

WWW — кратко обо всём

Автор: [/Roman Savochenko](#)

Оглавление

WWW — кратко обо всём	1
1. Введение	2
2. Язык гипертекстовой разметки (HTML, XHTML и XML)	2
2.1. Теги заголовка <HEAD>	3
2.2. Теги тела сообщения <BODY>	4
2.3. Фреймы	7
3. Каскадные таблицы стилей (CSS)	8
3.1. Атрибуты	8
3.2. Модификаторы тегов	10
4. JavaScript	11
4.1. Типы данных	11
4.2. Операции	12
4.3. Операторы	12
4.4. Функции	13
4.5. События и их обработка	14
4.6. Объекты	15
5. Приложение	19
5.1. Структура URL (Unifical resource locator)	19
5.2. CER (Character Entity Reference)	19
5.3. Цвета и их коды	19
5.4. Единицы размеров	20
5.5. Шрифты	20
6. Литература	21

1. Введение

Web технологии получают всё большее распространение, а также приобретают всё новые и новые возможности. Следовательно вопрос лаконичного изложения материала, в целом, является важным для адекватного восприятия технологии при изучении. Кроме этого немаловажным является решение проблемы поддержания знаний в актуальном состоянии.

Данный документ призван решить эти вопросы путем сжатого и лаконичного изложения материала граничащего со справочным представлением.

2. Язык гипертекстовой разметки (HTML, XHTML и XML)

Практически, язык гипертекстовой разметки появился первым, совместно с протоколом HTTP (Hyper Text Transfer Protocol). Следовательно изучение его, как основы Web-технологии является первоочередной задачей.

Язык разметки HTML является довольно мягким в плане требований к синтаксису, что и привело к значительным проблемам в стандартизации его обработки в различных WWW-браузерах. Для решения этой проблемы было принято решение на постепенный переход от HTML к более строгому языку XML. Промежуточной ступенью в этом переходе стал язык XHTML.

Основным элементом этих языков является контейнер, вид которого, условно, можно изобразить следующим образом: `<ИмяТега Атрибут1="Значение атрибута1" Атрибут2="Значение атрибута2"> Тело контейнера </ИмяТега>`

В языках XHTML и XML наличие закрывающего тега является обязательным требованием. В случае отсутствия тела контейнера вид контейнера примет вид: `<ИмяТега Атрибут1="Значение атрибута1" Атрибут2="Значение атрибута2" />`

Также, языками XHTML и XML запрещается пересечение тегов. Т.е. ранее открытые теги должны закрываться после тегов открытых в них.

Общая структура страницы на языке гипертекстовой разметки:

```
<?xml version='1.0' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF8"/>
    <title>Заголовок сайта</title>
  </head>
  <body bgcolor="#818181" text="#000000" link="#3366ff" vlink="#339999" alink="#33ccff">
    ...
    Тело страницы
    ...
  </body>
</html>
```

