

# HTML

[Слайд 1. Дескриптор <HEAD>](#)

[Слайд 2. Слайд 2. Слайд 2. Дескриптор <BODY>](#)

[Слайд 3. Атрибуты. Теги выравнивания](#)

[Слайд 4. Дескриптор <BODY>. Списки. Ссылки](#)

[Слайд 5. Дескриптор <BODY>. Дескриптор <TABLE>](#)

[Слайд 6. <FRAMESET> - фреймы \(кадры\)](#)

[Слайд 7. Пример](#)

[Слайд 8. Фреймы \(продолжение\)](#)

[Слайд 9. Формы](#)

[Слайд 10. Пример](#)

[Слайд 11. Наименование Формы](#)

[Слайд 12. <INPUT>](#)

[Слайд 13. Слайд 13. TEXT, PASSWORD, CHECKBOX](#)

[Слайд 14. Слайд 14. RADIO – сложный флажок](#)

[Слайд 15. RESET, SUBMIT, FILE](#)

[Слайд 16. Создание навигационных карт ссылок](#)

[Слайд 17. Слайд 17. Пример](#)

# XML

[Слайд 1. Реализация](#) Слайд 1. Реализация [XML](#)

[Слайд 2. Слайд 2. Слайд 2. Пролог](#) Слайд 2. Пролог [XML](#)

[Слайд 3. Комментарий](#)

[Слайд 4. Таблица](#)

[Слайд 5. Язык](#) Слайд 5. Язык [DTD \(Document type](#)

[Definition\)](#) Слайд 5. Язык DTD (Document type Definition)

[Слайд 6. Объявление атрибутов](#)

[Слайд 7. Примеры](#)

[Слайд 8. Объявление сущности](#)

[Слайд 9. Не разбираемая сущность](#)

[Слайд 10. Пример описания](#) Слайд 10. Пример описания

[DTD](#) Слайд 10. Пример описания DTD. [Записная книжка](#)

[Слайд 11. Язык](#) Слайд 11. Язык [XSD](#) Слайд 11. Язык XSD

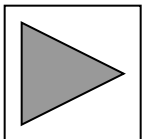
[Слайд 12. Сужение](#)

[Слайд 13. Слайд 13. Facets](#)

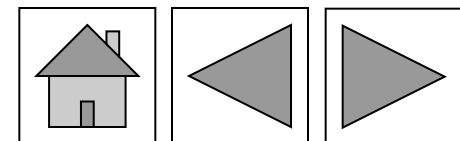
[Слайд 14. Слайд 14. Fundamental facets](#)

[Слайд 15. Список](#)

[Слайд 16. Объединение](#)



[Слайд 17. Объединение элементов](#)  
[Слайд 18. Схема](#) Слайд 18. Схема [XSD](#) Слайд 18.  
Схема XSD [книжки](#)  
[Слайд 19. Язык](#) Слайд 19. Язык [XLINK](#)  
[Слайд 20. Атрибут простая ссылка](#)  
[Слайд 21. Атрибут](#) Слайд 21. Атрибут [Resource](#)  
[Слайд 22.](#) Слайд 22. [Слайд 22. Атрибут](#) Слайд 22.  
Атрибут [ARC](#) Слайд 22. Атрибут ARC  
[Слайд 23.](#) Слайд 23. [Слайд 23. Атрибут](#) Слайд 23.  
Атрибут [Show](#)  
[Слайд 24.](#) Слайд 24. [Слайд 24. Атрибут](#) Слайд 24.  
Атрибут [actuate](#)  
[Слайд 25. Атрибут](#) Слайд 25. Атрибут [role](#)  
[Слайд 26. Уточненные ссылки](#) Слайд 26.  
Уточненные ссылки [XPointer](#)  
[Слайд 27. Использование указателей в ссылках](#)  
[Слайд 28. Схема](#) Слайд 28. Схема [xpointer](#)  
[Слайд 29.](#) Слайд 29. [Слайд 29. Схема](#) Слайд 29.  
Схема [xmlns](#)  
[Слайд 30. Язык](#) Слайд 30. Язык [XPath](#)  
[Слайд 31. Оси поиска](#)  
[Слайд 32. Области, определяемые осями](#)  
[Слайд 33. Тест по имени узла](#)  
[Слайд 34. Тест по виду узла](#)  
[Слайд 35. Предикаты](#)  
[Слайд 3](#) Слайд 36 Слайд 36. [Слайд 36. Циклы](#)  
[Слайд 3](#) Слайд 37 Слайд 37. [Слайд 37. Операции с](#)  
[множествами](#)  
[Слайд 3](#) Слайд 38 Слайд 38. [Слайд 38. Выражение в](#)  
[атрибутах конструктора](#)  
[Слайд 3](#) Слайд 39 Слайд 39. [Слайд 39. Выражение запроса](#)





# Лекция 2-3



<HTML>  
<HEAD>  
...  
</HEAD>  
<BODY>  
...  
</BODY>  
</HTML>

<HEAD>

<TITLE> → <TITLE> текст <TITLE>

<BASE> → <BASE HREF = "протокол://имя сервера/путь">

<META HTTP\_equiv (content, url) – поручает действие серверу

<META> →

пример

<META http\_equiv = "Refresh" content = '5  
url = http://wasm.ru/index.html

Принудительное обновление страницы каждые 5 сек.

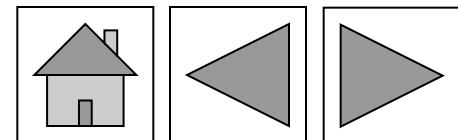
<META http\_equiv = "Expires" content = "Data"

Срок годности документа

<META name = "Keywords" lang = "ru" content = "слово1,слово2,слово3..."

<META name = "Description" content "Содержание"

</HEAD>



<HTML>  
<HEAD>  
...  
</HEAD>  
<BODY>  
...  
</BODY>  
</HTML>

## <BODY>

### атрибуты:

background – фон документа

bgcolor – цвет документа

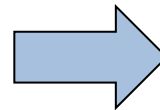
link – цвет гиперссылки

alink – цвет активной гиперссылки

vlink – цвет посещенной гиперссылки

topmargin – отступ сверху, right margin – отступ справа

leftmargin – отступ слева, bottommargin – отступ снизу



<BODY background="(URL)(путь)имя  
файла">

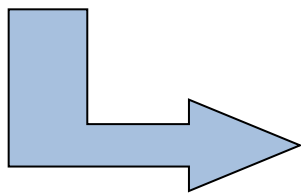
<BODY bgcolor="цвет">

<BODY link="цвет">

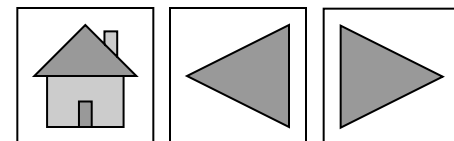
<BODY alink="цвет">

<BODY vlink="цвет">

[ <address> - идентификация автора



[ <address> описание <br> </address>



# Атрибуты тэги выравнивания:

H<номер> - тип заголовка

<BR> - переход на новую строку

<DIV> - выравнивание абзаца

<FONT> - управление внешним видом (FACE (гарнитура),

SIZE(размер), COLOR(цвет))

<IMG SRC= . . . > - графическое изображение

<P> - абзац

жирный - 

- <strong>** - индексируется поиск. системой
- <b>**
- <em>**

курсив - **<i>**

<u> - подчеркнутый , &nbsp; - много пробелов

h1 24p

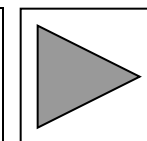
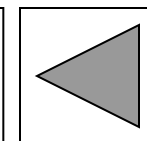
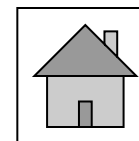
h2 18p

h3 14p

h4 12p

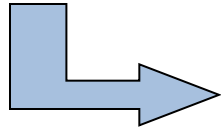
h5 10p

h6 8p



<HTML>  
<HEAD>  
...  
</HEAD>  
<BODY>  
...  
</BODY>  
</HTML>

## Списки:

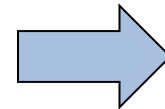


- <OL> - нумерованный список
- <UL> - маркерванный список
- <MENU> - меню
- <DL> - список определений

```
<OL type=1 start=1>  
<LI> эл.списка</LI>эл.списка </OL>  
<UL type=circle>  
<LI> эл.списка</LI>эл.списка </UL>  
<MENU>  
<LI> эл.списка</LI>эл.списка </MENU>  
<DL><DT>термин1<DD>опр.1  
<DT>термин2<DD>опр.2</DL>
```

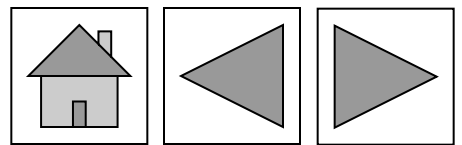
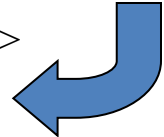
## <A> - Ссылки:

- name – метка для перехода внутри текста
- title – визуализация подсказки
- accesskey – указание горячей клавиши
- href – адрес ссылки



```
<A name=имя>текст</A>  
<A title="имя ссылки">ссылка</A>  
<A accesskey="имя ссылки">ссылки</A>  
<A href="URL">текст</A>  
<A href="#имя">текст</A>
```

```
<A href = "http://www.poshuk.com">Поисковый сервер</A>  
<A href = "ftp://ftp.poshuk.com/install.exe">Пример</A>  
<A href = mailto:name@domen.ru>Пример</A>
```



<TABLE> - Таблица:

<HTML>  
<HEAD>  
...  
</HEAD>  
<BODY>  
...  
</BODY>  
</HTML>  
  
**</BODY>**

атрибуты <TABLE>



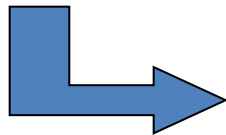
bgcolor - фон ячеек  
background - фоновый рисунок ячеек  
align - горизонтальное выравнивание  
width - ширина таблицы  
структура документа → { top - вверх  
middle - середина  
bottom - вниз  
valign - верт. выр. → {  
height - высота таблицы (в пикселях)  
cellpadding - отступ внутри ячейки  
cellspacing - расстояние между ячейками  
<sub> - нижний регистр, <sup> - верхний  
<frame> - отобразить/скрыть линии таблицы  
структура документа → { <Tthead>  
<Ttbody>  
<Ttfoot>

способ = { left  
center  
right

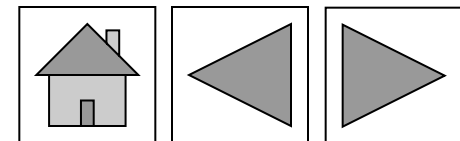
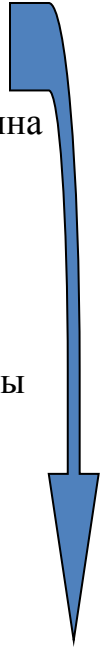


<TABLE align = способ>

<TR> - строка  
<CAPTION> - заголовок таблицы  
<TH> - заголовок в ячейке  
<TD> - ячейка



rowspan - объединение ячеек столбца -- <TD rowspan = 5>  
colspan - объединение ячеек строки -- <TD colspan = 5>





# <FRAMESET> - фреймы (кадры)

Заменяет тэг <BODY>

Атрибуты frameset:

cols – количество и размер колонок

rows – количество и размер строк

border – толщина обрамления

frameborder – наличие или отсутствие обрамления



<FRAMESET cols = “число,\*,%> cols=50%,50%

<FRAMESET rows = “число,\*,%> rows=100,20%,\*

<FRAME>

src – документ фрейма <FRAME src=“URL”>

frameborder – обрамление <FRAME frameborder = “1 или 0”>

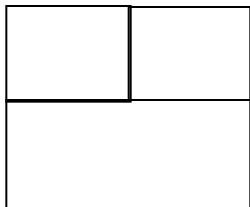
marginheight – толщина верхнего и нижнего обрамления

marginwidth – толщина левого и правого обрамления

name – задает имя фрейма для обращения к нему атрибутом target <A href>

noresize – лишает возможности изменения размеров фрейма

scrolling – задает наличие полосы прокрутки (yes, no, auto)



<frameset>

<frameset>

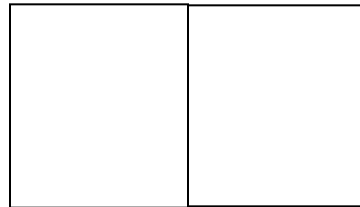
<frame>

<frame>

</frameset>

<frame>

</frameset>



<frameset>

<frame>

<frame>

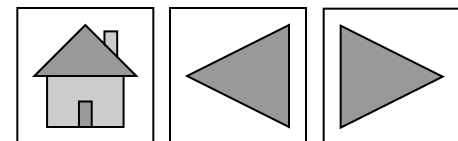
</frameset>

Контейнер

Index.htm

p1.htm – фрейм 1

p2.htm – фрейм 2



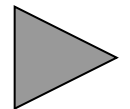
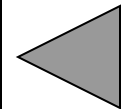
# Пример

```
<HTML>
<HEAD>
<TITLE>FRAMES</TITLE>
</HEAD>
<FRAMESET ROWS=150,*>
<FRAME SRC="page1.html" NAME="1">
<FRAMESET COLS=150,*>
  <FRAME SRC="page2.html" NAME="2">
  <FRAME SRC="page3.html" NAME="3">
</FRAMESET>
</FRAMESET>
</HTML>
```

## Файл page2.html

```
<HTML>
<HEAD>
<TITLE>page2</TITLE>
</HEAD>
<BODY>
<A HREF="page1.html" target="3"> Link</A>
</BODY>
</HTML>
```

Назад



# Фреймы (продолжение)

## Разрешенные имена

Зарезервированные неявные имена фреймов

- blank – загружает документ в новое окно, не имеющее имени

```
<A href="stuff.htm" target="_blank">
```

