



Основы конструкторско-технологической информатики

Верстов Владимир Александрович

E-mail: v.verstov@gmail.com

mob. +7 916 818 86 33



1 Введение в проектирование конструкторско-технологических информационных систем

1.1 ОСНОВЫ ЯЗЫКА HTML



Основные понятия

- HyperText Markup Language — «язык разметки гипертекста»
- Браузер
- Редактор исходного кода



Текстовые редакторы

- Notepad++



- PSPad



- HtmlReader



- EditPlus



Браузеры

- Internet Explorer



- Mozilla Firefox



- Opera



- Google Chrome



- Safari



Структура HTML-документа

- **Тэг:**
 - Открывающий – `< >`
 - Закрывающий – `</ >`
- **Дескриптор:**
 - `<дескриптор>`
- **Контейнер:**
 - `<дескриптор>Контейнер</дескриптор>`
- **Атрибут:**
 - `<дескриптор атрибут=“значение атрибута”>`



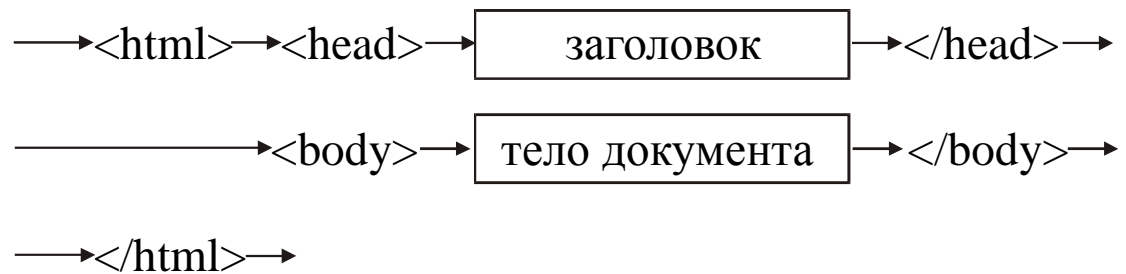
Структура HTML-документа

Условное обозначение	Наименование
N	Целочисленные константы
Char	Строковые константы, состоящие из символов (см. приложение 1 – символы и их обозначения в HTML).
Color	Задание цвета (приложение 2).
URL	Уникальный идентификатор ресурса (см. приложение 3).
FILENAME	Имя файла, это может быть как имя файла на локальном диске (относительное задание пути и наименование), так и его URL адрес (абсолютное задание пути и наименование документа). !!Внимание: Большинство серверов в интернете работают на ОС UNIX клона и имена файлов, указанные в различных регистрах – это не одно и тоже. Пример, index.htm и INDEX.HTM – ЭТО ДВЕ БОЛЬШИЕ РАЗНИЦЫ!!
%% %%	Зарезервированные слова



Структура HTML-документа

- Структура HTML документа
- Синтаксическая диаграмма структуры HTML документа



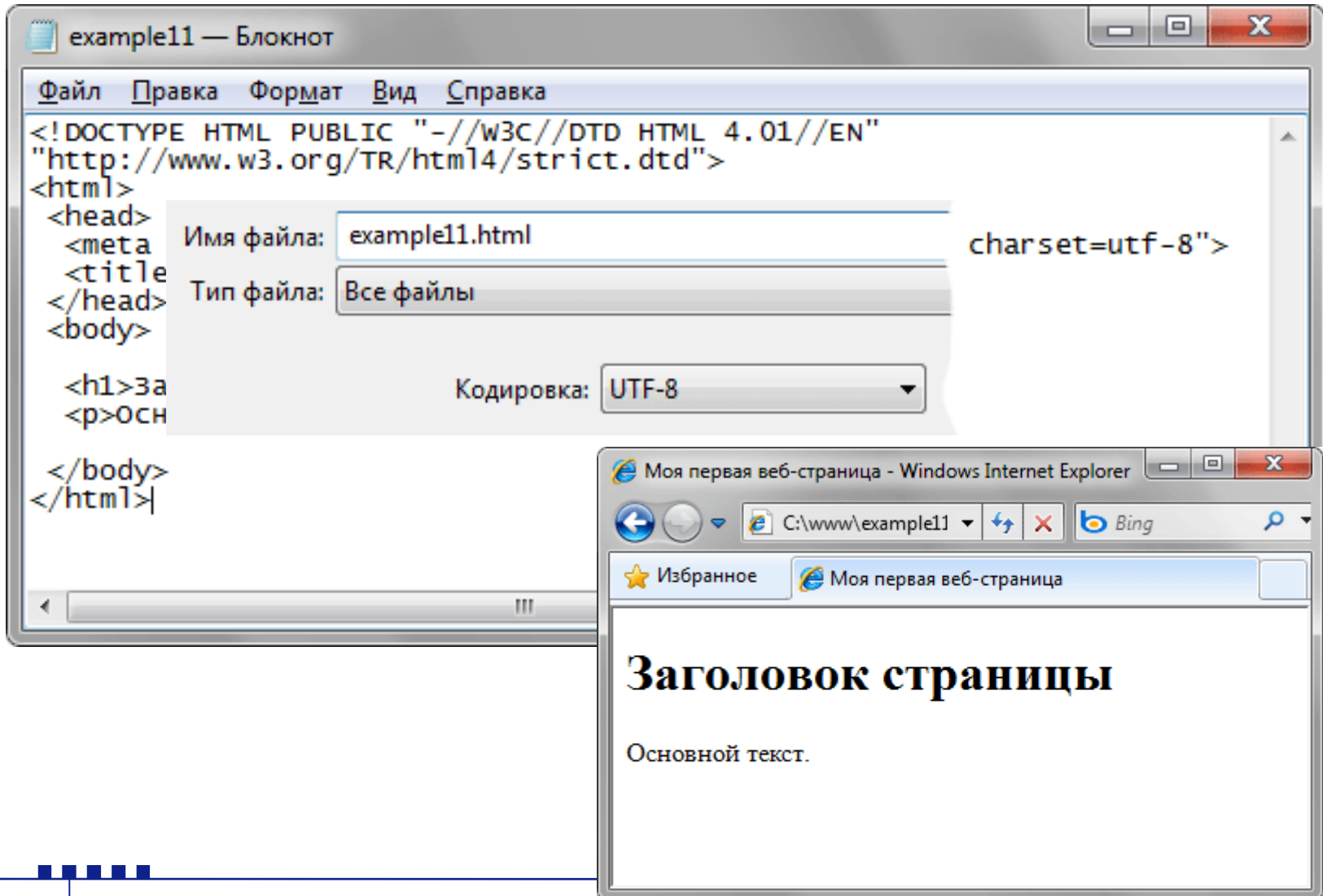
Простейшая веб-страница

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Моя первая веб-страница</title>
  </head>
  <body>

    <h1>Заголовок страницы</h1>
    <p>Основной текст.</p>


  </body>
</html>
```

Создание простейшей веб-страницы



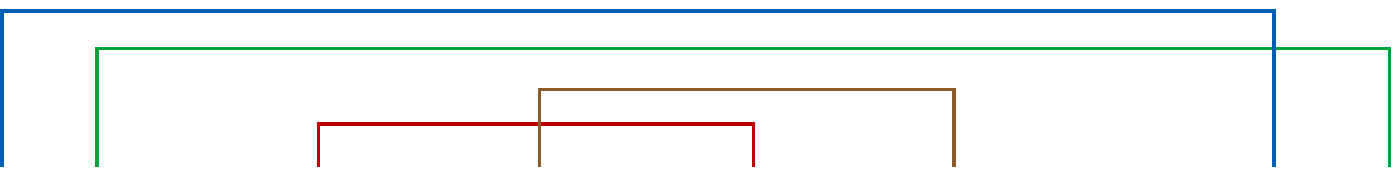
Задание структуры HTML-документа

- Правильное вложение



```
<div><p>Lorem <b>ipsum dolor</b> sit amet...</p></div>
```

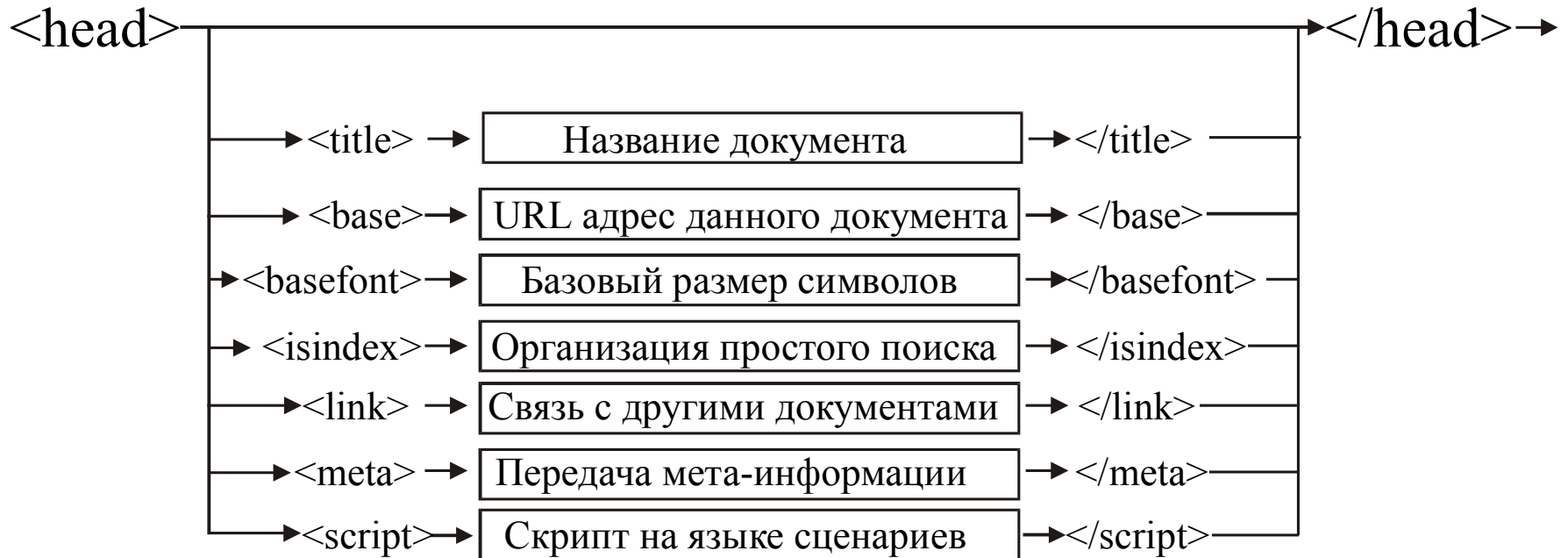
- Неверное вложение