В случае с HTML эта структура может быть упрощена до вида:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
  <TITLE>Заголовок</TITLE>
  ...
</HEAD>
<BODY>
  ...
</BODY>
</HTML>
```

Как можно видеть из примеров, страница состоит из двух основных частей:

- Заголовка страницы (HEAD)
- Тела страницы (BODY)

Каждая часть имеет свой набор тегов который может незначительно пересекаться. Т.е. существуют теги которые могут использоваться как в одной части так и в другой.

Основные теги языка и их атрибуты:

- <HTML> ... </HTML>** — Заключает тело документа.
- <HEAD> ... </HEAD>** — Заключает заголовка документа.
- <BODY> ... </BODY>** — Заключает тело страницы.
 - BackGround* — цвет фона или рисунок фона (BACKGROUND="image.gif");
 - BGcolor* — цвет фона страницы;
 - Text* — цвет текста;
 - Link* — цвет гипертекстовых ссылок;
 - ALink* — цвет активных гипертекстовых ссылок;
 - VLink* — цвет пройденных гипертекстовых ссылок;
 - LeftMargin* — задаёт ширину левого поля страницы (в пикселах);
 - TopMargin* — задаёт ширину верхнего поля страницы (в пикселах).
- <BASE Href="http://my.resource.org/test/">** — Определяет базовую точку входа. Используется относительно адресуемыми ссылками. Может использоваться в тегах: HTML, HEAD, BODY а также вне тела документа.
 - Href* — ссылка на базовую страницу;
 - Target* — базовое имя фрейма, для отображения в нём внутренних ссылок.
- <FRAMESET> ... </FRAMESET>** — Описание фреймовой структуры документа. Определяется только на одной странице являющейся точкой входа на документ. Страница с фреймовой структурой может не содержать тела страницы (<BODY>).
- <NOFRAMES> Текст </NOFRAMES>** — Текст сообщения для браузеров не поддерживающих фреймы.
- <!-- Комментарий -->** — Комментирование участков текста.

2.1. Теги заголовка <HEAD>

- <TITLE> Hello </TITLE>** — Устанавливает имя окна браузера.
- <META Name="Refresh" HTTP-EQUIV="Refresh" Content="1; URL=refresh.htm" />** — Предоставляет управляющую информацию.
 - Name* — имя метаинформации (обычно совпадает с HTTP-EQUIV);
 - Content* — параметр опции, тип которой указан в HTTP-EQUIV;
 - HTTP-EQUIV* — тип опции:
 - “Refresh” — перегрузка страницы указанной в “Content” через указанное количество секунд (Content="1; URL=refresh.htm");
 - “Content-Type” — указывает кодировку документа (Content="text/html; Charset=UTF8");
 - “Cache-Control” — управление кешированием:
 - Content="no-cache" — запретить кеширование документа;
 - Content="no-store" — запретить хранение документа после пересылки.
 - “Description” — описание документа (для поисковых систем) (Content="My home page");
 - “Keywords” — ключевые слова документа для поисковых систем (Content="home page; Vasa Pupkin").
- <LINK Rel="stylesheet" Href="../css/css.htm" Type="text/css" />** — Обслуживание различных связей. Обычно используется для загрузки стилей (CSS).
 - Rel* — тип отношения:
 - “stylesheet” — загрузка таблицы стилей;
 - “shortcut icon” — указание иконки сайта, высвечивается в закладках браузера для данного сайта.
 - Href* — URL;
 - Type* — тип содержания (text/css, text/javascript).
- <STYLE Type="ТипОписанияСтилей"> описание стиля/стилей </STYLE>** — Содержит описание стиля всего документа.
 - Type* — тип стилей (text/css, text/javascript).
- <SCRIPT Type="ТипЯзыкаПрограммирования" Src="URL"> JavaScript/VBScript-код**

</SCRIPT> — Размещает на страницу код скрипта из файла. Код может помещаться непосредственно в теле контейнера.

Type — типа языка (JavaScript, VBScript, JScript);

Src — файл со скриптами, указывается в случае загрузки скриптов из внешнего файла.

2.2. Теги тела сообщения <BODY>

2.2.1. Управления разметкой

<H1> Заголовок </H1> — Определение заголовка с указанным уровнем от 1 до 6.

<P/> — Начало нового абзаца.

Align — определяет способ выравнивания (justify-по ширине, left, right, center).

<CENTER> Objects </CENTER> — Центрирование объектов внутри тега.

**
** — Разрыв строки (начать с новой строки).

Clear — остановить обтекание объекта в указанной точке (left, right, all).

<NOBR> Text </NOBR> — Формирование неразрывной строки.

2.2.2. Управления отображением

<BIG> Текст </BIG> — Текст со шрифтом более стандартного.

<SMALL> Текст </SMALL> — Текст со шрифтом менее стандартного.

^{Индекс} — Текст в нижнем индексе.

_{Индекс} — Текст в верхнем индексе.

** Текст ** — Текст с указанными параметрами шрифта.

Size — размер шрифта (-1,+1,3 ...);

Color — цвет шрифта.

<I> Текст </I> — Текст курсивом.

** Текст ** — Усиление текста.

<TT> Текст </TT> — Текст телетайпом.

<U> Текст </U> — Подчёркивание текста.

<S> Текст </S> — Перечёркивание текста.

** Текст ** — Типографски усиленный текст.

<CITE> Текст </CITE> — Цитата.

** Текст ** — Усиленный текст.

<CODE> Текст </CODE> — Исходные тексты кода (например, «коды программы»).

<SAMP> Текст </SAMP> — Литералы.

<KBD> Текст </KBD> — Клавиатурные символы.

<VAR> Текст </VAR> — Переменная.

<DFN> Текст </DFN> — Определение.

<Q> Текст </Q> — Текст в скобках.

<PRE> Текст </PRE> — Преформатированный вывод текста (как есть).

<BLOCKQUOTE> Текст </BLOCKQUOTE> — Формирование блоков цитат. Добавляются поля слева и справа от текста.

<HR/> — Горизонтальная линейка.

<FIELDSET> Текст </FIELDSET> — Бордюр вокруг текста или произвольного участка страницы.

<LEGEND>Заголовок</LEGEND> — Заголовок области.

2.2.3. Списки

<DL>...</DL> — Формирует список определений.

<DT> Термин </DT> — Термин определения.

<DD> Определение </DD> — Определение.

... — Ненумерованный список.

Type — устанавливает тип маркера (Disk, Circle, Square).

** Элемент ** — Описание элемента перечисления.

- Type* — устанавливает тип маркера для последующих элементов (Disk, Circle, Square).
... — Нумерованный список.
Type — устанавливает тип маркера (l,A,a,l).
** Элемент ** — Описание элемента перечисления.
Type — устанавливает тип маркера для последующих элементов (l,A,a,l).

2.2.4. Таблицы

- <TABLE> ... </TABLE>** — Описание таблицы.
Align — общее выравнивание таблицы;
Border — тип и размер границы таблицы;
Width — общая ширина таблицы (в пикселах или процентах – 50%, 300);
Height — общая высота таблицы;
CellPadding — ширина пустого пространства между содержимым ячейки и ее границами;
CellSpacing — ширина промежутков между ячейками;
BgColor — цвет фона таблицы;
BackGround — фоновое изображение для таблицы.
<CAPTION>Подпись</CAPTION> — Указание подписи таблицы.
Align — выравнивание подписи (top, bottom).
<TR> ... </TR> — Формирование строки.
Width — ширина строки;
BgColor — цвет фона строки;
Align — выравнивание по горизонтали (bleedleft, left, center, right);
Valign — выравнивание по вертикали (top, middle, bottom).
<TD Colspan="3">Текст</TD> — Описание ячейки строки.
Align — выравнивание по горизонтали (bleedleft, left, center, right);
Valign — выравнивание по вертикали (top, middle, bottom);
ColSpan — объединение указанного количества колонок в одну;
RowSpan — объединение указанного количества строк в одну;
Nowrap — не переносить содержимое ячейки на следующую строку;
BgColor — цвет фона ячейки;
BackGround — фоновое изображение ячейки;
Width — ширина ячейки.
<TH> Текст </TH> — Заголовок.
Align — выравнивание по горизонтали (bleedleft, left, center, right);
Valign — выравнивание по вертикали (top, middle, bottom);
Colspan — объединение указанного количества колонок в одну;
RowSpan — объединение указанного количества строк в одну;
Nowrap — не переносить содержимое ячейки на следующую строку;
BgColor — цвет фона ячейки;
BackGround — фоновое изображение ячейки;
Width — ширина ячейки.

2.2.5. Гипертекстовые ссылки

- Моя ссылка ** – Якорь. Формирование текстовой ссылки.
Href — содержит адрес гиперссылки.
Обычная гипертекстовая ссылка
Ссылка на точку «point» в документе
**** —
 Рисунок в качестве ссылки.
Ссылка на адрес электронной почты.
Target — имя фрейма на который помещать данную ссылку (Target="main");
Name — объявление точки для ссылки внутри документа (Name="point").

2.2.6. Изображения

**** — Формирование рисунка.

Src — адрес файла изображения;

Alt — текст выводимый браузером вместо изображения (изображения выключены);

Align — способ обтекания изображения (top, texttop, middle, absmiddle, baseline, bottom, absbottom, left, right, all);

HSpace — ширина горизонтальных полей отделяющих рисунок;

VSpace — ширина вертикальных полей отделяющих рисунок;

Border — ширина рамки вокруг изображения.

Width — ширина изображения;

Height — высота изображения;

Usemap — имя карты данного изображения **<IMG Src="mage.png" Alt="Изображения"**

Usemap="#map">, карта описывает активные участки изображения.

<MAP Name="map"> ... </MAP> — Описание карты изображения.

Name — имя карты;

<AREA Shape="rect" Coords="0,0,100,100" Href="http://mydomain.org"/> — Описание активной области и ссылки.

Shape — вид активной области (rect, circle, poly, default);

Coords — координаты активной области;

Href — адрес гиперссылки;

Nohref — без ссылки.

2.2.7. Формы

<FORM Method="post" Action="http://mydomen.org"> ... </FORM> — Формирование формы.

Method — метод уведомления WWW-сервера (Get, Post);

Action — адрес сервера обрабатывающего данные формы.

<INPUT Name="Name" Size="35"/> — Поле ввода строки.

Name — наименование поля ввода;

Type — тип поля ввода:

“text” — ввод текста;

“checkbox” — флажок выбора (*Checked* — выбран);

“radio” — флажок выбора одного из группы флажков;

“password” — поле ввода пароля (без отображения введенных символов);

“reset” — кнопка сброса формы (*Value* — надпись на кнопке);

“submit” — кнопка завершения ввода данных (*Value* — надпись на кнопке);

“hidden” — скрытое поле (для скрытия атрибутов *Name* и *Value* и использование их в управляющих целях);

“file” — поле ввода/выбора файла;

“button” — кнопка;

“image” — выполняет функцию завершения ввода по клику на изображении (*Src* — содержит адрес изображения).

MaxLength — максимальное число символов вводимое в поле (по умолчанию число символов не ограничено);

Size — размер поля ввода (символов);

Value — начальное значение поля ввода или имя для checkbox и radio, а также надпись на кнопках;

Src — содержит путь к файлу изображения для кнопок;

Align — выравнивание для (*Type="image"*);

Checked — флажок выбран (для *Type="checkbox"*);

Disabled — поле ввода неактивно/отключено.