- self – загружает документ в текущее окно

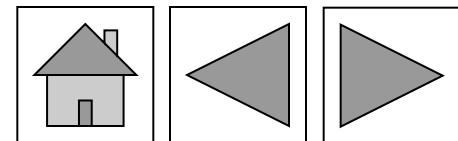
```
<A href="stuff.htm" target="_self">
```

- parent – загружает документ в окно родительской фреймовой структуры

```
<A href="stuff.htm" target="_parent">
```

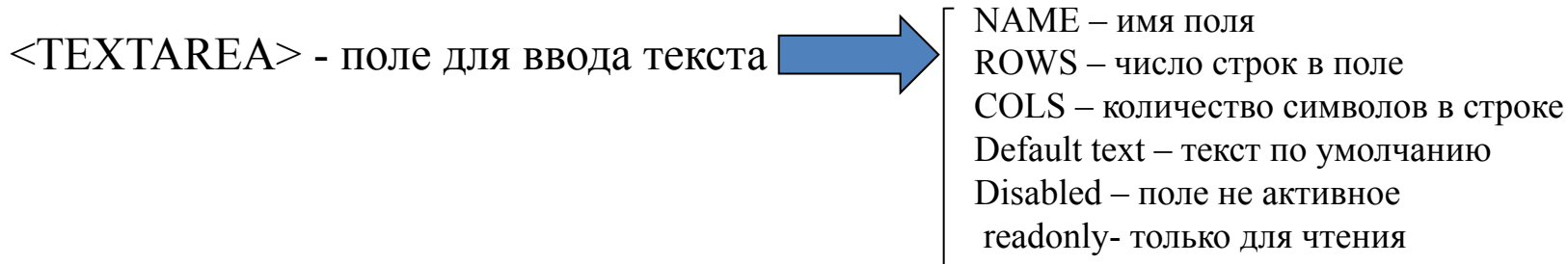
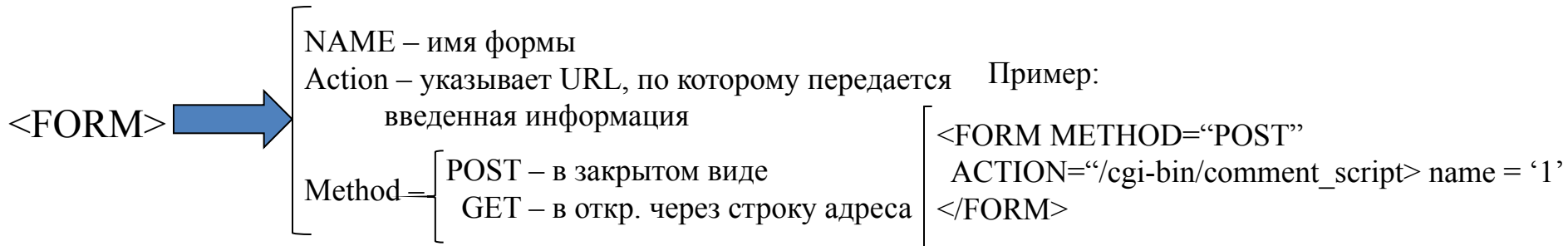
- top – загружает документ в окно фреймовой структуры верхнего уровня по отношению к данному фрейму

```
<A href="stuff.htm" target="_top">
```



# Формы

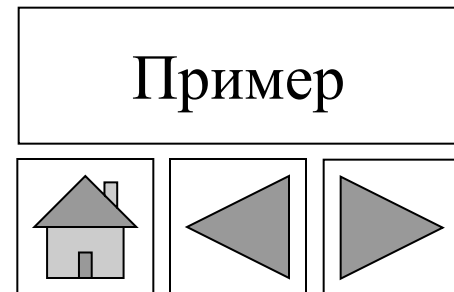
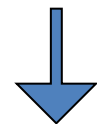
Передача в PHP `$_post` [`'<name'`]  
`$_get` [`'<name'`]



Пример:

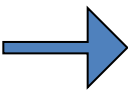
```
<FORM>
<TEXTAREA NAME="coments" ROWS=4 COLS=40>
Default text 1,2,3... - текст по умолчанию
</TEXTAREA>
```

Результат name=value

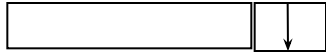


# Пример

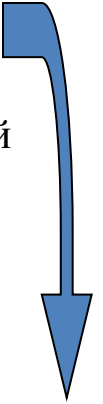
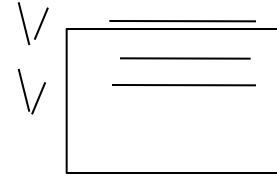
<SELECT> - выпадающее меню



Без MULTIPLE - однострочное меню



NAME – имя поля (обязательный)  
SIZE – число строк в многострочном меню  
MULTIPLE – режим выбора нескольких значений  
(многостр. меню)



```
<FORM>  
<SELECT NAME = "Network">  
<OPTION SELECTED VALUE="Ethernet"> ETHERNET  
<OPTION VALUE = "Token 16"> Token Ring – 16 MB  
<OPTION VALUE = "Token 4"> Token Ring – 4 MB  
<OPTION VALUE = "Localtalk"> Local Talk  
</SELECT>  
</FORM>  
</BODY>  
</HTML>
```

Результат:  
name = value  
(select) (option)

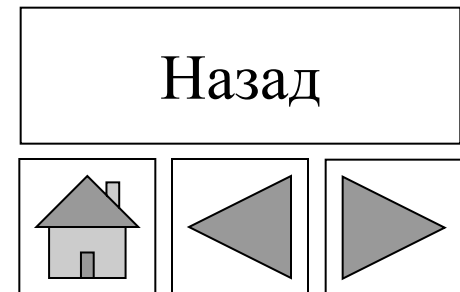
<OPTION>



VALUE – присваиваемое значение  
SELECTED – (по умолчанию)  
DISABLED - неактивный

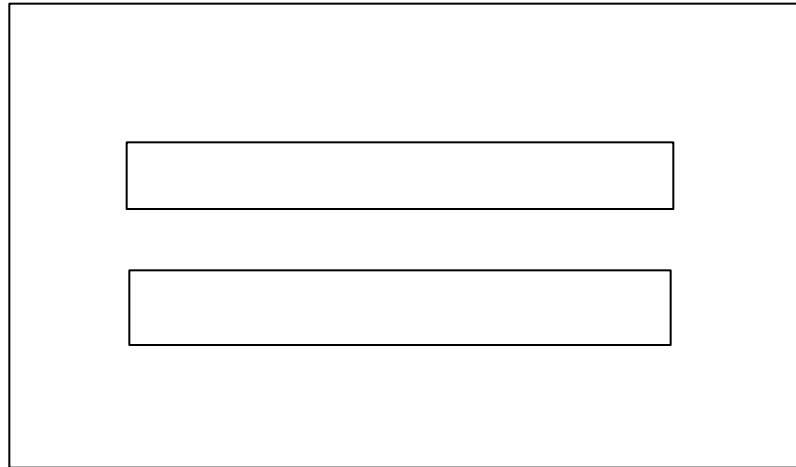
</SELECT>

</FORM>

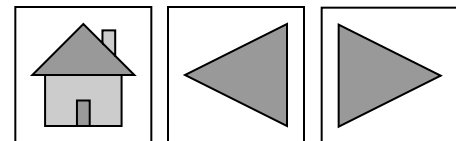


<fieldset> <legend>наименование формы </legend> </fieldset>

## Наименование Формы



The diagram illustrates a form structure. It consists of a large outer rectangle representing a fieldset. Inside this fieldset, there are two smaller horizontal rectangles stacked vertically, representing input fields. The text 'наименование формы' (form name) is positioned between the opening <legend> and closing </legend> tags in the code above, indicating that the legend is used to label the form.



## <INPUT>

Атрибуты

NAME - имя поля

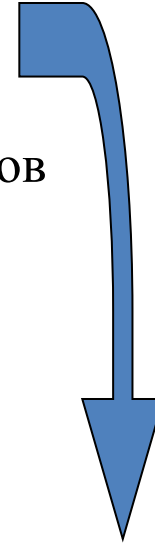
SIZE - размер поля

MAXLENGTH - максимальное количество слов

VALUE - значение по умолчанию

CHECKED - флажок/переключатель

TYPE - тип поля



TEXT – ввод строки

PASSWORD – ввод пароля

CHECKBOX – простой флажок

RADIO – сложный флажок

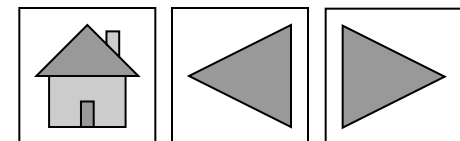
RESET – кнопка «отменить»

SUBMIT – кнопка «переслать»

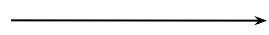
BUTTON - любое действие

IMAGE - аналог SUBMIT (с индивидуальным рисунком)

FILE - прикрепить файл

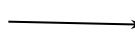


ТЕХТ – ввод строки



```
<FORM>  
<INPUT TYPE = "text" name="phone" value="текст">  
SIZE="15" MAXLENGTH="12" (по умолчанию)  
</FORM>
```

PASSWORD – ввод пароля



```
<FORM>  
Введите пароль <INPUT TYPE = "password">  
name="secret_word" SIZE="30"  
MAXLENGTH="30"  
</FORM>
```

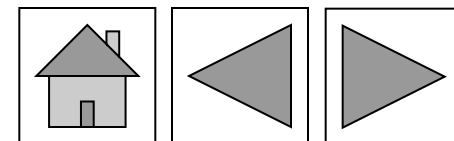
Результат    name=on      
                  name=off   

CHECKBOX – простой флажок



```
<FORM>  
<INPUT TYPE = "checkbox" name="checkbox1">  
флажок  
<INPUT TYPE = "checkbox" name="checkbox2"  
CHECKED>установленный флажок  
(по умолчанию флажок установленный)  
</FORM>
```

- флажок     - установленный флажок





## RADIO – сложный флажок

```
<FORM>
```

```
<INPUT TYPE = “radio” name=“choice”VALUE=“ch1”>Да
```

```
<INPUT TYPE = “radio” name=“choice”VALUE=“ch2”>Нет
```

```
</FORM>
```

```
<FORM>
```

```
<INPUT TYPE = “radio” name=“choice”VALUE=“ch1”>
```

```
CHACKED>Да
```

```
<INPUT TYPE = “radio” name=“choice”VALUE=“ch2”>Нет
```

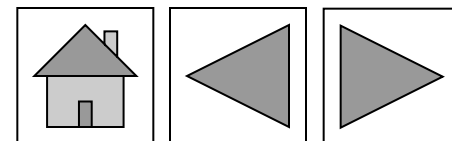
```
</FORM>
```

Результат name=value

Да     Нет

Да     Нет

Disabled – флажок не активный



RESET – кнопка «отменить» →

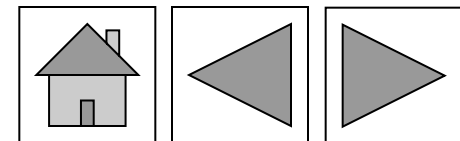
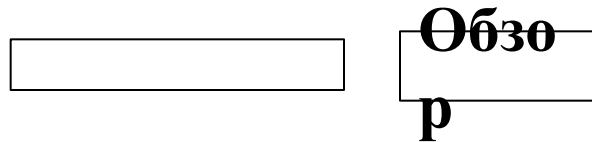
```
<FORM>  
<INPUT TYPE = “reset”>  
<BR>  
<INPUT TYPE=“reset” value=“Отменить!”>  
</FORM>
```

SUBMIT – кнопка «переслать» →

```
<FORM>  
<INPUT TYPE = “submit”>  
<BR>  
<INPUT TYPE=“submit”  
value=“Отослать данные”>  
</FORM>
```

FILE - прикрепить файл →

```
<FORM>  
<INPUT TYPE = “File” name=“file”>  
</FORM>
```



# Создание навигационных карт ссылок

1. Графическое изображение  
карты ссылок

2. Файл определения  
карты ссылок  
(форматы CERN  
NSCA)

Область по умолчанию: default <http://www.myserver.com/mypage/index.htm>  
Прямоугольная область: rect <http://www.myserver.com/mypage/rectangle.htm>  
50, 40, 100, 120  
Круглая область: circle <http://www.myserver.com/mypage/rectangle.htm>  
50, 40, 100, 60  
Многоугольная область: poly <http://www.myserver.com/mypage/rectangle.htm>  
10, 20 24, 70 84, 45 07, 11 10, 20

3. Программа или  
сценарий обработки  
карты ссылок

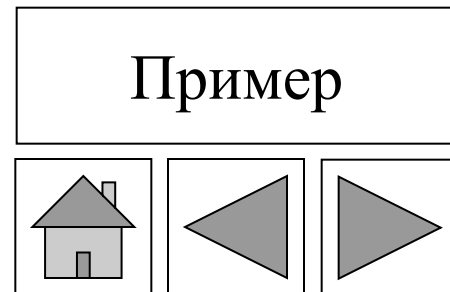
1. Вызов CGI-программы: `<A href = “http://www.myserver.com/cgi-bin/mapfile.map”>  
<IMG SRC = imagemap.gif ISMAP> </A>`  
2. Работа с провайдером: `<A href = “/cgi-bin/imagemap/~username/image.map”>  
<IMG SRC = “myimage.gif”> </A>`

4. Карты ссылок,  
Обработываемые  
клиентом

`<MAP NAME = “mapname”>  
<AREA [SHAPE = “shape”] COORDS = “x, y, ...” [HREF = “URL”/NOHREF]>  
</MAP>`

SHAPE – форма области (rect, poly, circle, default)  
COORDS – список координат  
HREF – URL, на который ссылается область  
NOHREF – область – мертвая зона