```
<div><p>Lorem <b>ipsum <i>dolor</i> sit</i> amet...</div></p>
```

Задание структуры HTML-документа

- Семантическая диаграмма заголовка HTML-документа

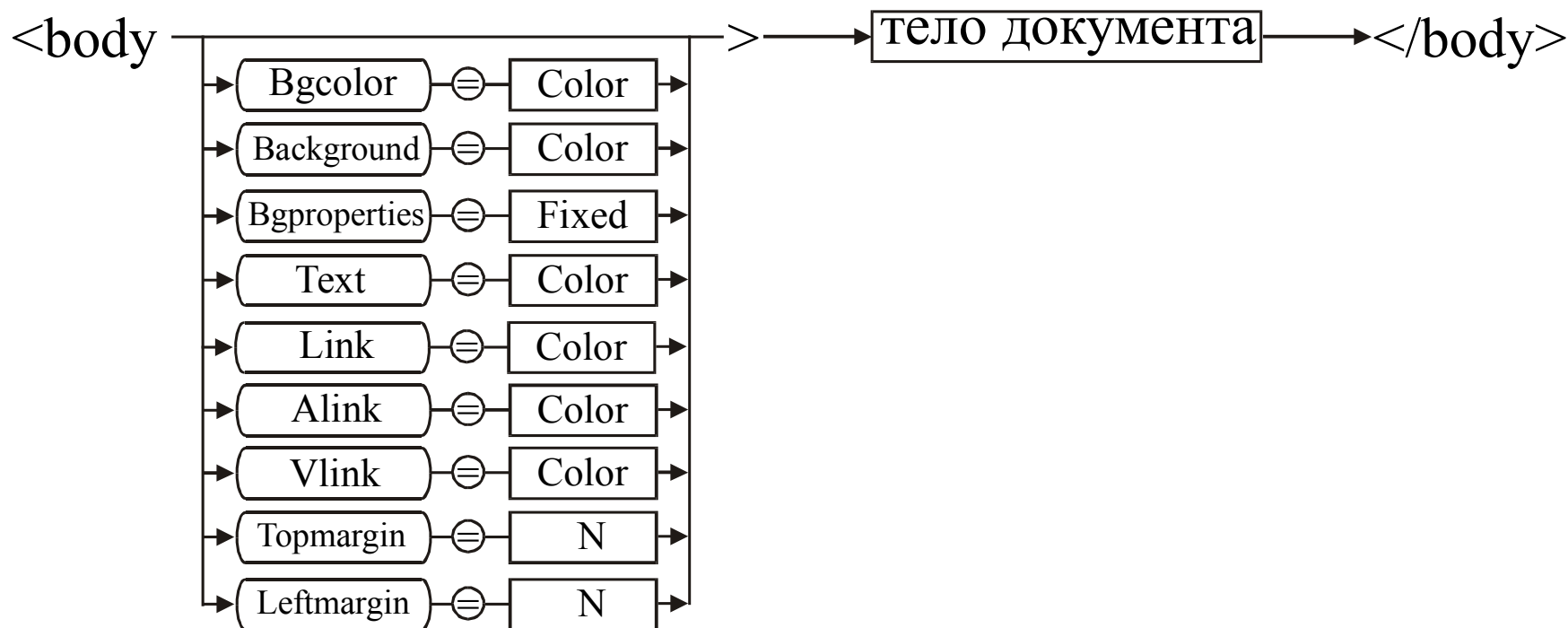


Задание структуры HTML-документа

Спецификация дескриптора <BODY>																									
Дескриптор	Назначение																								
<body></body>	Тело HTML – документа																								
	<table border="1"> <thead> <tr> <th>Атрибут</th> <th>Значение</th> </tr> </thead> <tbody> <tr> <td>bgcolor=color</td> <td>Цвет фона документа</td> </tr> <tr> <td>bgproperties=fixed</td> <td>Фиксирование фоновой картинки в IE.</td> </tr> <tr> <td>background=filename</td> <td>Указание имени графического файла, который будет использован в виде фона документа.</td> </tr> <tr> <td>link=color</td> <td>Цвет гиперссылки</td> </tr> <tr> <td>alink=color</td> <td>Цвет активизированной гиперссылки</td> </tr> <tr> <td>vlink=color</td> <td>Цвет посещенных гиперссылок</td> </tr> <tr> <td>text=color</td> <td>Цвет текста документа по умолчанию</td> </tr> <tr> <td>topmargin=n</td> <td>Отступ от верхнего края</td> </tr> <tr> <td>leftmargin=n</td> <td>Отступ от левого края</td> </tr> <tr> <td>marginheight=n</td> <td>Отступ между таблицами по высоте</td> </tr> <tr> <td>marginwidth=n</td> <td>Отступ между таблицами по ширине</td> </tr> </tbody> </table>	Атрибут	Значение	bgcolor=color	Цвет фона документа	bgproperties=fixed	Фиксирование фоновой картинки в IE.	background=filename	Указание имени графического файла, который будет использован в виде фона документа.	link=color	Цвет гиперссылки	alink=color	Цвет активизированной гиперссылки	vlink=color	Цвет посещенных гиперссылок	text=color	Цвет текста документа по умолчанию	topmargin=n	Отступ от верхнего края	leftmargin=n	Отступ от левого края	marginheight=n	Отступ между таблицами по высоте	marginwidth=n	Отступ между таблицами по ширине
Атрибут	Значение																								
bgcolor=color	Цвет фона документа																								
bgproperties=fixed	Фиксирование фоновой картинки в IE.																								
background=filename	Указание имени графического файла, который будет использован в виде фона документа.																								
link=color	Цвет гиперссылки																								
alink=color	Цвет активизированной гиперссылки																								
vlink=color	Цвет посещенных гиперссылок																								
text=color	Цвет текста документа по умолчанию																								
topmargin=n	Отступ от верхнего края																								
leftmargin=n	Отступ от левого края																								
marginheight=n	Отступ между таблицами по высоте																								
marginwidth=n	Отступ между таблицами по ширине																								

Задание структуры HTML-документа

- Семантическая диаграмма дескриптора «*body*»

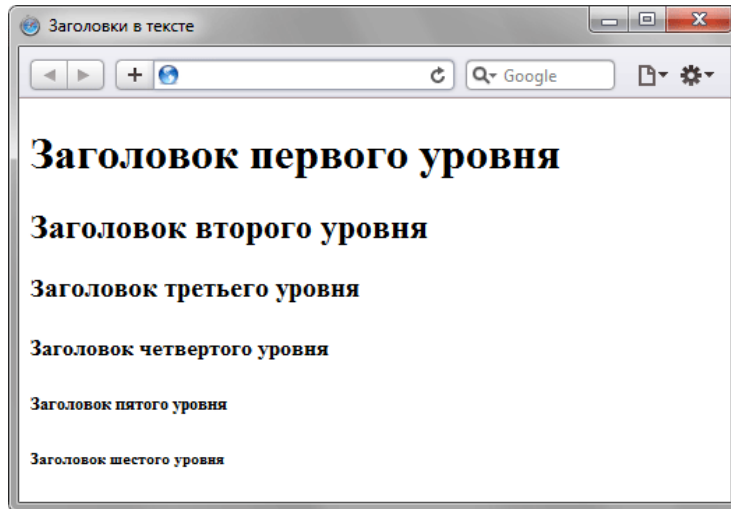


Форматирование текста

- Заголовки

Спецификация дескриптора определения заголовка <Hn>

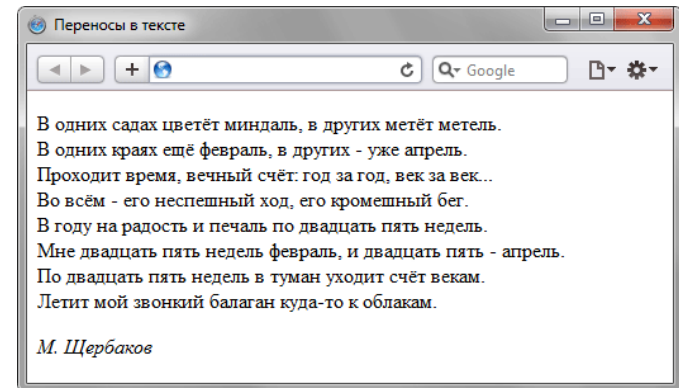
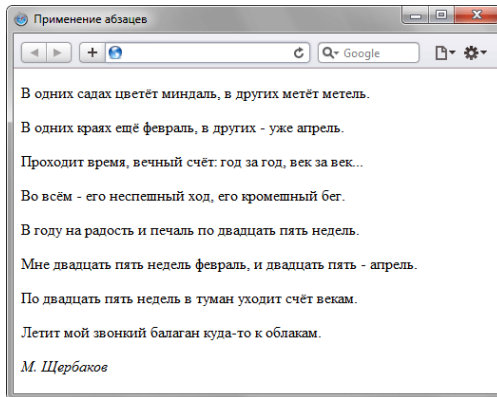
Дескриптор	Назначение	
<Hn></Hn>	Заголовок уровня n, где = 1 ... 6	
	Атрибут	Значение
	align={left, right, center, justify}	Выравнивание заголовка по горизонтали



Форматирование текста

- Абзацы

Спецификация дескрипторов выравнивания текста					
Дескриптор	Назначение				
<code>
</code>	Перенос на новую строку				
<code><nobr></nobr></code>	Неразрывная строка				
<code><wbr></code>	Место в котором браузеру можно осуществлять разрыв текста (практически не используется).				
<code><p></p></code>	Текстовый абзац				
	<table border="1"> <thead> <tr> <th>Атрибут</th> <th>Значение</th> </tr> </thead> <tbody> <tr> <td><code>align={left, right, center, justify}</code></td> <td>Выравнивание заголовка по горизонтали</td> </tr> </tbody> </table>	Атрибут	Значение	<code>align={left, right, center, justify}</code>	Выравнивание заголовка по горизонтали
Атрибут	Значение				
<code>align={left, right, center, justify}</code>	Выравнивание заголовка по горизонтали				



Форматирование текста

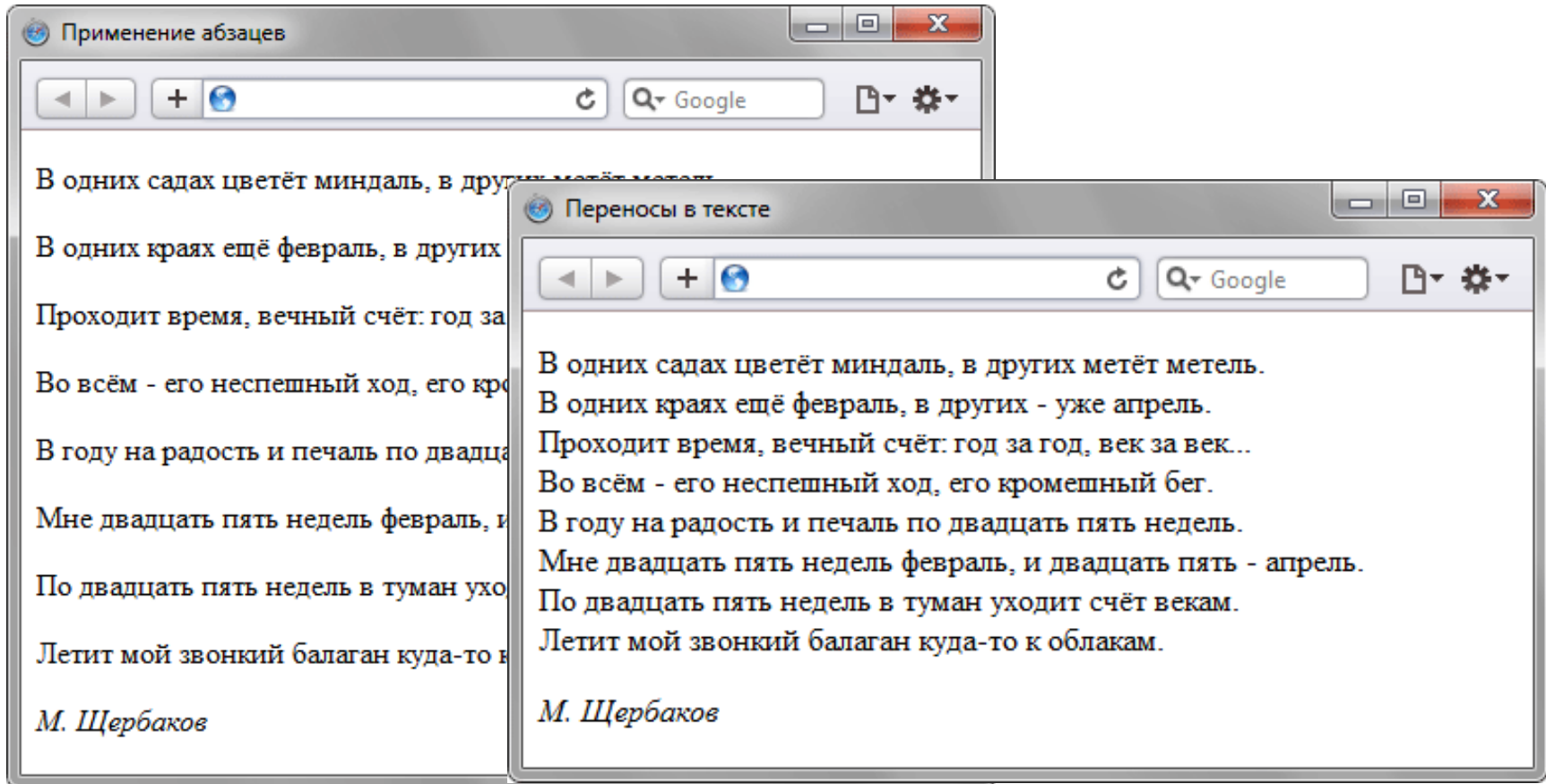
- Управление размещением

Спецификация дескрипторов выравнивания текста					
Дескриптор	Назначение				
<code>
</code>	Перенос на новую строку				
<code><nobr></nobr></code>	Неразрывная строка				
<code><wbr></code>	Место в котором броузеру можно осуществлять разрыв текста (практически не используется).				
<code><p></p></code>	Текстовый абзац				
	<table border="1"> <thead> <tr> <th>Атрибут</th> <th>Значение</th> </tr> </thead> <tbody> <tr> <td><code>align={left, right, center, justify}</code></td> <td>Выравнивание заголовка по горизонтали</td> </tr> </tbody> </table>	Атрибут	Значение	<code>align={left, right, center, justify}</code>	Выравнивание заголовка по горизонтали
Атрибут	Значение				
<code>align={left, right, center, justify}</code>	Выравнивание заголовка по горизонтали				



Форматирование текста

- Использование абзацев и тега `
`



Форматирование текста

- Управление размещением

Спецификация дескрипторов преформатирования		
Дескриптор	Назначение	
<code><pre> </pre></code>	Предварительно отформатированный текст	
	Атрибут	Значение
	<code>Width=n</code>	ширина (в пикселях)
<code><plaintext> </plaintext></code>	Обычный текст (практически не используется).	



Форматирование текста

- Физические стили

Спецификация дескрипторов физических стилей	
Дескриптор	Назначение
<code> </code>	Полужирный шрифт
<code><i> </i></code>	Курсив
<code><tt> </tt></code>	Моноширинный шрифт (пишущая машинка)
<code><u> </u></code>	Подчеркнутый шрифт
<code><sub> </sub></code>	Подстрочный текст (нижний индекс)
<code><sup> </sup></code>	Надстрочный текст (верхний индекс)
<code><s> </s></code> или <code><strike> </strike></code>	Перечеркнутый шрифт
<code><big></code>	Увеличенный шрифт
<code><small></code>	Уменьшенный шрифт
<code><blink></code>	Мерцающий ("моргающий") шрифт (Netscape)

Дескриптор **`<strike>`**, возможно, не будет поддерживаться в последующих версиях.

Форматирование текста

- Логические стили

Спецификация дескрипторов логических стилей	
Дескриптор	Назначение
<code> </code>	Выделенный текст
<code> </code>	Сильно выделенный текст
<code><code> </code></code>	Фрагмент HTML-кода (моноширинный шрифт)
<code><listing> </listing></code>	Фрагмент кода (сохраняет первоначальное форматирование) <i>Устаревший.</i> Аналог <code><pre width=132></code>
<code><dfn> </dfn></code>	Определение
<code><samp> </samp></code>	Пример (аналог фрагмента кода)
<code><xmp> </xmp></code>	Пример. <i>Устаревший.</i> Аналог <code><pre width=80></code>
<code><kbd> </kbd></code>	Название клавиши клавиатуры
<code><var> </var></code>	Переменная или значение
<code><acronym> </acronym></code> или <code><abbr> </abbr></code>	Аббревиатура (акроним) и её расшифровка

Форматирование текста

- **Адреса авторов**

Спецификация дескриптора указания адреса автора	
Дескриптор	Назначение
<code><address> </address></code>	Адрес Web-мастера/дизайнера/автора

- **Цитаты**

Спецификация дескрипторов указания цитат	
Дескриптор	Назначение
<code><blockquote> </blockquote></code>	Длинная цитата
<code><q> </q></code>	Короткая цитата
<code><cite> </cite></code>	Цитата, выделенная курсивом



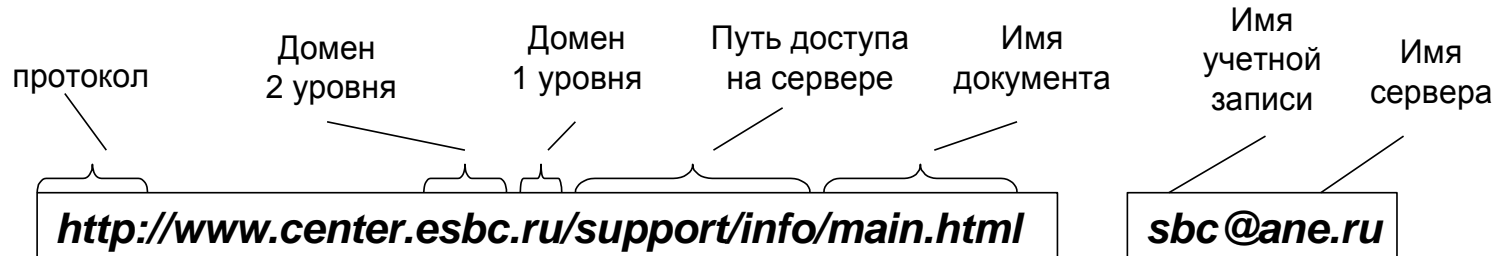
Форматирование шрифтов

Спецификация дескриптора форматирования шрифта

Дескриптор	Назначение	
<code> </code>	Форматирование шрифта для фрагмента текста в контейнере	
	Атрибут	Значение
	<code>Size=n</code>	Размер
	<code>Color=color</code>	Цвет (см. приложение 2)
	<code>Face="FontName"</code>	Гарнитура
	<code>Lang=cod</code>	Определение языка, код указан в приложении 4.

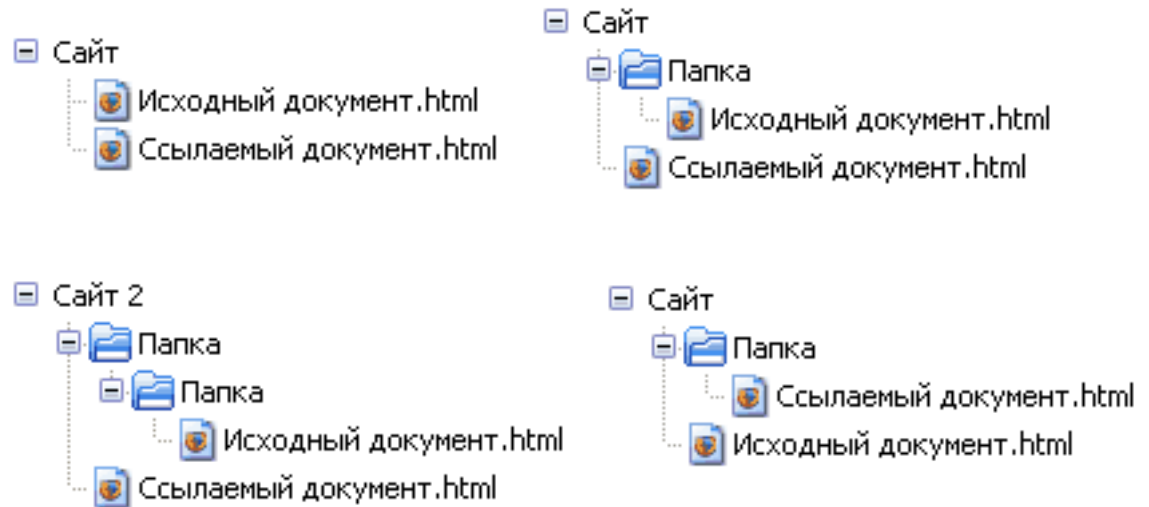
Организация гиперсвязей

• URL



• Ссылки

- Внутренние
- Внешние
- Относительные



Организация гиперсвязей

- Ссылки

Спецификация дескриптора создания гипертекстовой ссылки

Дескриптор	Назначение	
<code><a></code>	Гиперссылка.	
	<code>href=URL</code>	URL целевого документа
	<code>methods = char</code>	Метод обработки связи
	<code>name=char</code>	Локальное имя ссылки
	<code>rel={см. спецификацию}</code>	Связь к целевому документу
	<code>rev={см спецификацию}</code>	Связь от целевого документа
	<code>target=name</code>	Имя целевого фрейма (т.е. фрейма в котором будет открыт указанный в URL документ)
	<code>title=char</code>	Название ссылки

Организация гиперсвязей

- Связь с другими документами

Спецификация дескриптора создания гипертекстовой ссылки

Дескриптор	Назначение	
<code><link></link></code>	Гиперссылка.	
	<code>href=URL</code>	URL целевого документа
	<code>methods = char</code>	Метод обработки связи
	<code>name=char</code>	Локальное имя ссылки
	<code>rel={см. спецификацию}</code>	Связь к целевому документу
	<code>rev={см спецификацию}</code>	Связь от целевого документа
	<code>target=name</code>	Имя целевого фрейма (т.е. фрейма в котором будет открыт указанный в URL ресурс).
	<code>title=char</code>	Название ссылки

Организация гиперсвязей

- Связь с другими документами

Спецификация основных значений атрибутов rel и rev	
Тип связи	Обычная интерпретация: связь с...
