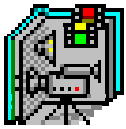




Мултимедийни документи в Internet

- [Езикът HTML](#)
- [Елементи на езика HTML](#)
- [Резервирани символи в HTML](#)
- [Структура на HTML документ](#)
 - [Описателна част](#)
 - [Изпълнима част](#)
 - [Имена на цветове и RGB стойности](#)
- [HTML елементи в секция HEAD](#)
- [HTML елементи в секция BODY](#)
- [Различни нива на заглавия \(heading commands\)](#)
- [Управление на текстовия поток \(text-flow commands\)](#)
 - [Форматиране на параграф - <P>](#)
 - [Преминаване на нов ред -
](#)
 - [Преформатиране на текст - <PRE>](#)
 - [Хоризонтална линия - <HR>](#)
 - [Коментар - <!-->](#)
 - [Пример за управление на текстов поток](#)
 - [Структуриране на документ <DIV>](#)
- [Форматиране на символи \(character-formatting commands\)](#)
 - [Физически атрибути](#)
 - [Пример за форматиране на символи](#)
 - [Логически атрибути](#)
- [Подредени и неподредени списъци \(list commands\)](#)
 - [Неподреден списък -](#)
 - [Подреден списък -](#)
 - [Списък от дефиниции - <DL> <DT> <DD> </DL>](#)
- [Създаване на хипервръзки \(anchor & link commands\)](#)
- [Компютърна графика \(Computer graphics\)](#)
- [Включване на графика \(in-line graphic commands\)](#)
- [Графики с горещи точки \(Hotspot Images\)](#)
- [Възпроизвеждане на звук \(Playing Audio\)](#)
- [Създаване на таблици \(Table commands\)](#)
- [Включване на формуляри \(Form commands\)](#)
- [Създаване на формуляри](#)
 - [Механизми за обработка на формуляри](#)
 - [Описание на формуляри](#)
- [Структуриране на секции \(Frame commands\)](#)
- [Начална страница \(home page\)](#)
- [Редактори и филтри за HTML](#)
- [Динамични прозорци](#)
- [Независима каскада от стилови формати в HTML 4.0 \(Cascading Style Sheets Level 1\)](#)
- [Каскада от стилови формати за позициониране на HTML елементи \(Cascading Style Sheets-Positioning - CSS-P\)](#)
- [Динамични шрифтове - OpenType & TrueDoc](#)
- [Включване на Java аплети](#)

- [Указател на HTML елементи](#)



МУЛТИМЕДИЙНИ ДОКУМЕНТИ В INTERNET

Мултимедийните документи представляват структура данни, включваща хипертекст, генерирани изображения, изображения от реалния свят, говор, цифрова информация с аудио и видео характер, от една страна и от друга, методи за представянето им (визуализацията им) на различни възпроизвеждащи нива.

Съществуват следните основни стандарти за преносимо структурно представяне на мултимедийни документи:

- **SGML** - *Standard Generalized Markup Language*;
- **DTD** - *Document Type Definition*;
- **HTML** - *Hypertext Markup Language*;
- **XML** - *Extensible Markup Language*;
- **SMIL** - *Synchronized Multimedia Integration Language*
- **HyTime** - *Hypermedia/Time-based Document Structuring Language*;
- **MHEG** - *Multimedia and Hypermedia Information Coding Experts Group*;
- **PDF** - *Portable Document Format* (PDF file - Adobe);
- **MIME** - *Multipurpose Internet Mail Extensions*;
- **AVI** - *Audio-Visual Interactive Services*.

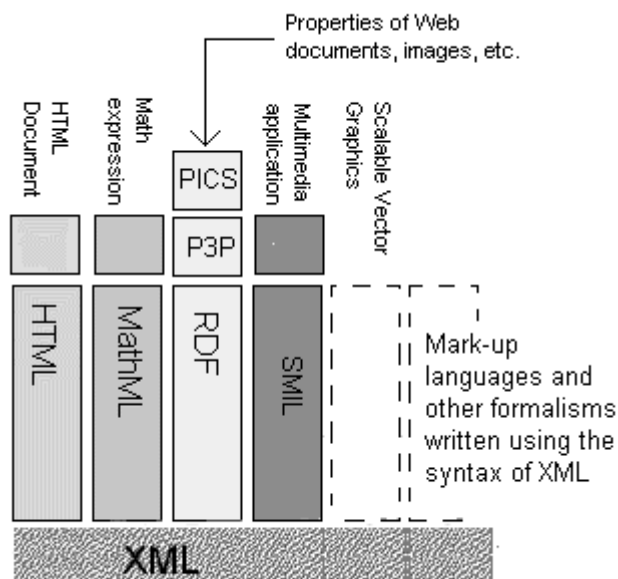
SGML е метаезик, който чрез маркиране разделя логическото и семантично съдържание в даден документ. На този метаезик се подготвят дефиниции на типове документи **DTD** (*Document Type Definitions*), описващи структурните елементи, които могат да се съдържат в мултимедийни документ и правилата за прилагането им.

Сходен на SGML е метаезикът **HTML**, но приликата се изразява само във възможността за използване на дефиниции на типове документи (DTD). В действителност HTML не използва пълните възможности на SGML, и благодарение на това създаването на HTML документи не изисква мощните средства за редактиране, необходими за работа с SGML. Изготвянето на HTML документи може да става на персонален компютър, на който има специализиран HTML редактор или обикновен текстов редактор.

Ориентацията към HTML се обяснява с факта, че той е използваем върху различни платформи в хетерогенни компютърни мрежи (посредством протокола **HTTP**). Процесът на създаване на документи на HTML е подобен на авторския процес при използване на системи за текстообработка, т.е. HTML документи се създават лесно и бързо.

HTML поддържа парадигмата за създаване на мултимедийни документи с хипервръзки. Това означава, че авторът може да включва в създаваните от него документи връзки към други подобни на неговия документ. Връзките се основават на т.нар. адреси **URL** (*Uniform Resource Locators*), като всеки такъв адрес специфицира уникален документ в цялото множество от документи. Документите са достъпни чрез връзката клиент-сървър. Понастоящем съществуват повече от 2000 WWW сървъри в Internet, които по заявка изпращат документи към съответните клиенти.

XML е нов (февруари 1998) метаезик подмножество на **SGML** и осигурява преносимост на SGML документи чрез Web с лекотата, простотата и удобството на езика HTML. Новата формула има за цел да стандартизира усилията в насока на по-комплексно и унифицирано представяне на документи чрез глобалната мрежа (*Фигура 1.0*).



Фигура 1.0 - Архитектура на езика XML

В структурно отношение езикът **XML** е съставен от три части:

- Дефиниции на типове от вида **DTD**;
- **XSL** - *Extensible Style Language* като средство за описание на стилови формати в **XML** документи;
- **XLL** - *Extensible Link Language* като средство за описание на хипервръзки в **XML** документи.

Появата на **XML** с принципно по-големи възможности не го прави директен конкурент на **HTML**. Общото между двата езика е твърде много и пълна съвместимост се постига лесно на ниво Web браузър. Предполага се, че сверите на влияние ще се разделят в зависимост от обема, важността и трайността на документите. За малки по обем и по-скоро с рекламно съдържание и кратък период на съществуване **HTML**, в чистия му вид, ще бъде предпочитан. За представителна, динамично изменяща се, но неизменна по своята важност и актуалност във времето информация се предвижда използването на **XML**.

Пълно описание на езика **XML** може да се намери на сайта на консорциума **W3C**:

<http://www.w3.org/XML/>

Предписанията на нормата, публикувана от **W3C** вече се прилага от Web браузърите *Netscape 4.7* и *Internet Explorer 5.0*.

SMIL е метаезик, дефиниран от норма на консорциума W3C и има за цел синхронизация във времето на мултимедийни елементи. Основата на езика е нормата на **XML** и дава възможност за вмъкване в рамките на един документ на различни мултимедийни

компоненти. Езикът осигурява смесване на различни типове като текст, графика, видео, звук, векторна анимация и тяхната синхронизация във времето. Файлове с **SMIL** инструкции са написани на **XML** и имат разширение **.smi**.

ЕЗИКЪТ HTML World Wide Web

Езикът HTML се използва в WWW от 1990 година. Той е пряк наследник на езика SGML. Езиковите средства са обвързани с определен начин на записване на текст с помощта на набор от стилови формати. Записът се интерпретира от WWW клиент (*Web Browser*) във вида създаден от автора.

В този текст се въвеждат основите на базирания на маркиране език HTML и се дават необходимите знания за създаване на собствени WWW страници. **HTML+** представлява разширение на HTML, с което езикът е обогатен с някои допълнителни възможности. В текста базовите елементи на **HTML** не се разграничават от допълненията на **HTML+**, т.е. в този текст под HTML трябва да се разбират HTML и HTML+. Средата за интерпретиране и визуализация на HTML документ тук се нарича **WWW клиент** (*Web Browser*).

HTML е система от прости команди, които се включват в стандартните текстови документи с цел да се осигури интегрирана визуализация. Всеки документ, създаден с произволна система за текстообработка, и съхранен в ASCII формат, може да се превърне в HTML страница чрез прибавяне на HTML команди. Основните категории команди, включени в версиите на езика (*HTML 3.2 /4.01*) са предназначени за:

- специфициране на стилови формати и управление на текстовия поток (*Flow Control*);
- включване на графични изображения (*Images*);
- създаване на хипервръзки (*links*);
- интегриране на аудио с външни графични обекти (*Sound and Maps*);
- създаване на интерактивни формуляри (*Forms*);
- разделяне на документа на отделни полета (*frames*);
- включване на външни приложения, написани на езика Java (*Applet*);
- осъществяване на връзка с външни информационни структури (*CGI-script*).

Четвъртото поколение Web клиенти като *Netscape Communicator 4.7* и *Internet Explorer 5.0* възприемат вече някои от новите елементи на Web технологиите при изобразяване и контрол на съдържанието на Web страниците. Новите технологични решения се събират под името **DHTML** (*Dynamic HTML*) и под формата на ново разширение на възможностите на този език. Част от тези решения бяха осъществими и досега чрез използване на допълнителни и външни програмни решения като *CGI* или *Java Applets*. Основните компоненти на тези нови технологични решения се подчиняват на общата формулата: **DHTML = HTML + CSS + scripting + DOM** и включват:

- **Cascading Style Sheets (CSS1)** - каскада от стилови формати чрез които се определят параметрите на всеки елемент при неговото изобразяване в полето на документа. Всеки HTML елемент може да унаследи или директно да придобие определен брой стилови параметри;
- **Cascading Style Sheets-Positioning (CSS-P)** - допълнение към каскадата от стилови формати, имащо за цел да се улесни максимално позиционирането на различните HTML елементи в полето на документа;
- **Document Object Model (DOM)** - обектно-ориентиран модел на документа е една тенденция която се маркира с въвеждането на обектно-скриптови езици като

Javascript. Стандартизацията на **DOM** включва и [второ ниво](#) от декември 1998г. и се очаква в следващите версии на Web клиентите;

- **Client-side scripting** - появата на обектните скриптови езици като **Javascript** формират базата за създаване на интерактивни страници при пълен контрол върху изобразяваните елементи в полето на документа;
- **Downloadable Fonts** - възможността да бъде зареден дистанционно необходимия шрифт за изобразяване на съдържанието на един HTML документ предполага разрешаването на множество неудобства и несъответствия между авторския стил на поднасяне на информацията и текущо установения шрифт при Web клиента.

Езикът **HTML** еволюира непрекъснато и след първоначалното си създаване с прости, предимно текстови компоненти, понастоящем предлага многофункционални възможности. Основна грижа за развитие и стандартизиране на езика има една работната група известна под името **W3C**. Техните отчети се публикуват на следния адрес:

<http://www.w3.org/>

Пълната спецификация на езика HTML версия 3.2 е публикувана от работната група **W3C** и може да се намери на адрес:

<http://www.w3.org/TR/REC-html32>

Предстояща е стандартизацията на HTML версия 4.01. Актуално описание на работната спецификация (*W3C Working Draft*) се намира на адрес:

<http://www.w3.org/TR/WD-html40-970708/>

Локално достъпното описание на работната спецификация на [HTML4.01](#) е от 24 април 1998.

Елементи на езика HTML

Под елемент се разбира команда на HTML. Елементът дава възможност да се промени начина, по който изглежда даден текстов документ, когато той се интерпретира от WWW клиент (*Web Browser*). HTML елементите осигуряват инструкции за WWW клиента, отнасящи се до форматирането и включването на външни елементи като графика и звук.

Даден HTML елемент винаги представлява дума или фраза, заградена между символите за по-малко и по-голямо ("**<**", "**>**"), например:

<P>

WWW клиентът игнорира всеки елемент, който не съдържа валидна команда. Това обстоятелство е важно и осигурява преносимост на HTML документите, като се интерпретират и от най-простите WWW клиенти чрез пропускане на командите, които не могат да обработват за сметка на влошено качество на визуализацията.

Съществуват два основни типа HTML елементи - *непразни* и *празни*. За да се обяснят тези два механизма, е необходимо да се покаже как даден HTML елемент се използва за модифициране на текст.

За модифициране на даден блок от текст трябва да се специфицира *НАЧАЛО* и *КРАЙ* на блока. Това става, като един HTML елемент се поставя в началото на блока, а друг - в края му. Текстът между двата елемента подлежи на обработка. Това е пример за *непразен* елемент.

Празните елементи не се нуждаят от краен елемент за завършване на командата. Например при вмъкване на графика в текстов документ, е достатъчно да се специфицира само началния.

Съществуват някои елементи, които допускат да се разглеждат като *празни*, и като *непразни*. Например елементът, който задава началото или края на параграф може да бъде *празен* елемент. Достатъчно е да се зададе само началото на командата. HTML елементът е логически блок и може да се разглежда като *непразен*. За да се реши този проблем новите версии на HTML поддържат както празни, така и непразни версии на някои команди (в такива случаи ще бъде използван по-простият вариант, т.е. празните версии). От чисто практическа гледна точка се препоръчва при оформяне на параграфи да се спазват правилата за непразните елементи.

Синтаксисът на крайния елемент при непразните елементи е същият като на началния. Разликата между двата елемента е, че крайният започва със символа наклонена черта ("*/*" - *slash*). Например, един параграф се оформя по следния начин:

```
<p>Пример</p>
```

Елементите могат да се задават с малки или с големи букви или с малки и големи заедно.

Много от HTML елементите съдържат параметри, които позволяват да се опише функцията на съответната команда. Например елемент, който специфицира, че е необходимо включването на графично изображение в даден документ (*IMG*), трябва да съдържа името на файла с изображението (*SRC=filename*). Последното представлява *параметър* или *аргумент*. Например, изобразяване на съдържанието на графичния файл "*globe.gif*" в полето на документа с подравняване на графиката към базовата линия на текста се извършва със следната команда:

```
<IMG SRC="globe.gif" ALIGN=baseline>
```

ПОДРЕЖДАМ

Параметрите винаги се изброяват след името на елемента, като редът на изброяване при команди с повече от един параметър е произволен. Данните (имена на файлове), включени в параметъра, се заграждат в кавички. В повечето случаи WWW клиентът приема данните, независимо от това дали са поставени в кавички или не. Използване на кавички се налага, когато в данните са включени специални символи или празни позиции. Общото правило е данните на параметрите да се поставят в кавички, за да се осигури съвместимост с всички Web клиенти.

Текстовите документи могат да съдържат допълнителни символи, например "CR" (преминаване на нов ред) или празни позиции, добавени с форматираща цел. WWW клиентът

игнорира допълнителните "CR" и последователните празни позиции, така че те не се появяват при визуализация на документа. В някои случаи това може да причини неудобства, свързани с трудност при постигане на специфично форматиране. Правилото за интерпретиране на празни позиции и "CR" е следното. Повече от една последователни празни позиции се интерпретират като една. Управляващите символи "CR" се отстраняват. С цел да се преодолеят ограниченията, наложени от автоматичното форматиране, HTML осигурява няколко специални елемента за това, например:

- ** ** е команда за поставяне на твърда шпация;
- **
** е елемент за преминаване на нов ред в рамките на единен параграф (мек "CR");
- **<PRE>** е елемент, с помощта на който се запазва предварително създадено форматиране на определен текст (например, при извеждане в полето на документа на текстове на програми).