<TEXTAREA Name="name" Cols="38" Rows="3">Текст</TEXTAREA> — Область ввода большого количества текста.

Name — наименование области ввода;

- Cols* — количество колонок ввода;
Rows — количество видимых строк текстового ввода.
<SELECT Name="myselect"> ... </SELECT> — Поле выбора значений (combobox).
Multiple — выбирать более одного наименования;
Name — наименование объекта;
Size — количество видимых строк списка (1 – как combobox).
<OPTION> Вариант 1 </OPTION> — Описание элемента списка.
Selected — элемент выбран;
Value — значение возвращаемое формой при выборе.

2.3. Фреймы

- <FRAMESET Cols="30%,*"> ... </FRAMESET>** — Формирование фреймовой структуры.
Cols — перечень (через “;”) колонок и их размеры (* – оставшееся пространство);
Rows — перечень (через “;”) строк и их размеры (* – оставшееся пространство).
<FRAMESET Cols="30%,*"> ... </FRAMESET> — Рекурсивное вложение фреймовых структур.
<FRAME Src="index.html" Name="Left" /> — Описание отдельно взятого фрейма.
Name — имя фрейма (по этому имени выполняются ссылки на фрейм);
Src — указывает на URL фрейма;
MarginWidth — горизонтальный отступ между содержимым кадра и его границами;
MarginHeight — вертикальный отступ между содержимым кадра и его границами;
Scrolling — наличие полосы прокрутки во фрейме (no,yes,auto);
NoResize — запретить изменение размера фрейма.
<NOFRAMES> Текст </NOFRAMES> — Содержит информацию для браузеров не поддерживающих фреймы.

3. Каскадные таблицы стилей (CSS)

Предназначены для расширения управляемости текстовой части документа, а также для выделения стиливаемой составляющей документа в отдельную часть, оставив языкам HTML/XHTML/XML декларативные функции в формировании документа.

Поддерживается несколько способов включения стилей в документ. Все они перечислены ниже в соответствии с иерархичностью:

- Загрузка из файла посредством тега `<LINK>` в заголовке `<HEAD>`:
`<LINK Rel="stylesheet" Href="./css/css.htm" Type="text/css" />`;
- Описание в теге `<STYLE>` заголовка документа `<HEAD>`:
`<STYLE>P { color:grey; text-align:left }</STYLE>`;
- Описание стиля непосредственно в теге назначения используя атрибут `Style`:
`<P Style="color:grey; text-align:left"></P>`.

Синтаксис описания стилей может быть двух типов:

`selector[, selector[, ...]] { attribute:value;[attribute:value;...] }` — перечисление селекторов, для которых действует данное описание стиля.

`selector selector [selector ...] { attribute:value;[attribute:value;...] }` — задание иерархии вложенности селекторов, для совокупности которых определен стиль.

При этом селектор `<selector>` может быть представлен следующими способами:

- Именем тега: `P { color:grey; text-align:left }`.
- Классом стиля: `.myclass { color:grey; text-align:left }`. Использование класса должно указываться в тегах с помощью атрибута `Class`: `<P Class="myclass"></P>`.
- Именем тега и классом стиля: `P.myclass { color:grey; text-align:left }`. Использование также должно указываться в тегах с помощью атрибута `Class`: `<P Class="myclass"></P>`.
- По идентификатору тега: `#myid { color:grey; text-align:left }`. Соответственно, у тега должен быть определен атрибут `Id`: `<P Id="myid"></P>`.
- По модификатору тега: `P:first-line { color:green }`.

Спецификация языков HTML/XHTML/XML содержит определение обобщающих элементов. Этими элементами являются блочный DIV и строковый SPAN. Все другие элементы текстовой разметки можно описать через DIV и SPAN используя CSS. Собственно, только в связке с CSS элементы DIV и SPAN имеют смысл.

В блочном элементе DIV можно манипулировать границами блока (*border*), отступами к ней от внешнего блока (*margin*) и от внутреннего (*padding*).

3.1. Атрибуты

3.1.1. Границы блока

- *border* — краткое описание параметров всей границы («1px solid blue»);
- *border-top* — краткое описание параметров границы сверху;
- *border-bottom* — краткое описание параметров границы снизу;
- *border-left* — краткое описание параметров границы слева;
- *border-right* — краткое описание параметров границы справа;
- *border-width* — общая ширина границы;
- *border-top-width* — ширина границы сверху;
- *border-bottom-width* — ширина границы снизу;
- *border-left-width* — ширина границы слева;
- *border-right-width* — ширина границы справа;
- *border-color* — общий цвет границы;
- *border-top-color* — цвет верхней границы;
- *border-bottom-color* — цвет нижней границы;

- *border-left-color* — цвет левой границы;
- *border-right-color* — цвет правой границы;
- *border-style* — общий стиль границы (none, dotted, dashed, solid, double, groove, ridge, inset, outset);
- *border-top-style* — стиль верхней границы;
- *border-bottom-style* — стиль нижней границы;
- *border-left-style* — стиль левой границы;
- *border-right-style* — стиль правой границы.

3.1.2. Отступы блока. Отступы от внешнего блока (margin) и от внутреннего, набивка (padding)

- *margin* — общий внешний отступ («5px» или перечисление: “5px 3px 5px 3px”);
- *margin-top* — верхний внешний отступ;
- *margin-bottom* — нижний внешний отступ;
- *margin-left* — левый внешний отступ;
- *margin-right* — правый внешний отступ;
- *padding* — общий размер набивки («4px» или перечисление: “4px 3px 2px 1px”);
- *padding-left* — размер набивки слева;
- *padding-right* — размер набивки справа;
- *padding-top* — размер набивки сверху;
- *padding-bottom* — размер набивки снизу.

3.1.3. Обтекание и выравнивание блока

- *float* — формирование плавающего блока и его расположение (left, right, both);
- *clear* — обтекание блока (right, left, none, both).

3.1.4. Цвет и шрифт в CSS

- *color* — цвет текста;
- *background* — краткое описание характеристик фона (background: blue http://mysite.org/bk.png no-repeat fixed center);
- *background-color* — цвет фона;
- *background-image* — адрес фонового изображения;
- *background-repeat* — ?
- *background-attachment* — ?
- *background-position* — ?
- *font* — краткое описание шрифта (font:bold 10pt verdana);
- *font-family* — семейство шрифта (verdana, arial, sans-serif);
- *font-style* — стиль шрифта (italic, normal);
- *font-weight* — усиление шрифта (normal, bold);
- *font-size* — размер шрифта, кегль (10pt, large, larger, small, smaller, medium, x-small, xx-small);
- *font-variant* — вариант начертания копитель (normal, small-caps).

3.1.5. Управление текстом

- *letter-spacing* — расстояние между буквами текста;
- *line-height* — расстояние между строками текста;
- *text-align* — выравнивание текста (justify, left, right, center);
- *text-transform* — преобразование текста (uppercase, lowercase, capitalize);
- *text-decoration* — оформление текста (none, underline, line-through);
- *text-indent* — отступ первой строки параграфа (абзац).

3.1.6. Управление видимостью. Слои. Положения и размеры блоков

- *position* — тип позиционирования:
 “absolute” — относительно левого верхнего угла документа;
 “relative” — относительно левого верхнего угла родительского или предыдущего блока.
- *top* — расстояние до блока от верха границы;
- *left* — расстояние до блока от левой границы;
- *height* — высота блока;
- *width* — ширина блока;
- *visibility* — видимость блока (hidden, visible);
- *z-index* — порядок перекрытия, большее значение атрибута определяет верхний блок;
- *clip* — выделение части блока (clip:rect(0,50,10,5)).

3.1.7. Другие атрибуты

- *display* — определяет видимость элемента (none);
- *list-style-type* — стиль значков списка (none, square, disk, circle, lower-roman, upper-alpha, lower-alpha);
- *white-space* — пустое пространство (nowrap).
- *font-face* — ?;
- *vertical-align* — вертикальное выравнивание (top).

3.2. Модификаторы тегов

Модификаторы используются для конкретизации группы свойств тега.

Текст :

- *:first-line* — модификатор первой строки параметра;
- *:first-letter* — модификатор первой буквы первой строки.

Гипертекстовые ссылки <a>:

- *:link* — ссылка по умолчанию;
- *:visited* — посещённая ссылка;
- *:hover* — под курсором;
- *:active* — активная.

4. JavaScript

JavaScript это язык созданный фирмой Netscape для программирования гипертекстовых страниц с исполнением программы на стороне браузера. Программирование заключается во внедрении текстов программ на языке JavaScript в тело гипертекстовых страниц и последующее исполнение этих программ браузером. Использование этого языка в создании сайтов позволяет сделать их более живыми и динамичными. Посредством языка JavaScript можно получить доступ не только к документу на котором программа размещена, но и к свойствам самого браузера. В последствии язык был стандартизирован ECMA (European Computer Manufacturers Association).

Язык обявывает объектную модель браузера и документов оперируя свойствами, методами и событиями объектов. Свойствами объекта являются атрибуты тегов страницы, а также атрибуты объектов браузера. Методами объектов выступают функции изменения свойств объектов документа и браузера. События объектов представлены функциями, которые активируются при выполнении того или иного действия над объектом. Все теги гипертекстовой страницы расширены атрибутами представляющими событийные функции. Для обработки событий код программы помещается в значение этих атрибутов.

В целом, код сценариев может размещаться на странице следующими способами:

- В гипертекстовой ссылки (URL-схема). Выполняется обработка программы вместо ссылки. Используется в атрибутах содержащих URL. Например:
`Message`
- В обработчике событий. Используется для размещения сценариев обработки событий. Например:
`<input type="button" value="Press" onClick="window.alert('Warning');"/>`
- Подстановка. Выполняется подстановка в атрибуты контейнеров документа результата исполнения скрипта. Например:
`<input type="text" value="{window.location.href}"/>`
- Вставка через контейнер *SCRIPT*. Посредством контейнера *SCRIPT* производится как загрузка файлов со сценариями так и непосредственное определение текста сценариев. Выполнение сценариев производится сразу, при обнаружении этого контейнера в процессе обработки документа. Данный способ можно использовать для размещения кода внутри документа и условной генерации документа на стороне браузера. Если указан исходный файл в атрибуте `<src>` и он присутствует, то тело контейнера *SCRIPT* не обрабатывается. Пример:

```
<HTML><HEAD>
  <SCRIPT type="text/javascript" src="http://myscript.js">
    //Мой сценарий
  </SCRIPT>
</HEAD></HTML>
```

4.1. Типы данных

Языком JavaScript поддерживается несколько простых типов:

- *Число*: любое целое или вещественное число (23, 05, 0x1A, 3.14159, -3.1E1).
- *Логический*: логический признак (true или false).
- *Строка*: строки и тексты («Строка»). В строке можно использовать специальные символы:
 - \b — забой;
 - \f — перевод страницы;
 - \n — новая строка;
 - \r — возврат каретки;
 - \t — табуляция;
 - \\ — двойной обратный слеш.
- *Ноль*: специальное ключевое слово для обозначения пустого значения (null).

Переменные в JavaScript определяются путём прямого присваивания в новую переменную ($x=3$) или посредством `var` (`var x=3`). Тип переменной меняется при присваивании значения другого типа. При смешанных операциях суммирования результатом является строка.

4.2. Операции

Символ	Описание
<i>Самый высокий приоритет</i>	
()	Вызов функции
[]	Выделение элемента массива
.	Выделение элемента структуры или объединения
++	Пост-приращение
—	Пост-декремент
+	Унарный плюс
—	Унарный минус
!	Логическое отрицание
~	Побитовое отрицание
*	Умножение
/	Деление
%	Остаток от деления
+	Сумма
—	Разность
<<	Сдвиг влево
>>	Сдвиг вправо
>>>	Сдвиг вправо, с заполнением нулями
<	Меньше
<=	Меньше и равно
>	Больше
>=	Больше и равно
==	Равно
!=	Неравно
&	Поразрядный «И»
^	Поразрядный «исключающий ИЛИ»
	Поразрядный «ИЛИ»
&&	Логический «И»
	Логический «ИЛИ»
?:	Условная операция(<code>var i=(val>=0)?val:-val;</code>)
=	Присваивание
+=, -=, *=, /=, %=, =, &=, <=, >=	Составное присваивание

4.3. Операторы

Управляющие операторы:

- `if...else` — Условные операторы.

Запись: `if(<condition>) <statements1>; [else <statements2>;]`

Действие: Если условие `<condition>` истинно, выполняется выражение `<statements1>`, иначе `<statements2>`. Условие может быть любым JavaScript-выражением, которое можно оценить как истинное или ложное. Выражения, которые нужно выполнить,

могут быть любыми JavaScript-операторами, включая условные. Более чем один оператор должен быть заключен в фигурные скобки, { }.

- *for*, *for...in*, *while*, *break* и *continue* — Операторы циклов. В теле цикла могут использоваться операторы `<break>` и `<continue>`. Первый для прерывания цикла, а второй для перехода к следующей итерации цикла.

Запись: `for (<initstatement>; <condition>; <poststatement>) <statement>;`

Действие: Выполнение выражения `<statement>` пока условие `<condition>` истинно. Перед запуском цикла выполняется выражение `<initstatement>`. После каждой итерации цикла выполняется выражение `<poststatement>`.

Запись: `while(<condition>) <statement>;`

Действие: Выполнение выражения `<statement>` пока условие `<condition>` истинно.

Запись: `for(<attr> in <obj>) <statement>;`

Действие: Выполнение выражения `<statement>` для всех атрибутов `<attr>` объекта `<obj>`.

- *new*, *this* и *with* — Операторы манипуляции с объектами.
- `// <coment>` и `/* <coment> */` — Комментарии. Первый это одно-строчный комментарий, а второй много-строчный.

Специальные операторы:

- *new {Type}* — Создание экземпляра объекта пользовательского или встроенного типа (`name = new typ(param1 [,param2] ...[,paramN])`).
- *typeof {Operand}* — Определение типа операнда. Возвращает тип операнда в виде строки (`typeof val`).
- *void (expression)* — Определяет выражение, которое не возвращает никакого значения (`void (0)`).

4.4. Функции

Функции – один из фундаментальных блоков JavaScript. Функция – JavaScript-процедура – набор операторов, который исполняет определенную задачу. Определяется и используется функция следующим образом:

```
function parWrite(str)
{
    document.write("<HR><P>" + str)
}
parWrite(2345);
```

Языком предусмотрена поддержка переменного числа параметров функций. Для доступа к произвольному параметру предусмотрен запрос: `<functionName>.arguments[i]`. Для определения числа доступных атрибутов используется запрос: `<functionName>.arguments.length`. Пример функции с переменным числом параметров и её вызов приведен ниже:

```
function list(type)
{
    document.write("<" + type + "L>")
    for( var i = 1; i < list.arguments.length; i++ )
        document.write("<LI>" + list.arguments[i]+"</LI>")
    document.write("</" + type + "L>")
}
list("O", "one", 1967, "three", "etc., etc...");
```

4.5. События и их обработка

Управление элементами документа пользователем передаются в программу посредством событий. Стандартами управления браузером предусмотрена группа событий для различных элементов, которые перечислены в таблице ниже.

Событие/обработчик	Область использования	Причина возникновения	Метод имитации
onAbort	images	Пользователь прерывает загрузку изображения.	
onBlur	windows, frames и все формы элементов	Пользователь убирает фокус ввода.	blur()
onChange	текстовые поля, текстовые области	Пользователь меняет значение элемента.	change()
onClick	button, radio button, checkboxes, submit button, reset button, links	Пользователь выбирает форму или ссылку.	click()
onDbClick	button, radio button, checkboxes, submit button, reset button, links	Двойной щелчок	
onError	images, windows	Ошибка загрузки документа или изображения.	
onFocus	windows, frames, и все формы элементов.	Пользователь передает фокус ввода.	focus()
onKeyDown	button, radio button, checkboxes, submit button, reset button, links	Нажата клавиша на клавиатуре	
onKeyPress	button, radio button, checkboxes, submit button, reset button, links	Нажата и отпущена клавиша на клавиатуре	
onKeyUp	button, radio button, checkboxes, submit button, reset button, links	Отпущена клавиша на клавиатуре	
onLoad	тело документа	Пользователь загружает страницу в браузер.	
onMouseDown	button, radio button, checkboxes, submit button, reset button, links	Нажата кнопка мыши в пределах текущего элемента.	
onMouseMove	button, radio button, checkboxes, submit button, reset button, links	Перемещение курсора мыши в пределах текущего элемента.	
onMouseOut	button, radio button, checkboxes, submit button, reset button, links	Пользователь перемещает указатель мыши по области.	
onMouseOver	button, radio button, checkboxes, submit button, reset button, links	Курсор мыши наведен на текущий элемент.	
onMouseUp	button, radio button, checkboxes, submit button, reset button, links	Отпущена кнопка мыши в пределах текущего элемента .	
onMove	window	Перемещение окна.	
onReset	form	Сброс данных формы (щелчок по кнопке <code><input type="reset"></code>).	reset()
onResize	window	Изменение размеров окна.	
onSelect	input, textarea	Выделение текста в текущем элементе.	
onSubmit	form	Отправка данных формы (щелчок по кнопке <code><input type="submit"></code>).	submit()
onUnload	body, frameset	Попытка закрытия окна браузера и выгрузки документа	

4.6. Объекты

В JavaScript объекты представлены объектной моделью браузера и встроенными не визуальными объектами. Кроме того, пользователем могут создаваться собственные объекты. Объект может содержать свойства и функции.

К свойствам объекта можно получить доступ двумя способами. Первый способ это иерархическое обращение (`<object>.<attr>`). Второй способ это доступ к объекту как к массиву (`<object>[<attr>]`). При доступе к элементу объекта как к массиву можно обращаться как по имени, так и по индексу.

Для адресации к текущему объекту используется ключевое слово `<this>`. Например при прямом описании обработчиков событий по текущему объекту:

```
<INPUT TYPE = "text" NAME = "age" SIZE = 3 onChange="validate(this, 18, 99)"/>
```

Для обращения к элементам родительского объекта можно использовать атрибут "parent" (`parent.color`).

Определение нового объекта производится посредством ключевого слова `<function>`, а создание экземпляра объекта посредством `<new>`. К экземпляру объекта можно свободно добавлять новые свойства, путём простого присваивания. Например:

```
function person(name,age,sex)
{
    this.name = name;
    this.age = age;
    this.sex = sex;
}

function setUser(user)
{
    this.user = user;
}

function book(name, author, year, publisher, person)
{
    this.name = name;
    this.author = author;
    this.year = year;
    this.publisher= publisher;
    this.user = person;
    this.setUser = setUser; //Подключение к объекту метода setUser()
}

pers1 = new person("Vasja",24,"M");
book1 = new book("JavaScript","Flenagan",2006,"O'REILLY",pers1);

userNm=book1.user.name;book.prototype.user=null; //Установка свойства в экземплярах объекта book
book1.setUser("Vasja");
```

4.6.1. Встроенные типы объектов

1. *Array* — Объект массива. Массив – упорядоченный набор значений, к которым можно обратиться по имени и индексу. Экземпляр массива можно создать двумя способами, путём указания количества элементов в массиве и прямого указания элементов:

```
aObjNm = new Array(<arrayLength>);
aObjNm = new Array(<element0>, <element1>, ..., <elementn>);
a = new Array(4);
for (i=0; i < 4; i++)
{
    a[i] = new Array(4);
    for(j=0; j < 4; j++) a[i][j] = "["+i+","+j+"]";
}
```

Свойства:

- *length* — размер массива;

Методы:

- *join()* — связывание всех элементов массива в строку, через знак запятой;
- *reverse()* — реверс порядка элементов в массиве;
- *sort()* — сортировка элементов массива в лексикографическом порядке;
- *concat()* — добавление к массиву значений;
- *splice(<beg>,<end>)* — получение части массива начиная с `<beg>` и заканчивая

<end>;

- *push()*, *pop()* — работа с массивом как со стеком;

2. *Boolean* — Данный объект используется когда необходимо преобразовать не-boolean значение в boolean.