Alternate	Другая версия документа
alternate (с атрибутом lang)	Тот же документ на другом языке
alternate (с атрибутом media)	Версия, подготовленная для другого носителя
bookmark	Закладка в документе
Contents	Оглавление
copyright	Информация об авторских правах на данный документ
Glossary	Толковый словарь терминов для данного документа
Help	Справочный документ
Index	Алфавитный указатель данного документа
Next	Документ, следующий сразу за данным документом
previous	Документ, находящийся непосредственно перед данным документом
Start	Первый документ в группе
stylesheet	Внешний лист стилей

Списки

- **Упорядоченные** – элементу списка соответствует запись на новой строке, начинающаяся с маркера.
- **Нумерованные** – элементу списка соответствует абзац, начинающийся с номера.
- **Неупорядоченные маркированные** – элементу списка соответствует абзац, начинающийся с маркера.
- **Список определений** - элементу списка соответствует абзац определения, расположенный с отступом.
- **Список элементов меню** – выглядит в виде неупорядоченного списка.
- **Список элементов каталогов** – предназначен для построения иерархического дерева каталогов и файлов, выглядит, как неупорядоченный список.

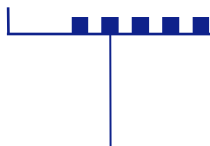


Списки

- **Маркированный список**

Спецификация дескриптора для задания неупорядоченных маркированных списков

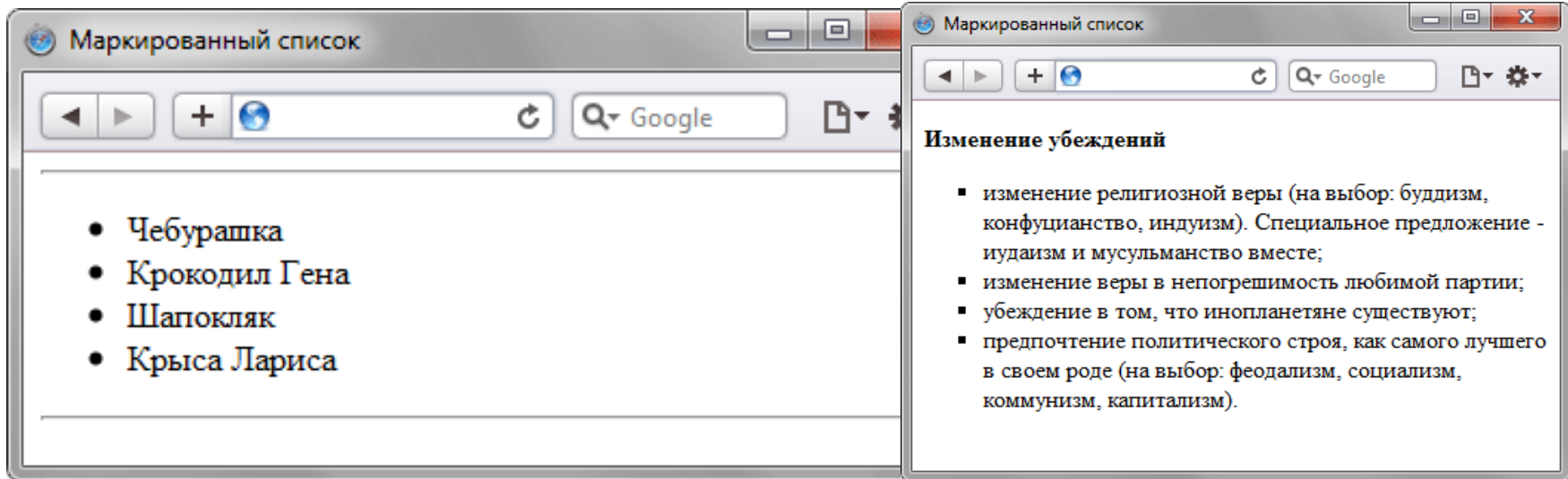
Дескриптор	Назначение	
 	Неупорядоченный список	
	Атрибут	Значение
	Compact	Измененный внешний вид
	Type	Тип вводного символа Disk - Круг Square - Квадрат Circle - Окружность



Списки

• Элемент списка

Спецификация дескриптора задания маркерных элемента списков	
Дескриптор	Назначение
<code> </code>	Элемент списка
Атрибут	Значение
<code>Type</code>	тип вводимого символа элемента списка
<code>Value=number</code>	текущий номер

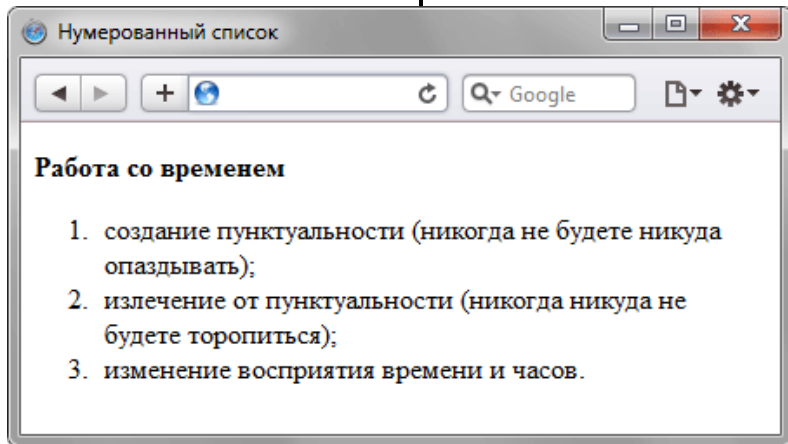


Списки

- **Нумерованный список**

Спецификация дескриптора задания нумерованного элемента списка

Дескриптор	Назначение	
<code> </code>	Нумерованный список	
	Атрибут	Значение
	<code>compact</code>	измененный внешний вид
	<code>start=number</code>	начальный номер
	<code>type=cod</code>	тип нумерации 1- арабские цифры (по умолчанию) i- строчные римские цифры I- прописные римские цифры a- строчные буквы латинского алфавита A- прописные буквы латинского алфавита



Списки

- Список определений

Спецификация дескриптора задания списка определений.

Дескриптор	Назначение	
<code><dl> </dl></code>	Список определений	
	Атрибут	Значение
	<code>compact</code>	измененный внешний вид

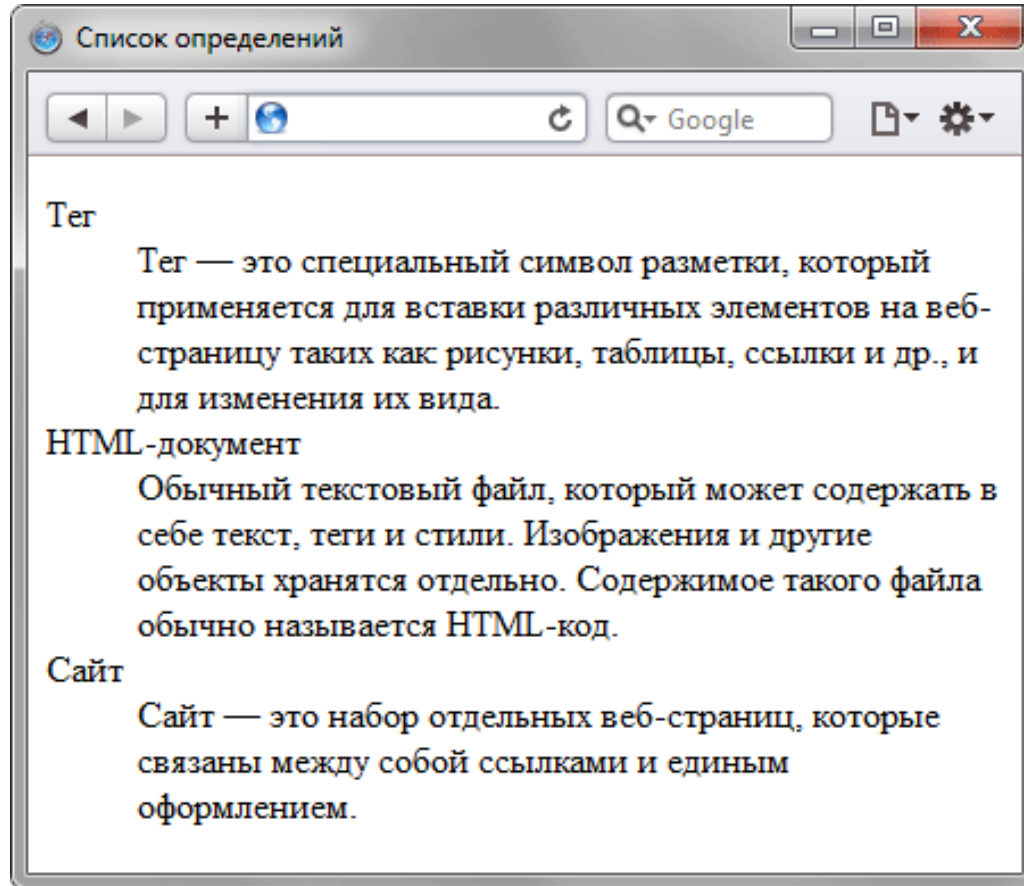
Спецификация дополнительных дескрипторов для списка определений

Дескриптор	Назначение
<code><dt> </dt></code>	Определяемый термин в списке определений
<code><dd> </dd></code>	Текст определения



Списки

- Список определений



Списки

- Меню

Спецификация дескриптора задания списка меню

Дескриптор	Назначение	
<code><menu> </menu></code>	Список элементов меню	
	Атрибут	Значение
	<code>compact</code>	измененный внешний вид
	<code>type</code>	тип вводного символа [Netscape]

- Список каталогов

Спецификация дескриптора задания списка каталогов

Дескриптор	Назначение	
<code><dir> </dir></code>	Каталог	
	Атрибут	Значение
	<code>compact</code>	измененный внешний вид
	<code>Type</code>	тип вводного символа [Netscape]

Таблицы

- Атрибуты таблицы

Спецификация дескриптора тела таблицы	
Дескриптор	Назначение
<table></table>	Задание тела таблицы
Атрибут	Значение
Align ={right center left}	Выравнивание по горизонтали
Bgcolor =color	Цвет фона
Border =N	Ширина рамки (border=0 – ее отсутствие).
Border color =color	Цвет рамки
Border color dark =color	Цвет затененной части рельефной рамки (IE)
Border color light =color	Цвет освещенной части рельефной рамки (IE)
Cellpadding =N	Расстояние между содержимым и рамкой ячеек таблицы
Cellspacing =N	Расстояние между ячейками таблицы
Width =N (%)	Ширина таблицы
Height =N (%)	Высота таблицы
Dir	Направление чтения текста
Lang	Язык

Таблицы

- Атрибуты строк

Спецификация дескриптора задания строк таблицы	
Дескриптор	Назначение
<tr></tr>	Задание тела таблицы
Атрибут	Значение
Align ={ right center left }	Выравнивание по горизонтали
valign ={ top bottom }	Выравнивание по вертикали
Bgcolor =color	Цвет фона
Border =N	Ширина рамки (border=0 – ее отсутствие).
Border color =color	Цвет рамки
Border color dark =color	Цвет затененной части рельефной рамки (IE)
Border color light =color	Цвет освещенной части рельефной рамки (IE)
Width =N (%)	Ширина строки таблицы
Height =N (%)	Высота строки таблицы

Таблицы

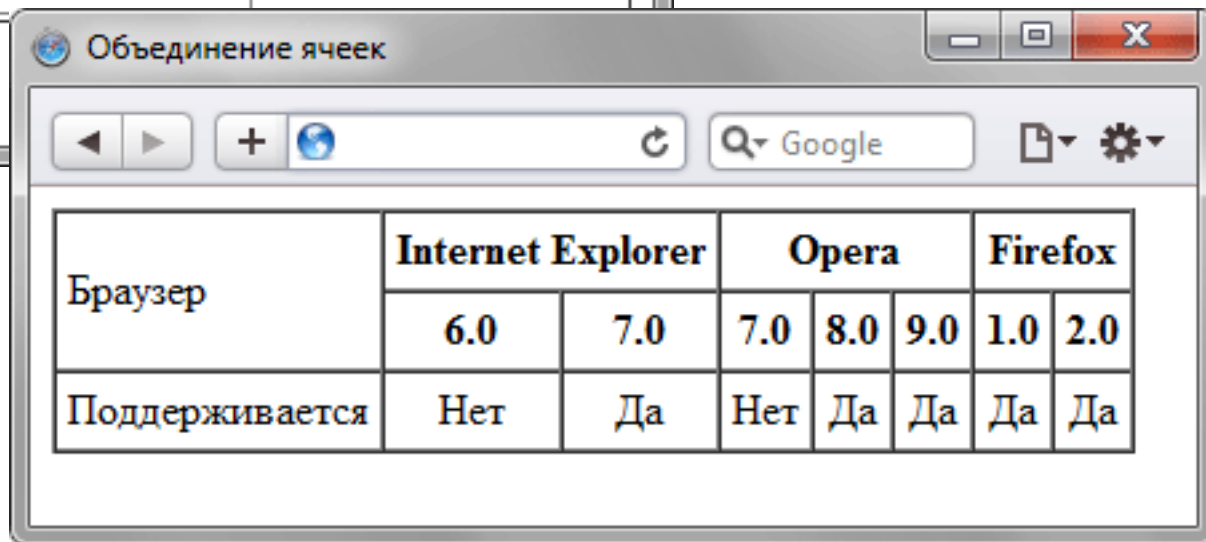
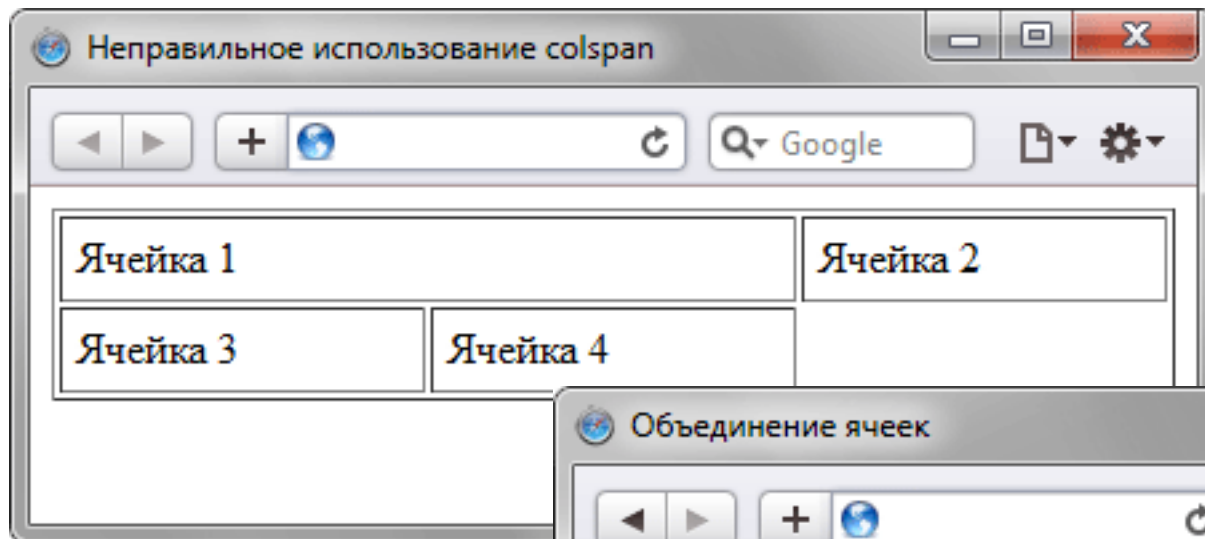
- Атрибуты ячеек

Спецификация дескриптора задания столбцов таблицы

Дескриптор	Назначение	
<td></td>	Задание тела таблицы	
	Атрибут	Значение
	Align ={ right center left }	Выравнивание по горизонтали
	valign ={ top bottom }	Выравнивание по вертикали
	Bgcolor =color	Цвет фона
	Border =N	Ширина рамки (border=0 – ее отсутствие).
	Border color =color	Цвет рамки
	Border color dark =color	Цвет затененной части рельефной рамки (IE)
	Border color light =color	Цвет освещенной части рельефной рамки (IE)
	Width =N (%)	Ширина строки таблицы
	Height =N (%)	Высота строки таблицы
	Colspan =N	Ширина ячейки, выраженная в столбцах
	Nowrap	Выключение разрыва строк
	Rowspan =N	Высота ячейки, выраженная в строках

Таблицы

- Объединение ячеек

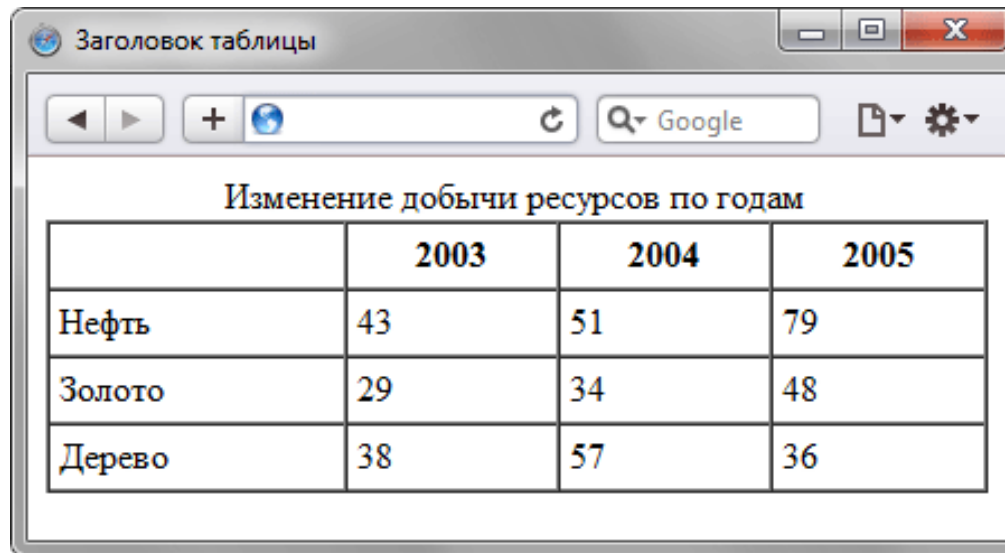


Таблицы

- Заголовок таблицы

Спецификация дескриптора задания названия таблицы

Дескриптор	Назначение
<code><caption> </caption></code>	Элемент списка/меню в форме.
Атрибут	Значение
Align ={ right center left }	Выравнивание по горизонтали
valign ={ top bottom }	Выравнивание по вертикали



Изображения

- **Форматы рисунков:**
 - JPEG – Joint Photographic Experts Group
 - GIF – Graphics Interchange Format
 - PNG – Portable Network Graphics
 - TIFF – Tagged Image File Format



GIF

- **Особенности**

- Количество цветов в изображении может быть от 2 до 256, но это могут быть любые цвета из 24-битной палитры.
- Файл в формате GIF может содержать прозрачные участки. Если используется отличный от белого цвета фон, он будет проглядывать сквозь «дыры» в изображении.
- Поддерживает покадровую смену изображений, что делает формат популярным для создания баннеров и простой анимации.
- Использует свободный от потерь метод сжатия



GIF

- **Область применения**
 - Текст
 - Логотипы
 - Иллюстрации с четкими краями
 - Анимированные рисунки
 - Изображения с прозрачными участками
 - Баннеры



JPEG

- **Особенности**

- Количество цветов в изображении — около 16 миллионов, что вполне достаточно для сохранения фотографического качества изображения.
- Основная характеристика формата — качество, позволяющее управлять конечным размером файла.
- Поддерживает технологию, так называемый прогрессивный JPEG, в котором версия рисунка с низким разрешением появляется в окне просмотра до полной загрузки самого изображения.

- **Область применения**

- Фотографии



PNG-8

- **Особенности**

- Использует 8-битную палитру (256 цветов) в изображении, за что и получил в своем названии цифру восемь. При этом можно выбирать, сколько цветов будет сохраняться в файле — от 2 до 256.
- В отличие от GIF, не отображает анимацию ни в каком виде.

- **Область применения**

- Текст
- Логотипы
- Иллюстрации с четкими краями



PNG-24

- **Особенности**

- Использует примерно 16,7 млн. цветов в файле, из-за чего этот формат применяется для полноцветных изображений.
- Поддерживает многоуровневую прозрачность, это позволяет создавать плавный переход от прозрачной области изображения к цветной, так называемый градиент.
- Из-за того, что используемый алгоритм сжатия сохраняет все цвета и пиксели в изображении неизменными, если сравнивать с другими форматами, то у PNG-24 конечный объем графического файла получается наибольшим.

- **Область применения**

- Фотографии
- Рисунки



Изображения

- **Добавление рисунка**
 - ``
- **Примеры размещения рисунка**



- **Изменение размеров рисунка**
 - Атрибут «width»
``
 - Атрибут «height»
``

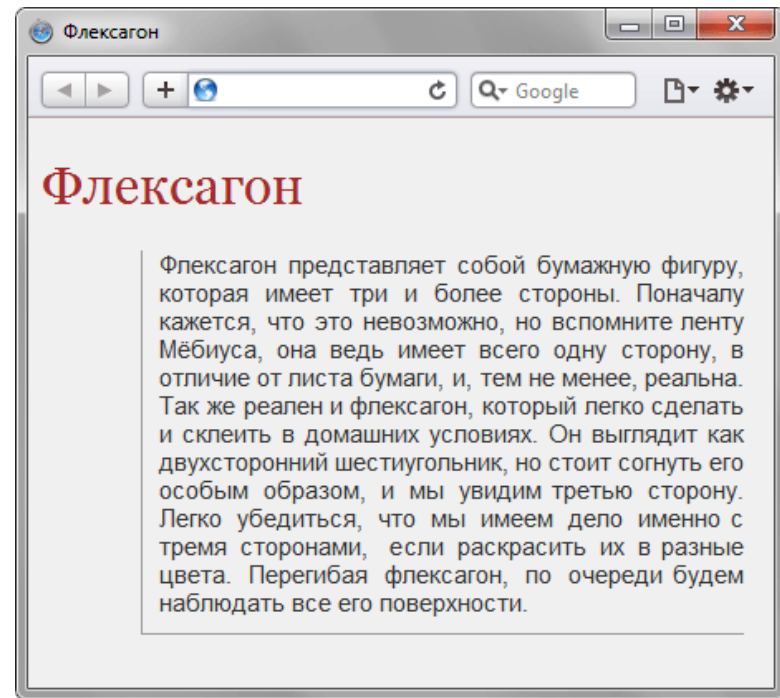
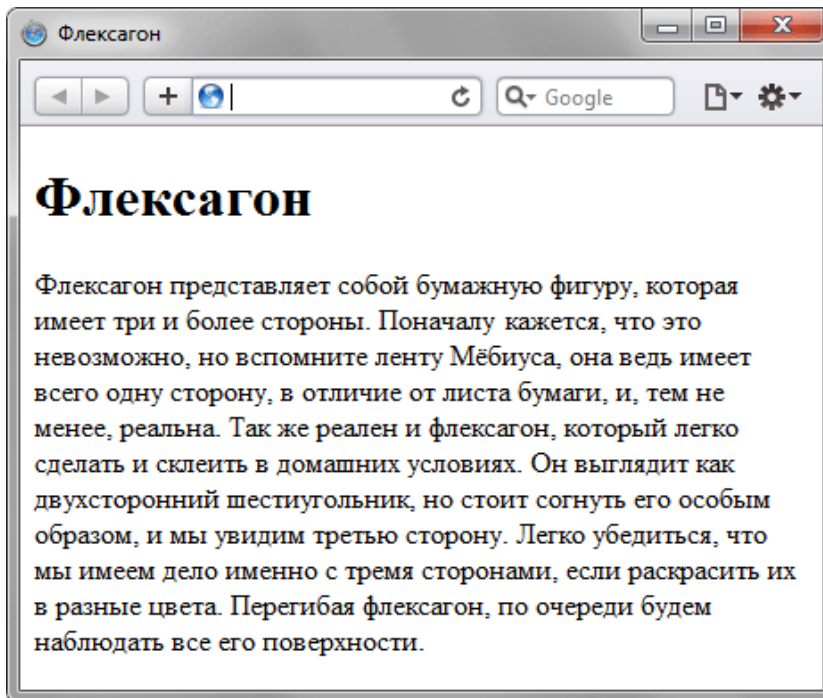
1 Введение в проектирование конструкторско-технологических информационных систем

1.2 ОСНОВЫ CSS



Основные понятия

- CSS (англ. Cascading Style Sheets — каскадные таблицы стилей)



Преимущества CSS

- Разграничение кода и оформления
- Разное оформление для разных устройств
- Расширенные по сравнению с HTML способы оформления элементов
- Ускорение загрузки сайта
- Единое стилевое оформление множества документов
- Централизованное хранение



Способы подключения CSS

- Связанные стили
 - `<link rel="stylesheet" type="text/css" href="mysite.css">`
- Глобальные стили
 - `<style type="text/css" >>`
H1 { font-size: 120%; }
`</style>`
- Внутренние стили
 - `<p style="font-size: 120%; color: #cd66cc">`
Пример текста
`</p>`



Базовый синтаксис

- Селектор

- Селектор { свойство1: значение; свойство2: значение; }