Резервирани символи в HTML

Резервирани символи в HTML са "<", ">", "&". Символът "&" се използва за извеждане на символи, които в общия случай не са директно достъпни от клавиатурата, включително и непечатуеми такива ([Таблица 2](#)). Използването на резервираните символи в даден документ като текстови компоненти, а не в качеството им на HTML команди става по следния начин ([Таблица 1](#)):

Таблица 1

Код	Заместване
&lt;	Изобразява символа <
&gt;	Изобразява символа >
&amp;	Изобразява символа &
&nbsp;	Изобразява шпация ' '
&quot;	Изобразява символа "
&cent; &pound; &yen;	Изобразява съответно символите: ¢ £ ¥
&copy; &reg;	Изобразява съответно символите: © ®
&deg; &divide; &pi;	Изобразява съответно символите: ° ÷ [Pi]
&#171;	Изобразява символа «
&#187;	Изобразява символа »
&#177;	Изобразява символа ±
&#176;	Изобразява символа °
&#185; &#178; &#179;	Изобразява съответно символите: 1 2 3
&#167;	Изобразява символа §
&#188; &#189; &#190;	Изобразява съответно символите: 1/4 1/2 3/4

Извеждането в полето на документа на специални символи като *copyright* или запазена марка става съответно с командите "©", "&trad;". Не всички WWW клиенти поддържат целия набор от специални символи, а някои от тях игнорират първата празна позиция (шпация) след специалния символ. За решаване на проблемите, свързани с шпациите, може да се добавят допълнителни такива около използваните специални символи с помощта на командата " ".

В езика HTML 4.0 е въведено цяло множество от специални символи, които са разделени в три категории:

- Множеството от международни типографски символи от Таблица 2;
- Множество от математически символи, силно желани от специалистите в научно-техническите области;
- Множество от специални символи с повече или по-малко общо предназначение, като валутни знаци, запазена марка, запазени права и други.

Посоченият списък от специални символи в Таблица 1 представлява кратка извадка от пълното му множество. Пълна информация за множеството от специални символи може да се намери на адрес:

<http://www.w3.org/TR/WD-entities>

Таблица 2 съдържа списък от специални символи, от разширената таблица на символите, които се извеждат чрез управляващия символ "&".

Таблица 2

<i>Kod</i>	<i>Символ</i>	<i>Kod</i>	<i>Символ</i>	<i>Kod</i>	<i>Символ</i>	<i>Kod</i>	<i>Символ</i>
æ	æ	&Aelig;	Æ	ß	ß		
á	á	Á	Á	ï	ï	Ï	Ï
â	â	Â	Â	ñ	ñ	Ñ	Ñ
à	à	À	À	ó	ó	Ó	Ó
ä	ä	Ä	Ä	ô	ô	Ô	Ô
ç	ç	Ç	Ç	ò	ò	Ò	Ò
é	é	É	É	ö	ö	Ö	Ö
ê	ê	Ê		õ	õ	Õ	Õ
è	è	È	È	ø	ø	Ø	Ø
ë	ë	Ë	Ë	ú	ú	Ú	Ú
í	í	Í	Í	û	û	Û	Û
î	î	Î	Î	ù	ù	Ù	Ù
ì	ì	Ì	Ì	ü	ü	Ü	Ü

За извеждане на малките букви голямата буква в "&string;" се замества със съответната малка. Например, **à** се извежда с командата **à**, а **À** с **À**.

Употребата на специални символи, вмъкнати в текста на документ изисква определен синтаксис при който символичното име на символа се загражда между "&" и ";". Произволен ASCII символ може да се представи и чрез числовия му формат - **&#Number**. В латинските езици се употребяват много ударени букви от разширеното множество символи на ASCII таблицата. Така последователността **<P>´lèves</P>** в един параграф предизвиква извеждането на думата **élèves**, а **<P>grölßer</P>** - на думата **größer**.

Структура на HTML документ

Стандартът HTML специфицира елементи, които разделят документа на описателна (дескриптивна) и функционална (изпълнима) части. Форматът на един HTML документ е следният:

```
<HTML>
  <HEAD>
    <TITLE>Тук се разполага заглавна информация</TITLE>
  </HEAD>
  <BODY>
    Тук се разполага тялото на документа
  </BODY>
</HTML>
```

Описателна част

HTML е елемент, указващ на WWW клиента, че следва код с HTML описатели. Документът започва с **<HTML>** и завършва с **</HTML>**. Съдържанието му се записва във файл с разширение **.html** или **.htm** за операционни системи с формат за имената на файловете 8.3.

HEAD е заглавен елемент, съдържащ информация за идентифициране на HTML документа. Негови елементи могат да бъдат **<TITLE>**, **<BASE>**, **<META>**. Заглавният елемент следва **<HTML>** елемента и предхожда **<BODY>** елемента, където се разполага съдържателната част на документа. Елементът се отваря с **<HEAD>** и се затваря с **</HEAD>**.

TITLE е част от елемента **HEAD**. Елементът се отваря с **<TITLE>** и се затваря с **</TITLE>**. Указаното заглавие на документ се появява при интерпретация от WWW клиент (Netscape, Internet Explorer, Mosaic) в заглавната линия на неговия прозорец. Текстът на заглавието не се включва и не се изобразява в полето на документа, но неговата функция не трябва да се пренебрегва. Това е синтезиран израз на съдържанието на документа, а не просто заглавие и се използва в *Bookmark* списъци, "горещи" точки в системите за търсене на информация по WWW като WAIS. Общоприето е текстът на заглавието да не надхвърля 64 символа.

Изпълнима част

BODY е втората, най-дълга и съдържателна част на един HTML документ. Това е тялото на документа, където се разполага изобразяваната от WWW клиента информация. Елементът се отваря с **<BODY>** непосредствено след **</HEAD>** и се затваря с **</BODY>** преди **</HTML>**. Заглавната част на тялото допуска употребата на атрибути за определяне на цялостния вид на документа. Общият вид на елемента е:

```
<BODY ALINK=#rrbbgg BACKGROUND=URL BGCOLOR=#rrbbgg LINK=#rrbbgg TEXT=#rrbbgg VLINK=#rrbbgg>
```

- **ALINK** определя цвета на активирана хипервръзка;
- **BACKGROUND** определя URL адреса на графичен файл (*gif*, *jpg* или *png*), използван за тапет при изобразяване на документа (фон на документа);
- **BGCOLOR** определя цвета на фона в полето на документа;
- **LINK** определя цвета на хипервръзка;
- **TEXT** определя цвета на буквите в текста;
- **VLINK** определя цвета на избирана вече хипервръзка;
- **rrbbgg** определя цвят на изобразявания атрибут от RGB цветната палитра, зададен като шестнадесетични цифри предхождани от символа "#" или с текст, отговарящ на английското наименование на цвета - [Таблица 5](#).
 - **rr** - шестнадесетичен байт, специфициращ червената компонента със значения от 00 до FF.
 - **bb** - шестнадесетичен байт, специфициращ зелената компонента със значения от 00 до FF.
 - **gg** - шестнадесетичен байт, специфициращ синята компонента със значения от 00 до FF.

При използване на атрибут **BACKGROUND** значението на **BGCOLOR** се подтиска. В HTML 4.0 са включени някои нови атрибути за управление на събития като: *onload*, *onunload*, *onclick*, *ondblclick*, *onmousedown*, *onmouseup*, *onmouseover*, *onmousemove*, *onmouseout*, *onkeypress*, *onkeydown*, *onkeyup*. Тяхното действие ще бъде разгледано в друг раздел.

Браузърът *MS Internet Explorer 5.0* поддържа в елемента **<BODY>** атрибут **BGPROPERTIES** за анимиране на фона в документа. Двойката атрибути **LEFTMARGIN** и **TOPMARGIN** задават в пиксели отместването на съдържанието на документа в полето на *Internet Explorer* спрямо горния ляв ъгъл. Тези три атрибута от елемента **<BODY>** не се поддържат от браузърите на *Netscape* и не са част от стандарта на езика HTML 4.0.

Таблица 5

Имена на цветове и RGB стойности					
Black (Черен)		#000000	Green (Зелен)		#008000
Silver (Сребрист)		#C0C0C0	Lime (Светло зелен)		#00FF00
Gray (Сив)		#808080	Olive (Тъмно зелен)		#808000
White (Бял)		#FFFFFF	Yellow (Жълт)		#FFFF00
Maroon (Кафяв)		#800000	Navy (Морско син)		#000080
Red (Червен)		#FF0000	Blue (Син)		#0000FF
Purple (Пурпурен)		#800080	Teal (Тъмно син)		#008080
Fuchsia (Бежов)		#FF00FF	Aqua (Светло син)		#00FFFF

[Таблица 5](#) съдържа стойностите на шестнадесетте основни цвята от *Windows VGA* палитрата, и съответните символични имена, а следните два записа са равностойни:

```
<BODY bgcolor=#FFFFFF text=#000000 link=#FF0000 vlink=#800000 alink=#FF00FF
background="blue.gif">
```

ИЛИ

```
<BODY bgcolor=white text=black link=red vlink=maroon alink=fuschia
background="blue.gif">
```

В [таблица 5.1](#) са посочени имената и шестнадесетичните кодове на 140 цвята, достъпни за използване от атрибутите *bgcolor*, *text*, *link*, *vlink*, *alink*.

HTML елементи в секция HEAD

Елемент HEAD

Съдържанието на един HTML документ се изобразява в прозорец, чието заглавие се формира от елемента **<TITLE>**. **TITLE** е част от елемента **HEAD**. Елементът се отваря с **<TITLE>** и се затваря с **</TITLE>**. Указаното заглавие на документ се появява при интерпретация от WWW клиент (*Netscape*, *Internet Explorer*, *Mosaic*) в заглавната линия на неговия прозорец. Текстът на заглавието не се включва и не се изобразява в полето на документа, но неговата функция не трябва да се пренебрегва. Това е синтезиран израз на съдържанието на документа, а не просто заглавие и се използва в *Bookmark* списъци, "горещи" точки в системите за търсене на информация по WWW като **WAIS**. Общоприето е текстът на заглавието да не надхвърля 64 символа.

Елемент META

Елементът **<META>** е незадължителна част от секцията **HEAD** и се използва за разполагане на служебна информация, интерпретирана от някои клиент-сървър системи. Параметрите му **NAME** и **CONTENT** описват информация за автора на документа, предназначение, дата на създаване, програмния продукт генератор на документа, ключови думи по които документът може да бъде намерен. Общият вид при използване на елемента е:

```
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=win-1251">
<META HTTP-EQUIV="refresh" CONTENT="n ; URL=url">
<META HTTP-EQUIV="pragma" CONTENT="no-cache">
<META HTTP-EQUIV="Window-Target" CONTENT="име на прозорец">
<META HTTP-EQUIV="expires" CONTENT="дата и час">
<META NAME="author" CONTENT="Автор на документа">
<META NAME="description" CONTENT="Кратко описание">
<META NAME="keywords" CONTENT="Ключови думи">
<META NAME="generator" CONTENT="WWW клиент">
<META NAME="Copyright" CONTENT="COPYRIGHT Име ...">
<META NAME="Publisher" CONTENT="Име фамилия">
<META NAME="Identifier-URL" CONTENT="url">
```

```
<TITLE> ... </TITLE>
<BASE HREF=URL>
</HEAD>
<BODY>
.....
</BODY>
</HTML>
```

META секцията не е задължителна, но не бива да се изпуска. Системите за търсене на информация (*Search Engines*) могат да ползват тези елементи за формиране на общи индекси в техните бази данни. Частта **Description** предполага едно кратко описание на съдържанието на документа. Частта **Keywords** може да съдържа списък от ключови думи, разделени със запетаи и да описват ключовите компоненти в документа.

Вида на документа и кодовата таблица на неговото съдържание се задава чрез следния команден ред:

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=win-1251">
```

Този команден ред трябва да бъде първи в поредицата от **<META>** елементи. Наличието на текстове, изписани на български език, се обозначава с параметъра *charset=win-1251*. Текстове, които са изписани само с букви от латинската азбука се декодират с *charset=ISO-8859-1*. Параметърът *charset* задава декодиращата таблица при визуализацията на документа.

Опцията *refresh* в **<META>** елемента предизвиква автоматично зареждане на нова страница, чийто адрес е указан с параметъра **URL** след изтичане на *n* секунди.

```
<META HTTP-EQUIV="refresh" CONTENT="n ; [URL=url]">
```

Циклично презареждане на текущата страница след изтичане на *n* секунди се постига при отсъствие на параметъра **URL**. Автоматично зареждане на нова страница при указан **URL** след изтичане на зададено време може да се използва за демонстрация на продукти, софтуер, реклама. Тази възможност е демонстрирана непосредствено при проследяване на [тази хипервръзка](#). В примера се зарежда страница със съдържание "Първа страница". Атрибутите на **META** елемента са настроени за преминаване към следваща страница на всеки 5 секунди. В страницата се променя цвета на текста и неговото съдържанието.

За предотвратяване зареждането на страницата в кеш-паметта на клиента се използва следния команден ред:

```
<META HTTP-EQUIV="pragma" CONTENT="no-cache">
```

С помощта на атрибута *Window-Target* се указва името на подрабитацият се фрейм.

```
<META HTTP-EQUIV="Window-Target" CONTENT="New-Frame">
```

Актуалността на страницата се определя до зададена дата и час. Тази информация е предназначена за *proxy server*, при което страницата трябва задължителна да бъде презаредена.

```
<META HTTP-EQUIV="expires" CONTENT="Wed, 10 Febr 1999 12:00:00 GMT">
```

Името на автора на документа се указва чрез атрибут *author*.

```
<META NAME="author" CONTENT="Тодор Иванов">
```

Атрибутът *description* спомага за кратко описание на съдържанието на документа.

```
<META NAME="description" CONTENT="Кратко описание на съдържанието">
```

Списък от ключови думи за индексирание на документа с помощта на атрибута *keywords*.

```
<META NAME="keywords" CONTENT="Ключови думи">
```

Чрез атрибут *generator* се посочва генераторът на HTML документа.

```
<META NAME="generator" CONTENT="WWW клиент">
```

Уточняване на правата *Copyright*.

```
<META NAME="Copyright" CONTENT="COPYRIGHT Име ...">
```

Уточняване на името на този, който публикува документа за случаите в които това не е неговият автор.

```
<META NAME="Publisher" CONTENT="Иван Иванов">
```

Пълният *URL* адрес на Web сайта собственик на документа

```
<META NAME="Identifier-URL" CONTENT="url">
```

Указания към роботите за индексирание на страници се задават с атрибут *robots*.

```
<META NAME="robots" CONTENT="all | none | index | noindex | follow | nofollow">
```

Множеството параметри имат следното значение:

- *all* - всичко подлежи на индексирание;
- *none* - забрана за индексирание;
- *index* - страницата е вече индексирана;

- ***noindex*** - индексират се само връзките от страницата;
- ***follow*** - извличане на връзките от страницата за последващо индексирание;
- ***nofollow*** - да не се извличанат връзките от страницата.

Указания за използвания език или езици в страницата:

```
<META NAME="site-languages" CONTENT="English Bulgarian">
```

Задава предметната област, категория за класификация на страницата:

```
<META NAME="subject" CONTENT="Internet">
```

Определяне на начина на разпространение:

```
<META NAME="distribution" CONTENT="global">
```

За този елемент дори е предвидена опция *"interplanetary"*.

Определяне на географско разположение за сайта или компанията:

```
<META NAME="geography" CONTENT="Град, Област, Държава, Пощенски код">
```

При различие в месторазположението на сървъра и компанията за препоръчване е да се укаже адреса на компанията.

Класификация на HTML страницата като вид ресурс:

```
<META NAME="resource-type" CONTENT="document">
```

Индикатор за важна или актуална информация:

```
<META NAME="highlight" CONTENT="Highlight information">
```

Елемент BASE

Елементът **<BASE>** е част от секцията на **HEAD** командите. Поставена там, командата указва на WWW клиента базовия абсолютен URL адрес, спрямо който в HTML документите ще се формира относителния.