Пример



# Пример

<MAP NAME = mymap>

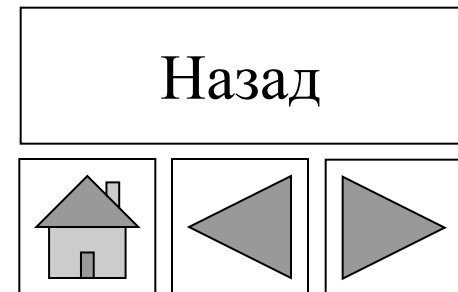
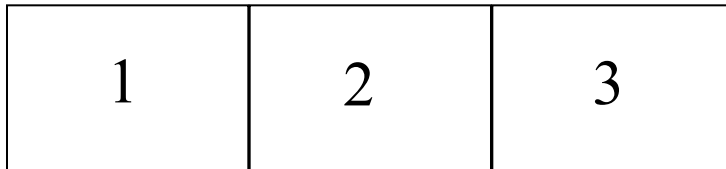
<AREA SHAPE = RECT COORDS = "0, 0, 100, 100" HREF = item1.html>

<AREA SHAPE = RECT COORDS = "101, 0, 200, 100" HREF = item2.html>

<AREA SHAPE = RECT COORDS = "201, 0, 300, 100" HREF = item3.html>

</MAP>

<IMG SRC = mymap.gif USEMAP = #mymap> - ссылка на карту ссылок

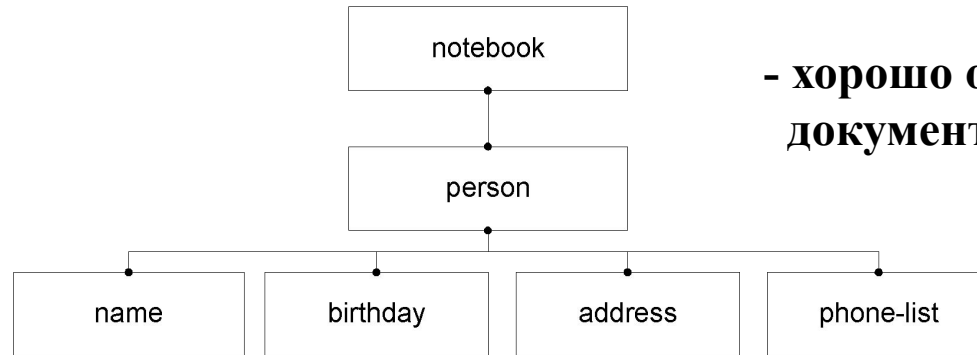


# XML

Реализации XML →

- XML - язык HTML, приведенный в соответствие с XML
- XML - язык записи математических формул
- XML - язык записи химических формул
- XML - язык записи звуков
- XML - язык, применяемый в беспроводной технологии

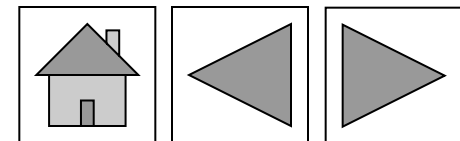
**Верный документ - документ, который соответствует описанию структуры документа или схеме документа (правила написания тегов ).**



**- хорошо оформленный документ**

**Для описания схемы документа используются язык DTD или XSD, где DTD - язык описания структуры документа; XSD - язык описания схемы документа.**

**(Extensible Markup Languages – расширенный язык разметок)**



# XML (*Extensible Markup Languages* – расширенный язык разметок)

Пролог XML →

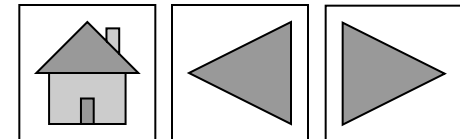
```
<?xml version="1.0"?>  
<?xml version="1.0" encoding="UTF-8"?>  
<?xml version="1.0" encoding="windows-1251"  
  standalone="yes"?>  
  yes – по умолчанию;  
  no – ссылка на внешний документ
```

Корневой элемент →  
(root)

```
<!DOCTYPE notebook [описание DTD]> - описание внутри  
  документа  
  
<!DOCTYPE notebook (имя root) [SYSTEM (если определение документа  
  записывается в отдельном файле) "ntb.dtd"]>
```

Элементы XML →

```
<surname> Сидорова </surname>  
<br /> для пустых элементов
```





# Таблицы

Й  
К  
К  
К  
К  
К  
К  
К  
К  
К  
К  
К  
К  
К  
Л

## SUPPLIER

| <i>SNUM</i> | <i>SNAME</i> | <i>LOC</i>      |
|-------------|--------------|-----------------|
| 10123       | Иванов       | Санкт-Петербург |
| 10212       | Петров       | Львов           |

## PRODUCT

| <i>PNUM</i> | <i>PNAME</i> | <i>Color</i> | <i>WEIGHT</i> |
|-------------|--------------|--------------|---------------|
| 012         | Болт         | Синий        | 25            |
| 013         | Гайка        | Черный       | 20            |
| ...         | ...          | ...          | ...           |

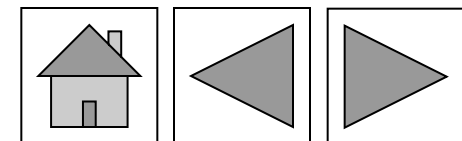
## SP

| <i>SNAME</i> | <i>PNAME</i> | <i>QTY</i> |
|--------------|--------------|------------|
| 10123        | гайка        | 500        |
| 10123        | болт         | 800        |
| 10212        | болт         | 300        |

## SQL Поставка

```
<? xml version = "1.0"
    encoding = "Windows - 1251"? >
<? DOCTYPE supplprod
    SYSTEM "suppl-prod.dtd" >
< sp : supplprod xmlns : sp = "http://..." >
< sp: suppl_prod
>
< sp: snum 10123
>
< /sp :
< sp : sname>Иванов </sp :
>
< sp : loc Санкт-Петербург /sp:loc
< sp: pname>гайка </sp :
>
< sp : qty > 500 </sp : qty
>
< /sp : suppl_prod
>
< sp : suppl_prod
>
M
< /sp : suppl_prod >
```

- корневой элемент – имя БД
- имя элемента – имя таблицы
- имя столбца – значение строки





## Язык DTD (*Document type Definition*)

Объявление  
типа элементов →  
<!ELEMENT

<!ELEMENT *br* EMPTY > - <*br* >

<!ELEMENT *something* ANY > -

<!ELEMENT *element* (#PCDATA) >

<!ELEMENT *supplprod* (sp : *snum*, sp : *sname*, sp : *loc*...) > -



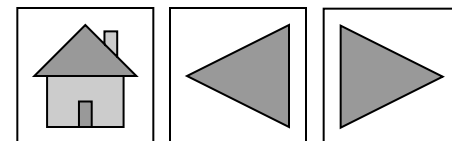
<!ELEMENT *supplprod* (sp : *supplprod*)\* >

<!ELEMENT *chapter* (bk : *title*, (bk : *section*, bk : *listing*\*, bk : *picture*\*)\*, bk : *conclusion*?) >

Элемент или список может встретиться один или ноль раз

Элемент или список может встретиться нуль или несколько раз

+ - элемент или список может встретиться один или несколько раз



Объявление  
атрибутов →  
<!ATTLIST

Типы атрибутов

*CDATA* – строка символов

*ID* – уникальный идентификатор

*IDREF* – идентификатор-ссылка на другие элементы

*ENTITY* – имя неповторяемой анализатором сущности



<!ELEMENT *name* (#PCDATA) >

<!ATTLIST *name* *reg\_num* *ID* #REQUIRED > ⇒

<!ELEMENT *post* (#PCDATA) >

<!ATTLIST *post* *ref* *IDREF* #IMPLIED >

< name reg\_num="1045">

Иванов </ name>

< name reg\_num="1052">

Петров </ name>

<<rost ref = " 1045">

Зав. отделом </ post>

Признак обязательности:

#REQUIRED – обязательно записывать в э

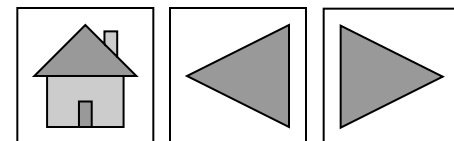
#IMPLIED – обязателен, у него нет значен

#FIXED – у атрибута только одно значение

Иванов = зав. отделом

элементе

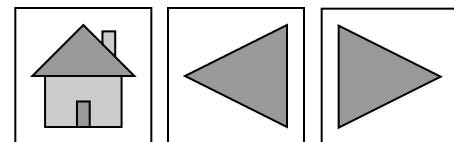
ия по умолчанию



## Примеры

```
< / ATTLIST name  
  first CDATA #REQUIRED  
  second CDATA#IMPLIED  
  surname CDATA#REQUIRED >
```

```
< / ATTLIST bk:listing  
  bk:number ID #REQUIRED  
  bk:head CDATA #REQUIRED >
```



объявление  
сущности  
(*entity*)

Внутренние сущности: - задаются при объявлении сущности

```
<!ENTITY #author "
    &ссылка на сущность
    <!ENTITY #cover " ; Название &author ;"
    title. >
```

Внешние сущности: -содержаться в отдельных файлах

```
<!ENTITY tel
    SYSTEM "http://www.tty.com/TelDef.xml" >
<!ENTITY tel
    PUBLIC "_//DTD/Tele..." "http://www.tty.com/TelDef.xml" >
```

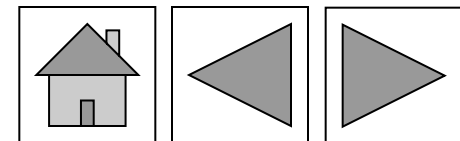
*PUBLIC* – общеизвестное объявление

Параметризованная сущность:- используется только внутри объявления DTD

```
<!ENTITY % lang "ru_Ru" >
```

*%lang* : на параметризованную сущность

Используется только внутр и описания *DTD*



не  
разбираемая  
программой  
аниматором

→

Сущность которая не обрабатывается средствами *XML*

```
<!ENTITY tel_logo
  SYSTEM "/ image / TelDef logo 32x16.gif" _NDATA image - gif >
```

*NDATA* – определяет не разбираемую сущность  
*image - gif* – программа, которая открывает изображение

Объявление  
обозначения  
(NOTATION)

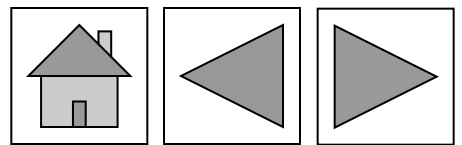
→

й

```
<!NOTATION image_gif SYSTEM "viewer.exe"
```

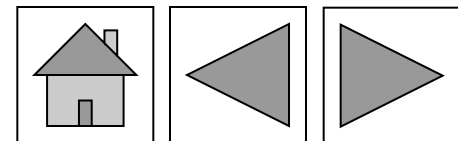
к  
к связывает *image - gif* с программой обработки  
к изображений в файле *viewer.exe*

к  
л



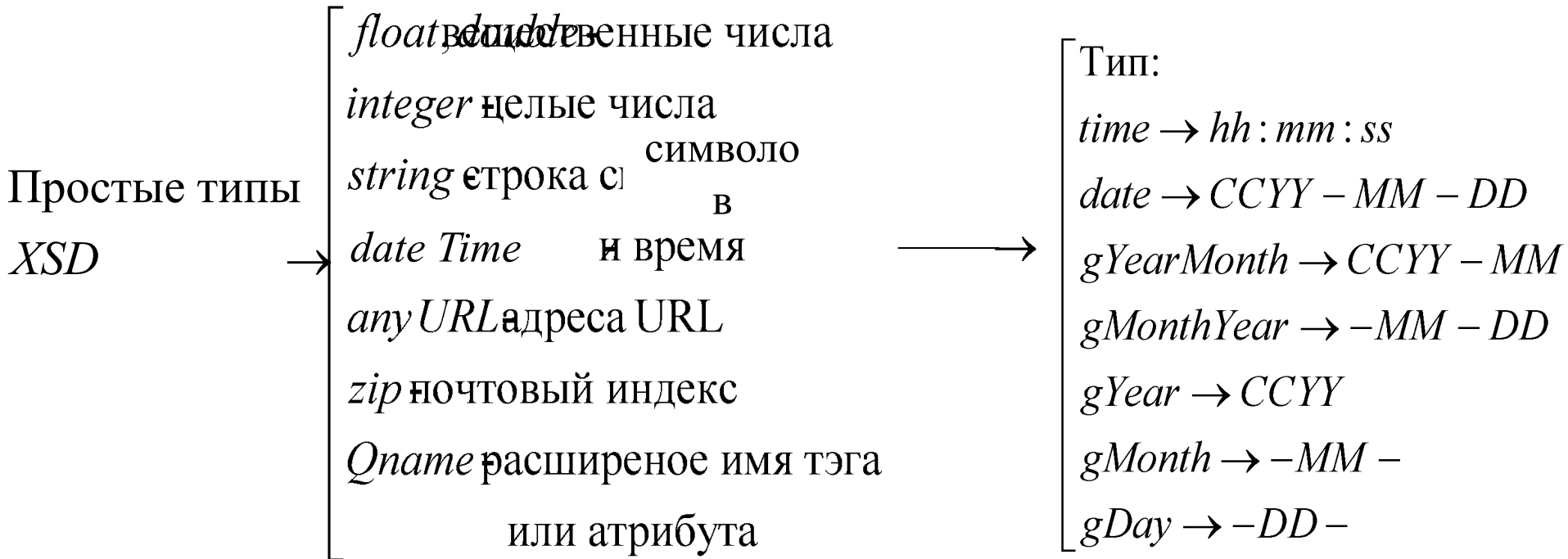
## Пример описания DTD. Записная книжка

```
<!ELEMENT notebook (person)* >
<!ELEMENT person (name, birthday?,
  address*, phone - list?) >
<!ELEMENT name EMPTY >
<!ATTLIST name
  first CDATA# IMPLIED
  second CDATA# IMPLIED
  surname CDATA# REQUIRED >
<!ELEMENT birthday (# PCDATA)
<!ELEMENT address (street ,city, zip)? >
<!ELEMENT street (# PCDATA) >
<!ELEMENT city (# PCDATA) >
<!ATTLIST city
  type (поселок деревня город) " " >
<!ELEMENT zip (# PCDATA) >
<!ELEMENT phone - list (work - phone*,
  home - phone*) >
<!ELEMENT work - phone (# PCDATA) >
<!ELEMENT home - phone (# PCDATA) >
```

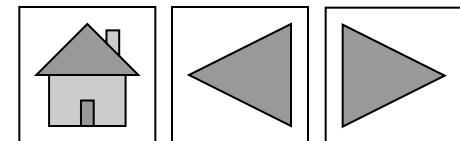


# Язык XSD (Описание схемы документа – XML Schema Definition)

XSD – реализация XML. Корневой элемент schema.