```
h2 {  
  color: olive;  
  border-bottom: 2px solid black;  
}
```

- Комментарии

```
div {  
  width: 200px; /* Ширина блока */  
  margin: 10px; /* Поля вокруг элемента */  
  float: left; /* Обтекание по правому краю */  
}
```



Базовый синтаксис

- Селектор

- Селектор { свойство1: значение; свойство2: значение; }

```
h2 {  
  color: olive;  
  border-bottom: 2px solid black;  
}
```

- Комментарии

```
div {  
  width: 200px; /* Ширина блока */  
  margin: 10px; /* Поля вокруг элемента */  
  float: left; /* Обтекание по правому краю */  
}
```



Значение стилевых свойств

- Строки

```
'Гостиница "Турист"'  
"Гостиница "Турист"  
"Гостиница \"Турист\""
```

- Числа

```
p {  
  font-weight: 600; /* Жирное начертание */  
  line-height: 1.2; /* Межстрочный интервал */  
}
```

- Проценты

```
TABLE {  
  width: 100%; /* Ширина таблицы в процентах */  
  background: #f0f0f0; /* Цвет фона */  
}
```



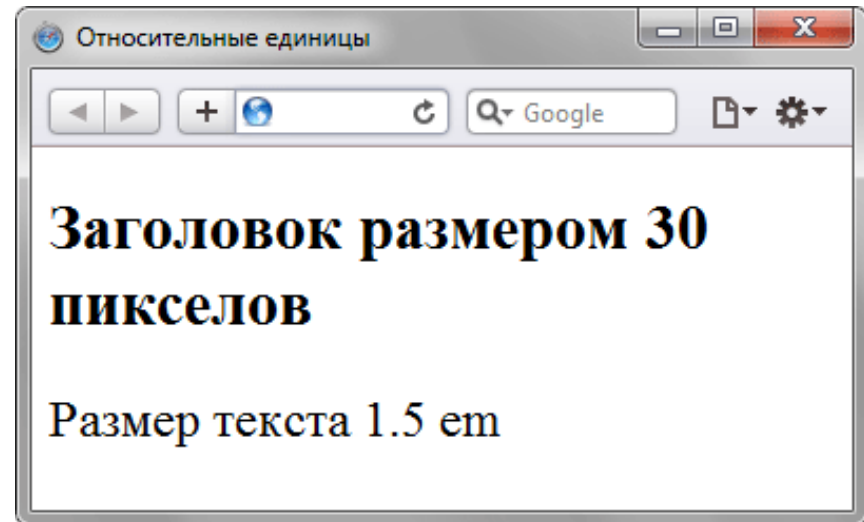
Значение стилевых свойств

- Относительные размеры

Единица	Описание
em	Размер шрифта текущего элемента
ex	Высота символа x
px	Пиксел
%	Процент

- Пример

```
H1 { font-size: 30px; }  
P { font-size: 1.5em; }
```



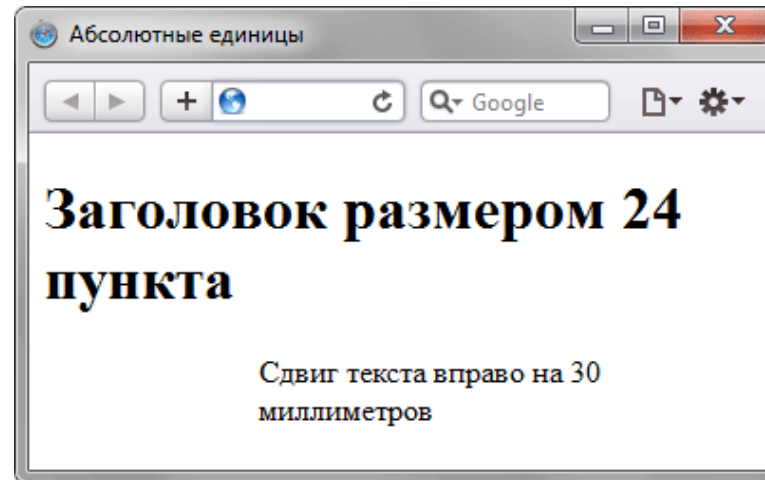
Значение стилевых свойств

- Относительные размеры

Единица	Описание
in	Дюйм (1 дюйм равен 2,54 см)
cm	Сантиметр
mm	Миллиметр
pt	Пункт (1 пункт равен 1/72 дюйма)
pc	Пика (1 пика равна 12 пунктам)

- Примеры

```
H1 { font-size: 24pt; }  
P { margin-left: 30mm; }
```



Значение стилевых свойств

- Цвет

```
BODY {  
  background-color: #3366CC; /* Цвет фона веб-страницы */  
}  
H1 {  
  background-color: RGB(249, 201, 16); /* Цвет фона под заголовком */  
}  
P {  
  background-color: maroon; /* Цвет фона под текстом абзаца */  
  color: white; /* Цвет текста */  
}
```

- Ключевые слова

Правильно: P { text-align: right; }

Неверно: P { text-align: "right"; }

Значение стилевых свойств

- Адреса

```
body {  
  background: url('http://webimg.ru/images/156_1.png') no-repeat;  
}
```

```
div {  
  background: url(images/warning.png) no-repeat;  
  padding-left: 20px;  
  margin-left: 200px;  
}
```



Классы

- Тег.Имя класса { свойство1: значение; свойство2: значение; ... }

```
P.cite { /* Абзац с классом cite */
  color: navy; /* Цвет текста */
  margin-left: 20px; /* Отступ слева */
  border-left: 1px solid navy; /* Граница слева от текста */
  padding-left: 15px; /* Расстояние от линии до текста */
}
```

- .Имя класса { свойство1: значение; свойство2: значение; ... }

```
.term {
  border-bottom: 1px dashed red; /* Подчеркивание под текстом */
}
```



Идентификаторы

- #Имя идентификатора { свойство1: значение; свойство2: значение; ... }

– CSS

```
#help {  
  position: absolute; /* Абсолютное позиционирование */  
  left: 160px; /* Положение элемента от левого края */  
  top: 50px; /* Положение от верхнего края */  
  width: 225px; /* Ширина блока */  
  padding: 5px; /* Поля вокруг текста */  
  background: #f0f0f0; /* Цвет фона */  
}
```

– HTML

```
<div id="help">
```

Классы VS идентификаторы

- **Идентификаторы**

- В коде документа каждый идентификатор уникален и должен быть включён лишь один раз.
- Имя идентификатора чувствительно к регистру.
- Через метод `getElementById` можно получить доступ к элементу по его идентификатору и изменить свойства элемента.
- Стиль для идентификатора имеет приоритет выше, чем у классов.

Классы VS идентификаторы

- **Классы**

- Классы могут использоваться в коде неоднократно.
- Имена классов чувствительны к регистру.
- Классы можно комбинировать между собой, добавляя несколько классов к одному тегу.



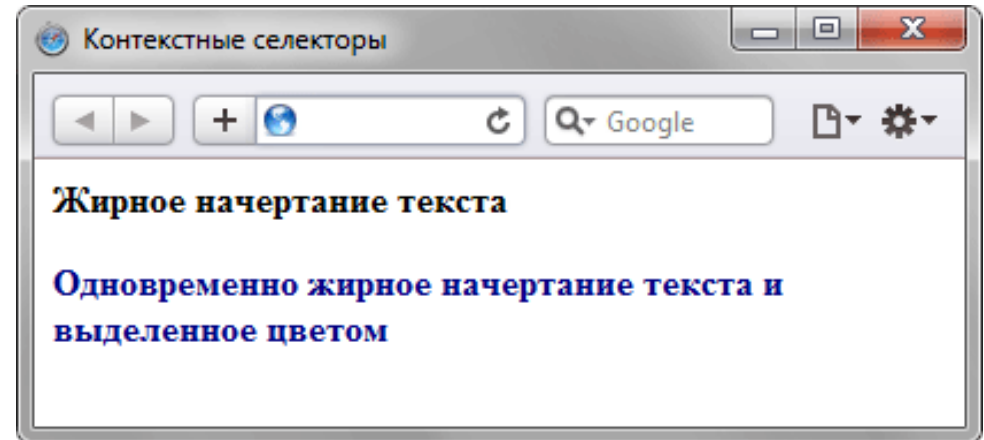
Контекстные селекторы

- Описание
 - Тег1 Тег2 { ... }
- Применение