```
<BASE HREF=URL>
```

С командата **<BASE HREF="http://mulmed.vmei.acad.bg/">** се установява абсолютен адрес, след което в документите хипервръзките се задават като относителни URL. Така препратката **** практически замества ****. При изграждането на един HTML проект от множество хора на различни места винаги се появяват затруднения при интегрирането на

отделните му елементи. С командата **<BASE>** много от неудобствата се преодоляват по естествен начин.

HTML елементи в секция BODY

Различни нива на заглавия (heading commands)

Авторът на HTML документи може да задава различен размер на текста. Повечето WWW клиенти изпълняват коректно тези команди, но някои (предимно ориентирани към извеждане само на текст) игнорират или променят тези команди.

Командите за задаване на *размер на текст* се използват за заглавия или в случаи, когато трябва да се привлече вниманието към определен пасаж. Тези команди са непражни, т.е. изисква се завършващ елемент. Форматът на елемента е следният:

```
<H1|2|3|4|5|6 ALIGN=LEFT|CENTER|RIGHT>Текст на заглавие</H1|2|3|4|5|6>
```

където, "?" се замества със число от едно до шест. <H1>задава най-големият достъпен размер, <H2> - вторият по големина, и т.н., <H6> - най-малкият размер. Действителният размер на текста зависи от WWW клиента - илюстрация на [Екран 1](#). Примерът може да бъде изпълнен и директно при проследяване на [тази хипервръзка](#).

HTML е лесен за усвояване - (H1)

HTML е лесен за усвояване - (H2)

HTML е лесен за усвояване - (H3)

HTML е лесен за усвояване - (H4)

HTML е лесен за усвояване - (H5)

HTML е лесен за усвояване - (H6)

Екран 1 - Нива на заглавия

Отделните атрибути имат следното значение:

- **H1|2|3|4|5|6** - значенията от 1 до 6 определят размера на заглавието.
- **ALIGN=LEFT|CENTER|RIGHT** - задава относителното подравняване на заглавния ред (в ляво или центрирано или в дясно).

Управление на текстовия поток (text-flow commands)

По правило WWW клиентът реорганизира разположението на текста в даден документ, като отстранява всички управляващи символи "CR" и допълнителните празни позиции (шпации).

В езика съществуват специални команди, които позволяват даден текст да се визуализира точно по начина по който е създаден.

Форматиране на параграф - **<P>**

<P> е команда за начало на нов параграф. По принцип всички параграфи трябва да започват и да завършват с този елемент. Командата не изисква краен елемент **</P>**, въпреки, че много WWW клиенти го съблюдават. В края на параграфа се поставят два символа "CR". Следващият параграф започва през един ред. Увеличаване на разстоянието между параграфи може да стане с използване на празна дефиниция на параграф като между **<P>** и **</P>** не се поставя текст, а специален символ (* *). Командата за начало на параграф се използва в два формата - съкратен и пълен, където се поддържат атрибути за подравняване на текста - **ALIGN**.

Съкратен запис:

```
<P>Text</P>
```

Разширен запис:

```
<P ALIGN=left|center|right> ... </P>
```

- **ALIGN=CENTER** - Текстът в параграфа ще бъде центриран;
- **ALIGN=LEFT** - Текстът в параграфа ще бъде изравнен в ляво;
- **ALIGN=RIGHT** - Текстът в параграфа ще бъде изравнен в дясно.

Преминване на нов ред - **
**

**
** е празен елемент, който налага извеждане на "CR" в точката, в която е зададен. Елементът осигурява нормално междуредие в рамките на параграф или списък когато текстът трябва да се разположи на повече самостоятелни редове.

Съкратен запис:

```
<BR>
```

Разширен запис:

```
<BR CLEAR=left|all|right|none>
```

Използването на елемента с параметри дава добри възможности за разполагане на текст около графични обекти и таблици в полето на HTML документа. Параметрите определят по какъв начин да бъде разположен новият ред, формиран след изпълнението на командата **
**. Значението на параметрите е следното:

- **CLEAR=left** - редът започва от първата допустима линия под обекта, подравнен в ляво;

- **CLEAR=all** - редът започва от първата допустима линия под обекта, подравнен спрямо центъра;
- **CLEAR=right** - редът започва от първата допустима линия под обекта, подравнен в дясно;
- **CLEAR=none** - следващият ред започва нормално (значение по подразбиране).

За блокиране преминаването на нова линия текстът може да се ограда от елементите **<NOBR>** и **</NOBR>**. Използването им трябва да става внимателно, защото може да предизвика формиране на дълга линия на един ред и да принуждава потребителя да използва асансьора за хоризонтално преместване, което нарушава комфорта при четене.

Преформатиране на текст - **<PRE>**

<PRE> е непразен елемент. Заграденият от елемента текст се извежда в полето на документа без промяна в шрифт с фиксирана широчина (Courier). Командата се използва винаги, когато е необходимо текстът да изглежда по същият начин, по който е зададен във източника, като се съблюдават специфицираните команди, символи "CR" и празни позиции.

Съкратен запис:

<PRE>

Разширен запис:

<PRE WIDTH=n>, където **n** определя броя на символите на един ред.

Хоризонтална линия - **<HR>**

<HR> е команда за извеждане на хоризонтална линия в текущата позиция. Дебелината и формата на изобразяване на линията се контролира от параметри. При повечето WWW клиенти тази линия е съпроводена от празен ред преди и след линията. Командата е полезна за разделяне на документа на отделни части или глави.

Съкратен запис:

<HR>

Разширен запис:

<HR ALIGN=LEFT|RIGHT|CENTER NOSHADE SIZE=n WIDTH=n|n% COLOR="#rrggbb">

- **ALIGN=LEFT|RIGHT|CENTER** - определя хоризонталното подравняване на линията - ляво|дясно|центрирано.
- **NOSHADE** - стандартно линията се изтегля със сянка (3D). Атрибутът премахва това условие.
- **SIZE=n** - определя дебелината на линията в зависимост от стойността на **n**, зададена в пиксели.

- **WIDTH=n|n%** - задава дължината на линията определена от стойността на **n**, зададена в пиксели или проценти.
- **COLOR="#rrggbb"** - задава цвета на линията (поддържа се от MS Internet Explorer само).

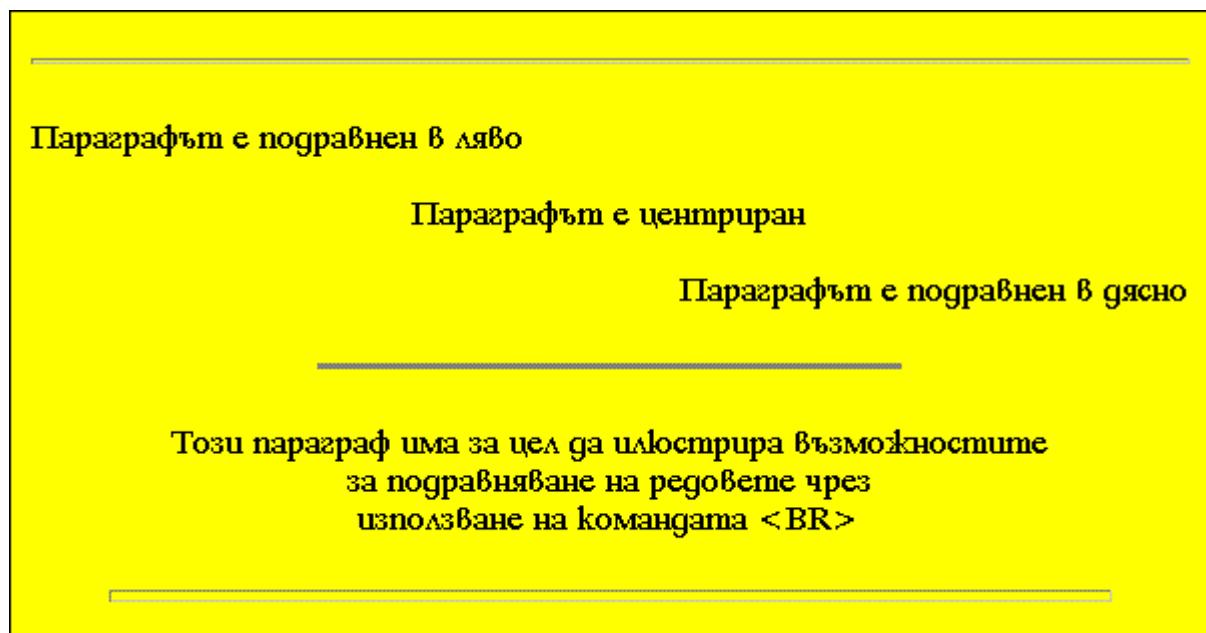
Коментар - `<!-->`

`<!-- Text -->` е елемент за въвеждане на коментар в кода на HTML документа, който няма да бъдат интерпретирани и съответно изобразяван в полето на документа.

Пример с команди за управление на текстовия поток - код:

```
<HR WIDTH="100%">
<P ALIGN=left>Параграфът е подравнен в ляво</P>
<P ALIGN=center>Параграфът е центриран</P>
<P ALIGN=right>Параграфът е подравнен в дясно</P>
<HR NOSHADE SIZE=3 WIDTH="50%">
<P ALIGN=center>Този параграф има за цел да илюстрира възможностите<BR>за
подравняване на редовете чрез<BR>използване на командата &lt;BR&gt;</P>
<HR SIZE=5 WIDTH="500%">
```

Визуализация след интерпретация на кода:



Екран 2 - Управление на текстовия поток

Структуриране на документ `<DIV>`

`<DIV>` е команда за логическо разделяне на текста в един документ на отделни блокове като глави, приложения, библиография и други. Командата е от непразен тип, като заключените текст или параграфи между началния и крайния елемент се подчиняват на дефинираните в командата атрибути.

Съкратен запис:

```
<DIV> ... </DIV>
```

Разширен запис:

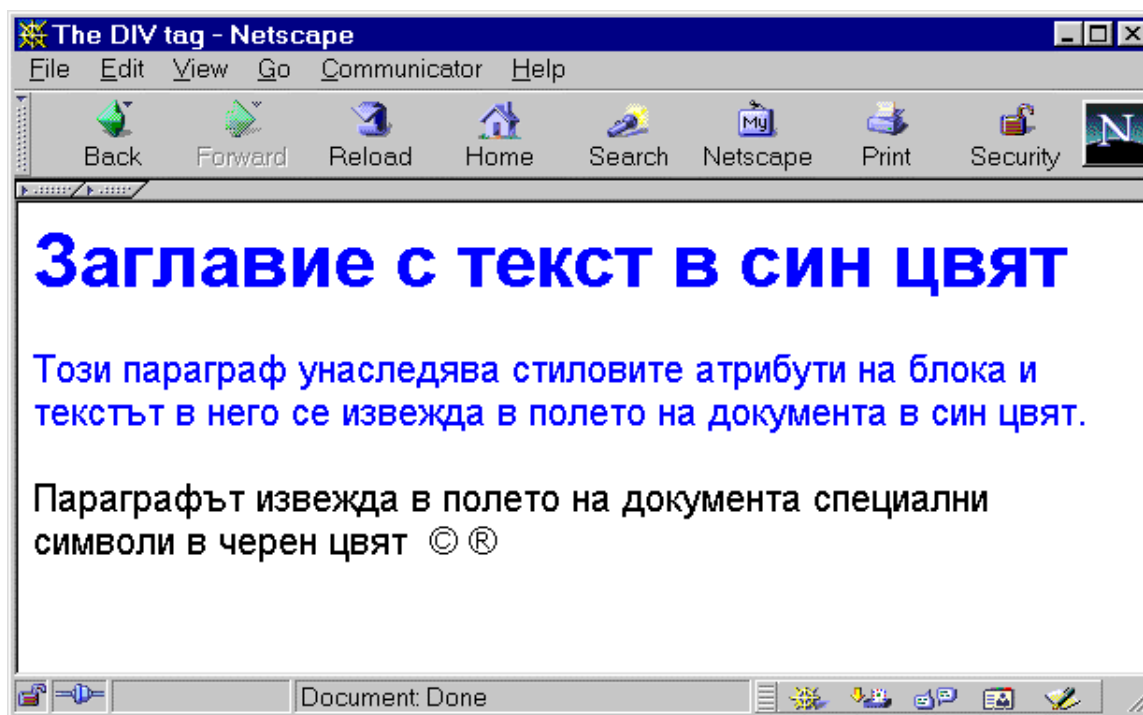
```
<DIV ALIGN=LEFT|RIGHT|CENTER ID|CLASS=name STYLE=style-data CLEAR=LEFT|RIGHT|ALL NOWRAP> ... </DIV>
```

- **ALIGN=LEFT|RIGHT|CENTER** - определя хоризонталното подравняване на блока - ляво|дясно|центрирано.
- **ID|CLASS=name** - атрибутът задава единствен идентификатор за елемента (*ID*) или име на клас/класове към този елемент (*CLASS*).
- **STYLE=style-data** - атрибутът определя стиловата дефиниция за елемента.
- **CLEAR=LEFT|RIGHT|ALL** - начало на блока с празни полета - ляво|дясно|от двете страни.
- **NOWRAP** - включва/изключва пренасянето на текста в блока на нов ред.

Следващият пример илюстрира приложението на команда **<DIV>** и унаследяване на неговите атрибути от командите в блока:

```
<HTML>
<HEAD>
<TITLE>The DIV tag</TITLE>
</HEAD>
<BODY>
<DIV STYLE="COLOR:BLUE" CLEAR=ALL>
<H1>Заглавие с текст в син цвят</H1>
<P>Този параграф унаследява стиловите атрибути на блока и текстът в него се
извежда в полето на документа в син цвят.</P>
<P><FONT COLOR="BLACK">Параграфът извежда в полето на документа специални
символи в черен цвят &nbsp;&copy;&nbsp;&reg;</FONT></P>
</DIV>
</BODY>
</HTML>
```

Визуализация след интерпретация на кода (Екран 2.1):



Екран 2.1 - Приложение на команда <DIV>

Форматиране на символи (character-formatting commands)

Елементите за форматиране на символи са винаги непразни и могат да бъдат разделени на две групи - за задаване на физически и логически атрибути на текст. Тези елементи формират стиловия израз на символите в текста при избран шрифт. Употребата им е възможна в рамките на текстов параграф или списъчна структура.

Физически атрибути

Физическите атрибути са същите както в текстообработващите системи. Употребата на завършващ елемент е задължителна (непразни елементи). Поставеният между началния и краен елемент текст се форматира в зависимост от употребената команда, например:

- **Boldface** - извежда символи в удебелен (получер) шрифт;
- **<I>Italic</I>** - извежда символи в курсив;
- **<U>Underline</U>** - извежда символи с подчертаване;
- ** [Text] ** - задава цвят и размер на символите за избран шрифт - илюстрация на Екран 2;
 - **COLOR=#rrbbgg** определя цвят на изобразявания текст от RGB цветната палитра, зададен като шестнадесетични цифри предхождани от символа "#" (виж Таблица 5).
 - **SIZE=n** - задава размера на шрифта, указан чрез *n* или относителната му промяна спрямо текущо установения размер чрез $\pm n$.
 - **FACE = list** - задава се списък от имена на шрифтове, разделени със запетаи. Текстът в полето на документа се изобразява в първия намерен шрифт от списъка.
- **<TT>Teletype text-fixed-width font (Courier)</TT>** - заграденият текст между началния и крайния елемент се извежда с шрифт с фиксирана широчина, сходен с този на пишеща машина, обикновено Courier.

Пример 1 с команди за форматиране на символи - код:

```
<P><FONT SIZE=-2>Размер n=1 - Размер n=-2<BR></FONT>
<FONT SIZE=-1><U>Размер n=2, подчертан - Размер n=-1</U><BR> </FONT>
<FONT FACE="Hebar" SIZE=+0>Размер n=3, шрифт Hebar - Размер n=+0
<BR></FONT>
<B><FONT SIZE=+1>Получер с размер n=4 - Размер n=+1<BR></FONT> </B>
<I><FONT SIZE=+2>Курсив с размер n=5 - Размер n=+2<BR></FONT> </I>
<FONT SIZE=+3>Размер n=6 - Размер n=+3<BR></FONT>
<FONT SIZE=+4>Размер n=7 - Размер n=+4</FONT><BR>
<TT>Teletype text-fixed-width font Courier</TT></P>
```

Визуализация след интерпретация на кода:

Размер n=1 - Размер n=-2

Размер n=2, подчертан - Размер n=-1

Размер n=3, шрифт Hebar - Размер n=+0

Получер с размер n=4 - Размер n=+1

Курсив с размер n=5 - Размер n=+2

Размер n=6 - Размер n=+3

Размер n=7 - Размер n=+4

Teletype text-fixed-width font Courier

Екран 2 - Стилови формати

Възможна е комбинация от различни команди за форматиране на текст, например:

```
<B><I><U> Определен текст </U></I></B>
```

В подобни случаи поведението на различните WWW клиентите не е гарантирано. Някои съблюдават всички елементи, други избират един от атрибутите и го прилагат, а трети игнорират изцяло зададената комбинация.

Шрифтове

Вида на шрифта се определя автоматично от браузъра при визуализация на документа. Този т.н. автоматизъм всъщност произлиза от настройките на WWW клиента, където се задават двойка предпочитани шрифта - един пропорционален и един с фиксирана широчина. Този подход е много рационален и не създава неудобства на клиента при разчитане на изтегления документ.

Тенденцията за динамизиране съдържанието на HTML страниците, често пъти, изисква съобразяване с авторския подход. Авторът на HTML страници може да задава конкретни имена или фамилии от шрифтове, с които да се визуализира съдържанието на неговите документи при клиента. За удовлетворяване на тази цел се използва елемента ****. Елементът се прилага за текстови фрагменти и определя шрифта, големината му и цвета на изобразявания текст.

```
<FONT FACE="font_name" SIZE=n COLOR="rrggbb" ... </FONT>
```

където:

- **FACE** = "*font_name*" - задава името на шрифта;
- **SIZE** = *n* - определя големината на символите от шрифта;
- **COLOR** = "*rrggbb*" - задава цвета на изобразявания текст.

Атрибутът **FACE** може да съдържа списък от шрифтове или фамилия шрифт. Това осигурява безпроблемно изобразяване на документа при клиента при отсъствие на някои от тях.

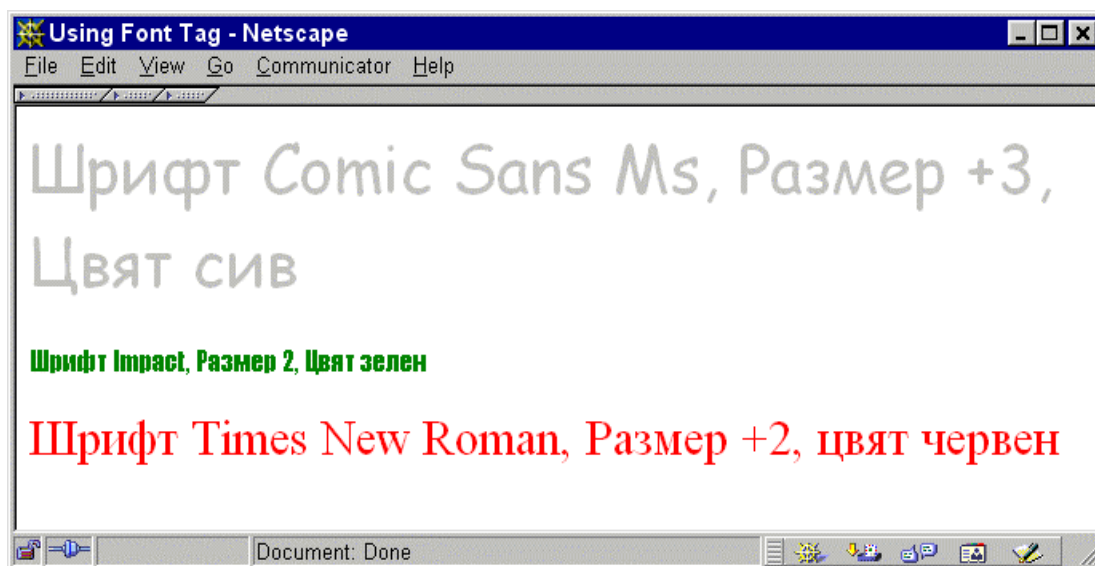
Стойността за елемента **SIZE** се задава в относителни (**SIZE=+2**) или абсолютни (**SIZE=2**) единици.

За цвета на текста чрез атрибут **COLOR** се указва името или шестнадесетичната стойност на избран цвят от *RGB* палитра.

Например:

```
<html>
<head><title>Using Font Tag</title>
</head>
<body>
<p><FONT FACE="Comic Sans Ms, arial" SIZE=+3 COLOR="#C0C0C0" >
Шрифт Comic Sans Ms, Размер +3, Цвят сив
</FONT>
<p><FONT FACE="Impact, verdana" SIZE=2 COLOR="green">
Шрифт Impact, Размер 2, Цвят селен
<p><FONT FACE="Times New Roman" SIZE=+2 COLOR="red">
Шрифт Times New Roman, Размер +2, Цвят червен
</FONT>
</body>
</html>
```

Визуализация след интерпретиране на кода - Екран 2.1.



Екран 2.1 - Различни шрифтове, големина и цвят на текста

Използването на различни шрифтове подобрява визията на документа. Желанието на някои автори да направят страниците си по-атрактивни се натъква на затруднения при визуализацията им при клиента, особено в случаите, когато използваният шрифт липсва. Тогава, той е заменен автоматично с шрифт от същата фамилия, но търсения ефект се губи. Проблемът отчасти се решава с използване на *DHTML* и стилови формати (виж раздел "[Независима каскада от стилови формати в HTML 4.0 Cascading Style Sheets Level 1](#)"). Цялостно решение се постига при представяне на характерния надпис като графичен компонент във формат *gif*. Много разпространена практика е използването на банер-панели с анимирани надписи. Те се съставят като графични елементи и комбинират графика с текст.

За указване на шрифт, общ за целия документ, се използва елементът **<BASEFONT>**. Тази директива се разпознава от браузъра на *Microsoft Internet Explorer 5.0*. Атрибутите не елемента и начина на използване е същият както при елемента ****, но без затварящ елемент.

```
<BASEFONT FACE="font_name" SIZE=n COLOR="rrggbb">
```

Елементът се поставя в секцията **<HEAD>** след **<TITLE>** на HTML документа. За постигане на същия ефект при *Netscape Communicator 4.x*, елемента **** с неговите атрибути се поставя на същото място без затварящ елемент.

Логически атрибути

Логическите атрибути са по-абстрактни и по-многобройни. При използването им не е възможно да се предвиди как конкретният WWW клиент ще интерпретира командата. Понастоящем повечето клиенти правят съответствие между различните логически елементи и физическите елементи ****, **<I>**, **<U>**, **<TT>** или въобще не интерпретират логическия атрибут. Браузърите от четвърто поколение (*Netscape 4.x*, *Internet Explorer 4.x*) интерпретират нормално логическите атрибути включени в HTML 4.0. Основното предназначение на логическите атрибути е постигане на пряка съвместимост с текстообработващите системи като например *Microsoft Word*. Съществуващите в момента конвертиращи програми от един файлов формат към HTML формат широко използват логическите атрибути.

В Таблица 6 е даден списък на логически и някои допълнителни стилови атрибути:

Таблица 6

Атрибути	Тълкувание
<ADDRESS> </ADDRESS>	Адрес на автора на документ
<BIG> </BIG>	Изобразяване на текст в едър шрифт
<BLINK> </BLINK>	Създава ефект на мигащ текст
<BLOCKQUOTE> </BLOCKQUOTE>	Форматиране на текст като цитат
<CITE> </CITE>	Цитиране - текст в курсив

<code><CODE> </CODE></code>	Използува се за HTML директиви
<code><DFN> </DFN></code>	Задава дефиниция
<code><KBD> </KBD></code>	Посочва въвеждане на текст от клавиатурата
<code> </code>	Подчертаване (най-често <i>italic</i>)
<code><LISTING> </LISTING></code>	Дублира функцията на <code><PRE></code>
<code><PLAINTEXT> </PLAINTEXT></code>	Дублира функцията на <code><PRE></code>
<code><SAMP> </SAMP></code>	Посимволно извеждане на текст
<code><SMALL> </SMALL></code>	Извеждане на текст в малък шрифт
<code> </code>	Подчертаване (удебелен шрифт)
<code><STRIKE> </STRIKE></code>	Зачертаване (хор. черта през текста)
<code><STYLE> </STYLE></code>	Употреба на външен стилев формат
<code><SUB> </SUB></code>	Задава ^{долен} индекс в текст
<code><SUP> </SUP></code>	Задава ^{горен} индекс в текст
<code><VAR> </VAR></code>	Указва променлива на програма
<code><XMP> </XMP></code>	Вмъкване на пример - сходно с PRE

Тенденцията е в бъдеще делът на логическите атрибути в езика HTML да нараства, тъй като тези елементи съдържат информация за вида на текста и за начина на неговата визуализация. Част от описаните в Таблица 6 HTML команди са взаимозаменяеми и имат сходни функции. Основното им

Подредени и неподредени списъци (list commands)

Версиите 3.2 и 4.0 на езика HTML поддържат три типа списъци:

- подредени (*ordered lists*) - ` `;
- неподредени (*unordered*) - ` `;
- списъци от дефиниции (*definition lists*) - `<DL> </DL>`.

Отделните елементи в подредените и неподредени списъци се задават чрез командата ``.

Подредените списъци представят последователни операции или алгоритми и WWW клиентът автоматично генерира последователни числа пред всеки елемент в списъка.

При неподредените списъци пред всеки елемент се поставя графична точка.

Списъците от дефиниции осигуряват форматиране на текста подобно на това в речници - за всеки елемент (термин) се дава съответно описание или обяснение. В списъците от дефиниции елементите се задават чрез `<DT>` и `<DD>`, като по този начин се отделя терминът

от описанието му. В тялото на списъка от дефиниции могат да се задават параграфи, подредени или неподредени списъци, форматиращи команди за текст.

Неподреден списък - `` ``

Съкратен запис:

```
<UL> ... </UL>
```

Пълен запис:

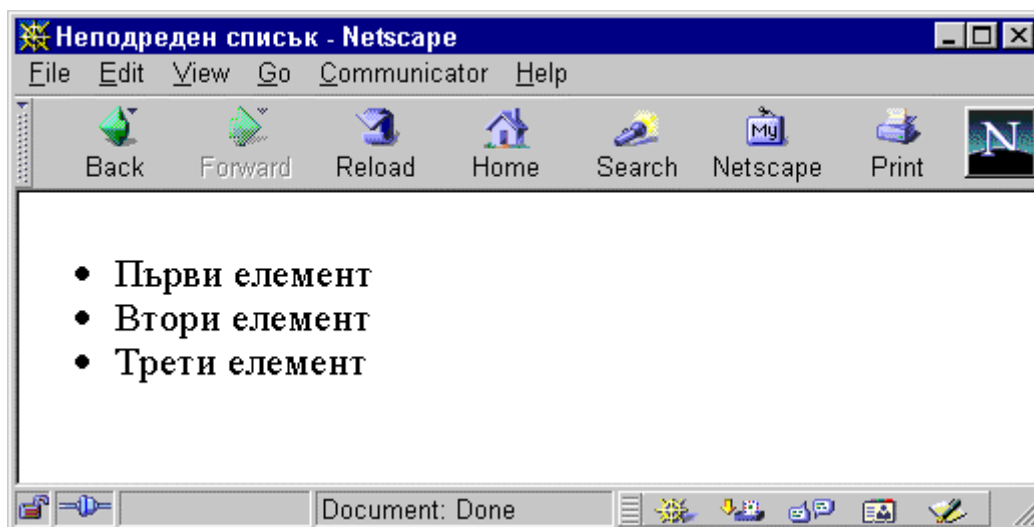
```
<UL COMPACT TYPE=disc|square|circle> ... </UL>
```

- **COMPACT** - представяне в съгъстен формат;
- **TYPE** - задава вида на водещия символ на точка от списъка както следва:
 - *disc* - елемент във вид на плътен диск;
 - *square* - елемент във вид на квадрат;
 - *circle* - елемент във вид на кръг.

Последователността от следните HTML команди задават неподреден списък:

```
<HTML>
<HEAD>
<TITLE>НЕПОДРЕДЕН СПИСЪК</TITLE>
</HEAD>
<BODY>
<UL COMPACT TYPE=disc>
<LI> Първи елемент
<LI> Втори елемент
<LI> Трети елемент
</UL>
</BODY>
</HTML>
```

Списъкът се визуализира по следния начин:



Екран 3 - Неподреден списък

Подреден списък - `` ``

Съкратен запис:

```
<OL> ... </OL>
```

Пълен запис:

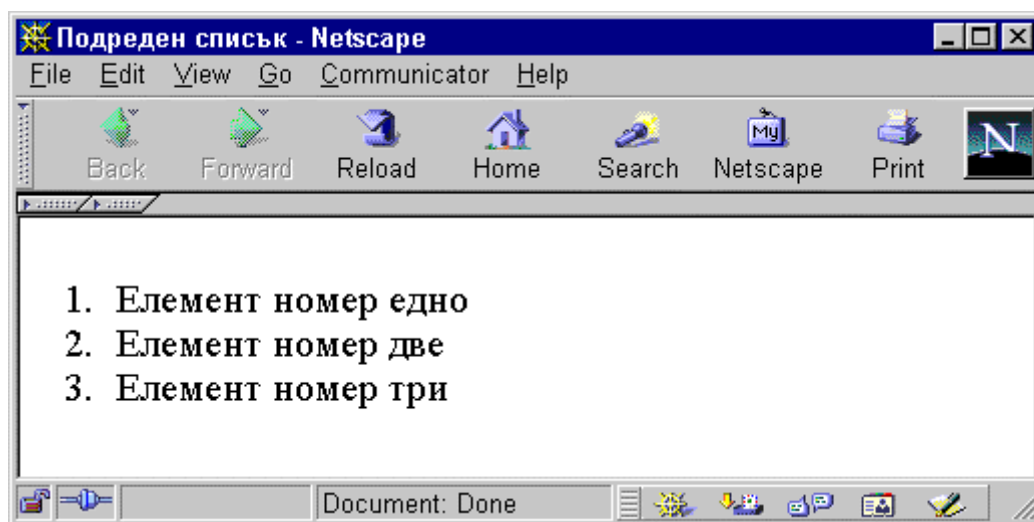
```
<OL COMPACT START=NUMBER TYPE=1|a|A|i|I> ... </OL>
```

- **COMPACT** - представяне в съгъстен формат;
- **START=NUMBER** - задава началната стойност на първия елемент в списъка;
- **TYPE=1|a|A|i|I** - определя символното множество от знаците за изброяване.

Последователността от следните HTML команди представлява подреден списък:

```
<HTML>
<HEAD>
<TITLE>Подреден списък</TITLE>
</HEAD>
<BODY>
<OL>
<LI> Елемент номер едно
<LI> Елемент номер две
<LI> Елемент номер три
</OL>
</BODY>
</HTML>
```

Визуализацията на списъка е дадена на *Екран 4*:



Екран 4 - Подреден списък

Списък от дефиниции - `<DL>` `<DT>` `<DD>` `</DL>`

Елементите формиращи списъци от дефиниции нямат параметри. Елементът **<DL>** задава началото на списъка, елементът **<DT>** задава етикета на дефиницията, а елементът **<DD>** задава описанието на дефиницията. Само началният елемент **<DL>** има завършващ - **</DL>**.

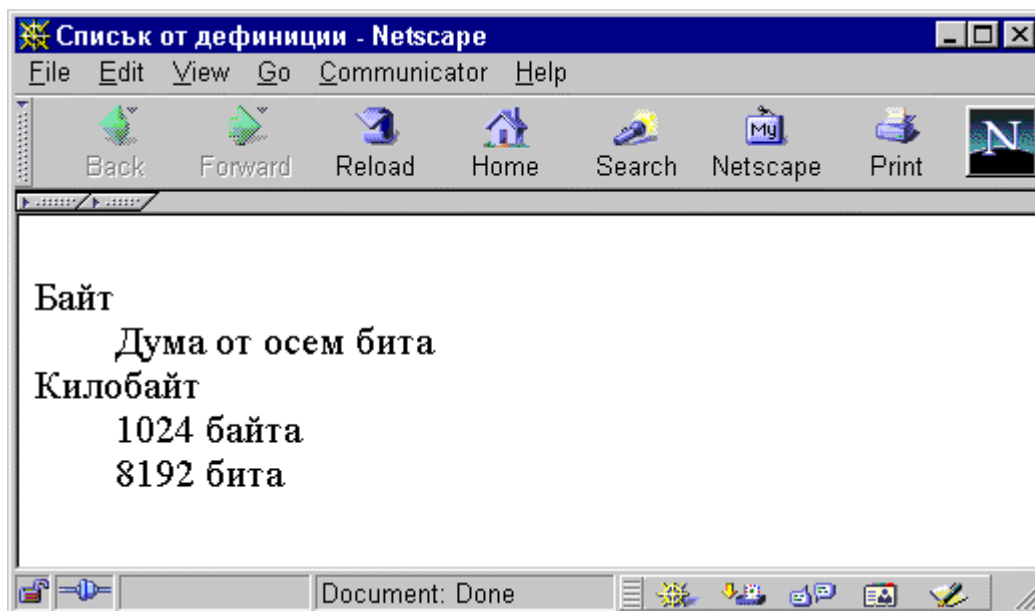
Съкратен запис:

```
<DL>
<DT> item
<DD> text
</DL>
```

Последователността от следните HTML команди представлява списък от дефиниции:

```
<HTML>
<HEAD>
<TITLE>Списък от дефиниции</TITLE>
</HEAD>
<BODY>
<DL>
<DT> Байт
<DD> Дума от осем бита
<DT> Килобайт
<DD> 1024 байта
<DD> 8192 бита
</DL>
```

Визуализацията след интерпретация на кода е дадена на *Екран 5*.



Екран 5 - Списък от дефиниции

Поддържат се вложени списъци, а някои WWW клиенти използват различни по форма или цвят точки за неподредените списъци. Авторът не може да задава разстояния и отстъпи в списъците. Пример:

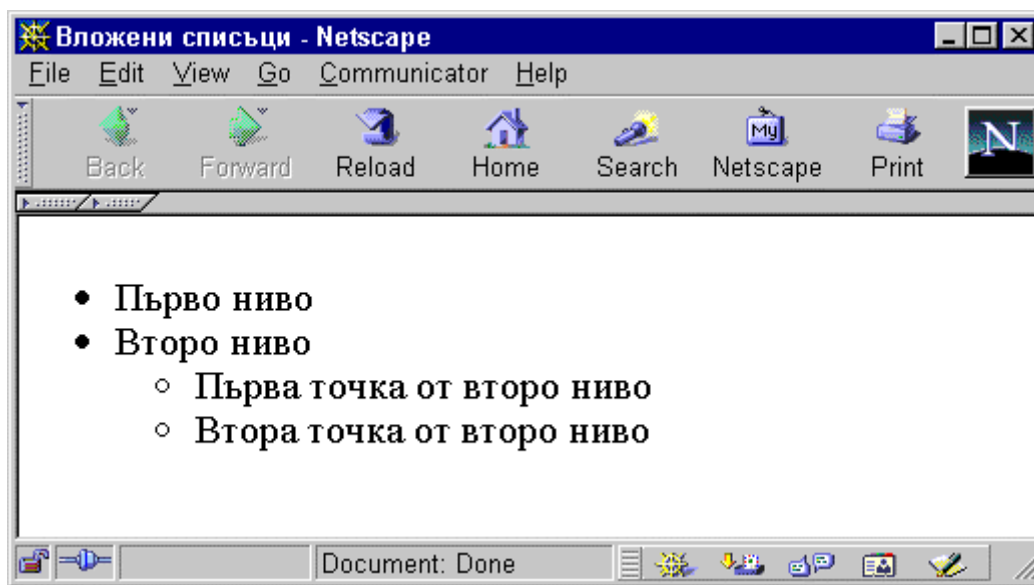
<HTML>

```

<HEAD>
<TITLE>Вложени списъци</TITLE>
</HEAD>
<BODY>
<UL>
<LI>Първо ниво
<LI>Второ ниво
<UL>
<LI>Първа точка от второ ниво
<LI>Втора точка от второ ниво
</UL></UL>
</BODY>
</HTML>

```

Визуализацията след интерпретация на кода е дадена на *Екран 6*.



Екран 6 - Вложен списък

Заглавие за списък <LH>

```
<LH>Text Title
```

Елементът <LH> задава заглавие за всеки един от видовете списъци. Елементът се поставя като първи след определителя на списъка и няма завършващ елемент.

Създаване на хипервръзки (anchor & link commands)

Една от най-важните възможности на HTML е създаването на хипервръзки (*anchors* според стандарта HTML, но също *links* или *hyperlinks*) към други обекти. Езикът HTML поддържа връзки към следните типове единици:

- Друг HTML документ;
- Друг, различен от HTML документ (графични изображения, двоични файлове, текст, звукови файлове, анимация, видео клипове и др.);
- Произволна точка от текущия HTML документ;
- Друг протокол - FTP, Telnet, Mail, News, Gopher, Wais.

Командите `<A>` и `` се използват за създаване на връзки към други файлове или позиции в текущия файл. При условие, че тези елементи заграждат някакъв *текст*, той става избираем, гореща дума по смисъла на хайпертекста. След избирането му се активира връзката свързана с него.

В случай, че елементите `<A>` и `` заграждат *графично изображение*, то става избираемо и след избирането му се реализира съответната връзка. Ако връзката сочи към друг документ, той се извлича и визуализира.

Връзки към *позиция в същия документ* осъществяват преход към съответната част от документа.

В случай, че връзката е към файл с формат различен от HTML, то файлът се извлича и визуализира, като се използват възможностите на съответния WWW клиент да активира външни визуализиращи програми, например *Adobe Acrobat Reader* за *PDF* файлове. Възпроизвеждането на звукови файлове (*.wav*) или видео клипове (*.mov*) става възможно по пътя на асоциираните имена, активиращи съответните приложения. За тази цел WWW клиентът, например *Netscape Navigator*, *Internet Explorer* или *Netscape Communicator*, чрез своя програмен интерфейс активира необходимото приложение (*Plug-Ins*).

Елементът `<A>` почти във всички случаи се употребява с параметри. Форматът на командата е:

```
<A HREF=URL | NAME=cdata REL=cdata REV=cdata TITLE=cdata> ... </A>
```

- **HREF** - Описва хипервръзката чрез URL адрес;
- **NAME** - Задава името на етикет в хипервръзка (*Anchor*);
- **REL** - Задава връзката от **HREF** към *Anchor*;
- **REV** - Задава връзката от *Anchor* към **HREF**;
- **TITLE** - информативно - описва предназначението на **HREF** обекта.

При използването на елемента не съществуват ограничения нито по отношение на броя на връзките, нито към какъв тип документ да сочи дадена връзка, но хипервръзки не могат да се влагат една в друга.

За да се създаде връзка към друг документ или файл (*Link*) се използва параметърът **HREF** на командата `<A>`. Пример:

```
<A HREF="file.html"> Текст за хипервръзка </A>
```

В примера текстът *"Текст за хипервръзка"* е избираем, като след избирането му ще бъде извлечен и визуализиран указаният файл *"file.html"*. Параметърът **HREF** сочи относителен или абсолютен **URL** адрес на документ.

В следващия пример активното за избиране поле е графично изображение, активирано в полето на документа чрез командата ``.

```
<A HREF="http://www.server.com/gen.exe"><IMG SRC="graf.gif"></A>
```

При избор на хипервръзката от WWW сървъра "www.server.com" се извлича файлът "gen.exe". Той не е в HTML формат и ще бъде изпратен на потребителя като двоичен файл и ще бъде съхранен върху текущото устройство. Този подход се използва много често от Web сървъри за прехвърляне на файлове към потребители и по същество замества функциите на FTP-услугата.

Установяването на хипервръзка към позиция във външен документ или към текущия (*anchor*) става чрез задаване на етикети. Връзката се задава по следния начин:

```
<A HREF="#label1"> Текст за избор </A>
```

За да се осъществи тази хипервръзка е необходимо в съответната позиция на документа да се вмъкне следният ред (етикет):

```
<A NAME="label1"> Точка, към която се прави преход </A>
```

В качеството на етикет може да се използва произволен текст, който задължително започва със символа "#" и не съдържа специални символи. Параметърът *NAME* задава самия етикет, като не се поставя символът "#". Когато потребителят избере текста "*Текст за избор*", ще бъде осъществен преход към онази част от документа, която съдържа параметъра *NAME*, т.е. текста "*Точка, към която се прави преход*". Възможностите за преход към различни части на един документ или към произволна точка от външен документ е основна характеристика на хипертекста като технология. Тези възможности могат да се използват в един HTML документ под формата на обръщения от индекс към тълкувателната част на изложението. Етикетите са зависими към малки и големи букви.

Създаването на връзки към етикети, намиращи се в други документи се осъществява по следния начин:

```
<A HREF="myfile.html#L1">За повече информация, натиснете тук</A>
```

В този пример ще бъде извлечен файл "myfile.html" и ще се визуализира частта от него, съдържаща етикета *L1*.

Хипервръзка от HTML документ към E-Mail адрес се задава чрез *Mailto*:

```
<A HREF="mailto:user@host">Text</A>
```

където, *user@host* е електронният адрес на лицето, получател на съобщението. Хипервръзката "*Text*" се визуализира в полето на документа. При активирането ѝ, в зависимост от възможностите на WWW клиента, на екрана се отваря прозорец с формуляр за редактиране на електронна поща. При *Netscape Navigator* това е интегрална част от приложението. Към изпозвания Web клиент може да се асоциира предпочитания Mail клиент в опциите *Preferences*.

Едно електронно съобщение съдържа множество параметри. Със средствата на хипервръзка и протокола *mailto*: в HTML кода може да се направи пълна подготовка на съобщението, което само да бъде изпратено от потребителя. Всяко съобщение има получател (*mail address*), част описваща предмета на съобщението (*subject*), тяло със съдържание на самото

съобщение (*body*), и при необходимост, изпращане на копие до друг кореспондент (*cc*).

Пример:

```
<A HREF="mailto:ivanov@vmei.acad.bg?cc=totev@hotmail.com
&subject=предмет&BODY=текст на съобщението">Вашето мнение</A>
```

По-голямата част от HTML клиентите разпознават определен брой файлови формати по техните разширения. Наборът от разпознаваните файлови формати се задава в секцията *Helpers* на съответния HTML клиент. В секцията *Preferences* се указват имената на приложенията, обработващи съответния формат. Този механизъм може да бъде използван за възпроизвеждане на звукови файлове и видео клипове в полето на един HTML документ чрез хипервръзки, които да сочат адресите на тези файлове. Активирането на една такава връзка предизвиква възпроизвеждане на съдържанието им. Например, за възпроизвеждане на звуковия файл "*Musica.wav*" може да се използва следния код:

```
<P>Възпроизвеждане на звук: <A HREF="Musica.wav">Музика</A></P>
```

Възпроизвеждането на видео клип по този механизъм принципно не се различава, например:

За възпроизвеждане на видео клипа, изберете иконата:

```
<P><A HREF="video.avi"></A></P>
```

Посочените два примера могат да бъдат включени към произволна HTML страница с адресиране на наличен *.wav*, *.mov*, или *.avi* файл.

Мултимедийните документи не са само от вида на HTML. Разработеният стандарт от *Adobe* за формат на файлове **PDF**, позволява включването им чрез хипервръзка в един HTML документ. Описание на този формат, както и свързаните с него приложения на *Adobe* може да се намери на следния адрес:

<http://www.adobe.com/acobat/>

Обръщение към съдържанието на един Adobe-PDF документ се осъществява по стандартната конвенция за хипервръзки. Неговото изобразяване, обаче, зависи от разширените (Plug-Ins) възможности на конкретния Web клиент, а не от HTML езика (виж частта "WWW КЛИЕНТ - NETSCAPE NAVIGATOR", "[Разширение на възможностите - Plug-Ins](#)"). Следният пример илюстрира обръщение към документ в PDF формат:

```
<A HREF="Filename.pdf">
```

В таблица са представени примери за възможното приложение на елемента *<A>*.

Връзка	Пример
Преход в рамките на страницата	
Етикет на точка от страницата	
Преход към външна страница	

Преход към точка от външна страница	
Преход към друг Web site	
Връзка с електронна поща	
Връзка с UseNet	
Връзка към FTP сървър	
Връзка с Gopher услуга	
Активиране на Telnet връзка	
Връзка към база данни Wais	

Независима каскада от стилови формати в HTML 4.0 (Cascading Style Sheets Level 1)

Новост в HTML 4.0 е възможността да се раздели съдържанието на една HTML страница от форматните спецификации за нейното представяне във Web пространството. Основните препоръки на W3C за HTML 4.0 за поддържане на **CSS1** се обслужват от четвърто поколение браузъри на *Microsoft Internet Explorer 4.x* и *Netscape Communicator 4.x*. W3C задава препоръките на механизма за интерпретиране на стиловите формати в HTML 4.0 и как този механизъм да бъде обвързан със съдържанието на един документ.

META дефиниция

При използване на езика за описание на стилови формати **CSS1** в заглавната част на **HTML** файла се добавя **<META>** дескриптор със следното съдържание:

```
<META http-equiv="Content-Style-Type" content="text/css">
или
Content-Style-Type: text/css
```

Свързването на стилови формати с HTML документ може да се приложи по три начина:

- независим файл (**.css**) с описание на стиловите формати;
- вмъкване на стиловите формати в тялото на HTML файла;
- използване на елемент **<STYLE> . . . </STYLE>**.

LINK

Първият подход е по-перспективен и практичен. Той дава възможност да бъде наложен определен стил на оформяне на поредица от документи, които обикновено са съставна част от един цял Web Site. Администратора на Web сървъра (*Webmaster*) проектира поредицата от стилови формати и ги разполага във текстов файл с разширение **.css**, достъпен за всички потребители на сървъра. Всеки, който създава своя HTML страница може да ползва това описание чрез вмъкване на команда **<LINK>** в кода на страницата, например:

```
<LINK REL=STYLESHEET HREF="http://www.mulmed.bg/sheets.css" TYPE="text/css">
```

STYLE

Вторият подход изисква вмъкването на стиловите дескриптори във всеки HTML файл, с което може да се постигне разнообразие в оформянето на отделни страници. Описанието на поредица от стилови формати, валидни в рамките на една HTML страница, се извършва чрез команда <STYLE>, която е от непразен тип. Командата и нейните атрибути се разполагат между командите <HEAD> и <BODY>.