Пример: `<xsd:simpleType base="string" name="mySimpleType" />`



Сужение

*restriction* →

```
< xsd : simpleType name = " zip " >  
< xsd : restriction base = " xsd : string " >  
< xsd : pattern value = "[0-9]{6}" / >  
pattern – почтовый индекс zip
```

6 арабских цифр

фасетка “регулярное выражение”

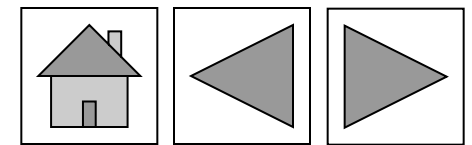
6 арабских цифр

```
< /xsd : restriction >  
< xsd : simpleType >  
< xsd : simpleType name = " zip " >  
< xsd : restriction base = " xsd : positive Integer " >  
< xsd : min Inclusive value = "100000" / >  
< xsd : max Inclusive value = "999999" / >  
< /xsd : restriction >  
< /xsd : simpleType >
```

→

100000 < zip < 999999

Фасетка (facets) “наибольшее значение”





$\mathbb{K}$   
 $\mathbb{K}$   
 $\mathbb{K}$  *maxExclusive*- наибольшее значение, которое уже не  
входит в определяемый тип.

$\mathbb{K}$  *maxInclusive*- наибольшее значение определяемого типа

$\mathbb{K}$   
 $\mathbb{K}$  *minExclusive*- наименьшее значение, которое уже не входящее  
в определяемый тип

$\mathbb{K}$  *minInclusive*- наименьшее значение определяемого типа

*facet* =  $\mathbb{K}$  *totalDigits*- общее количество цифр в числовом типе  
 $\mathbb{K}$

*s*  $\mathbb{K}$  *fractionDigits* количество цифр в дробной части числа

$\mathbb{K}$   
 $\mathbb{K}$  *length*- длина значений определяемого типа

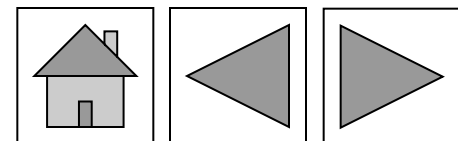
$\mathbb{K}$  *maxlength*- наибольшая длина значений определяемого типа

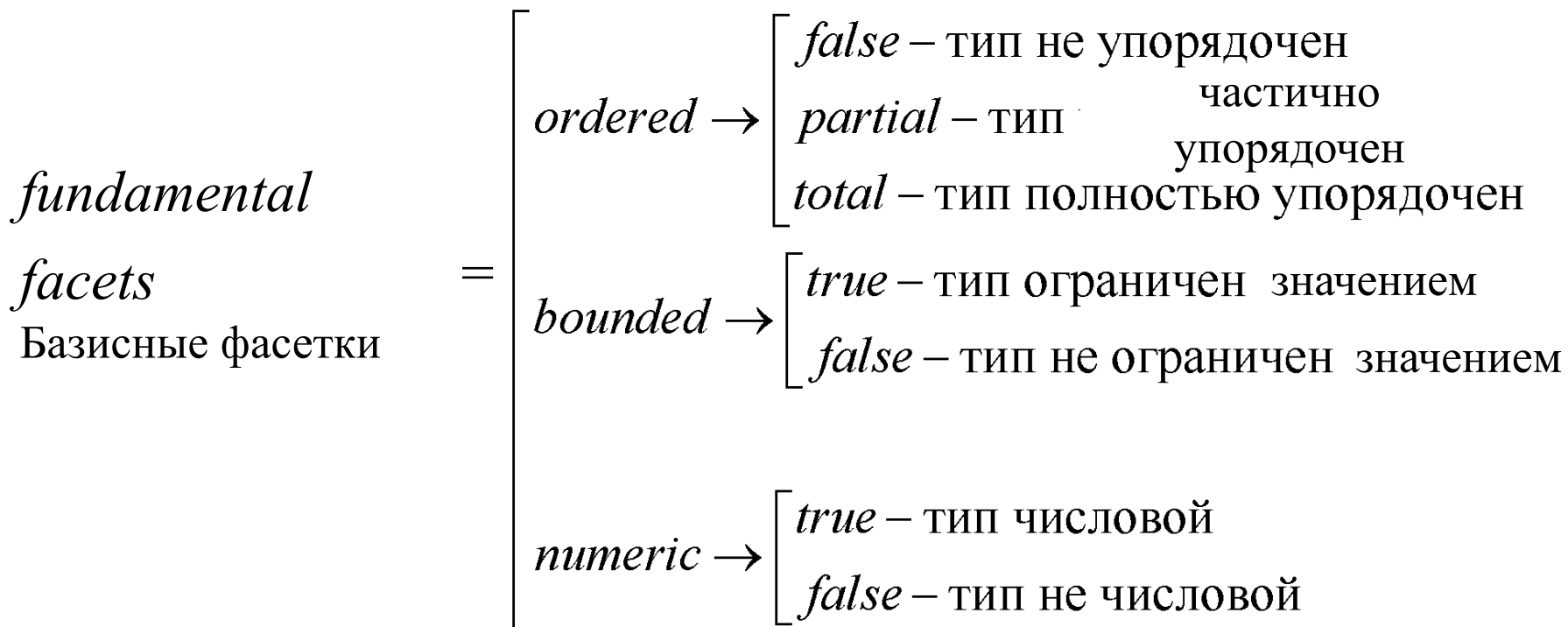
$\mathbb{K}$   
 $\mathbb{K}$  *minlength*- наименьшая длина значений определяемого

$\mathbb{K}$  *pattern*- регулярное выражение типа

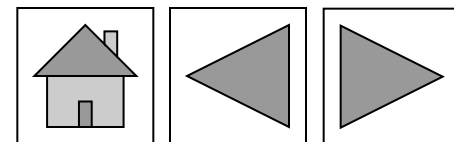
$\mathbb{K}$   
Л

М





Используются как атрибуты тэгов в фасетках



Список

*list* →

*< days > 21 34 55 46 < /days >*

↓ схема

*< xsd : simpleType name < xsd : element name = "days"  
type = "list of Integer" />* → список из целых чисел < 5

*< xsd : restriction >*

*< xsd : simple Type >*

*< xsd : list item Type = "xsd : Integer" />*

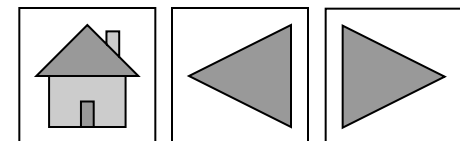
*< xsd : Simple Type >*

→ ТИП ЭЛЕМЕНТОВ

*< xsd : max Length value = "5" />*

*< /xsd : restriction >*

*< xsd : Simple Type >*

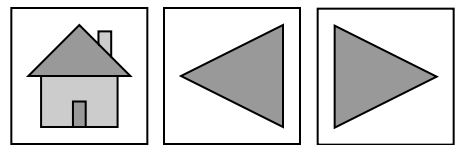


Объединение  
*union* →

```
xsd : union member Type = "xsd : string  
xsd : integer Listofinteg  
er" />  
-----  
< xsd : attribute name = "size" >  
< xsd : simpleType >  
< xsd : union >  
< xsd : simpleType >  
< xsd : restriction base = "xsd : positive Integer" >  
< xsd : min Inclusive value = "8" />  
< xsd : max Inclusive value "72" />  
< /xsd : restriction >  
< /xsd : SimpleType >  
+ < xsd : SimpleType >  
< xsd : restriction base = "xsd : NMTOKEN" >  
< xsd : enumeration value = "smale" / >  
< xsd : enumeration value = "medium" / >  
< xsd : enumeration value = "large" / >  
< /xsd : restriction >  
< /xsd : simpleType >  
< /xsd : union >  
< /xsd : Simple type >  
< /xsd : attribute >
```

→

```
< font size = "large" >  
Глава 1 < / font >  
  
< font size = "12" >  
Простой текст < / font >
```



Объявление элементов *element* →

```

< xsd:element name = " degree " type = " xsd:nonPositiveInteger "
  minOccurs = " 1 " maxOccurs = " 1 " />

```

по умолчанию = 1

Объявление атрибутов *attribute* →

```

< xsd:attribute name = " id " type = " positiveInteger "
  use = " required " />

```

use =

- optional* – необязателен
- required* – обязателен
- prohibited* – неприменим (определение подтипа с целью отменить атрибуты базового типа)

↓

```

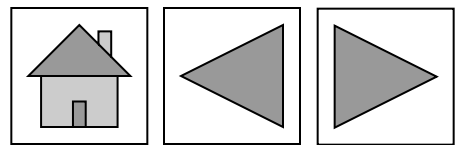
< xsd:attribute name = " id " type = " positiveInteger "
  use = " required " />

```

```

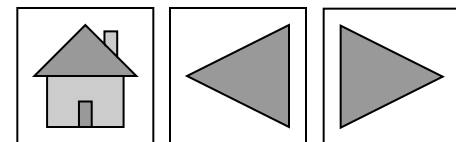
< xsd:attribute name = " name " type = " NCName "
  default = " anonymous " />

```



## Схема XSD книжки

```
<xsd:shema xmlns:xsd="http://...XMLSchema" >
  <xsd:element name="notebook" type="notebook Type" /
  >
  <xsd:complexType name="notebook Type" >
    <xsd:element name="person" type="person Type"
      minOccurs="0" maxOccurs="unbounded" /
    >
  </xsd:complexType >
<xsd:complexType name="person Type" >
  <xsd:sequence >
    <xsd:element name="name" >
      <xsd:complexType >
        <xsd:attribute name="first" type="xsd:string"
          use="optional" /
        >
        <xsd:attribute name="second" type="xsd:string"
          use="optional" /
        >
      </xsd:complexType >
    </xsd:complexType >
  </xsd:sequence >
</xsd:complexType >
</xsd:shema >
```



## Язык XLINK

Пространство имен  
*XLINK*

→

Содержание элемента

→

*< some Element xmlns : xlink = "http://www.w3.org/1999/xlink" >*  
*< /some Element >*

Атрибуты  
*XLINK*

→

*type* – тип ссылки

→

*href* – адрес ресурса

*show* – способ показа ресурса

*actuate* – момент активации ссылки

*label, from, to* – начальные и конечные пункты ссылок

*role, arcrole, title* – смысл ссылки

→

*simple* – простая ссылка

*extended* – расширенная ссылка

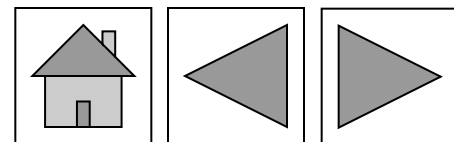
*resource* – информационный ресурс

*locator* – указатель на ресурс

*arc* – дуга графа

*title* – заголовок

*none* – остальные элементы не имеют отношения к ссылкам



Атрибут

*Simple*

простая ссылка



```

Simple – аналог a, img
< some Lnk xmlns : xlink = "http : ..."
  xlink : type = " simple "
  xlink : href = " http : ... k1d2012.xml " >
< /some Lnk >

```

Атрибут

*Extended*

расширенная  
ссылка



```

Extended – содержит внутри себя другие ссылки
< mult link xmlns : xlink = "http : ..."
  xlink : type = " extended "
  < src xlink : type = " resource " xlink : label = " s0012 " / >
  < tgt xlink : type = " locator "
    xlink : href = " http : ... udr01.xml "
    xlink : label = " f0012 " / >
  < ref xlink : type = " arc " from = " s0012 " to " f0012 " / >
< /mult link >

```

элемент

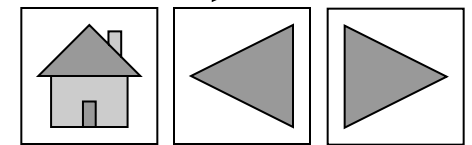
ссылка



*Ключевые слова документов*

информ. ресурс

указатель на ресурс





Атрибут

*RESOURCE* → [Отмечает локальный ресурс, в котором записана ссылка

Атрибут

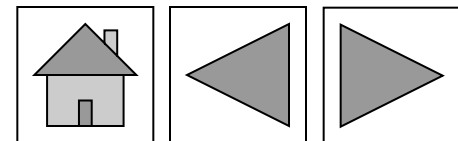
*LOCATOR* →

Описывает удаленный ресурс

Используется при создании ссылки дуги типа *arc*

```
< my Biogr xlink : type = "locator"  
                               указатель на ресурс  
  xlink : href = "http : ....xml"  
  xlink : label = "L1234" / >
```

Ссылка типа *locator* применяется только внутри *extended*



Атрибут

*ARC* →

Описание связи между двумя ресурсами

*From* используется в дугах указания начальной и конечной точек

```
<tplink xlink:type="extended" >
```

```
  <scresource xlink:type="locator" xlink:label="loc"
    xlink:href="http://...reso24.xml" / >
```