```
<Тег1>  
    <Тег2> ... </Тег2>  
</Тег1>
```

- Пример

```
P B {  
    font-family: Times, serif; /* Семейство шрифта */  
    font-weight: bold; /* Жирное начертание */  
    color: navy; /* Синий цвет текста */  
}
```



Контекстные селекторы и классы

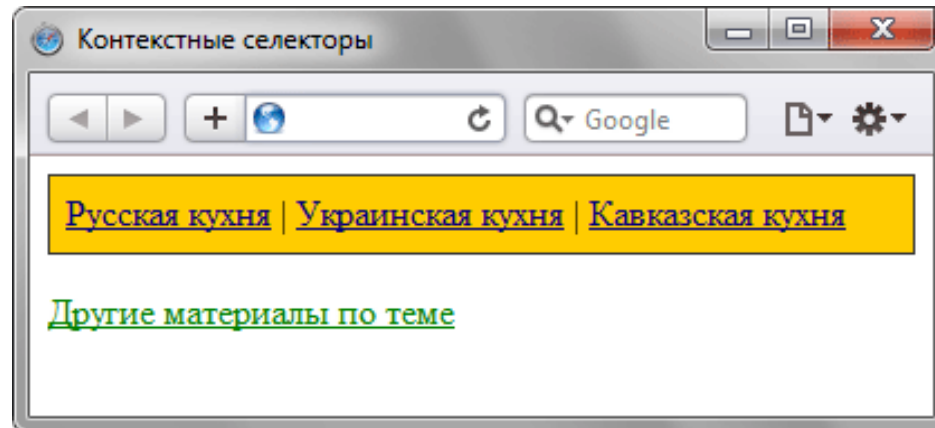
- Пример

```
A {  
  color: green; /* Зеленый цвет текста для всех ссылок */  
}  
.menu {  
  padding: 7px; /* Поля вокруг текста */  
  border: 1px solid #333; /* Параметры рамки */  
  background: #fc0; /* Цвет фона */  
}  
.menu A {  
  color: navy; /* Темно-синий цвет ссылок */  
}
```

Контекстные селекторы и классы

- Пример

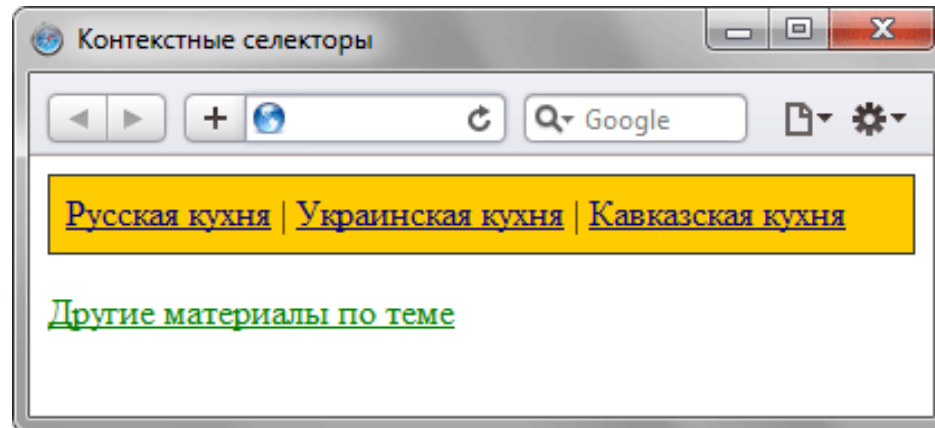
```
<body>  
  <div class="menu">  
    <a href="1.html">Русская кухня</a> |  
    <a href="2.html">Украинская кухня</a> |  
    <a href="3.html">Кавказская кухня</a>  
  </div>  
  <p><a href="text.html">Другие материалы по теме</a></p>  
</body>
```



Соседние селекторы

- Пример

```
<body>  
  <div class="menu">  
    <a href="1.html">Русская кухня</a> |  
    <a href="2.html">Украинская кухня</a> |  
    <a href="3.html">Кавказская кухня</a>  
  </div>  
  <p><a href="text.html">Другие материалы по теме</a></p>  
</body>
```



Виды селекторов

- Соседние селекторы

Селектор 1 + Селектор 2 { Описание правил стиля }

- Дочерние селекторы

Селектор 1 > Селектор 2 { Описание правил стиля }

- Селекторы атрибутов

- Атрибут

[атрибут] { Описание правил стиля }

Селектор[атрибут] { Описание правил стиля }

- Атрибут со значением

[атрибут="значение"] { Описание правил стиля }

Селектор[атрибут="значение"] { Описание правил стиля }

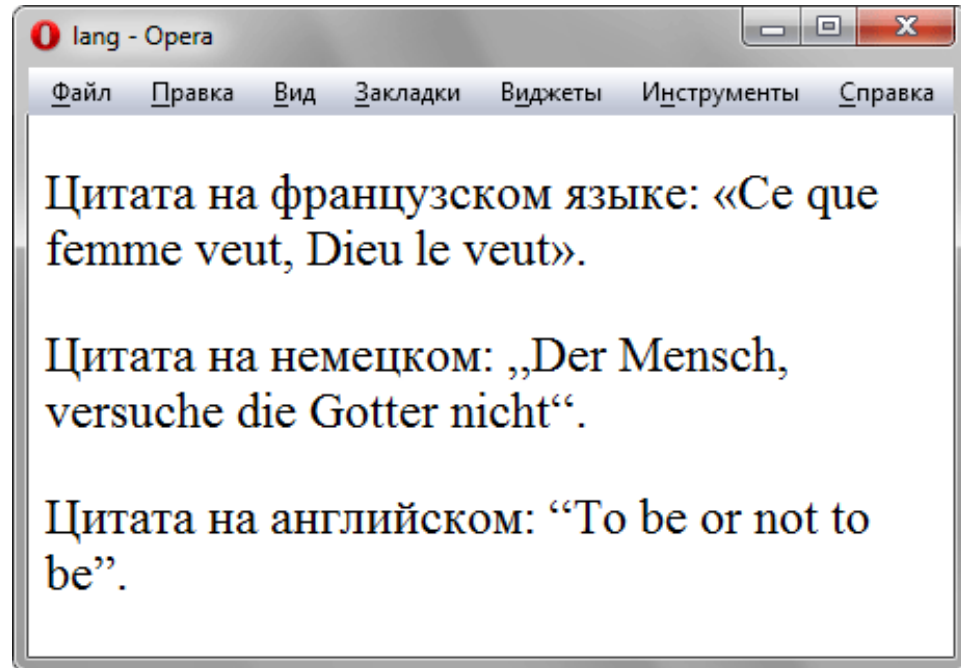
Псевдоклассы

- Синтаксис

Селектор:Псевдокласс { Описание правил стиля }

- Виды

- :active
- :link
- :focus
- :hover
- :visited
- :first-child
- :lang



Псевдоэлементы

- Синтаксис

Селектор:Псевдоэлемент { Описание правил стиля }

- Виды

- :after

- :before

```
P.new:after {  
    content: " - Новьё!"; /* Добавляем после текста абзаца */  
}
```

- :first-letter

- :first-line

```
.foo:first-letter { color: red }  
.foo:first-line { font-style: italic }
```



1 Введение в проектирование конструкторско-технологических информационных систем

1.3 Основы РНР

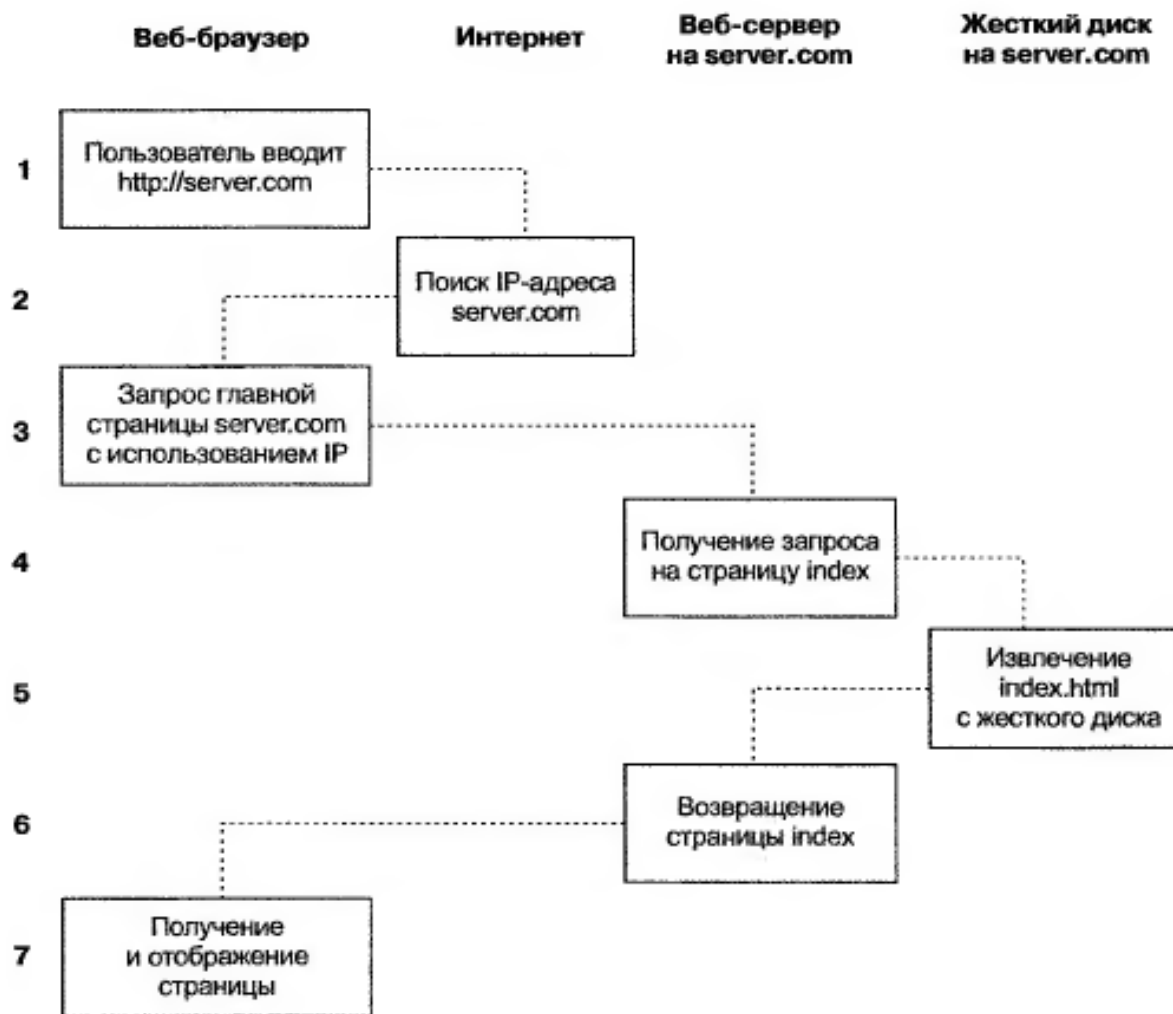


1 Введение в проектирование конструкторско-технологических информационных систем

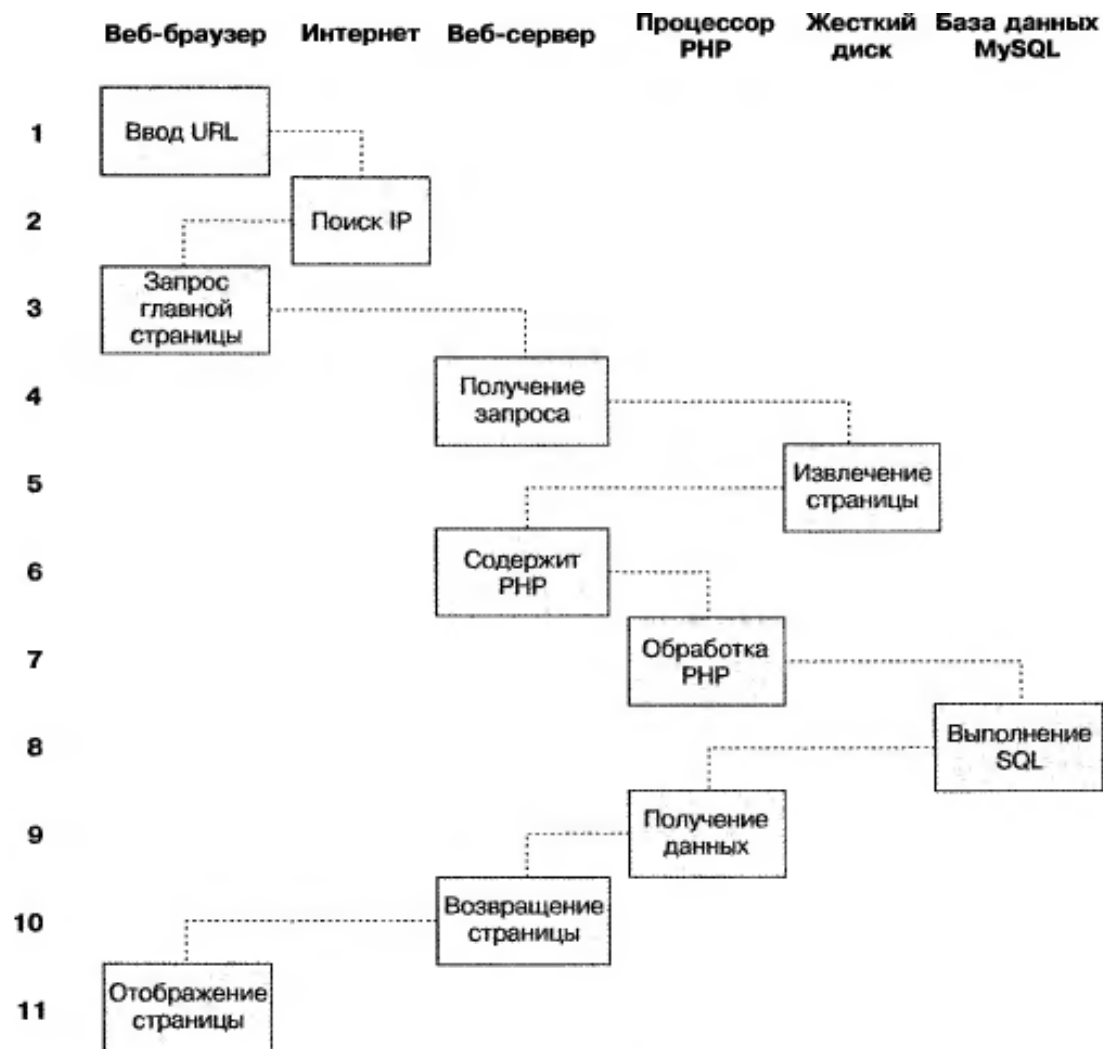
1.3.1 Введение в динамическое содержимое web-страницы



Процедура «запрос-ответ»



Процедура «запрос-ответ»



Формы

- Задание формы

Спецификация дескриптора задания форм	
Дескриптор	Назначение
<form> </form>	Формуляр
	Атрибут Значение
	Action URL-адрес для отправки заполненного формуляра
	Enctype кодирование передаваемых данных
	Method способ передачи формуляра

Формы

- Поля ввода

Спецификация дескриптора задания форм	
Дескриптор	Назначение
<code><input> </input></code>	Поле для ввода строки
Атрибут	Значение
Action= URL	URL адрес для отправки заполненной формы
Align= bottom	выравнивает нижний край кнопки по базовой линии строки
left	выравнивает кнопку-иллюстрацию по левому краю текста
middle	центрирует кнопку-иллюстрацию в текстовой строке
right	выравнивает кнопку-иллюстрацию по правому краю текста
top	выравнивает верхний край кнопки-иллюстрации по верхней линии строки текста

Формы

- Поля ввода

Спецификация дескриптора задания форм	
Дескриптор	Назначение
<code><input> </input></code>	Поле для ввода строки
Атрибут	Значение
Checked	Установленный флажок или выбранное положение переключателя
Maxlength	Максимальная длина вводимых текстов
Name	Название элемента ввода/управления
Src	Источник графического файла картинки кнопки
Value	Установленное по умолчанию значение
Type =	Тип элемента управления /ввода
checkbox	флажки, независимы друг от друга, их можно установить/сбросить в любой комбинации поле ввода для имени файла,

Формы

- Поля ввода

Спецификация дескриптора задания форм											
Дескриптор	Назначение										
<code><input> </input></code>	Поле для ввода строки										
	<table border="1"> <thead> <tr> <th>Атрибут</th> <th>Значение</th> </tr> </thead> <tbody> <tr> <td>Type =</td> <td>Тип элемента управления /ввода</td> </tr> <tr> <td>file</td> <td>рядом отображается кнопка, "Пролистать/Browse" открывающая стандартное диалоговое окно выбора файла</td> </tr> <tr> <td>image</td> <td>пользовательская командная кнопка вместо стандартных, получаемых с помощью type=submit или type=reset</td> </tr> <tr> <td>hidden</td> <td>параметры передаваемые на сервер, которые не могут быть изменены пользователем</td> </tr> </tbody> </table>	Атрибут	Значение	Type =	Тип элемента управления /ввода	file	рядом отображается кнопка, "Пролистать/Browse" открывающая стандартное диалоговое окно выбора файла	image	пользовательская командная кнопка вместо стандартных, получаемых с помощью type=submit или type=reset	hidden	параметры передаваемые на сервер, которые не могут быть изменены пользователем
Атрибут	Значение										
Type =	Тип элемента управления /ввода										
file	рядом отображается кнопка, "Пролистать/Browse" открывающая стандартное диалоговое окно выбора файла										
image	пользовательская командная кнопка вместо стандартных, получаемых с помощью type=submit или type=reset										
hidden	параметры передаваемые на сервер, которые не могут быть изменены пользователем										

Формы

- Поля ввода

Спецификация дескриптора задания форм		
Дескриптор	Назначение	
<code><input> </input></code>	Поле для ввода строки	
	Атрибут	Значение
	Типе =	Тип элемента управления /ввода
	password	текстовое поле, вводимые данные отображаются "звездочками"
	radio	селекторные кнопки (переключатели), из группы можно выбрать только одну
	reset	командная кнопка, возвращает формуляр к исходному состоянию; данные не пересылаются
	submit	командная кнопка, отправляет на сервер всё внесенное в формуляр
	text	однострочное текстовое поле

Формы

- Поля ввода многострочных текстов

Спецификация дескриптора ввода многострочного текстового поля	
Дескриптор	Назначение
<textarea></textarea>	Многострочное поле для ввода текста в форме
Атрибут	Значение
Cols=N	Количество символов /столбцов в поле ввода
Name=char	Имя поля ввода
Rows=N	Количество строк поля ввода
Wrap= off	верстка не выполняется (сервер получает текст одним куском)
physical	автоматическая верстка, с переносом строк по мере достижения правого края (сервер получает текст с разрывами строк)
virtual	автоматическая верстка, с переносом строк по мере достижения правого края (сервер получает текст одной строкой - без разрывов)

Формы

- Выпадающие меню и поля списков

Спецификация дескриптора меню или поля списка в форме		
Дескриптор	Назначение	
<code><select></select></code>	Меню или поле списка в форме	
	Атрибут	Значение
	Multiple	Возможность выбора нескольких опций
	Name=char	Имя элемента
	Size=N	Количество одновременно отображаемых элементов

Спецификация дескриптора задания элемента списка/меню в форме		
Дескриптор	Назначение	
<code><option> </option></code>	Элемент списка/меню в форме.	
	Атрибут	Значение
	Selection	Выбран по умолчанию
	Value	Параметры элемента

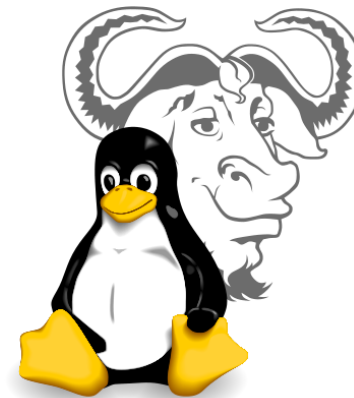
1 Введение в проектирование конструкторско-технологических информационных систем

1.3.2 Установка сервера для разработки

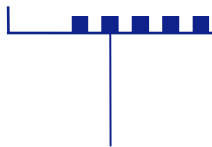


Сервер для разработки

- WAMP
 - Windows
 - Apache
 - MySQL
 - PHP
- LAMP
 - Linux
 - ...
- MAMP
 - Mac OS
 - ...



OS X



WAMP

- easyphp.org
- <http://www.apachefriends.org/ru/xampp.html><http://www.apachefriends.org/ru/xampp.html>
- <http://www.wampserver.com/ru/>
- <http://glossword.biz/glosswordwamp/>



Проверка установки

- `http://localhost/`
- `http://127.0.0.1`



Работа с удаленным сервером

- Вход в систему
 - Протокол ssh
 - Программа Putty
- Работа с файлами
 - Протокол ftp
 - Программа TotalCommander и др.



Среда разработки – текстовый редактор

- Notepad++
 - <http://notepad-plus-plus.org/>
- Editra
 - <http://editra.org>



Среда разработки – IDE

- Eclipse PDT
 - <http://www.eclipse.org/projects/project.php?id=tools.pdt>
- NetBeans
 - <http://netbeans.org/>
- PHPEclipse
 - <http://www.phpeclipse.com/>





1 Введение в проектирование конструкторско-технологических информационных систем

1.3.3 Введение в РНР



Определения

- Hypertext Preprocessor (PHP)— препроцессор гипертекста, скриптовый язык программирования общего назначения, интенсивно применяемый для разработки веб-приложений.
- Скриптовый язык – язык программирования, разработанный для записи «сценариев», последовательностей операций, которые пользователь может выполнять на компьютере.
- Интерпретируемый язык программирования – язык программирования, в котором исходный код исполняется с помощью специальной программы-интерпретатора.

Включение php

- Расширение файла - .php
- Вызов парсера php
 - `<?php /* php код */ ?>`
 - `<? /* php код */ ?>`



Классический пример «Hello, world!»

