его аромат, как я
обнаружил, исходит из желтой зоны между коричневыми
чашелистиками и белыми лепестками.</p>
```

</body>

Чтобы указать, что следующий блок не должен содержать «обтекающий» текст, можно явно указать это с помощью атрибута `clear` с возможными значениями `right`, `left`, `both`, например:

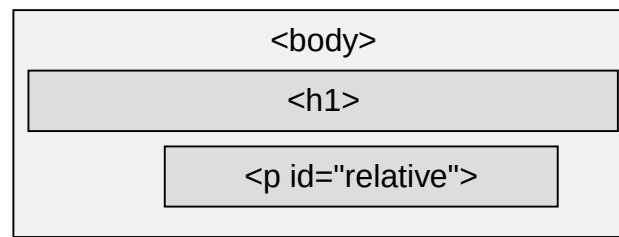
```
p.wide-paragraph { clear: both; }
```

# Позиционирование блоков (продолжение)

Второй способ вывода блока из общего алгоритма размещения блоков – явное позиционирование на странице относительно начала страницы или других блоков.

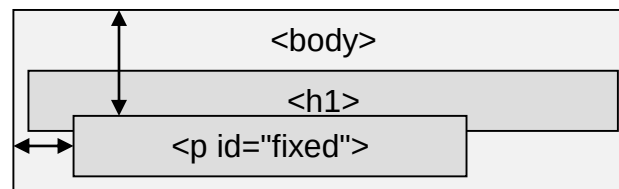
```
#relative { position: relative; right: 10%; width: 200px; }
```

Задание позиции `relative` означает, что блок смещается относительно своего «естественного» положения.



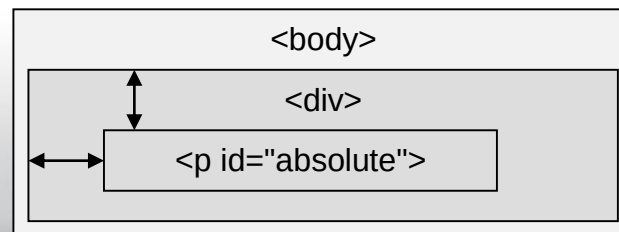
```
#fixed { position: fixed; top: 300px; left: 5em; width: 15cm; }
```

Задание позиции `fixed` означает, что блок располагается на фиксированном месте страницы.



```
#absolute { position: absolute; top: 200px; width: 200px; }
```

Задание позиции `absolute` означает, что положение блока указано относительно охватывающего его блока.



# Позиционирование блоков (продолжение)

Пример.

```
<div>
 <p>Это самый обычный параграф. Таких параграфов ...</p>
</div>
<div>
 <p style="position: relative; left: 10%; width: 80%">
 Это тоже самый обычный параграф, но ...</p>
</div>
<div style="position: fixed; left: 100px; width: 80%;">
 <p style="position: absolute; left: 5em; top: 200px; width: 50%;">
 В этом параграфе указана абсолютная позиция ...</p>
</div>
<div style="position: fixed; left: 100px; width: 80%;">
 <p style="position: fixed; left: 100px; top: 300px; width: 50%;">
 Наконец, для этого параграфа указано его фиксированное ...</p>
</div>
```

[positioning.html](#)



# Порядок видимости блоков

Порядок расположения блоков в случае, когда они перекрывают друг друга, определяется атрибутом стиля z-index. Чем больше значение этого атрибута, тем выше находится соответствующий объект в «стопке» перекрывающих друг друга объектов.

```
#advertising {
 width: 350px; top: 100px; left: 50px; padding: 20px;
 position: fixed; z-index: 100;
 background-color: red; opacity: 0.5;
 font-family: arial; font-size: 24pt; }

<body>
 <h1>Это совершенно нормальный заголовок</h1>
 <p>The CSS positioning properties...</p>
 <p>FirePublish is the first multi-platform...</p>
 <p>Anchor Pseudo-classes: A link...</p>
 <div id="advertising">
 <p style="color: white;">Эту песню не задушишь, не убьешь!</p>
 </div>
</body>
```

# Вопросы?



# Web-технологии и web-дизайн

HTML

Язык описания  
Web-страниц

CSS

Язык описания  
представлений



Беркунский Е.Ю., кафедра ИУСТ, НУК  
eugeny.berkunsky@gmail.com  
<http://www.berkut.mk.ua>