указ. на ресурс удалённый

```
  <scresource xlink:type="locator" xlink:label="base"
    xlink:href="http://...reso43.xml" / >
```

указ. на ресурс удалённый

```
  <load xlink:type="arc" xlink:from="loc"
    xlink:to="base" / >
```

```
</tplink >
```

Атрибут

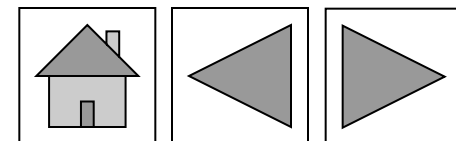
*TITLE* →

Дает описание расширенной ссылке

```
<rem xlink:type="title" >
```

Другие материалы по теме

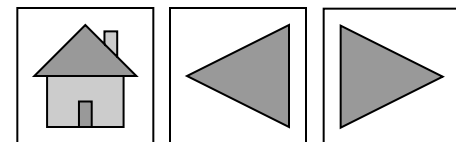
```
</rem >
```



Атрибут *Show* → Используется в ссылках типа *simple* и *arc* для отображения на экране  
Значение атрибута *Show* :

- new* – показать в новом месте
- replace* – показать в месте где была ссылка на ресурс
- embed* – при показе первоначальный ресурс не подвергается преобразованию
- other* – способ представления описания в других элементах документа
- none* – способ представления не описывается ни в одном элементе документа

```
<img xlink:type="simple"
xlink:href="http://...sigma.gif" >
xlink:show="embed"
xlink:style="display:none" / >
```



Атрибут  
*actuate* →

Используется в ссылках *simple* и *arc* для определения момента времени активации ссылки:

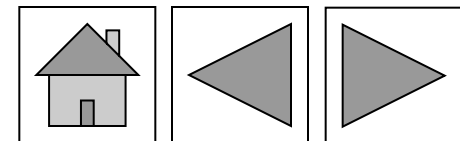
*on load* – при загрузке документа

*on Request* – по событию

*other* – способ активации описан в других элементах

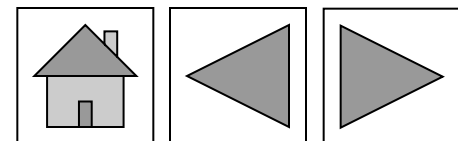
*none* – способ активации не описан ни в одном элементе документа

```
< sing xlink : type = " simple "  
    xlink : href = " http : // ... intro.xml "  
    xlink : actuate = " on load " / >
```



Атрибут *role* → Не используется в типах: *arc*  
Атрибут *role* указывает на ресурс, описывающий ссылку  
< *problem xlink : type = "simple"*  
    *xlink : href = "..."*                   - адрес ссылки  
    *xlink : role = "..."*                   - адрес ресурса описания ссылки  
Решение интегрального уравнения  
< / *problem* >

Атрибут *arcrole* → Аналогично *role* применяется в ссылках, *arc*  
< *load xlink : type = "arc"*  
    *xlink : arcrole = "..."*  
    *xlink : from = "cont" xlink : to = "base" xlink : actuate = "onload" />*



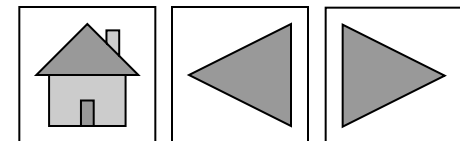
## Уточненные ссылки

Простые  
указатели →

XPointer  
< my Link xlink : type = "simple"  
xlink : href = "#id02" / > – ссылка с указанием ресурса  
р ресурса  
< my Link xlink : type = "simple"  
xlink : href = "#id02" / > – ссылка внутри документа

Указатели,  
основанные  
на схеме →

xpointer (/ book / chapter / section) element (color / 3)  
↓ ↓  
< book > < attr id = "color" >  
< chapter name = "ch5" title = "Chapter 5" > < rgb ... < / rgb >  
< section name = "sect1" > < hsb ... < / hsb >  
... < cmyk c = "25" ... / >  
< / section > < icc ... < / icc >  
< / chapter >  
< / book > < / attr >





Схема

*xpointer* ( ) →

*xpointer* (/contract)  
*xpointer* (/contract / section)  
*xpointer* (/contract / section / paragraph)  
*xpointer* (/contract / section / \*)  
*xpointer* (/contract / \* / paragraph)  
*xpointer* (/ / \*)  
*xpointer* (/contract / section[1])  
*xpointer* (/contract / section [connt(\*) = 2])  
*xpointer* (/contract / section[2] / paragraph[1])  
Все  
текущий элемент  
родительский элемент  
порядковый номер из выбранных элемент

< /contract >

---

< contract numb = "5" >

< section id = ... >

< paragraph > ... < / paragraph >

⊠

< /section >

< remark > ... < / remark >

< section id = "sect 2a" >

⊠

< /section >

< remark > ... < / remark >

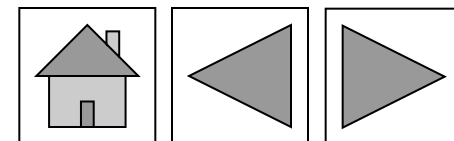
< section id = "sect 3a" ... >

⊠

< /section >

ОВ

< /contract >





Cxema

*xmlns* ( ) →

*xmlns* (*abc* = *http://example.com/ns/abc*)

1) *xmlns* (*img* = *http://example.org/image*)

*img:rect* (10,10,50,50)

2) *<customer xmlns="http://example.org/customer">*

*<name xmlns=http://example.org/personal-info">*

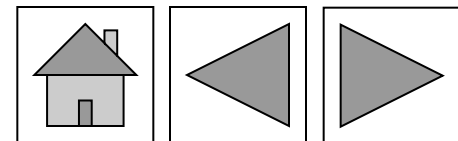
*<customer >*

*John Doe </name >*

*xmlns* (*c* = *http://example.org/customer*)

*xmlns* (*p* = *http://example.org/personal-info*)

*xpointer* (*/c:customer / p:name*)



## Язык XPath

Узлы дерева

документа →

узлы документы ( *document nodes* )

узлы элементы ( *element nodes* )

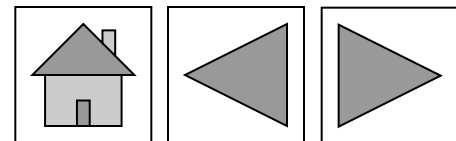
узлы атрибуты ( *attribute nodes* )

узлы пространства имен ( *namespace nodes* )

узлы инструкций по обработке ( *processing instruction nodes* )

узлы комментария ( *comment nodes* )

текстовые узлы ( *text nodes* )



Оси поиска →

*ось* :: *текст узла*[*предикат*] → *child* :: *section*[1] → *section*[1]

⊠

*self* – текущий узел

*child* – узлы потомки, кроме узлов-атрибутов и узлов пространств имен

*descendant* – узлы потомки с их потомками

*descendant – or – self* – объединение *descendant* и *self*

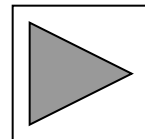
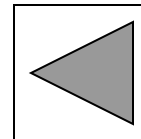
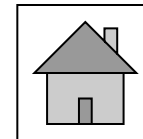
*following* – все узлы, лежащие "ниже" текущего узла

*ancestor* – все узлы-предки текущего узла

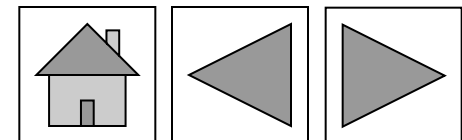
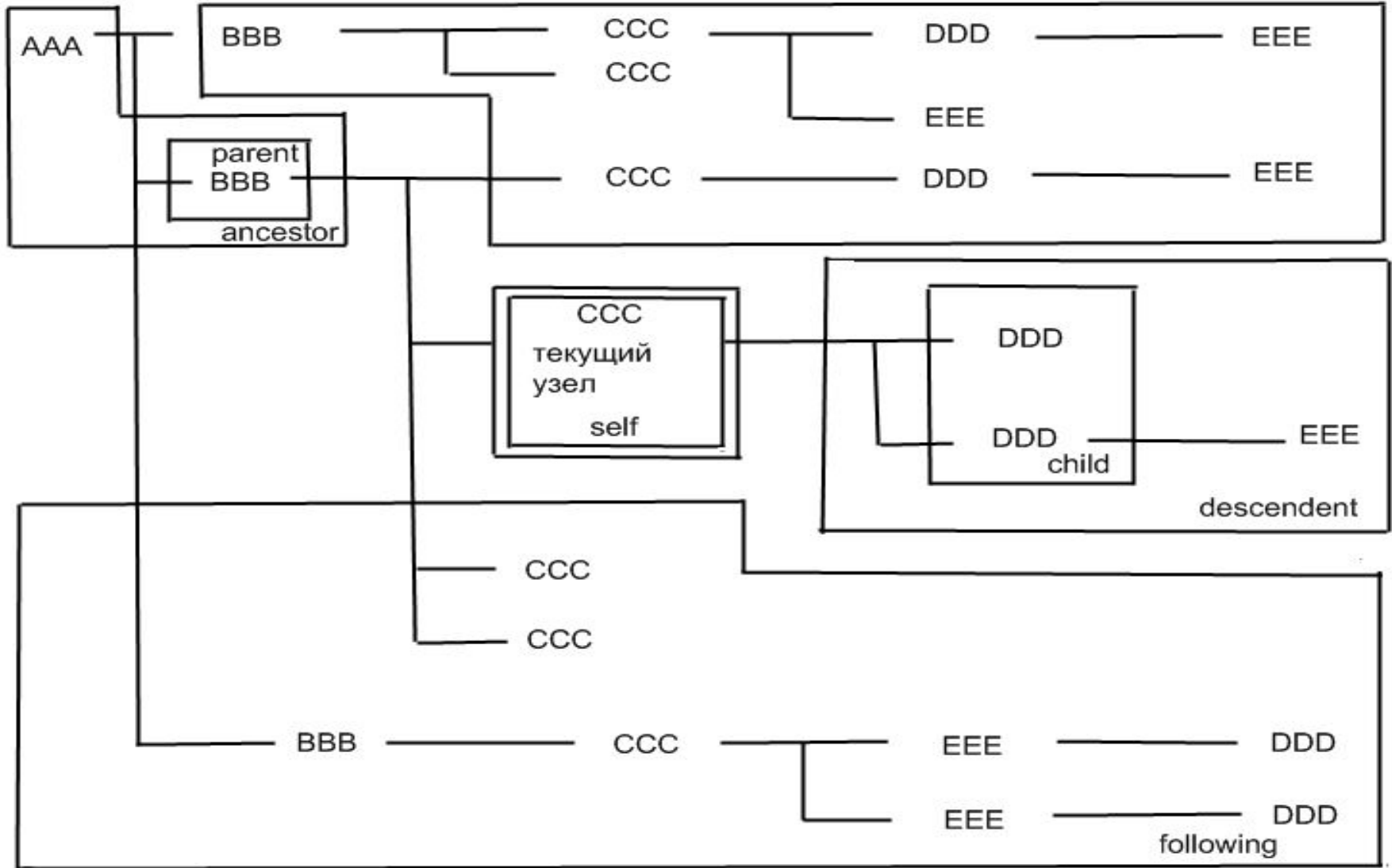
*preceding* – узлы, предшествующие текущему узлу

*parent* – непосредственный предок текущего узла

По умолчанию *child*



# Области, определяемые осями



Тест по имени

узла

*name test*

→

*self* текущий, узел дерева

*child* непосредственные потомки узла

*person*

*descendant* все потомки узла

*person*

*descendant – or – self* все потомки узла

*person*

и са

м узел *person*

*parent* родительский узел узла

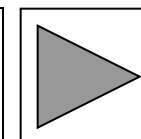
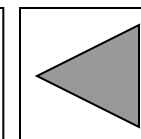
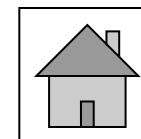
*person*

*ancestor* все предки узла

*person*

*attribute* все атрибуты узла

*person*



Тест по виду  
узла →  
(*kind test*)

*node* () отбирает узел любого вида

*text* () отбирает текстовые узлы

*element* () отбирает все узлы элементы

*element* (*name*) отбирает узлы элементы с именем *name*

*element* (*name*, *type*) отбирает узлы элементы с именем *name*

и типом *type*

*attribute* () отбирает все узлы атрибуты

*attribute* (*name*) отбирает все узлы атрибуты с именем *name*

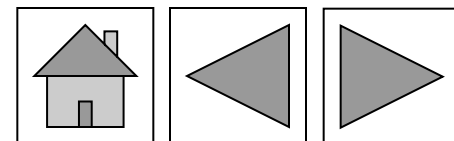
и типом *type*

*document node* () отбирает все корневые узлы документа

Пример

*child :: element* () – отбирает узлы потомки вида узлы-элементы

*self* отбирает текущий узел-комментарий



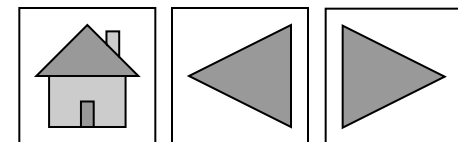
Предикаты →

- $\exists x \forall y [VVV \text{ элемент} [VVV] \text{ , вложенный в } CCC]$
- $\exists x \forall y [VVV \text{ элемент} [VVV] \text{ – , вложенный в } CCC]$
- $\forall x [VVV \text{ элемент} [VVV] \text{ документа с двумя в } CCC \text{ вложенными элементами } CCC]$
- $\forall x [VVV \text{ элемент} [VVV] \text{ документа с двумя } CCC \text{ любыми вложенными документам } CCC]$
- $\forall x [VVV \text{ элемент} [VVV] \text{ – , имеющие атрибут } CCC \text{ } name]$
- $\exists x [VVV \text{ элемент} [VVV] \text{ – } CCC \text{ с атрибутом } id = 3]$
- $\forall x [VVV \text{ элемент} [VVV] \text{ – имеющие атрибут } CCC \text{ } name]$

*if* →

- if* (выражение1) *then* (выражение2) *else* (выражение3)
- if* (\$plsex = "M") *then* "father" *else* "mother"
- if* (условие) *then* / count / price *else* () –

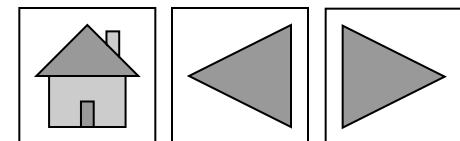
Ветвь *else* – обязательна!!!