```
<HTML>
  <HEAD>
    Заглавна част и мета-описатели
  </HEAD>
  <STYLE TYPE="text/css">
    <!--
      Стиливи дефиниции
    -->
  </STYLE>
<BODY>
  Съдържание на документа
</BODY>
</HTML>
```

Общият вид на командата за описание на стилови дефиниции е следния:

```
<STYLE [TYPE = mimetypes] [MEDIA = name_media_list] [lang = language_code] [dir
= LTR | RTL]>
<!--
selector1{property1:setting1; property2:setting2; ... }
selector2{property1:setting1; property2:setting2; ... }
. . . .
-->
</STYLE>
```

където:

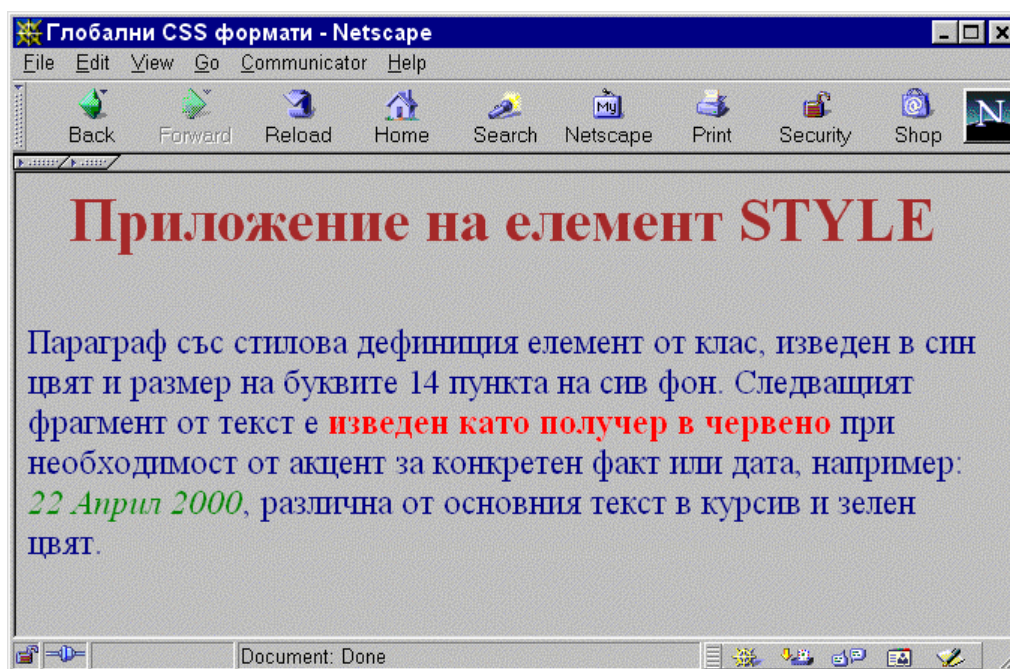
- **TYPE** = mimetypes - атрибут задаващ символен низ, с който се определя езиковата интерпретация на стиловите дефиниции, например за CSS1 това е "text/css".
- **MEDIA** = name_media_list - атрибут чрез който се определя устройството или списъкът от устройства, разделени със запетая, за които се отнасят стиловите формати. Списъкът от допустими устройства е следния:
 - *screen* - изходът е ориентиран към нестраниран документ, например екрана на монитора (подразбираща се стойност);
 - *print* - изходът е ориентиран към странирани документи, например предварителен преглед преди печат (*print preview mode*).
 - *projection* - изход, ориентиран за прожектиране.
 - *speech* - изход, ориентиран към работа със звуков синтезатор.
 - *all* - изход, ориентиран към всички видове устройства.
- **lang** = language_code - атрибут, с помощта на който се указва езиковия код на някой от естествените езици. Пълно описание на кодовете е публикувано в документа [RFC1766](#).
- **dir** = LTR | RTL - атрибут, с който се указва посоката на разположение на текста (от ляво да дясно или от дясно на ляво).
- **selector** - може да бъде:
 - HTML елемент;
 - *class* - име на клас от елементи;

- *id* - идентификатор за определен елемент.
- **property** - свойство.
- **setting** - значение (стойност).
- **<!-- -->** - огражда елемента **<STYLE>** в коментар за преодоляване на неудобството на клиенти, които не интерпретират стилови формати.

Пример:

```
<html>
<head>
  <title>Глобални CSS формати</title>
  <style type="text/css">
    <!--
      BODY {background-color:silver}
      H1 {color:brown}
      P.bigblue {color:navy; font-size:14pt}
      .price {font-style:italic; color:green}
      #SDC {font-weight:bold; color:red}
    -->
  </style>
</head>
<body>
<center><h1>Приложение на елемент STYLE</h1></center>
<br clear="all">
<p class="bigblue">Параграф със стилова дефиниция елемент от клас, изведен в син
цвят и размер на буквите 14 пункта на сив фон. Следващият фрагмент от текст
е&nbsp;<span id="SDC">изведен като получер в червено</span> при необходимост от
акцент за конкретен факт или дата, например:&nbsp;<span class="price">22 Април
2000</span>, различна от основния текст в курсив и зелен цвят.
</body>
</html>
```

На Екран 6.0 е показан резултатът от интерпретацията на кода. Примерът е изпълним директно от сървъра при проследяване на [тази връзка](#).



Екран 6.0 - Приложение на елемента STYLE

Дефиниции

Стиловите дефиниции се прилагат за HTML елементи, които управляват текстовия поток, например `<H1>`, `<H2>`, `<P>`, ``, `` или ``. Всяка стилова дефиниция формира едно *правило*, което се състои от *селектор* (HTML елемент) и дефиниция (стилови формати). Стиловата дефиниция съдържа две части - свойство и значение.

```
HTML tag {property name:value}
```

например:

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

където,

- **color** е наименование на свойството цвят;
- **blue** е значение за това свойство (син).

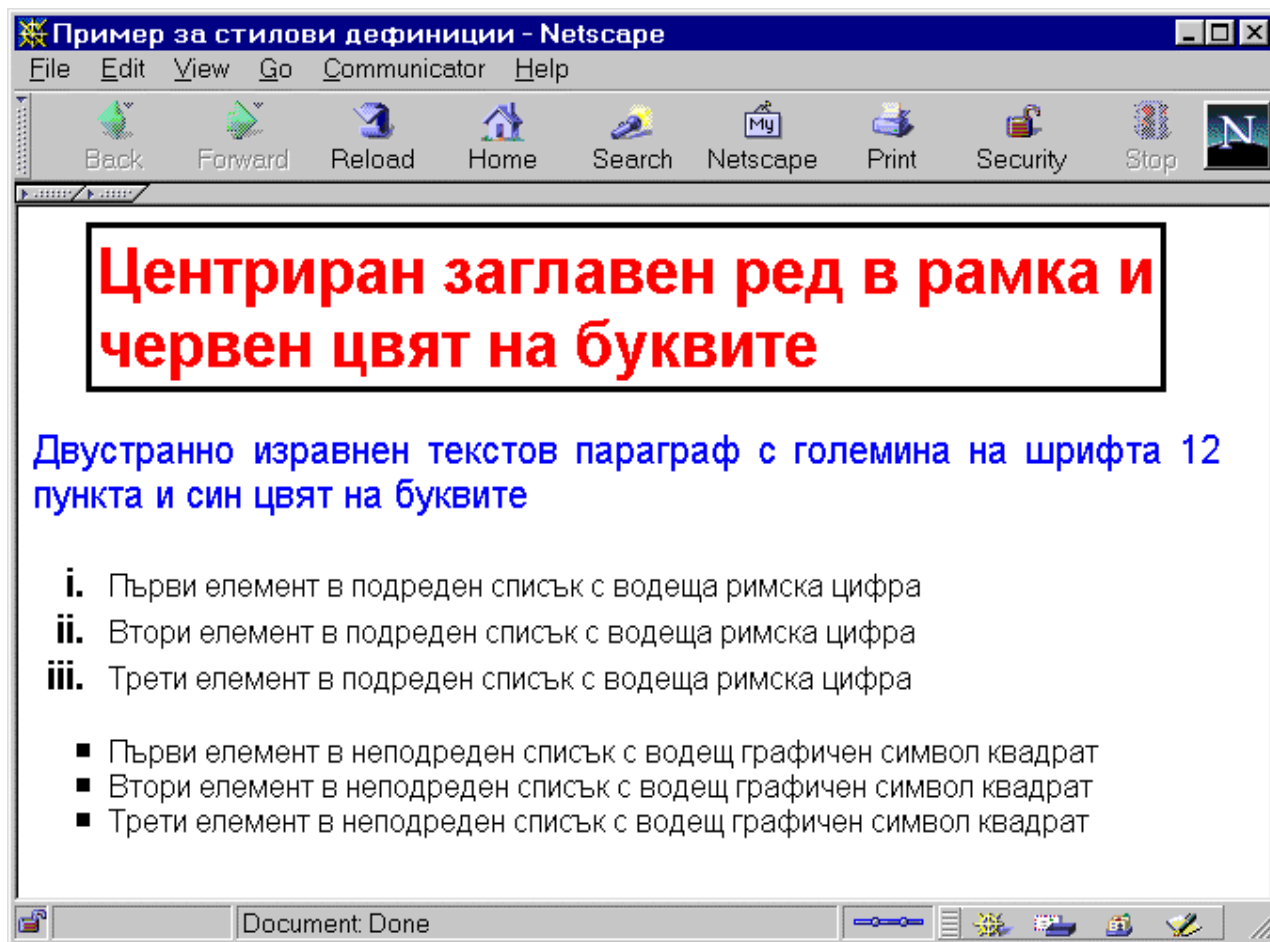
Стиловата дефиниция от примера се отнася за всички HTML елементи `H1` употребени в текущия документ и предизвиква извеждане на свързания с елемента текст с букви в син цвят.

Следващият пример представя комбинираното използването на различни стилови дефиниции в рамките на един HTML документ. Отделните стилови свойства и техните допустими значения са дадени в отделна [таблица](#).

```
<HTML>
<HEAD>
  <TITLE>Пример за стилови дефиниции</TITLE>
  <STYLE type="text/css">
    <!--
      H1 {border-width: 1; border: solid; text-align: center; color: red}
      P  {text-align: justify; color: blue; font-size: 12pt}
      OL LI {list-style-type: lower-roman; font: bold normal 12pt times, serif}
      UL LI {list-style-type: square}
    <--
  </STYLE>
</HEAD>
<BODY>
  <H1>Центриран заглавен ред в рамка и червен цвят на буквите</H1>
  <P>Двустранно изравнен текстов параграф с големина на шрифта 12 пункта и син
  цвят на буквите</P>
  <OL>
    <LI> Първи елемент в подреден списък с водеща римска цифра
    <LI> Втори елемент в подреден списък с водеща римска цифра
    <LI> Трети елемент в подреден списък с водеща римска цифра
  </OL>
  <UL>
    <LI> Първи елемент в неопределен списък с водещ графичен символ квадрат
    <LI> Втори елемент в неопределен списък с водещ графичен символ квадрат
    <LI> Трети елемент в неопределен списък с водещ графичен символ квадрат
  </UL>
</BODY>
</HTML>
```

Интерпретацията на HTML кода е представена на *Екран 6.1*. Стилът дефиниция се оформя в самостоятелен блок непосредствено преди **<BODY>** секцията. По този начин той добива глобално значение за съдържателната част на документа. Илюстрацията обхваща дефиниции на заглавия, текстов параграф и списъчни структури с различни стилови свойства и атрибути. Заглавният елемент **<H1>** е в рамка с дебелина 1 пиксел (*border-width: 1; border: solid*), подравняването на текста е центрирано (*text-align: center*), с червен цвят на буквите (*color: red*). Дефиницията за параграф **<P>** включва двустранно изравняване на текста (*text-align: justify*), син цвят на буквите (*color: blue*) и големина на активния шрифт 12 пункта (*font-size: 12pt*).

Стиловите дефиниции са едно нововъведение в HTML 4.0 и не всички Web браузъри поддържат и разпознават свързаните с тях свойства. Например, версиите на *Netscape Navigator* до 3.04 не поддържат стилови дефиниции. Непознатите команди се игнорират от браузърите, както и свързаните с тях атрибути. В случая командата **<STYLE>** ще бъде игнорирана, а самите стилови дефиниции ще бъдат изобразени като текст в полето на документа. За предотвратяване на този негативен ефект е полезно редовете с дефиниции да бъдат заградени в коментар (**<!-- -->**), както е показано в HTML кода по-горе. Това предизвиква "скриване" на тези редове при интерпретацията им от браузъри, които не поддържат стилови формати.



Екран 6.1 - Използване на стилови дефиниции

При формирането на стиловите правила могат да се описват отношения от типа **родител-наследник**. Това особено ясно се вижда при дефиниране на стилови парила за списъчни структури. Елементът **** задава съдържанието на компонент от списък независимо от неговата структура. Известно е, че списъците в HTML езика биват подредени, неподредени и

дефиниции. Например, ако е необходимо да бъде дефинирано различно правило за елемент **** за всеки отделен вид списък, то той се явява наследник на съответния списъчен вид. В HTML кода, посочен по-горе, за елемента **** са дадени две дефиниции, едната с родител подреден списък **** и друга с родител не подреден списък ****. От интерпретацията (Екран 6.1) се вижда, че дефиницията за подредения списък предизвиква извеждане на елементите му, предхождани от римски цифри (*list-style-type: lower-roman*) в получер нормален серифен шрифт с големина 12 пункта (*font: bold normal 12pt times, serif*). Дефиницията за не подредения списък съдържа указание за стила на водещия графичен маркер (*list-style-type: square*) - квадрат. Ефект на унаследяване на свойства се получава когато в документа се използват HTML елементи, за които липсва изрична стилова дефиниция. В този случай се унаследяват подразбиращите се значения от елемента **<BODY>**.

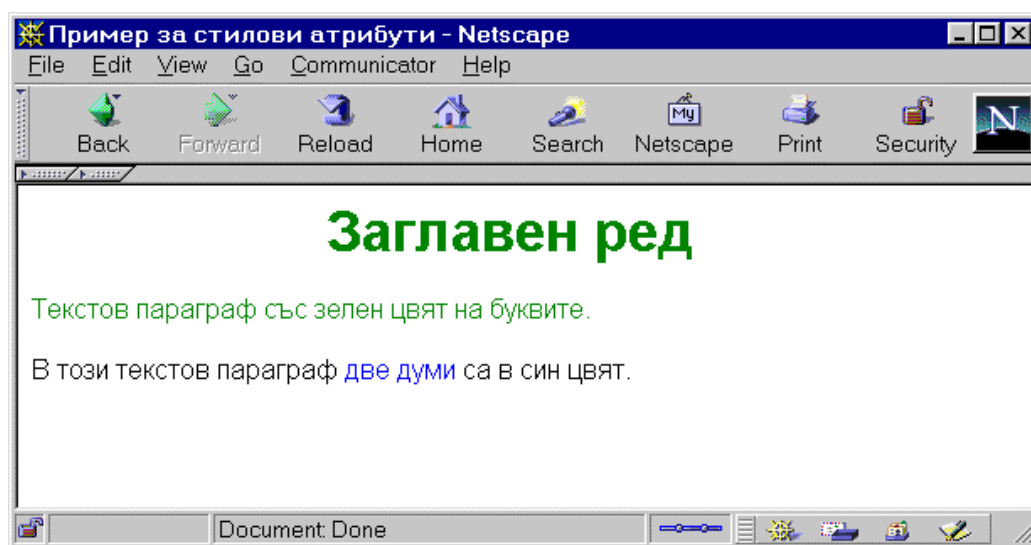
Стиловите дефиниции могат да бъдат задавани като самостоятелни елементи чрез **команда <STYLE>** или като **стилов атрибут** на HTML елемент, например **<H1>**, **<P>**, **<DIV>**, **** и **други**.

Елементът **<DIV>** се прилага за оформяне на блок от HTML команди в рамките на един документ. По принципа на унаследяване на свойствата, дефинираните в командата стилови атрибути се разпростират върху всички елементи в блока, а елементите вън от него унаследяват свойствата установени по подразбираен в **<BODY>**.

Следният пример илюстрира използването на стилови атрибути в HTML елементи.