```
<html>
  <head>
    <title>Пример</title>
  </head>
  <body>
    <?php echo "Hello, world!"; ?>
  </body>
</html>
```



Включение php

- Расширение файла - .php
- Вызов парсера php
 - `<?php /* php код */ ?>`
 - `<? /* php код */ ?>`



Структура php

- Использование комментариев

```
// Это комментарий  
// echo "X equals $x";  
$x += 10; // Увеличение значения $x на 10
```

```
<?php  
/* Это область  
   многострочного комментария,  
   которая не будет  
   подвергаться интерпретации */  
?>
```

Основной синтаксис PHP

- Точки с запятыми

```
$x += 10;
```

- Символ \$

```
<?php  
$mycounter = 1;  
$mystring = "Hello";  
$myarray = array("One", "Two", "Three");  
?>
```

Определение переменной

- **Переменная** – поименованная, либо адресуемая иным способом область памяти, адрес которой можно использовать для осуществления доступа к данным.
- **Значение переменной** – данные, находящиеся в переменной (то есть по данному адресу памяти).



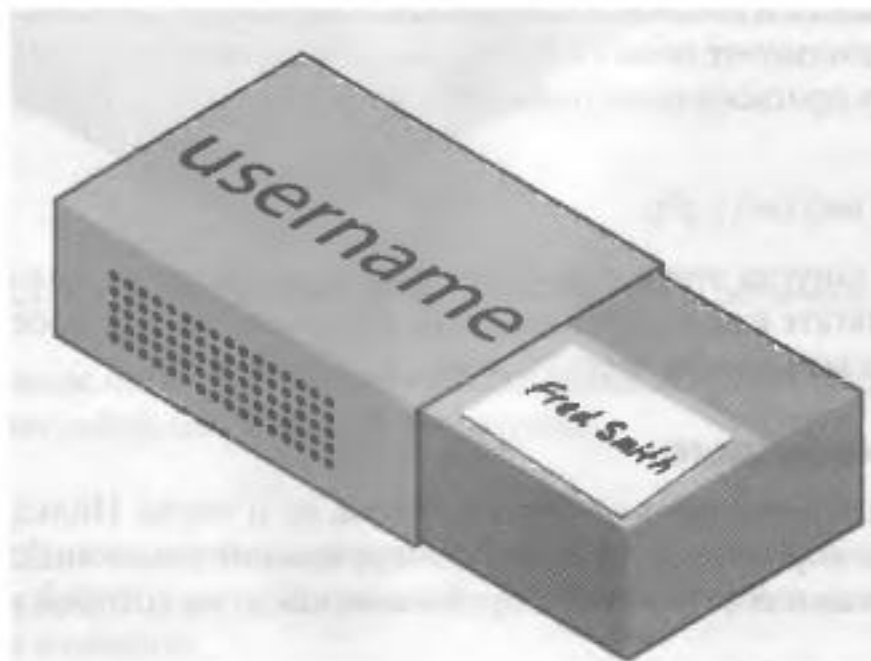
Осмысление переменных

- Строковые переменные

```
$username = "Fred Smith";
```

```
echo $username;
```

```
$current_user = $username;
```



Осмысление переменных

- Числовые переменные

```
$count = 17;
```

```
$count = 17.5;
```



Осмысление переменных. Массивы

- **Массив** – именованный набор однотипных переменных, расположенных в памяти непосредственно друг за другом, доступ к которым осуществляется по индексу (в отличие от списка).
- **Индекс массива** — целое число, либо значение типа, приводимого к целому, указывающее на конкретный элемент массива.
- Массивы:
 - Одномерные
 - Двумерные
 - Многомерные



Осмысление переменных

- Массивы

```
$team = array('Bill', 'Joe', 'Mike', 'Chris', 'Jim');
```



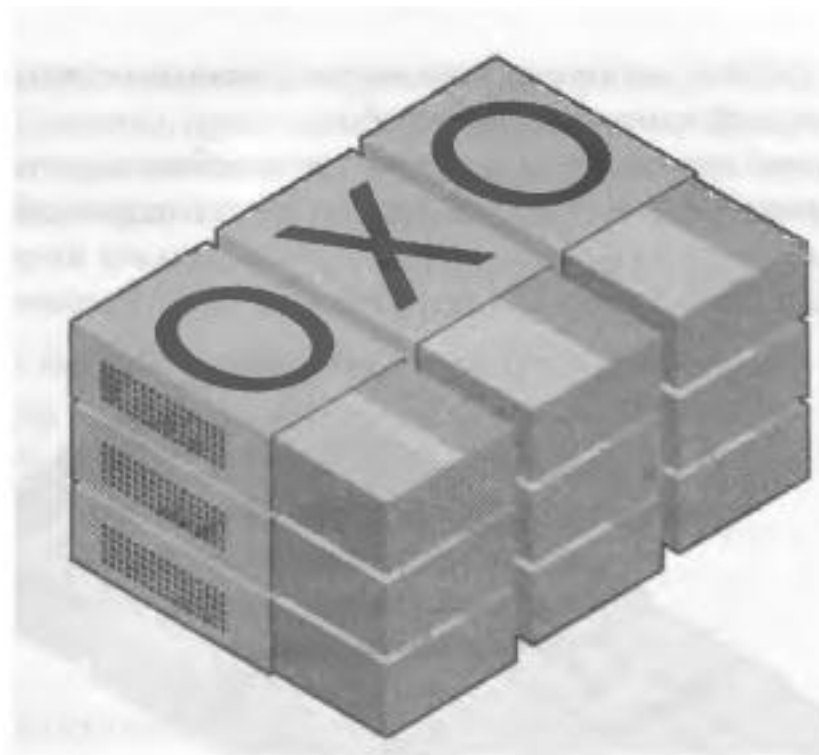
```
echo $team[3]; // Эта команда отображает имя Chris
```

Осмысление переменных

- Двумерные массивы

```
<?php
$oxo = array(array('x', '', 'o'),
              array('o', 'o', 'x'),
              array('x', 'o', '' ));
?>

echo $oxo[1][2];
```



Правила присваивания имен переменных

- Переменные начинаются буквы латинского алфавита или с символа «_»
- Имена переменных могут содержать символы «a-z, A-Z, 0-9, _»
- Имена переменных не должны содержать пробелов, если имя составляют два слова, то в качестве разделителя используется «_»
- Имена переменных чувствительны к регистру: \$x и \$X – разные переменные

Операторы. Определение

- **Оператор** – наименьшая автономная часть языка программирования.



Операторы

- Арифметические

```
<?php
```

```
$c = $a + $a;
```

```
$c = $a - $a;
```

```
$c = $a * $a;
```

```
$c = $a / $a;
```

```
$c = $a % $a;
```

```
++$a;
```

```
--$a;
```

```
?>
```

Операторы

- Присваивания

Оператор	Пример	Эквивалент
=	$\$j = 15$	$\$j = 15$
+=	$\$j += 5$	$\$j = \$j + 5$
-=	$\$j -= 3$	$\$j = \$j - 3$
*=	$\$j *= 8$	$\$j = \$j * 8$
/=	$\$j /= 16$	$\$j = \$j / 16$
.=	$\$j .= \k	$\$j = \$j . \$k$
%=	$\$j \% = 4$	$\$j = \$j \% 4$

Операторы

- Сравнения

Оператор	Описание	Пример
<code>==</code>	Равно	<code>\$j == 4</code>
<code>!=</code>	Не равно	<code>\$j != 21</code>
<code>></code>	Больше чем	<code>\$j > 3</code>
<code><</code>	Меньше чем	<code>\$j < 100</code>
<code>>=</code>	Больше чем или равно	<code>\$j >= 15</code>
<code><=</code>	Меньше чем или равно	<code>\$j <= 8</code>

Строки

- Объединение строк

```
echo "У вас " . $msgs . " сообщений.";
```

```
$bulletin .= $newsflash;
```

- Типы строк

```
$info = 'Предваряйте имена переменных символом $, как в данном примере: $variable';
```

```
echo "За всю историю было $count президентов США ";
```

- Экранирование

```
$text = 'My sister\'s car is a Ford';
```

```
$text = "Моя мама всегда говорит: \"Ешь зелень\".";
```

```
$heading = "Дата\tИмя\tПлатеж";
```



Операторы

- Многострочные команды

```
<?php
$author = "Альфред Ньютен";
echo "Это заголовок
Это первая строка.
Это вторая строка.
Автор $author.";
?>
```

```
<?php
$author = "Альфред Ньютен";
$out = <<<_END
Это заголовок
Это первая строка.
Это вторая строка.
- Автор $author.
_END;
?>
```

```
<?php
$author = "Альфред Ньютен";
$text = "Это заголовок
Это первая строка.
Это вторая строка.
Автор $author.";
?>
```

```
<?php
$author = "Альфред Ньютен";
echo <<<_END
Это заголовок
Это первая строка.
Это вторая строка.
- Автор $author.
_END;
?>
```

Типы переменных

- Преобразование числа в строку

```
<?php
$number = 12345 * 67890;
echo substr($number, 3, 1);
?>
```

- Преобразование строки в число

```
<?php
$pi = "3.1415927";
$radius = 5;
echo $pi * ($radius * $radius);
?>
```



Константы

- Пользовательские
константы

```
<?php

// Valid constant names
define("FOO",      "something");
define("FOO2",     "something else");
define("FOO_BAR",  "something more");

// Invalid constant names
define("2FOO",     "something");

?>
```

Константы

- Предопределенные константы
 - `__LINE__`
 - `__FILE__`
 - `__DIR__`
 - `__FUNCTION__`
 - `__CLASS__`
 - `__METHOD__`
 - `__NAMESPACE__`



Функции

- Простое объявление функции

```
<?php
function foo($arg_1, $arg_2, /* ..., */ $arg_n)
{
    echo "Example function.\n";
    return $retval;
}
?>
```

- Передача параметров

```
<?php
function takes_array($input)
{
    echo "$input[0] + $input[1] = ", $input[0]+$input[1];
}
?>
```

Область видимости переменной

- **Локальные переменные** – переменные, созданные внутри функции.

```
<?php
function longdate($timestamp)
{
    $temp = date("l F jS Y", $timestamp);
    return "Дата: $temp";
}
?>
```

```
<?php
$temp = "Дата: ";
echo $temp . longdate(time());
function longdate($timestamp)
{
    return date("l F jS Y", $timestamp);
}
?>
```

```
<?php
$temp = "Дата: ";
echo longdate(time());
function longdate($timestamp)
{
    return $temp . date("l F jS Y", $timestamp);
}
?>
```

```
<?php
$temp = "Дата: ";
echo longdate($temp, time());
function longdate($text, $timestamp)
{
    return $text . date("l F jS Y", $timestamp);
}
?>
```

Область видимости переменной

- Глобальные переменные – переменные, доступные из любого места программы.