```
bfalse = new Boolean(false);
```

Методы:

- *valueOf()* — Значение объекта.

3. *Date* — Дата. Хранит значение даты в миллисекундах с 1 января 1970г, 00:00:00.

```
today = new Date(); //Текущее время
tmFromStr = new Date("November 3, 2007 13:54:00"); //Из строки
tmSet = new Date(2007,11,3,13,54,00); //Прямая установка
```

Методы:

- *<mtm> getTime()* — получение времени в миллисекундах с 01.01.1970;
- *setTime(<mtm>)* — установка времени в миллисекундах с 01.01.1970;
- *<dt_part> get[Year]|[Month]|[Day]|[Days]|[Hours]|[Minutes]|[Seconds]()* — получения года, месяца, дня недели, дня месяца, часов, минут и секунд соответственно;
- *set[Year]|[Month]|[Days]|[Hours]|[Minutes]|[Seconds](<dt_part>)* — установка года, месяца, дня месяца, часов, минут и секунд соответственно;
- *<mtm> parse(dtStr) static* — получение даты путём разбора строки вида «November 3, 2007 13:54:00»;

4. *Function* — Объект функции. Используется для закрепления за переменной или свойством функции какой либо процедуры. Формат определения экземпляра объекта функции: **functionObjectName = new Function ([<arg1>, <arg2>, ... <argn>,) *functionBody***).

Пример использования:

```
var setBGColor = new Function("document.bgColor='antiquewhite'");
document.form1.colorButton.onclick=setBGColor;
<INPUT NAME="colorButton" TYPE="button" VALUE="Change background color" onClick="setBGColor()"/>
```

5. *Math* — Встроенный объект доступа к математическим функциям и константам. Все его свойства и методы являются статическими, а значит доступны без создания экземпляра объекта и посредством имени объекта (Math.PI).

Свойства:

- *PI* — константа π (3.14159276);

Методы:

- *abs()* — абсолютное значение;
- *sin()* — синус;
- *cos()* — косинус;
- *tan()* — тангенс;
- *acos()* — арккосинус;
- *asin()* — арксинус;
- *atan()* — арктангенс;
- *exp()* — экспонента;
- *log()* — натуральный логарифм;
- *ceil()* — возвращает наименьшее целое, большее или равное аргументу;
- *floor()* — возвращает наибольшее целое, меньшее или равное аргументу;
- *min()*, *max()* — возвращает меньшее или большее (соответственно) из двух аргументов;
- *pow()* — возведение в степень, первый аргумент основание, второй показатель степени;
- *round()* — округляет аргумент до ближайшего целого;
- *sqrt()* — квадратный корень.

6. *Number* — Содержит статические свойства цифровых констант.

Свойства:

- *MAX_VALUE* — наибольшее значение;
- *MIN_VALUE* — наименьшее значение;
- *NaN* — неопределенность;
- *NEGATIVE_INFINITY* — отрицательная бесконечность;

- *POSITIVE_INFINITY* — положительная бесконечность.

Методы:

- *toFixed(<num>)* — Округление числа до указанного знака после запятой.
- *toString(<base>)* — Преобразование в строку с указанной базой (2–36).

7. *String* — Объект для работы со сроками в приложениях. Фактически данный объект представляет строковые литералы и переменные строкового типа, а значит его свойства и методы характерны для них.

Свойства:

- *length* — длина строки;

Методы:

- *<str> anchor()* — создает HTML-якорь;
- *<str> big() | blink() | bold() | fixed() | italics() | small() | strike() | sub() | sup()* — создает строку HTML, соответствующего вида;
- *<el> charAt(<simb>)* — возвращает символ по его номеру в строке;
- *<pos> indexOf(<substr>) | lastIndexOf(<substr>)* — возвращают первую (последнюю) позицию специфицированной подстроки;
- *<str> link(<addr>)* — создает гипертекстовую ссылку для адреса *<addr>*;
- *<array> split(<sep>)* — разбивает объект *String* на массив подстрок по разделителю *<sep>*;
- *<str> substring(<pos>, <len>)* — извлекает подстроку из строки начиная с позиции *<pos>* и длиной *<len>*;
- *<str> toLowerCase() | toUpperCase()* — возвращает строку в нижнем или верхнем регистре, соответственно.

4.6.2. Глобальные свойства и методы

- *self, window* — ссылка на объект текущего окна/фрейма;
- *top* — ссылка на объект самого верхнего окна;
- *parent* — ссылка на родительский объект;
- *{winName}* — глобальное имя окна, используемое для прямой адресации к нему;
- *<bool> isNaN(<val>)* — оценка аргумента на неопределенность;
- *<str> toString(<obj>)* — преобразование объекта в строку;
- *<tid> setTimeout(<call_func>, <tm>)* — установка интервала *<tm>*, в миллисекундах, по истечению которого вызвать функцию *<call_func>*;
- *clearTimeout(<tid>)* — останов ранее созданного интервального вызова с идентификатором *<tid>*;
- *<numb> parseInt(<str>[, <radix>])* — преобразование строки *<str>* к целому с основанием *<radix>*;
- *<numb> parseFloat(<str>)* — преобразование строки к вещественному;
- *<rez> eval(<str>)* — вычисление выражения в *str* и возврат результата в *<rez>*.

4.6.3. Иерархия объектов браузера

window — корневой объект окна;

- *status* — содержимое и доступ к строке статуса;
- *alert(<string>)* — сформировать диалоговое окно с предупреждением;
- *confirm(<string>)* — сформировать диалоговое окно подтверждения с кнопками: “Ok” и “Cancel”;
- *prompt(<string>)* — сформировать диалоговое окно с полем ввода значения;
- *<wid> open(<url>, <title>, <add_params>)* — открытие нового окна для *<url>* с названием *<title>* и группой дополнительных параметров *<add_params>* (*width=400,height=150*).
Дополнительные параметры: *width, height, status=no, menu, toolbar=yes*;
- *close()* — закрытие окна;
- *focus()* — установка фокуса на данное окно;
- *blur()* — снятие фокуса с данного окна;

- *scroll()* — прокрутка окна к указанной позиции;

window.location — URL страницы данного окна, может быть изменен для перехода на другую страницу;

- *href* — полный текст URL;
- *protocol* — протокол URL;
- *host* — адрес сервера;
- *pathname* — адрес к ресурсу;
- *search* — атрибуты запроса.

window.navigator — объект браузера;

window.navigator.plugins[] — программы браузера;

window.navigator.mimeTypes[] — поддерживаемые MIME-типы;

window.frames[] — фреймы окна и сами окна;

- *location* — URL фрейма, может быть изменен для отображения другой страницы;

window.history — история визитов;

window.document — документ;

- *bgColor*, *fgColor*, *linkColor*, *alinkColor*, и *vlinkColor* — свойства глобальных цветов документа;
- *lastModified* — дата последней модификации документа;
- *referrer* — предыдущий почтаемый URL;
- *URL* — URL документа;
- *open()* — открытие документа окна для записи в него;
- *close()* — закрытие документа на запись;
- *write(<str_el1>[,<str_el2>[,<str_el3>, ..., <str_eln>]])* — запись строки, состоящей из элементов *<str_el{x}>* в конец документа, используется для динамического формирования документа;
- *writeln(<str_el1>[,<str_el2>[,<str_el3>, ..., <str_eln>]])* — выполняет ту же функцию, что и функция *<write>*, а также добавляет перевод строки в конце строки;

window.document.forms[] — формы документа;

- *document* — ссылка на объект документа-владельца.

window.document.forms[].elements[] — элементы формы;

- *form* — ссылка на объект формы-владельца.

window.document.anchors[] — якоря (ссылки) документа;

window.document.links[] — ссылки документа;

- *href* — адрес ссылки;

window.document.images[] — изображения документа;

window.document.applets[] — апплеты документа;

window.document.embeds[] — ???

window.packages[] — пакеты.

window.event — глобальный объект событий Event

- *type* — тип события (keydown, ;
- *target* — узел – источник события;
- *currentTarget* — узел – обработчик события;
- *timeStamp* — время возникновения события (объект Date);
- *button* — номер кнопки мыши (0-левая, 1-средняя, 2-правая);
- *altKey*, *ctrlKey*, *metaKey*, *shiftKey* — признаки удержания клавиши модификатора в момент клика мышью;
- *clientX*, *clientY* — положение курсора мыши относительно клиентской области;
- *pageX*, *pageY* — положение курсора мыши относительно левого верхнего угла окна (NN);
- *screenX*, *screenY* — положение курсора мыши относительно экрана;
- *offsetX*, *offsetY* — положение курсора мыши относительно исходного элемента (IE);
- *keyCode* — код нажатой клавиши (IE);
- *which* — кнопка мыши или код нажатой клавиши (NN).

5. Приложение

5.1. Структура URL (Unifical resource locator)

В общем виде Unifical resource locator(URL) записывается следующим образом:

http://user:password@domain.ru:port/path/some.html?query_string

где:

http — протокол обмена данными (Hyper Text Transfer Protocol);

user — идентификатор пользователя;

password — пароль;

domain.ru — доменное имя сервера;

port — номер TCP-порта сервера;

path — путь в корневом каталоге сервера к ресурсу;

some.html — ресурс;

query_string — параметр запроса.

5.2. CER (Character Entity Reference)

CER (Character Entity Reference) – соглашение о кодировании и представлении специальных символов:

Числовой код	Именная замена	Символ	Описание
"	"	"	Кавычка
&	&	&	Амперсант
<	<	<	Меньше
>	>	>	Больше
 	 		Неразрывный пробел
¡	¡	¡	Перевернутый восклицательный знак
¢	¢	¢	Цент
£	£	£	Фунт
¤	¤	¤	Валюта
¥	¥	¥	Йена
¨	¨	¨	Умляут
©	©	©	Копирайт
«	«	«	Левая угловая кавычка
®	®	®	Зарегистрированная торговая марка
±	±	±	Плюс или минус
»	»	»	Правая угловая кавычка

5.3. Цвета и их коды

Цвет	Код
aqua	#00FFFF
navy	#000080
black	#000000
olive	#808000
blue	#0000FF
purple	#800080
fuchsia	#FF00FF
red	#FF0000

Цвет	Код
gray	#808080
silver	#C0C0C0
green	#008000
teal	#008080
lime	#00FF00
white	#FFFFFF
maroon	#800000
yellow	#FFFF00

5.4. Единицы размеров

Размеры в документе HTML/ХHTML/XML можно указывать несколькими способом.

Единица	Описание
100px	Пиксели (100 пикселов)
50pt	Типографические пункты (50 пунктов, 1pt = 0,35mm)
10em	Условные единицы (10 условных единиц)
100%	В процентах от опорной величины

5.5. Шрифты

Семейства шрифтов:

Шрифт	Описание
serif	Шрифт с засечками (Times)
sans-serif	Шрифт без засечек (Arial, Helvetica)
monospace	Моно-ширинный (Courier)

6. Литература

1. Б.П. Храмцов, С.А.Брик, А.М.Русак, А.И.Сурин; «Основы WEB-технологий.»; Интернет-Университет; Москва; 2003
2. Молли Э. Хольцшлаг; «Использование HTML и XHTML. Специальный справочник.»; Вильямс; Москва, Санкт-Петербург, Киев: 2003