Циклы →  $\left[ \begin{array}{l} \text{for } \$n \text{ in } / BBB \text{ return } \$y / CCC - , \quad CCC \\ \text{выражение1} \quad \text{выражение2} \\ \text{вложенные в узлы} \quad BBB \\ \text{for } \$X \text{ in } // BBB \text{ return } (\text{for } \$y \text{ in } \$x / CCC \text{ return } \$y / DDD) \\ \text{вложенный цикл} \end{array} \right.$

Кванторы (quantifiers) существования →  $\left[ \begin{array}{l} \text{exists } \$n \text{ in } / person? name / satisfies \$n = " \quad " \\ \text{существует хотя бы один сотрудник по имени "Федор"} \end{array} \right.$

Квантор все общности →  $\left[ \begin{array}{l} \text{every } \$n \text{ in } / person / education satisfies \$n = " \quad ee" \\ = true \text{ если выражение2 всегда равно } true \\ = false \text{ если выражение2 один раз принимает значение } false \\ \text{все ли сотрудники имеют высшее образование} \end{array} \right.$





Операции с

множествами →

`doc("bids.xml")/*/bid` выбирает все элементы и  
`doc("bids.xml")/*/bid[bid_date > date("2002-01-01")]`  
 вложенные во все элементы AAA  
 выделяет узлы продукции стоимостью > 1000 и датой > 2002-01-01  
`doc("bids.xml")/*/bid[1]` первая последовательности  
`except` – дополнение узлов, не содержащиеся в о второй  
`doc("bids.xml")/*/bid[2]` последовательности

### Язык запросов XQuery

Прямой конструктор  
элемента →

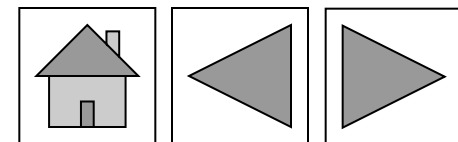
```

< person id = "92-3456" sex = "male" >
  < name > Иван Петрович < /name >
  < age > 30 < /age >
< /person >

```

Выражение в  
содержимом

конструктора → `< age > {10 + 20} < /age >`



Выражение в

атрибутах

конструктора →

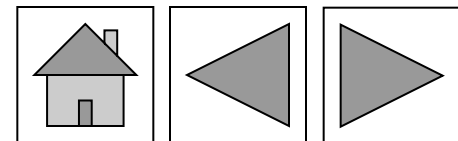
$\left[ \begin{array}{l} \langle person id = "92 - 3456" sex = "{/notion / sex[1]}" \rangle \\ \langle person id = 9\{1+1\} - 3456" sex = "male" \rangle \end{array} \right.$

Вычисляемые

конструкторы →

$\left[ \begin{array}{l} element\ person\{ \\ \quad attribute\ id\ \{"92 - 3456"\} \\ \quad attribute\ sex\ \{/notion / sex[1]\} \\ \quad \text{Элемент} \{ " \\ \quad \} \\ \quad element\ age\ \{10 + 20\} \\ \} \end{array} \right.$

$\Rightarrow$   $\left[ \begin{array}{l} element \\ \quad \{/notion / name[1]\} \\ \quad \{attribute\ id\ \{"92 - 3456"\} \\ \quad attribute \\ \quad \quad \{/notion / attr[1]\} \\ \quad \quad \{/notion / sex[1]\} \\ \quad \text{Элемент} \{ " \\ \quad element\ age\ \{10 + 20\} \\ \} \end{array} \right.$



Выражение  
запроса  
FLWOR →

*for* \$p := ("Иванов", "Петр") ⇒ <name> Петр Иванов </ name>  
*return* < name > { \$n } < / name >

*for* \$n in ("Иванов ", "Петр ") ⇒ <name> Петр Иванов </ name>  
*return* < name > { \$n } < / name > <name> Иванов </ name>

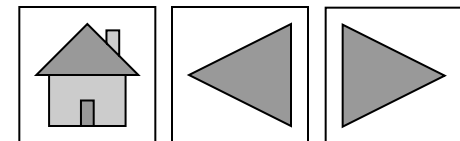
*for* \$i at "Иванов" ⇒ (\$n= "Петр", \$i=1)  
(\$n= "Иванов", \$i=2)

*for* \$i in (10,20), \$j in (1 to 3) ⇒ <x> 10,1 </ x>  
*return* < x > { \$i, ", ", \$j } < / x > <x> 10,2 </ x>  
<x> 10,3 </ x>  
<x> 20,1 </ x>  
<x> 20,2 </ x>  
<x> 20,3 </ x>

*for* \$x in / count / incr  
*where* \$x > 0  
*return* \$x

Если выражение истина, то  
Выполняется выражение *return*

*for* \$p in ("Иванов", "Петр") *order by* - сортировка  
*order by* \$n *descending* { *ascending* – по возрастанию  
*descending* – по убыванию  
*return* < name > { \$p } < / name >



< *bib* >

< *book* *year* = "2003" >

< ~~*title*~~ *Протоколы TCP IP* / *title* < / >

< *author* >

< ~~*last*~~ *Стивенс* *last* < / >

< ~~*first*~~ *W.* / *first* >

< / *author* >

< ~~*publisher*~~ *Дуалект* *publisher* < / >

< *price* > 220.00 < / *price* >

< / *book* >

< *book* *year* = "1992" >

< *title* > *Advanced Programming in the...* < / *title* >

< *author* >

< *last* > *Stevens* < / *last* >

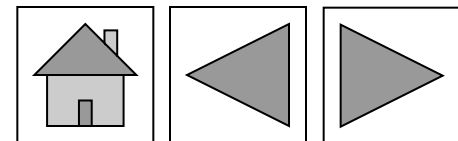
< *first* > *W.* < / *first* >

< / *author* >

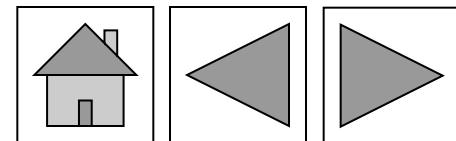
< *publisher* > *Addison - Wesley...* < / *publisher* >

< *price* > 65.95 < / *price* >

< / *book* >



```
< book year = "2000" >
  < title > Data on the Web... < /title >
  < author >
    < last > Abiteboul < /last >
    < first > Serge < / first >
  < /author >
  < author >
    < last > Buneman < /last >
    < first > Peter < / first >
  < /author >
  < author >
    < last > Sucin < /last >
    < first > Dan < / first >
  < /author >
  < publisher > Morgan Kaufmann... < / publisher >
  < price > 39.95 < / price >
< /book >
< book year = "1999" >
  ☒
< /book >
< /bib >
```



Выражение  
запроса  
FLWOR →

$\left[ \begin{array}{l} \text{Выражение } 1 < \$a1 > \text{Выражение } < \$b1 > \\ \text{Выражение } 2 < \$a2 > \text{Выражение } < \$b2 > \\ \text{Выражение } 3 < \$a3 > \text{Выражение } < \$b3 > \\ \text{Выражение } 4 < \$a4 > \text{Выражение } < \$b4 > \\ \text{Выражение } 5 < \$a5 > \text{Выражение } < \$b5 > \\ \text{Выражение } 6 < \$a6 > \text{Выражение } < \$b6 > \\ \text{Выражение } 7 < \$a7 > \text{Выражение } < \$b7 > \end{array} \right.$

Пример 1: Выделить из списка книг название и год издания книг, опубликованных

```

Addison_Wesley 1992
<bib> {
  for $b in doc("bib.xml")/bib/book
  where $b/publisher = "Addison_Wesley" and $b/@year > 1991
  return
    <book year="{ $b/@year }">
      { $b/title }
    </book>
  }
</bib>

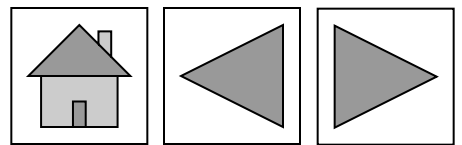
```

⇒

```

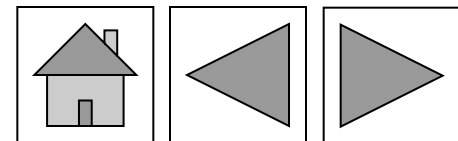
<bib>
  <book year="1992">
    <title>Advanced... </title>
  </book>
</bib>

```



Пример 2: Получить список названий книг и их авторов

```
<results >{  
  for $b in doc ("bib.xml") / bib / book,  
    $t in $b / title,  
    $a in $b / author  
  return  
    <result >  
      {  
        $t  
        $a  
      }  
    </result >  
}  
</results >
```



Пример 3: Выбрать книги название которых заканчивается на "02"  
и узлы, содержащие слово "suciu".

```
for $b in doc("bib.xml") // book
let $l := $b /*[contains (string(.), "suciu")
and nds - with(local - name(.), "02")]
where exists ($l)
return
```

```
< book >
  {$b / title}
  {$l}
< / book >
```

```
< book >
  < title > Data on the Web < / title >
  < author >
    < last > Suciu < / last >
    < first > Dan < / first >
  < / author >
< / book >
```

Оператор

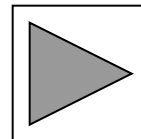
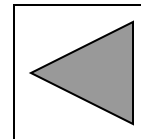
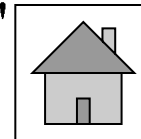
варианта →

```
typeswitch (// address)
```

```
case element(*, USAddress) return // address / state
```

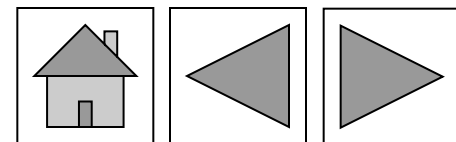
```
case element(*, RussiaAddress) return // address / region
```

```
default return "Unknow address type"
```





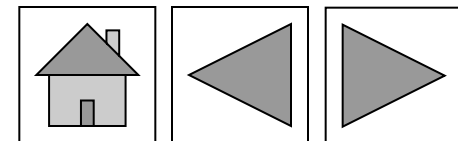
Пролог →  $\left[ \begin{array}{l} \textit{xquery version "1.0"} \\ \textit{declare namespace xyz = "http://some.domain/myns"} \\ \text{определение пространства имен} \end{array} \right.$



# Каскадные стили (CSS)

## Cascading style sheets

- Определение CSS →  $\left[ \begin{array}{l} \text{< Мета } http\_equiv = "Content\_Style\_Type" \\ \text{content} = "text / CSS" > \end{array} \right.$
- Компоновка CSS →  $\left[ \begin{array}{l} \text{связь файла CSS с HTML документа} \\ \text{< LINK REL = STYLESSHET HREF =} \\ \text{"http://www...mysheet.CSS" TYPE = "text.CSS" >} \\ \text{(описание стилей находится в отдельном файле)} \end{array} \right.$
- Внедрение CSS  
описание CSS внутри  
файла →  $\left[ \begin{array}{l} \text{< /HEAD >} \\ \text{< STYLE TYPE = "text / CSS" >} \\ \text{Здесь находятся определения стиля} \\ \text{< STYLE >} \\ \text{(определения стилей работают только внутри файла)} \\ \text{< BODY >} \end{array} \right.$



Встроенный CSS

описание CSS →

внутри тэга

```
<H1 STYLE = "color : blue" > ... </H1>
```

```
<DIV STYLE = "color : blue" >
```

```
<H1 > Заголовок </H1 >
```

```
<P > Абзац будет выведен браузером  
синим цветом </P >
```

```
</DIV >
```

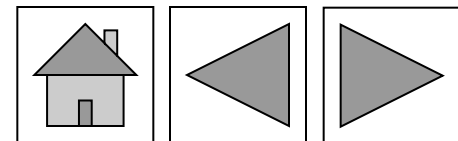
---

```
<SPAN STYLE = "color : blue" >
```

```
</SPAN >
```

встроенный CSS имеет приоритет над  
внедренным или связанным)

SPAN задает стиль для нескольких символов



Объединение листов →  
стилей

```
< LINK TYPE = "text / CSS" REL = "alternate stylesheet"  
TITLE = " Example" HREF = first.CSS >  
< LINK TYPE = "text/CSS" REL = "alternate stylesheet"  
TITLE = " Example" HREF = second.CSS >
```

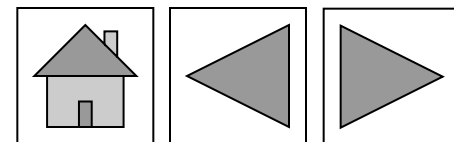
Конкурирующие CSS →

1. CSS в браузере  
2. CSS пользователя  
3. CSS браузера

Оперативные CSS внедренные CSS связанные CSS

3. H1 (color : red !important font - weight : bold  
font - family : sans - serif !important)  
(!important) → max

4. Классы > переопределение свойств тэга



Создание CSS для разных типов представлений →

```
< style type = "text / CSS" media = "screen" >
```

Определение стилей

```
< /style >
```

media →

screen – дисплей

print – принтер

⋮

all – все устройства

```
< STYLE TYPE = "text / CSS" media = " print " >
```

Определение стилей

```
< /STYLE >
```

Правила CSS →

```
H1 {color : blue}
```

```
P {font - size : 1 pt}
```

```
H1 {color : blue, font - size : 2 pt, text - align : center}
```

```
P, UL, LI {font - size : 2 pt}
```

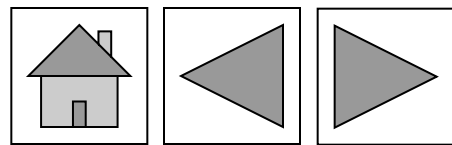
```
Strong {font-weight: normal}
```

pt – пункты (1..7) 1pt - 8px  
2pt - 10px



7pt – 36px

не будет видна жирность, но будет индексация



# Наследование CSS

(контекстные селекторы) →

- OL LI (*list-style-type: square*) — из нумерованного в маркерованный  
(для LI в списке OL)
- UL LI (*list-style-type: decimal*) — из маркерованного в нумерованный  
(для LI в списке UL)
- p span {color: blue}* - переопределение свойств тэга span внутри параграфов.