```
<HEAD>
  <TITLE>Пример за стилови атрибути</TITLE>
</HEAD>
<BODY>
  <DIV STYLE="color: green">
    <H1 ALIGN=center>Заглавен ред</H1>
    <P>Текстов параграф със зелен цвят на буквите.</P>
  </DIV>
  <P>В този текстов параграф <SPAN STYLE="color: blue">две думи </SPAN> са в син
  цвят.</P>
</BODY>
```

Интерпретацията на кода е дадена на Екран 6.2.



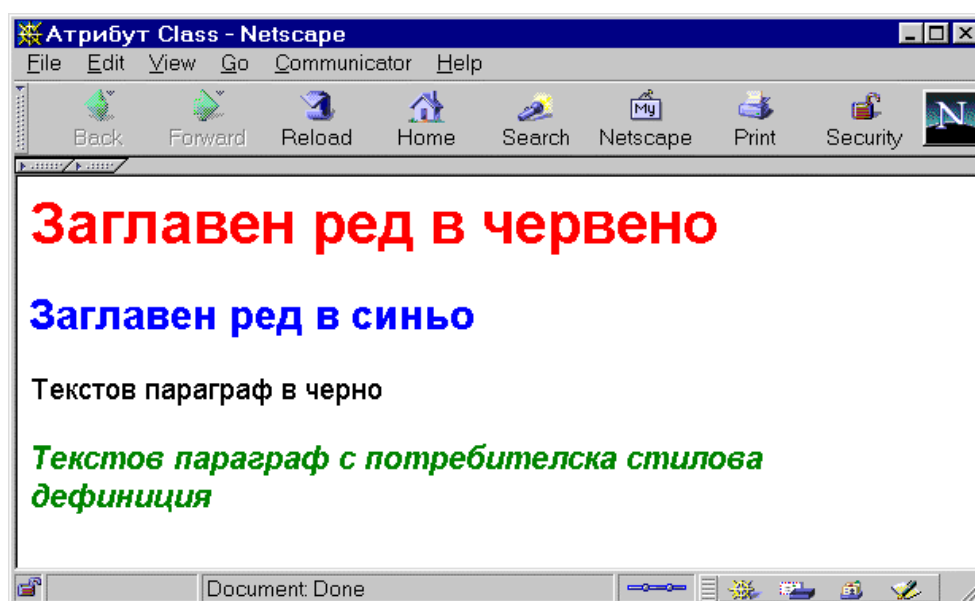
Екран 6.2 - Използване на стилови атрибути в HTML елементи

Използването на стилови атрибути вместо стилови дефиниции е оправдано за малки или чисто персонални HTML документи. В общия случай използването на командата `<STYLE>` по двата възможни начина е за предпочитане. Това улеснява поддържането на страниците в един Web Site, особено когато броят им е значителен и се търси единен стил в представянето на информацията.

С помощта на атрибутът `CLASS` могат да се задават варианти на стилови дефиниции. В примера по-долу са дефинирани стилови формати за елемента `<H1>` и `<H2>`. Отделно е дефиниран стилов формат от типа `class` без да бъде на първо време обвързан с конкретен HTML елемент (`.black {color: black}`). В секцията `<BODY>` тази дефиниция се използва за елемента параграф (`<P>`), като със същия успех може да се ползва и за произволен друг елемент.

```
<HTML>
<HEAD>
  <TITLE>Атрибут Class</TITLE>
</HEAD>
<STYLE type="text/css">
  H2.blue {color: blue}
  H1.red {color: red}
  .black {color: black}
  #mydef {color: green; font: bold italic 12pt times, serif}
</STYLE>
<BODY>
  <H1 CLASS=red>Заглавен ред в червено</H1>
  <H2 CLASS=blue>Заглавен ред в синьо</H2>
  <P CLASS=black>Текстов параграф в черно</P>
  <P ID=mydef>Текстов параграф с потребителска стилова дефиниция</P>
</BODY>
</HTML>
```

Дефиниция от вида `#mydef {color: green; font: bold italic 12pt times, serif}` определя потребителско правило и може да се прилага в текста на документа чрез обръщение към неговия идентификатор, например - `<P ID=mydef>`. Интерпретацията на този HTML код е дадена на Екран 6.3.



Екран 6.3 - Силови дефиниции с Class атрибут

Свойства

В таблица са представени свойствата на стиловите дефиниции и допустимите им атрибути. Част от тях са представени като размери в различни мерни единици:

- **em** - процент спрямо размера на шрифта (*0.5em*);
- **ex** - процент спрямо размера на буквата x (*1ex*);
- **px** - процент спрямо разделителната способност на изображение в пиксели (*12px*);
- **%** - процент спрямо общото пространство (*50%*);
- **pt** - размер на шрифт в пунктове (*12pt*);
- **in** - размер за разстояния в инчове (*1in = 2.54cm*);
- **cm** - размер за разстояния в сантиметри (*1cm*);
- **mm** - размер за разстояния в милиметри (*10mm*);
- **pc** - размер за шрифт в picas.

Наименование	Описание	Дефиниция	Пример
<i>font-family</i>	Име и фамилия на шрифта	[[<name> <family>],*]	P {font-family: courier, helvetica, sans-serif }
<i>font-style</i>	Стилов формат на шрифта	normal italic oblique	H1, H2, H3 {font-style: italic }
<i>font-variant</i>	Големина на шрифта	normal small-caps	H3 { font-variant: small-caps }
<i>font-weight</i>	Дебелина на шрифта	normal bold bolder lighter 100 200 300 400 500 600 700 800 900	EM {font-weight: bolder }
<i>font-size</i>	Размер на шрифта	< xx-small x-small small medium large x-large xx-large> < larger smaller> <taille> <процент%>	P { font-size: 12pt; } EM { font-size: 150%; }
<i>font</i>	Шрифт и неговите свойства	[<font-style> <font-variant> <font-weight>]?<font-size> [/ <line-height>]?<font-family>	P { font: 12pt/14pt sans-serif } P { font: normal small-caps 120%/120% fantasy }
<i>color</i>	Цвят на текста	aqua black blue fuchsia gray green lime maroon navy olive purple red silver teal white yellow	P { color: red } EM {color: rgb(255,0,0)}
<i>background-color</i>	Цвят за фона на екрана	<цвят> transparent	H1 { background-color: #F00 }
<i>background-image</i>	Графика за фон на екрана	<url> none	BODY { background: url (http://www.svr.bg/gif/logo.gif) }
<i>background-repeat</i>	Схема на повторение на	repeat repeat-x repeat-y o-repeat	BODY { background-image: url(sky.gif); }

	графиката за фон на екрана		background-repeat: repeat-y; }
background-attachment	фиксиран или подвижен фон	scroll fixed	BODY { background: red url(deckblue.gif); background-repeat: repeat-y; fixed;}
background-position	Разположение на графиката за фон спрямо горния ляв ъгъл на прозореца	[<размер> <процент>] {1,2} [top center bottom] [left center right]	BODY { background:url (http://www.svr.bg/marta.jpg) right top }
background	Фон на екрана и неговите свойства	<background-color> <background-image> <background-repeat> <background-attachment> <background-position>	P { background: url(http://www.svr.bg/marta.jpg) green 50% repeat fixed }
word-spacing	Разстояние между думите	normal <размер>	H1 { word-spacing: 0.4em}
letter-spacing	Разстояние между буквите	normal <размер>	EM { letter-spacing: 0.1em }
text-decoration	Атрибути на текста	none [underline overline line-through blink]	A:link, A:visited, A:active {text-decoration: none }
vertical-align	Вертикално подравняване на текста	baseline sub super top text-top middle bottom text-bottom <%>	EM {vertical-align: sub}
text-transform	Трансформиране на текст голями, малки букви или първа голяма буква	capitalize uppercase lowercase none	H2 { text-transform: uppercase }
text-align	Подравняване на текста	left right center justify	P { text-align:justify}
text-indent	Отместване на първата линия	<размер> <процент>	P { text-indent: 3em }
line-height	Разстояние между две линии	normal <число> <размер> <процент>	DIV { line-height: 1.2; font-size: 10pt }
margin-top	Горен маргин	<размер> <процент> auto	H2 { margin-top: 3px }
margin-right	Десен маргин	<размер> <процент> auto	H3 { margin-rigth: 3px }
margin-bottom	Долен маргин	<размер> <процент> auto	H4 { margin-bottom: 3px }
margin-left	Ляв маргин	<размер> <процент> auto	H5 { margin-left: 3px }

<i>margin</i>	Определя разположението на всички маргини	<размер> <процент> auto] {1,4}	BODY { margin: 1em 2em 3em }
<i>padding-top</i>	Запълване отгоре	<размер> <процент>	H1 { padding-top: 3px }
<i>padding-right</i>	Запълване отдясно	<размер> <процент>	H1 { padding-right: 3px }
<i>padding-bottom</i>	Запълване отдолу	<размер> <процент>	H1 { padding-bottom: 3px }
<i>padding-left</i>	Запълване отляво	<размер> <процент>	H1 { padding-left: 3px }
<i>padding</i>	Групира всички стойности на запълването в една команда	[<размер> <процент>]{1,4}	H1 { padding: 1em 2em }
<i>border-top- width</i>	Дебелина на горната рамка	thin medium thick <стойност>	H1 { border-top-width: 0.5em }
<i>border-right- width</i>	Дебелина на рамката в дясно	thin medium thick <стойност>	H1 { border-right-width: 0.5em }
<i>border-bottom- width</i>	Дебелина на долната рамка	thin medium thick <стойност>	H1 { border-bottom-width: 0.5em }
<i>border-left- width</i>	Дебелина на рамката в ляво	thin medium thick <стойност>	H1 { border-left-width: 0.5em }
<i>border-width</i>	Дебелина на рамка	[thin medium thick <стойност>]{1,4}	H1 { border-width: thin }
<i>border-color</i>	Цвят на рамка	<ЦВЯТ>{1,4}	H1 { border-color: red }
<i>border-style</i>	Вид на рамката	none dotted dashed solid double groove ridge inset outset	#xy34 { border-style: solid dotted }
<i>border-top</i>	Свойства за рамка отгоре	<border-top-width> <border-style> <ЦВЯТ>	H1 { border-top: thick }
<i>border-right</i>	Свойства за рамка отляво	<border-right-width> <border-style> <ЦВЯТ>	H1 { border-right: 1em }
<i>border-bottom</i>	Свойства за рамка отдолу	<border-bottom-width> <border-style> <ЦВЯТ>	H1 { border-bottom: thick solid red }
<i>border-left</i>	Свойства за рамка отляво	<border-left-width> <border-style> <ЦВЯТ>	H1 { border-left: thick solid red }
<i>border</i>	Свойства за рамка	<border-width> <border-style> <ЦВЯТ>	P { border: solid red }
<i>width</i>	Размер на текст или графика в широчина	<размер> <процент> auto	logo.ico { width: 100px }
<i>height</i>	Размер на текст или графика в височина	<размер> auto	logo.ico { height: 100px }

<i>float</i>	Изравняване на елемент в ляво или в дясно	left right none	logo.ico { float: left; margin-left: 0; }
<i>display</i>	Място за изобразяване на елемент	block inline list-item none	P { display: block }
<i>white-space</i>	Вид на празните позиции	normal pre nowrap	PRE { white-space: pre }
<i>list-style-type</i>	Вид на предхождащия списък елемент	disc circle square decimal lower-roman upper-roman lower-alpha upper-alpha none	OL { list-style-type: lower-roman }
<i>list-style-image</i>	Заместване на водещия графичен символ в списък със съответното графично изображение	<url> none	UL { list-style-image: url(bull.gif) }
<i>list-style-position</i>	Външно или вътрешно поставяне на водещия в списъка маркер	inside outside	UL { list-style-position: outside }
<i>list-style</i>	Включва всички свойства за оформяне на списъчни атрибути	(disc circle square decimal lower-roman upper-roman lower-alpha upper-alpha none) <url> none inside outside	UL { list-style: upper-roman inside }

Компютърна графика (Computer graphics)

Компютърната графика може да бъде разделена на две категории:

- растерна графика** - (*bit-map graphics*) този вид графика представлява карта от битове. Едно изображение **bit-map** е разделено на фрагменти или битове с информация за цвета. Всеки бит е наречен пиксел (**pixel**) и представлява елемент от изображението. В изображението пикселите се определят хоризонтално и вертикално. Едно изображение **bit-map**, което се състои от 400 пиксека в хоризонтален размер и 300 пиксела във вертикален размер притежава 120 000 пиксела, като всеки един от тях съдържа значение за цвят. Цветовата палитра на изображение в растерна графика може да варира ([Таблица 6.2](#)). Едно 8-битово изображение може да съдържа до 256 цвята, т.е. всеки отделен пиксел може да съдържа някой от тези 256 цвята. Едно 24-битово цветно изображение съдържа повече от 16.7 милиона цвята изобразяеми за всеки отделен пиксел. Колкото цветовата палитра е по-богата, толкова размера на файла с **bit-map** изображението е по-голям. Сканираните фотографски снимки представляват **bit-map** изображения. [Adobe Photoshop](#) е може би най-добрият **bit-map**

редактор. Притежава значителни възможности за обработка на сканирани изображения или за създаване на собствени такива. Изискванията на този редактор към ресурсите на компютърната система са значителни. Пример за графичен редактор с по-ограничени изисквания, но с богати възможности е [Paint Shop Pro](#). Версията му shareware е достъпна за копиране по Internet.

- **векторна графика** - (*Vector graphics*) представлява последователност от компютърни команди за "изчертаване" на изображението. Векторната графика се разделя на графични примитиви, като линии, окръжности, дъги, правоъгълници и други. Цветният компонент се задава като запълване или шриховка. Размерът на файла с векторна графика винаги е по-малък от съответния файл със същото изображение като **bit-map**. Основното предимство на векторната графика е нейната мащабируемост. Промяна в размерите на изображението не довежда до промяна в неговото качество при визуализация. Същото не важи за едно **bit-map** изображение. При разширяване на неговите стандартни размери се проявява ефект на пикселизация на изображението с помътняване на контурите и мозаечен ефект при визуализация. Векторните изображения са подходящи за представяне на фирмени знаци, диаграми и схеми. Списъкът от редактори за векторна графика е доста дълъг. Този вид редактори са част от т.н. *Drawing Systems* или системи за изчертаване. Редактор с изключителни възможности в това отношение и в значителна степен познат е [CorelDRAW](#). Изискванията на този редактор към ресурсите на компютърната система са значителни. С не по-малко изисквания и възможности е [Adobe Illustrator](#), редактор ориентиран предимно към компютърната илюстрация.

Брой на битовете	Брой на цветовете
1-bit	2 цвята
2-bit	4 цвята
4-bit	16 цвята
8-bit	256 цвята
16-bit	65.5 хиляди цвята
24-bit	16.7 милиона цвята

Таблица 6.2 - Схема на цветовата наситеност

Каскада от стилови формати за позициониране на HTML елементи (Cascading Style Sheets-Positioning - CSS-P)

CSS-P представлява независим опит за стандартизиране на позиционирането на HTML елементи в полето на документа. Въвеждането на стилови формати в [DHTML](#) чрез [CSS1](#) създаде базата на независимост на описанието на HTML страницата от нейното изобразяване в полето на браузъра. Въвеждането на стилови формати за позициониране (**CSS-P**) на HTML елементи се осъществява чрез разширяване на възможностите на CSS1 с цел прецизното позициониране на компонентите в една страница. Появата на стиловите формати се свързва с появата на четвъртото поколение Web браузъри и преди всичко с *Netscape Navigator 4.x*. Към момента на написване на тези редове не всички атрибути на CSS1 и CSS-P са адекватно интерпретирани от *Internet Explorer 4.x*. Оригиналната W3C спецификация, която третира стиловите формати за позициониране, може да се намери на следния адрес:

<http://www.w3.org/pub/WWW/TR/WD-positioning>

Коректното позициониране на различни компоненти в полето на визуализация на документа винаги е създавало проблеми на дизайнерите на Web страници. Коректното позициониране включва създаване на визия, която слабо или изобщо не се влияе от разделителната способност и размера на прозореца, с които клиентът изобразява съдържанието на страницата.

Позиционирането включва не само задаване на точка от полето на страницата, в която или към която определен елемент ще бъде свързан, а и параметри, чрез които се определя кои негови части ще покриват частично или изцяло части на друг елемент, кои части ще бъдат скрити, кои ще бъдат прозрачни. Най-добри резултати, за получаване на динамична HTML страница, се постигат с използване на възможностите на езика **JavaScript** като скриптово допълнение към HTML (**DHTML** - динамична версия на стандартния HTML). Стилите формати за позициониране се управляват по-прецизно и става възможно динамичното преместване на графични компоненти по полето на страницата.

Параметрите за дефиниране на позиционирането на HTML компоненти се задават чрез команда `<STYLE>`. В една стилска секция могат да бъдат дефинирани множество слоеве (*layer*), предхождани от етикет (`#label-id { ... }`). Втора възможност е описание на позиционирането със средствата на стилския атрибут към определен набор от HTML елементи. В стилската спецификация могат се използват всички дефинирани от нормата [CSS1](#) елементи: стилски идентификатор, клас атрибут, стилска дефиниция в елементите `<P>`, `<DIV>`, `` и др .

Контролът върху слоя и позиционирането се осъществява от определен брой избираеми параметри:

- **position:** *absolute|relative;* - абсолютно или относително позициониране на слоя в документа. Абсолютното позициониране се определя спрямо указани абсолютни координати. Относителното позициониране се определя спрямо текущата позиция в документа.
- **top:** *Ypx;* **left:** *Xpx;* - задава отместването на слоя в брой пиксели ([px](#)) спрямо горния ляв ъгъл на екрана при абсолютно позициониране или спрямо текущата позиция в документа при относително позициониране.
- **color:** *color-name;* - задава [наименованието](#) на цвета за текста, изобразен в полето на слоя.
- **background-color:** *color-name;* - определя цвета на фона на слоя по неговото [наименование](#).
- **border-width:** *Npx;* - широчина на рамката около слоя, зададен в пиксели ([px](#)).
- **border-color:** *color-name;* - цвят на рамката, зададен чрез неговото [наименование](#).
- **border-style:** *style-name;* - определя стилското оформяне на рамката около слоя по неговото [наименование](#) (*none | dotted | dashed | solid | double | groove | ridge | inset | outset*).
- **padding:** *N%;* - определя в проценти празен контур във вътрешната част на слоя.
- **source-include:** *url("URL");* - възможност за вмъкване в полето на слоя на съдържанието на външен HTML файл (параметърът не е част от предписанията на W3C и се интерпретира само от Netscape 4.x). Външният файл може да съдържа код на Javascript или да дефинира собствени слоеве, които в контекста на слоя родител ще бъдат третирани като дъщерни. Използването на този параметър е много подходящо в случаите, когато в съдържанието на една HTML страница за определена рубрика или секция се извършват ежедневни промени.
- **width:** *Npx|N%;* - определя в пиксели или в проценти от размера на слоя широчината на полето, в което се разполага съдържанието. Изобразяването на графични елементи

не се влияе от този размер ако широчината на изображението надвишава указаната стойност. Ако параметърът е изпуснат съдържанието се разполага до границите на слоя.

- **height:** *Npx|N%*; - определя в пиксели или в проценти от размера на слоя началната височина на отрязана част от слоя. Съдържанието на слоя се разполага независимо от този размер, който има значение при управлението на дъщерни слоеве.
- **clip:** *rect(left, top, right, bottom)*; - определя размерите на правоъгълна видима част от слоя. Размерите се задават в пиксели спрямо горния ляв ъгъл на слоя. Ако този параметър не е зададен, се приемат стойностите указани в *width* и *height*. Ако и те са изпуснати, горният ляв ъгъл на слоя се определя от неговата дефиниция, а размера от обема на съдържанието.
- **z-index:** *N*; - задава поредния номер (*N*) на слоя в стека. Използва се при управление на превключването между различни слоеве.
- **visibility:** *show|hide|inherit*; - атрибут за управление на видимостта на слоя. Слойт е видим при указано за параметъра значение *show* и невидим при *hide*. Унаследяване на атрибутите от родителския слой се установява чрез *inherit*.
- **background-image:** *url("URL")*; - задава [URL](#) адрес (абсолютен или относителен) на графичен файл, който ще послужи за тапет на слоя.

Следващите примери илюстрират приложението на стиловите атрибути за позициониране на HTML елементи. Използвано е разнообразие от изразни средства за демонстрация на възможностите на тази техника. В кода, показан по-долу, е използвана стилова секция за описание на слоевете *lr1* и *lr2*, дефиниция от типа [CLASS](#) се използва в комбинация с елемента `` за динамична промяна в стиловата дефиниция при извеждане на един рен на екрана на потребителя.

Позициониране:

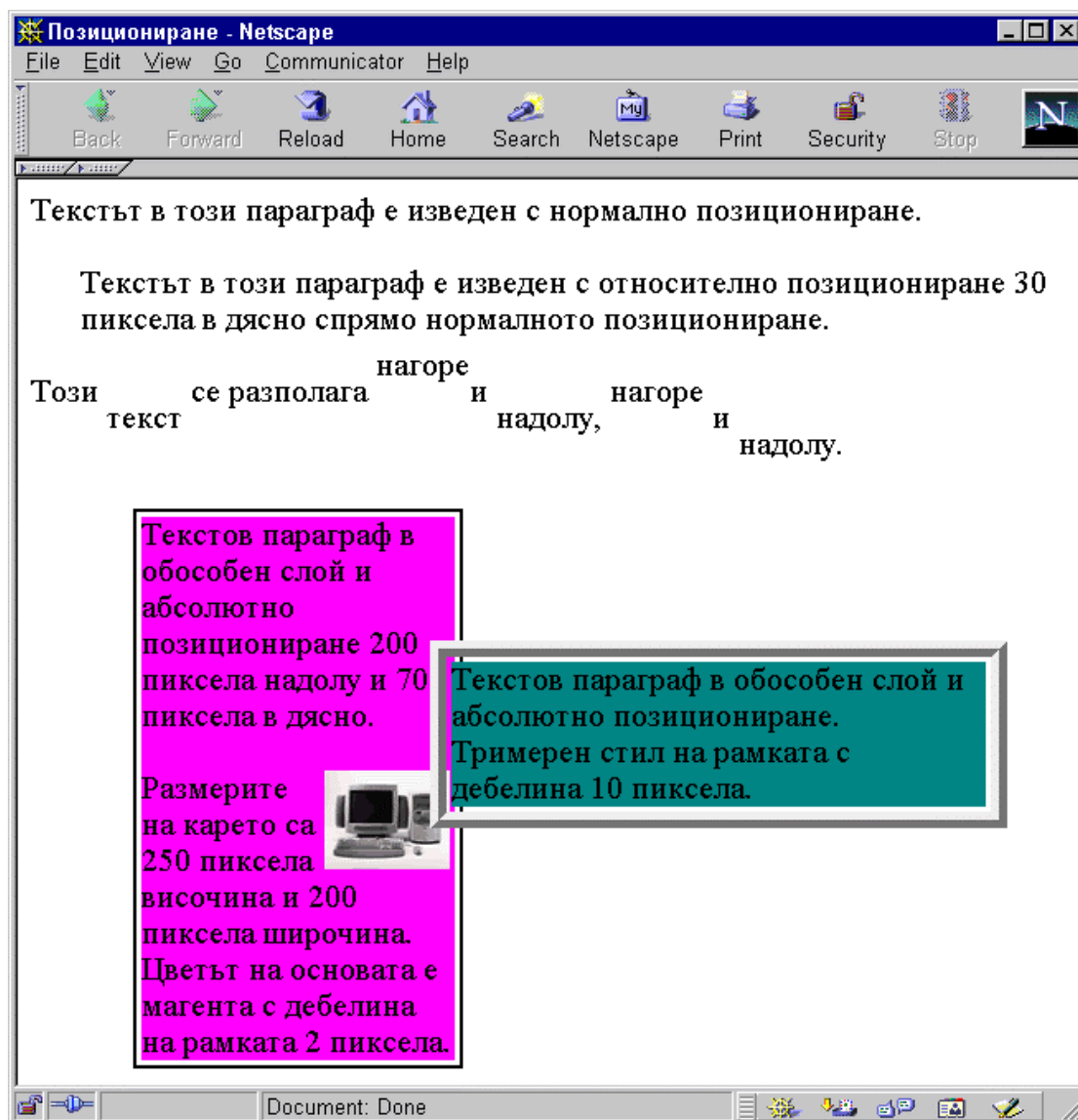
```
html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
    <title>Позициониране</title>
    <STYLE TYPE="text/css">
      <!--
      #lr1 {
        position:absolute; background-color:magenta;
        top:200px; left:70px; width:200px;
        height:250px; border-width:2px;
      }
      #lr2 {
        position:absolute; background-color:teal;
        top:280px; left:250px; width:350px;
        height:50px; border-width:10px;
        border-color:silver; border-style:ridge;
      }
      all.UP {position:relative; top:-10pt;}
      all.DOWN {position:relative; top:10pt;}
      -->
    </STYLE>
  </head>
  <BODY>
    <P>Текстът в този параграф е изведен с нормално позициониране.</P>
    <P STYLE="position:relative; left:30px;">Текстът в този параграф е
      изведен с относително позициониране 30 пиксела в дясно спрямо
      нормалното позициониране.</P>
    <P>Това е <SPAN CLASS=DOWN>малък </SPAN> <SPAN>се забелязва </SPAN>
```