```
global $is_logged_in;
```

- Статические переменные – переменные, значение которых сохраняется между вызовами функции.

```
<?php
function test()
{
    static $count = 0;
    echo $count;
    $count++;
}
?>
```

```
<?php
static $int = 0;           // Допустимо
static $int = 1+2;       // Недопустимо (вызовет ошибку синтаксического
                          // разбора (Parse error))
static $int = sqrt(144); // Недопустимо
?>
```


Область видимости переменной

- **Суперглобальные переменные** – ряд predefined переменных
 - \$GLOBALS
 - \$_SERVER
 - \$_POST
 - \$_GET
 - \$_FILES
 - \$_COOKIE
 - \$_SESSION
 - \$_REQUEST
 - \$_ENV



1 Введение в проектирование конструкторско-технологических информационных систем

1.3.4 Выражения и управляющие конструкции



Определения

- **Выражение** – сочетание значений, переменных, операторов и функций, в результате которого вычисляется новое значение.
- **Литерал** – простейшее выражение, вычисляющееся само в себя.




Выражения

- Базовая часть любого языка программирования

$y = 3(\text{abs}(2x) + 4)$  `$y = 3 * (abs(2*$x) + 4);`

```
<?php
echo "a: [" . (20 > 9) . "]<br />";
echo "b: [" . (5 == 6) . "]<br />";
echo "c: [" . (1 == 0) . "]<br />";
echo "d: [" . (1 == 1) . "]<br />";
?>
```



```
a: [1]
b: []
c: []
d: [1]
```

```
<?php // test2.php
echo "a: [" . TRUE . "]<br />";
echo "b: [" . FALSE . "]<br />";
?>
```



```
a: [1]
b: []
```

Литералы и переменные

- 5 типов литералов

```
<?php
$name = "Brian";
$age = 37;
echo "a: " . 73 . "<br />"; // Числовой литерал
echo "b: " . "Hello" . "<br />"; // Строковый литерал
echo "c: " . FALSE . "<br />"; // Литерал константы
echo "d: " . $name . "<br />"; // Литерал строковой переменной
echo "e: " . $age . "<br />"; // Литерал числовой переменной
?>
```

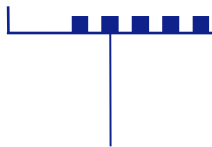
a: 73
b: Hello
c:
d: Brian
e: 37

- Инструкция = присваивание + управляющая конструкция + выражение

```
<?php
$days_to_new_year = 366 - $day_number; // Выражение
if ($days_to_new_year < 30)
{
    echo "Скоро Новый Год!"; // Инструкция
}
?>
```

Операторы

- Унарные операторы
 - Декремент
 - Инкремент
- Бинарные операторы
 - Арифметические
 - Логические
- Трехкомпонентный оператор
 - ? X : Y



Операторы

- Приоритет операторов

Оператор(ы)	Тип
()	Скобки
++ --	Инкремент-декремент
!	Логический
* / %	Арифметические
+ -	Арифметические и строковые
<< >>	Побитовые
< <= > >= <>	Сравнения
== != === !==	Сравнения
&	Поразрядный (и ссылочный)
^	Поразрядный
	Поразрядный
&&	Логический
	Логический
? :	Трехкомпонентный
= += -= *= /= .= %= &= != ^= <<= >>=	Присваивания
and	Логический
xor	Логический
or	Логический

Операторы

- Взаимосвязь операторов (вычисление справа-налево)

Оператор	Описание
new	Создание нового объекта
!	Логическое НЕ
~	Поразрядное НЕ
++ --	Инкремент и декремент
+ -	Унарный плюс и изменение знака числа
(int)	Преобразование в целое число
(double)	Преобразование в число с плавающей точкой
(string)	Преобразование в строковое значение
(array)	Преобразование в массив
(object)	Преобразование в объект
@	Подавление сообщения об ошибке
? :	Условный оператор
=	Присваивание

- Пример

```
<?php
$level = $score = $time = 0;
?>
```


Операторы отношений

- Равенство
 - Присваивание и проверка

```
<?php
$month = "Март";
if ($month == "Март") echo "Весна наступила";
?>
```

- Равенство и тождество

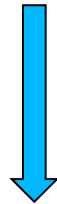
```
<?php
$a = "1000";
$b = "+1000";
if ($a == $b) echo "1";
if ($a === $b) echo "2";
?>
```



Операторы отношений

- Сравнение

```
<?php  
$a = 2; $b = 3;  
if ($a > $b) echo "$a больше $b<br />";  
if ($a < $b) echo "$a меньше $b<br />";  
if ($a >= $b) echo "$a больше или равно $b<br />";  
if ($a <= $b) echo "$a меньше или равно $b<br />";  
?>
```



2 меньше 3
2 меньше или равно 3



Операторы отношений

- Логические

Входные данные		Операторы и результаты		
a	b	AND	OR	XOR
TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	TRUE
FALSE	TRUE	FALSE	TRUE	TRUE
FALSE	FALSE	FALSE	FALSE	FALSE

- Пример

```
<?php
$a = 1; $b = 0;
echo ($a AND $b) . "<br />";
echo ($a or $b) . "<br />";
echo ($a XOR $b) . "<br />";
echo !$a . "<br />";
?>
```

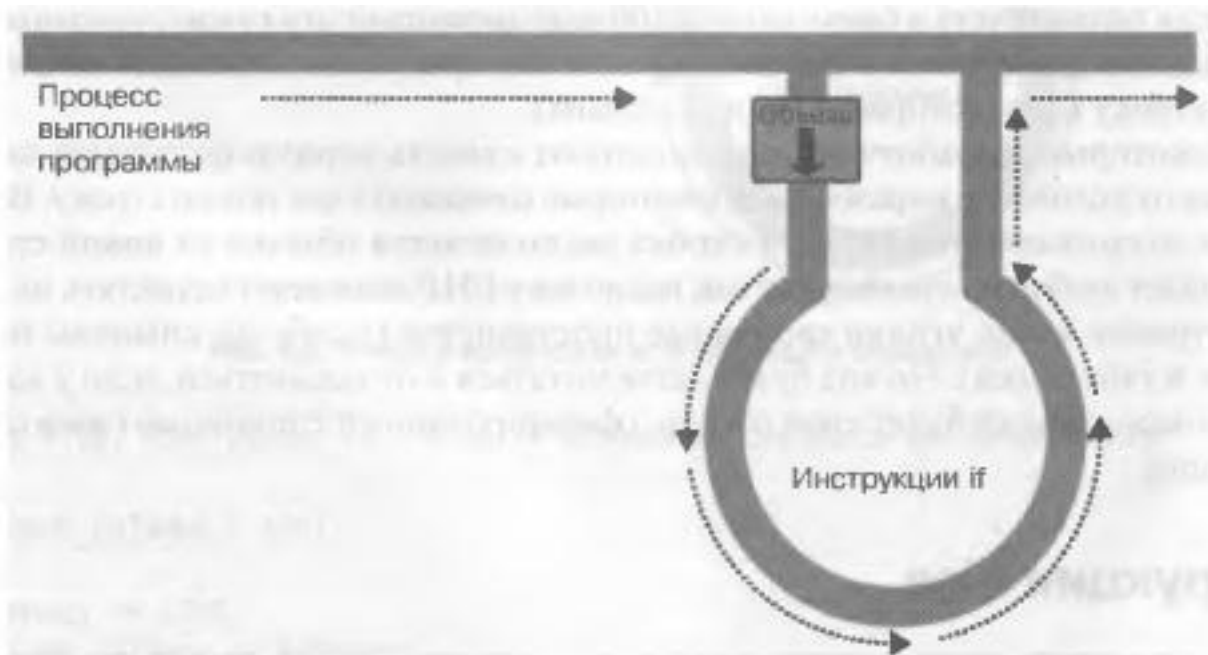
Условие. Определение

- **Условный переход** – команда на изменение порядка выполнения программы в соответствии с результатом проверки некоторого условия.

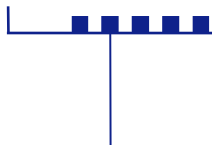


УСЛОВИЯ

- Инструкция **if**



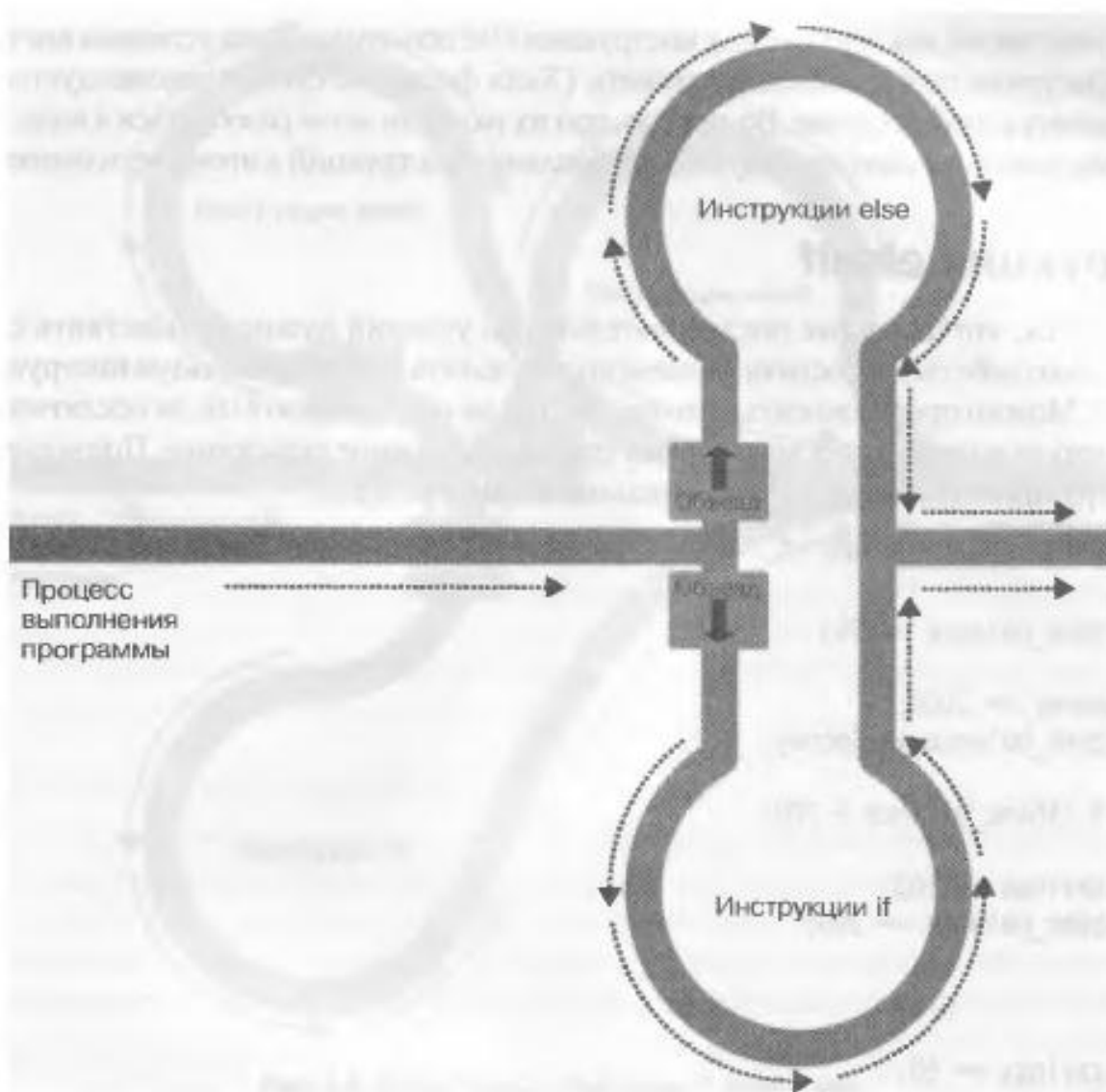
```
<?php
if ($bank_balance < 100)
{
    $money += 1000;
    $bank_balance += $money;
}
?>
```



УСЛОВИЯ

- Инструкция `else`

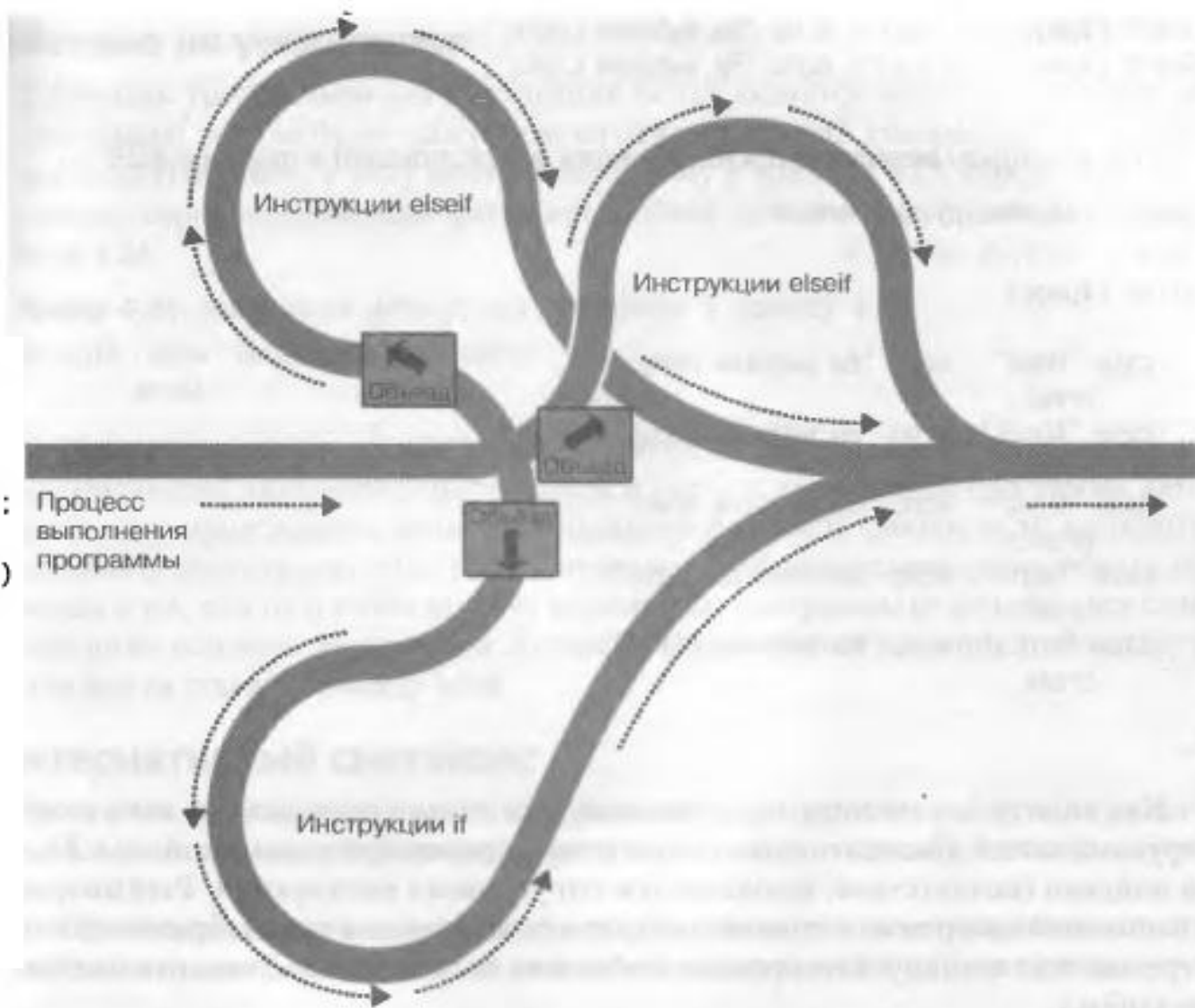
```
<?php
if ($bank_balance < 100)
{
    $money += 1000;
    $bank_balance += $money;
}
else
{
    $savings += 50;
    $bank_balance -= 50;
}
?>
```



УСЛОВИЯ

- **elseif**

```
<?php
if ($bank_balance < 100)
{
    $money += 1000;
    $bank_balance += $money;
}
elseif ($bank_balance > 200)
{
    $savings += 100;
    $bank_balance -= 100;
}
else
{
    $savings += 50;
    $bank_balance -= 50;
}
?>
```



УСЛОВИЯ

- Инструкция **switch**

```
<?php
switch ($page)
{
    case "Home": echo "Вы выбрали Home";
                break;
    case "About": echo "Вы выбрали About";
                break;
    case "News": echo "Вы выбрали News";
                break;
    case "Login": echo "Вы выбрали Login";
                break;
    case "Links": echo "Вы выбрали Links";
                break;
}
?>
```

```
<?php
if ($page == "Home") echo "Вы выбрали Home";
elseif ($page == "About") echo "Вы выбрали About";
elseif ($page == "News") echo "Вы выбрали News";
elseif ($page == "Login") echo "Вы выбрали Login";
elseif ($page == "Links") echo "Вы выбрали Links";
?>
```


Условные выражения

- Оператор «?»