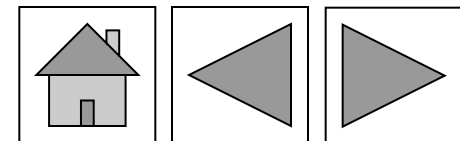
Классы CSS →

- H1.blue {color: blue}*
- H1.red {color: red}* — определение класса для дескриптора
- H1.black {color: black}*
- < H1 CLASS = red > Красный заголовок < /H1 >*
- .red {color: red}* — определение класса для любого дескриптора
- Ul li.forlist {background: blue}* - изменение свойств li для маркерованного списка

Специфический элемент CSS →

- < P ID = test > текстовый абзац < /P >*
- #test {color: red}* — определение класса для данного элемента

На 1 странице не может быть 2 одинаковых ID. Определяются глобальные классы



Псевдоклассы CSS →

Псевдокласс - специальный селектор, определяющий вид *HTML* в определенный момент

селектор: псевдокласс - { свойство: значение }

*A: link {color: blue}* - гиперссылки синие

*A: active {color: red}* - гиперссылки красные

*A: visited {color: yellow}* - гиперссылки желтые

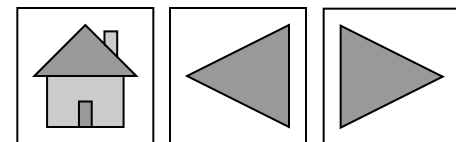
*A: hover {color: red}* → при наведении мышки

*P: first - line {font - weight : bold}* - первая строка жирная

*P: first - letter {font - weight : bold}* - первая буква жирная

Скрытие CSS от старых браузеров →

```
⋮  
< style type = "text / CSS " >  
< ! --  
H1 {color : red}  
-- >  
< /style >
```



# PHP

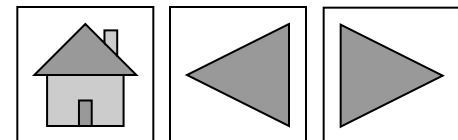
Интеграция  
PHP и HTML



- Переход в PHP
1. <php - стандартные тэги
  2. <? – короткие тэги
  3. <script language="php">  
<?php print "Welcome"; ?>  
</script>
  4. <% - тэги в стиле ASP

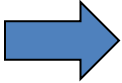


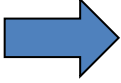
```
<?
$site_title = "PHP Recipies";
$bg_color = "white";
$user_name = "ivanov";
?>
<html>
<head>
<title> <? print $site_title; ?> <title>
</head>
<body bgcolor = "<? print $bg_color; ?>">
<?
//Присутствие тэгов HTML в команде
print "<h3>PHP/HTML integration</h3>"
<?
</body>
</html>
```



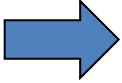


# Выражения , операторы и управляющие конструкции

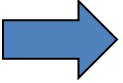
Elseif  If (выражение) {блок}  
elseif (выражение) {блок}

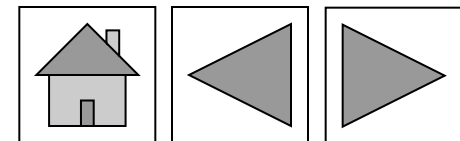
While  While (выражение):  
Блок  
Endwhile  
(выход -> выражение -> ложь)

```
$n=5;  
$ncopy=$n;  
$factorial=1;  
while($n>0):  
    $factorial = $n*$factorial;  
    $n--;  
endwhile;  
print "The factorial of $ncopy is  
$factorial"
```

Do ... While  Do:  
{Блок}  
While (выражение);

```
$n=5;  
$ncopy=$n;  
$factorial=1;  
do {  
    $factorial = $n*$factorial;  
    $n--;  
}while($n>0)  
print "The factorial of $ncopy is  
$factorial"
```

For  for (инициализация; условие; приращение){  
 блок}



Foreach → Foreach(массив as \$элемент) {  
    блок  
}

→ \$menu=array("pasta","steak", "potatoes","fish")  
foreach(\$menu as \$item {  
    print "\$item<BR>";  
}

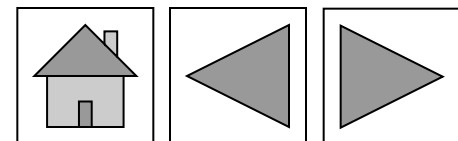
Foreach(массив as \$ключ =>элемент ) {  
    блок  
}

Switch → Switch (выражение)  
Case(условие):  
    блок  
Case(условие):  
    блок  
    ...  
    ...  
default:  
    блок  
}

→ switch(\$user\_input)  
    case("search"):  
        print "search";  
        break;  
    case("dictionary"):  
        print "dictionary";  
        break;  
    default:  
        print "here is the menu...";  
        break;  
}

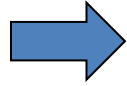
Break -> прерывает выполнение конструкций while , for или switch

Continue -> пропускаются оставшиеся команды цикла и начинаются новые итерации

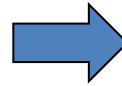


# Массивы

Индексируемые

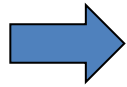


\$имя [индекс];

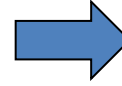


```
$meat[0]="chicken"  
$meat[1]="steak"  
$meat[2]="turkey"
```

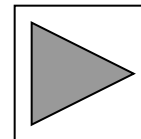
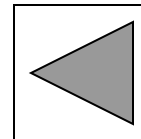
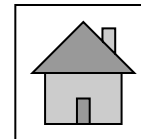
Ассоциативные



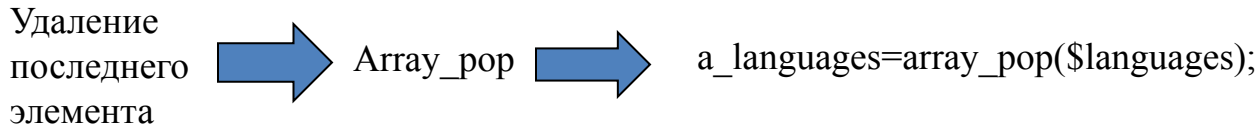
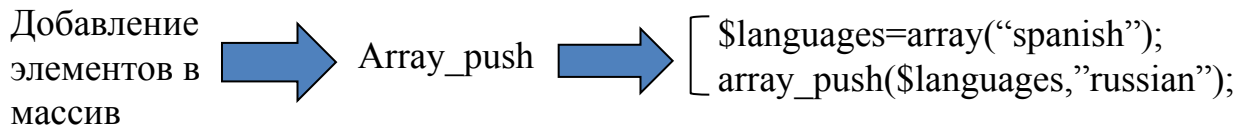
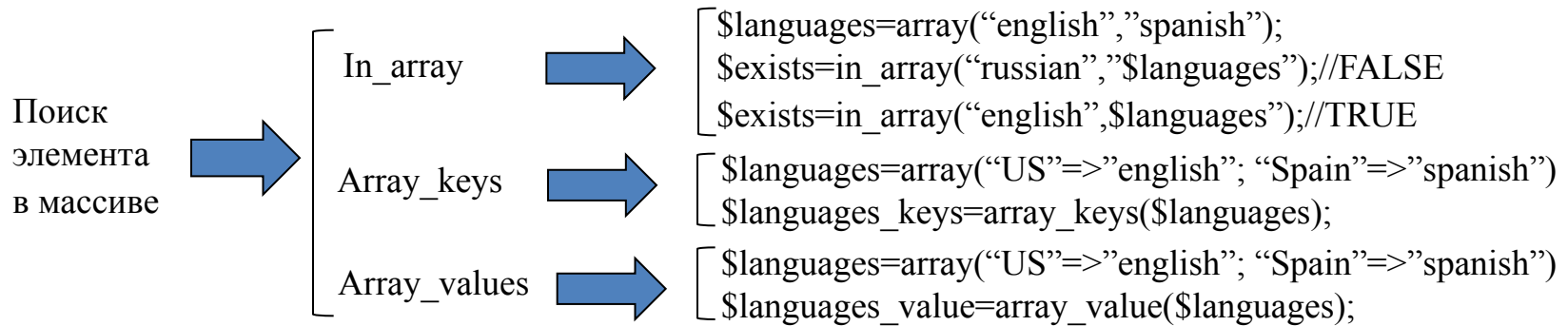
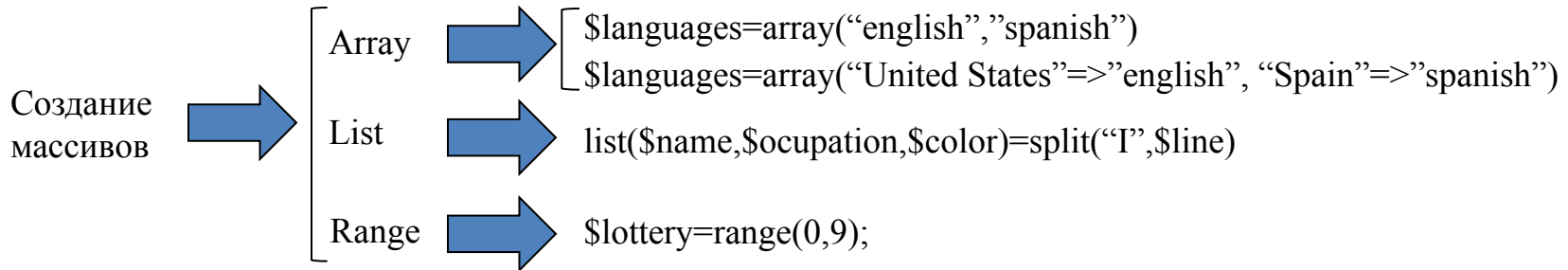
\$имя [ключ];



```
$languages["spain"]="spanish"  
$languages["france"]="french"
```



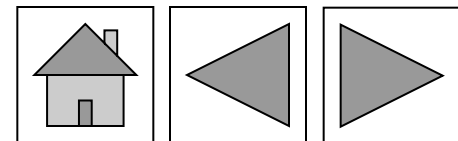
# Массивы



Sizeof( ) - количество элементов в массиве

Sort( ) } сортировка

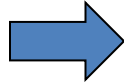
Rsort( ) } элементов массива



# Файловый ввод/вывод

Проверка  
существования  
файлов

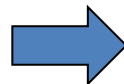
file\_exists



```
if(!file_exists ($filename)):
    print"File $filename does not exist!";
endif;
```

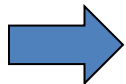
is\_file

(проверяет  
кроме  
существования  
операции  
чтения/записи)

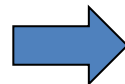


```
$file="somefile.txt";
if(is_file($file)):
    print"The file $file is valid and exists!";
else:
    print"The file $file does not exist or it is not a valid file!";
endif;
```

Открытие и  
закрытие  
файлов



fopen



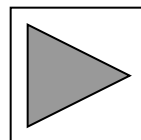
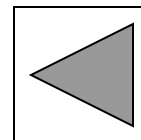
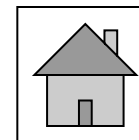
fopen(файл,режим)

Файл

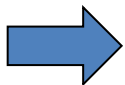
- имя локального файла
- php(стандартный поток I/O)
- http(подключение http к серверу)
- ftp(подключение ftp к серверу)

Режим

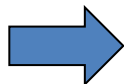
- r только чтение. Указатель в начало
- r+ чтение и запись. Указатель в начало
- w только запись. Указатель в начало  
содержимое файла уничтожается
- w+ чтение и запись. Указатель в начало  
содержимое файла уничтожается
- a только запись. Указатель в конец
- a+ чтение и запись. Указатель в конец



Открытие и  
закрытие  
файлов

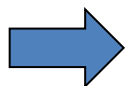


fclose

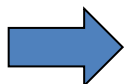


```
$file="userdata.txt";  
if(file_exists($file)):  
    $th=fopen($file,"r");  
    ...  
    fclose($th);  
else:  
    print"File $file does not exist!";  
endif;
```

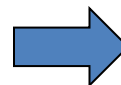
Запись  
в файл



fwrite  
is\_writable  
(существует  
и разрешена  
запись)

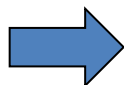


fwrite(манипулятор, переменная)

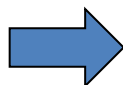


```
$data="08:13:00"  
$filename="somefile.txt"  
if(is_writable($filename)):  
    $th=fopen($filename,"a");  
    $success=fwrite($th,$data);  
    fclose($th);  
else:  
    print "could not open  
$filename for writing"  
endif;
```

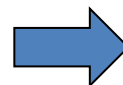
Чтение  
из файла



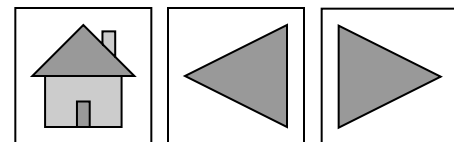
fread  
is\_readable  
(существует  
и разрешено  
чтение)



fread(манипулятор, длина [в байтах])

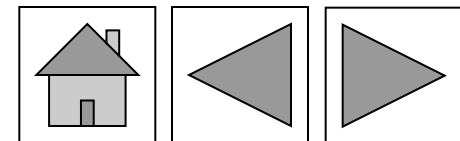


```
$th=fopen('pastry.txt','r');  
file=fread($th,filesize($th));  
print $file  
fclose($th);
```

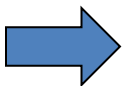


# Работа с файловой системой

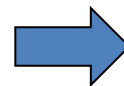
|   |   |                              |  |
|---|---|------------------------------|--|
| copy  | → | copy(источник,приемник)      |  |
| rename  | → | rename(старое_имя,новое_имя) |  |
| unlink<br>(удаление файла)                            | → | unlink(файл)                 |  |
| basename<br>(извлекает имя файла<br>из полного имени) | → | basename(полное_имя)         | → [ \$path="/usr/local/index.php";<br>\$title=basename(\$path);            |
| dirname<br>(извлекает путь из<br>полного имени файла) | → | dirname(путь)                | → [ \$path="/usr/local/index.php";<br>\$file=dirname(\$path);              |
| is_dir<br>(файл -> каталог?)                          | → | is_dir(имя_файла)            | → [ \$isdir =is_dir("insex.htm");//FALSE<br>\$isdir =is_dir("book");//TRUE |
| opendir   | → | opendir(путь)                | → [ открывает манипулятор<br>для работы с каталогом                        |
| closedir  | → | closedir(манипулятор)        | → [ закрывает манипулятор<br>работы с каталогом                            |



**readdir**  
(возвращает  
очередной  
элемент  
каталога)

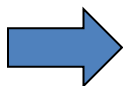


`readdir(манипулятор_каталога)`

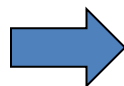


```
$path="/usr/local/index.php";  
$title=basename($path);
```

**chdir**  
(переход  
в каталог)



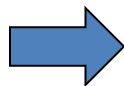
`chdir(каталог)`



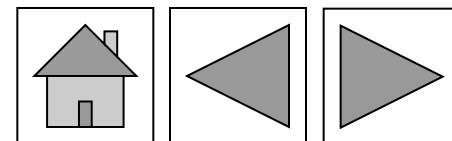
```
$newdir="book";  
chdir($newdir)  
$dh=opendir('.');  
print"files:";  
while ($file=readdir($dh)):  
    print("$file<br>");  
endwhile;  
closedir($dh);
```

**rewinddir**

(переводит указатель  
текущей позиции в  
начало каталога,  
открытого функцией  
`opendir()`)

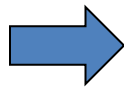


`rewinddir(манипулятор_каталога)`





## включение файлов в сценарий PHP



include → include(файл) →

```
if(some_condition):  
    include('text91a.txt');  
else:  
    include('text91b.txt');  
endif;
```

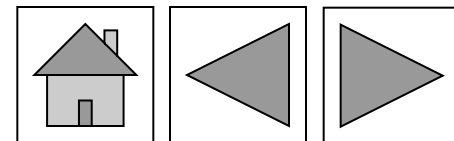
include\_once → include\_once(файл)

(проверяет, не был ли  
он включен ранее,  
если файл уже был  
включен ранее,  
то вызов игнорируется)

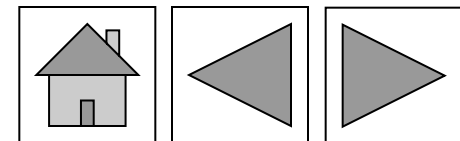
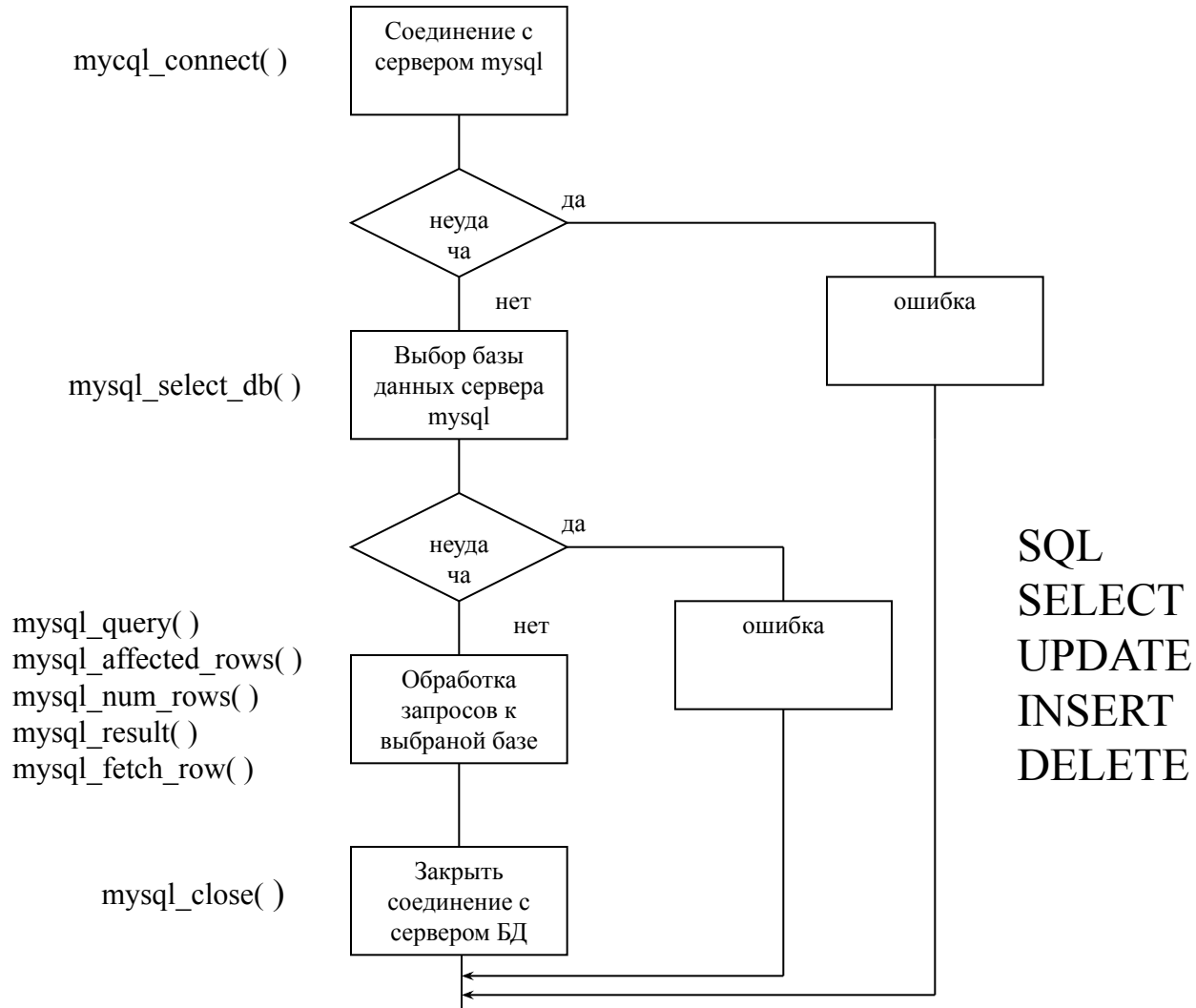
require → require(файл)

(включается всегда,  
даже если стоит  
в ветке "ложь")

require\_once(файл)



# Базы данных

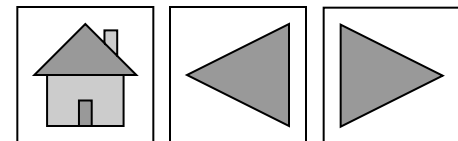


mysql\_connect →  $\left[ \begin{array}{l} \text{mysql\_connect}(\text{хост}, [\text{имя\_пользователя}], [\text{пароль}]) \\ \left[ \begin{array}{l} \text{Хост-имя хостового компьютера (по умолчанию локальный компьютер)} \\ \text{имя\_пользователя} \\ \text{пароль} \end{array} \right] \text{ в соответствии с} \\ \text{таблицами привелегий MYSQL} \end{array} \right.$

mysql\_connect ("localhost", "web", "4tf9222f")

mysql\_select\_db →  $\left[ \begin{array}{l} \text{mysql\_select\_db}(\text{имя\_базы\_данных}, [\text{идентификатор\_соединения}]) \\ \text{(обязательный для более одной открытой БД)} \\ \text{mysql\_select\_db}(\text{"company"}) \end{array} \right.$

mysql\_close → mysql\_close([идентификатор\_соединения])



mysql\_query



mysql\_query(запрос,[идентификатор\_соединения])

Запрос – запрос текста на SQL. Возвращает 0 в случае ошибки.  
При отсутствии идентификатора\_соединения запрос передается  
последнему открытому соединению

!!!mysql\_query – не выполняет запрос и работает совместно с  
mysql\_result и mysql\_affected\_rows

mysql\_affected\_rows



mysql\_affected\_rows([идентификатор\_соединения])

Определяет количество записей в запросе SQL  
с командами INSERT , UPDATE , REPLACE и DELETE

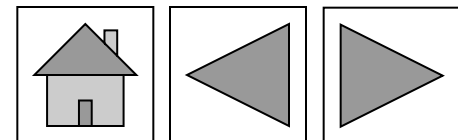
!не работает с SELECT



```
$query = "UPDATE products SET prod_name='cantaloupe'  
        WHERE prod_id='10001pr'";
```

```
$result = mysql_query($query);
```

```
print "Total row updated:". mysql_affected_rows( );
```

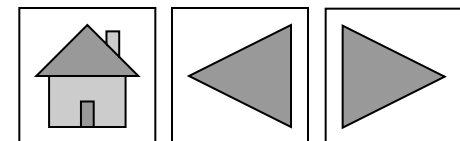


mysql\_num\_rows → mysql\_num\_rows(результат) → \$query="SELECT prod\_name FROM products WHERE ..."  
 Количество записей в команде  
 SELECT \$result=mysql\_query(\$query);  
 Print"Total rows selected:  
 ".mysql\_num\_rows(\$result);

mysql\_result → mysql\_result(идентификатор\_результата),запись,[поле]  
 получает результат SQL запроса  
 смещение в таблице  
 поле имя\_поля  
 имя\_поля.имя\_таблицы  
 \$query="SELECT \* FROM products"  
 \$result=mysql\_query(\$query);  
 \$id=mysql\_result(\$result,\$x,'prod\_id');  
 \$name=mysql\_result(\$result,\$x,'prod\_name');  
 \$price=mysql\_result(\$result,\$x,'prod\_price');

mysql\_fetch\_row → mysql\_fetch\_row(результат) → \$query="SELECT \* FROM products";  
 \$result=mysql\_query(\$query);  
 while  
 (\$row=mysql\_fetch\_row(\$result)):  
 print("\$row["prod\_id"]);  
 print("\$row["prod\_name"]);  
 print("\$row["prod\_price"]);

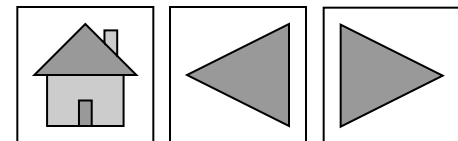
mysql\_fetch\_array - для ассоциативного массива.  
 Возвращает ассоциативный массив.



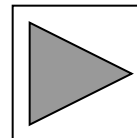
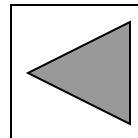
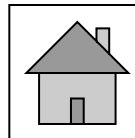
# Usability

## Категории

- Проблема первой категории, которая утратила первоначальную актуальность. Подобных ошибок при дизайне сайта желательно избегать, но их присутствие не приведет к краху сайта.
- Проблема второй категории и средней важности. При создании сайтов крайне желательно избегать таких проблем, но они больше не имеют первоочередного значения.
- Проблема третьей категории. Остается весьма актуальной. Очень важно, чтобы дизайнеры продолжали уделять ей самое пристальное внимание.



| Категория | Проблема  |
|-----------|---|
| 3         | Ссылки, которые не изменяют цвет после их использования     |
| 3         | Невозможность отмены выполненных действий                   |
| 3         | Открытие нового диалогового окна браузера                   |
| 3         | Всплывающее меню  |
| 3         | Элементы дизайна, похожие на рекламу                        |
| 3         | Нарушение основных принципов и традиций дизайна             |
| 3         | Странички с отсутствием понятного или полезного содержимого |
| 3         | Страницы, перенасыщенные текстом                            |
| 1         | Низкая скорость загрузки                                    |
| 1         | Кадры   |
| 2         | Флеш технологии   |
| 2         | Малоэффективные результаты поиска                           |
| 2         | Мультимедия и видео   |
| 2         | Фиксированная ширина страницы                               |
| 2         | Отсутствие межплатформенности                               |
| 1         | Неуверенные щелчки  |
| 2         | Прокрутка страниц   |
| 1         | Регистрация   |



| Категория | Проблема   |
|-----------|--|
| 1         | Регистрация  |
| 2         | Сложные url адреса   |
| 1         | Раскрывающиеся и каскадные меню  |
| 1         | Подключаемые модули и новые технологии                                 |
| 1         | Пользовательский интерфейс, созданный с помощью 3D технологий          |
| 1         | Перенасыщенный дизайн  |
| 1         | Заставки   |
| 1         | Изменяющиеся графические изображения и прокручивание текста            |
| 2         | Нестандартные элементы управления графическим интерфейсом пользователя |
| 1         | Отсутствие данных о поставщике информации                              |
| 1         | Придуманные термины  |
| 2         | Устаревшее содержимое  |
| 2         | Непоследовательная подача материала на сайте                           |
| 2         | Преждевременные требования о вводе конфиденциальной информации         |
| 2         | Многочисленные сайты одной компании                                    |