```
<?php  
echo $fuel <= 1 ? "Требуется дозаправка" : "Топлива еще достаточно";  
?>
```

```
<?php  
$enough = $fuel <= 1 ? FALSE : TRUE;  
?>
```

```
$saved = $saved >= $new ? $saved : $new;
```

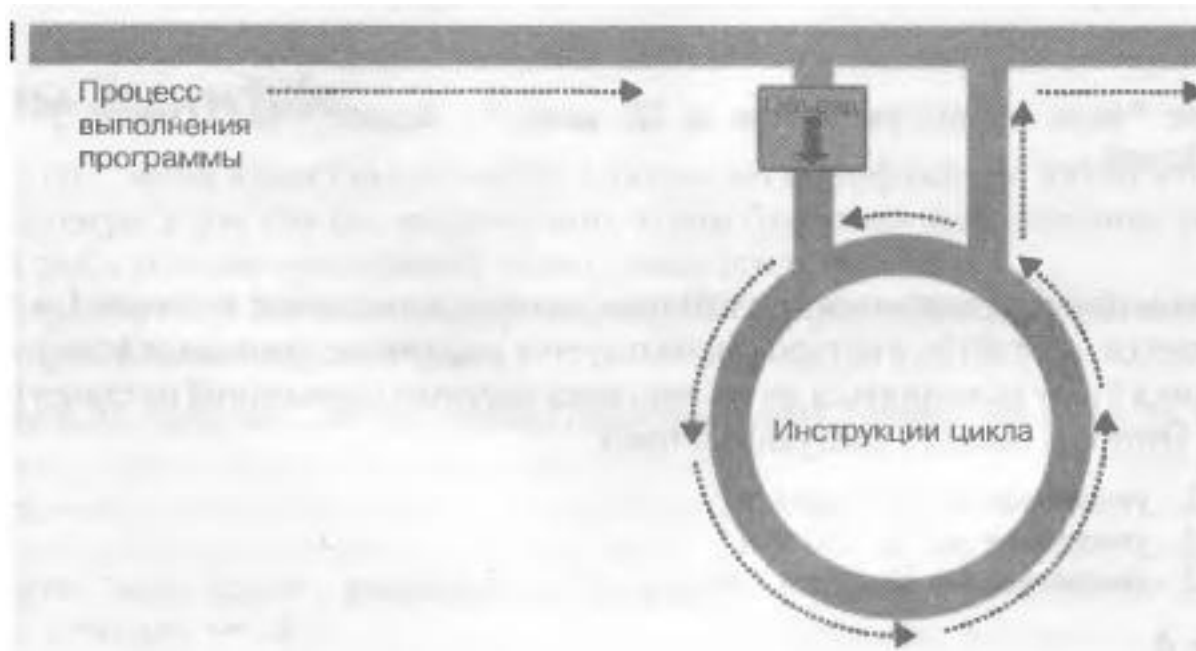
```
$saved =           // Присваивание значения переменной $saved  
    $saved >= $new // Сравнение $saved и $new  
    ?              // Если сравнение выдает истинный результат ...  
    $saved         // ... ей присваивается текущее значение $saved  
    :              // Если сравнение выдает ложный результат ...  
    $new;         // ... ей присваивается значение переменной $new
```

Циклы. Определение

- **Цикл** — разновидность управляющей конструкции, предназначенная для организации многократного исполнения набора инструкций.
- **Тело цикла** – последовательность инструкций, предназначенная для многократного исполнения
- **Итерация** – единичное выполнение тела цикла называется итерацией.
- **Условие выхода** – выражение, определяющее, будет в очередной раз выполняться итерация, или цикл завершится
- **Счетчик цикла** – переменная, хранящая номер текущей итерации цикла

Организация циклов

- Аналогия



Виды циклов

- **Безусловный цикл** – циклы, выход из которых не предусмотрен логикой программы.
- **Цикл с предусловием** – цикл, который выполняется пока истинно некоторое условие, указанное перед его началом.
- **Цикл с постусловием** – цикл, в котором условие проверяется после выполнения тела цикла.
- **Цикл со счетчиком** – цикл, в котором некоторая переменная изменяет своё значение от заданного начального значения до конечного значения с некоторым шагом, и для каждого значения этой переменной тело цикла выполняется один раз.


Организация циклов

- Циклы **while**

```
<?php
$fuel = 10;
while ($fuel > 1)
{
    // Продолжение поездки ...
    echo "Топлива еще достаточно";
}
?>
```

Число 1, умноженное на 12, равно 12
Число 2, умноженное на 12, равно 24
Число 3, умноженное на 12, равно 36

```
<?php
$count = 1;
while ($count <= 12)
{
    echo "Число $count, умноженное на 12, равно " . $count * 12 . "<br />";
    ++$count;
}
?>
```



Организация циклов

- Циклы **do...while**

```
<?php
$count = 1;
do
    echo "Число $count, умноженное на 12, равно " . $count * 12 . "<br />";
while (++$count <= 12);
?>
```

```
<?php
$count = 1;
do {
    echo "Число $count, умноженное на 12, равно " . $count * 12;
    echo "<br />";
} while (++$count <= 12);
?>
```

Организация циклов

- Циклы **for**
 - Выражение инициализации
 - Выражение условия
 - Выражение модификации

```
<?php
for ($count = 1 ; $count <= 12 ; ++$count)
    echo "Число $count, умноженное на 12, равно " . $count * 12 . "<br />";
?>
```

```
for ($i = 1, $j = 1 ; $i + $j < 10 ; $i++ , $j++)
{
    // ...
}
```

Организация циклов

- Операторы **break** и **continue**
 - Прекращение работы цикла

```
<?php
$fp = fopen("text.txt", 'wb');
for ($j = 0 ; $j < 100 ; ++$j)
{
    $written = fwrite($fp, "data");
    if ($written == FALSE) break;
}
fclose($fp);
?>
```

- Переход на следующую итерацию



Преобразование типов

- Неявное преобразование

```
<?php
$a = 56;
$b = 12;
$c = $a / $b;
echo $c;
?>
```

- Явное преобразование

```
$c = (int) ($a / $b);
```

Тип преобразования	Описание
(int) (integer)	Преобразование в целое число путем отбрасывания десятичной части
(bool) (boolean)	Преобразование в булево значение
(float) (double) (real)	Преобразование в число с плавающей точкой
(string)	Преобразование в строку
(array)	Преобразование в массив
(object)	Преобразование в объект

1 Введение в проектирование конструкторско-технологических информационных систем

1.3.5 Функции в РНР



Определение

- Функция – набор инструкций, который позволяет выполнять конкретную задачу и в дополнение к этому может вернуть какое-либо значение
- Преимущества
 - Экономия времени программирования
 - Сокращение количества ошибок
 - Сокращение времени выполнения
 - Возможность использовать функции в разных случаях



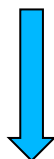
Функции

- Функция print

```
print("print является функцией");
```

- Пример вызова функций

```
<?php
echo strrev(" .dlrow olleH"); // Реверсирование строки
echo str_repeat("Hip ", 2);   // Повторение строки
echo strtoupper("hooray!");  // Преобразование символов строки в верхний
                             // регистр
?>
```



```
Hello world. Hip Hip HOORAY!
```

Функции

- Определение функции

```
function имя_функции([параметр [, ...]])
{
    // Инструкции
}
```

- Возвращаемое значение

```
<?php
echo fix_names("WILLIAM", "henry", "gatES");
function fix_names($n1, $n2, $n3)
{
    $n1 = ucfirst(strtolower($n1));
    $n2 = ucfirst(strtolower($n2));
    $n3 = ucfirst(strtolower($n3));
    return $n1 . " " . $n2 . " " . $n3;
}
?>
```



William Henry Gates

Функции

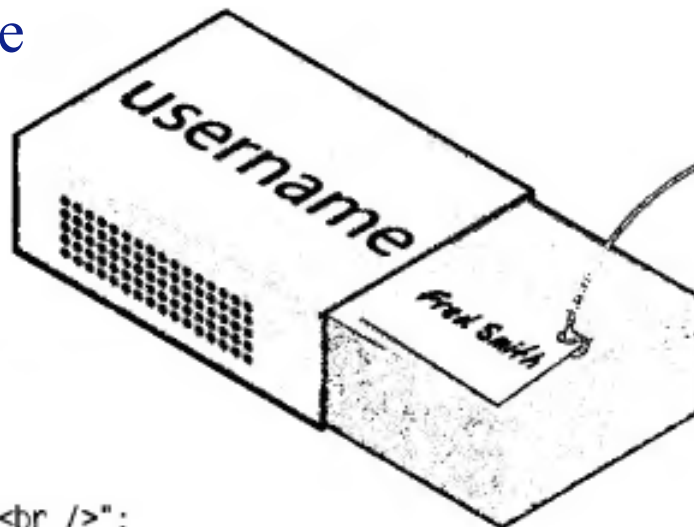
- Возвращение массива

```
<?php
$names = fix_names("WILLIAM", "henry", "gatES");
echo $names[0] . " " . $names[1] . " " . $names[2];
function fix_names($n1, $n2, $n3)
{
    $n1 = ucfirst(strtolower($n1));
    $n2 = ucfirst(strtolower($n2));
    $n3 = ucfirst(strtolower($n3));
    return array($n1, $n2, $n3);
}
?>
```

Функции

- Передача по ссылке

```
<?php
$a1 = "WILLIAM";
$a2 = "henry";
$a3 = "gatES";
echo $a1 . " " . $a2 . " " . $a3 . "<br />";
fix_names($a1, $a2, $a3);
echo $a1 . " " . $a2 . " " . $a3;
function fix_names(&$n1, &$n2, &$n3)
{
    $n1 = ucfirst(strtolower($n1));
    $n2 = ucfirst(strtolower($n2));
    $n3 = ucfirst(strtolower($n3));
}
?>
```



```
function( )
{
    //Code...
}
```



WILLIAM henry gatES
William Henry Gates

Функции

- Возвращение глобальных переменных

```
<?php
$a1 = "WILLIAM";
$a2 = "henry";
$a3 = "gatES";
echo $a1 . " " . $a2 . " " . $a3 . "<br />";
fix_names();
echo $a1 . " " . $a2 . " " . $a3;
function fix_names()
{
    global $a1; $a1 = ucfirst(strtolower($a1));
    global $a2; $a2 = ucfirst(strtolower($a2));
    global $a3; $a3 = ucfirst(strtolower($a3));
}
?>
```


Включение и запрос файлов

- Инструкция **include**

Пример 5.6. Включение файла PHP

```
<?php
include "library.php";
// Сюда помещается ваш код
?>
```

- Инструкция **include_once**

```
<?php
include_once "library.php";
// Сюда помещается ваш код
?>
```

Включение и запрос файлов

- Инструкция `require`
- Инструкция `require_once`

```
<?php  
require_once "library.php";  
// Сюда помещается ваш код  
?>
```

1 Введение в проектирование конструкторско-технологических информационных систем

1.3.6 Массивы в PHP



Основные подходы к массивам

- Массивы с численной индексацией

```
<?php
$paper[] = "Copier";
$paper[] = "Inkjet";
$paper[] = "Laser";
$paper[] = "Photo";
print_r($paper);
?>
```

```
<?php
$paper[0] = "Copier";
$paper[1] = "Inkjet";
$paper[2] = "Laser";
$paper[3] = "Photo";
print_r($paper);
?>
```



```
Array
(
    [0] => Copier
    [1] => Inkjet
    [2] => Laser
    [3] => Photo
)
```

```
<?php
$paper[] = "Copier";
$paper[] = "Inkjet";
$paper[] = "Laser";
$paper[] = "Photo";
for ($j = 0 ; $j < 4 ; ++$j)
    echo "$j: $paper[$j]<br>";
?>
```



```
0: Copier
1: Inkjet
2: Laser
3: Photo
```

Основные подходы к массивам


- Ассоциативные массивы

```
<?php
$paper['copier'] = "Copier & Multipurpose";
$paper['inkjet'] = "Inkjet Printer";
$paper['laser'] = "Laser Printer";
$paper['photo'] = "Photographic Paper";
echo $paper['laser'];
?>
```

- Присваивание

```
<?php
$p1 = array("Copier", "Inkjet", "Laser", "Photo");
echo "Элемент массива p1: " . $p1[2] . "<br>";
$p2 = array('copier' => "Copier & Multipurpose",
            'inkjet' => "Inkjet Printer",
            'laser' => "Laser Printer",
            'photo' => "Photographic Paper");
echo "Элемент массива p2: " . $p2['inkjet'] . "<br>";
?>
```

Элемент массива p1: Laser
Элемент массива p2: Inkjet Printer



Основные подходы к массивам

- **Совместный цикл** – цикл, задающий выполнение некоторой операции для объектов из заданного множества, без явного указания порядка перечисления этих объектов.



Основные подходы к массивам

- Цикл `foreach...as`
 - Для массива с численной индексацией

```
<?php
$paper = array("Copier", "Inkjet", "Laser", "Photo");
$j = 0;
foreach ($paper as $item)
{
    echo "$j: $item<br>";
    ++$j;
}
?>
```

Основные подходы к массивам

- Цикл `foreach...as`
 - Для ассоциативного массива

```
<?php
$paper = array('copier' => "Copier & Multipurpose",
               'inkjet' => "Inkjet Printer",
               'laser' => "Laser Printer",
               'photo' => "Photographic Paper");
foreach ($paper as $item => $description)
    echo "$item: $description<br>";
?>
```



```
copier: Copier & Multipurpose
inkjet: Inkjet Printer
laser: Laser Printer
photo: Photographic Paper
```


Основные подходы к массивам

- Функции **list()** и **each()**

```
<?php
$paper = array('copier' => "Copier & Multipurpose",
              'inkjet' => "Inkjet Printer",
              'laser' => "Laser Printer",
              'photo' => "Photographic Paper");
while (list($item, $description) = each($paper))
    echo "$item: $description<br>";
?>
```

```
<?php
list($a, $b) = array('Alice', 'Bob');
echo "a=$a b=$b";
?>
```



a=Alice b=Bob

Многомерные массивы

- Массивы с числовой индексацией

```
<?php
$chessboard = array(
    array('r', 'n', 'b', 'k', 'q', 'b', 'n', 'r'),
    array('p', 'p', 'p', 'p', 'p', 'p', 'p', 'p'),
    array(' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '),
    array(' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '),
    array(' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '),
    array(' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '),
    array(' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '),
    array(' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '),
    array('P', 'P', 'P', 'P', 'P', 'P', 'P', 'P'),
    array('R', 'N', 'B', 'K', 'Q', 'B', 'N', 'R'));
echo "<pre>";
foreach ($chessboard as $row)
{
    foreach ($row as $piece)
        echo "$piece ";
    echo "<br />";
}
echo "</pre>";
?>
```

Многомерные массивы

- Массивы с числовой индексацией

```
<?php
$products = array(
    'paper' => array(
        'copier' => "Copier & Multipurpose",
        'inkjet' => "Inkjet Printer",
        'laser' => "Laser Printer",
        'photo' => "Photographic Paper"),
    'pens' => array(
        'ball' => "Ball Point",
        'hilite' => "Highlighters",
        'marker' => "Markers"),
    'misc' => array(
        'tape' => "Sticky Tape",
        'glue' => "Adhesives",
        'clips' => "Paperclips") );
echo "<pre>";
foreach ($products as $section => $items)
    foreach ($items as $key => $value)
        echo "$section:\t$key\t($value)<br>";
echo "</pre>";
?>
```



paper:	copier	(Copier & Multipurpose)
paper:	inkjet	(Inkjet Printer)
paper:	laser	(Laser Printer)
paper:	photo	(Photographic Paper)
pens:	ball	(Ball Point)
pens:	hilite	(Highlighters)
pens:	marker	(Markers)
misc:	tape	(Sticky Tape)
misc:	glue	(Adhesives)
misc:	clips	(Paperclips)

Многомерные массивы

- Функции для работы с массивами
 - `is_array()`
 - `count()`
 - `sort()`
 - `shuffle()`
 - `explode()`
 - `extract()`
 - `compact()`
 - `reset()`
 - `end()`