```

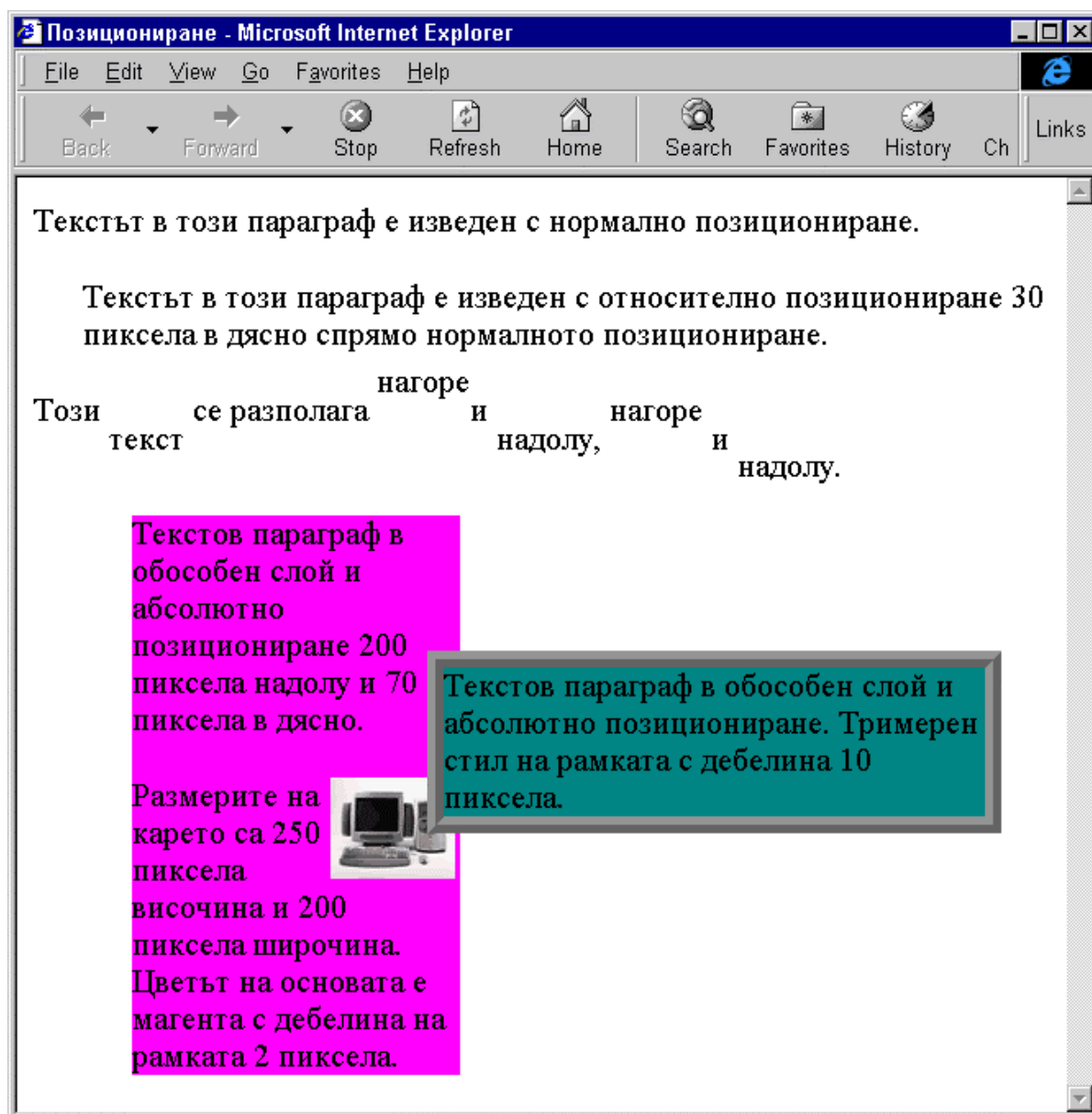
CLASS=UP>нагоре</SPAN>и <SPAN CLASS=DOWN>надолу, </SPAN>нагоре <SPAN
CLASS=DOWN>и <SPAN CLASS=DOWN>надолу.</SPAN></SPAN></P>
<DIV ID=lr1>
<P>Текстов параграф в обособен слой и абсолютно позициониране 200
пиксела надолу и 70 пиксела в дясно.</P> <IMG SRC="images/computer.gif"
align=right> <P>Размерите на карето са 250 пиксела височина и 200 пиксела
широчина. Цветът на основата е магента с дебелина на рамката 2
пиксела.</P>
</DIV>
<DIV ID=lr2>
<P>Текстов параграф в обособен слой и абсолютно позициониране.
Тримерен стил на рамката с дебелина 10 пиксела.</P>
</DIV>
</BODY>
</html>

```

Резултатът от изпълнението на кода е изведен на *Екран 6.4-1* с помощта на Web браузъра *Netscape Navigator 4.5*. Същият пример, изпълнен с *Internet Explorer 4.01* (*Екран 5.4-2*) дава различен резултат. Примерът е достъпен за изпълнение и в директен режим: [Пример позициониране](#).



Екран 6.4-1 - Позициониране: абсолютно и относително с Netscape 4.5



Екран 6.4-2 - Позициониране: абсолютно и относително с IE4

Позициониране на HTML елементи с команда <LAYER>

Web клиентът Netscape 4.x поддържа алтернативно средство за позициониране на HTML елементи, осъществено чрез команда **<LAYER>**. Елементите за позициониране и съдържание на слоя се поместват в секцията **<BODY>** на HTML документа. Дефиницията на слоя включва атрибутите за позиционирането му и елементи за формиране на неговото съдържание. Командата **<LAYER>** в качеството си на HTML елемент съдържа списък от атрибути за определяне на позиционирането на слоя сходни по значение и функция на съответните атрибути от стиловите дефиниции на [CSS-P](#). Различията са в синтаксиса на съвпадащите атрибути и наличие на няколко атрибута, които нямат аналог в [CSS-P](#). Абсолютното позициониране слой се определя от команда **<LAYER>**, относителното позициониране се постига с команда **<ILAYER>**. Допустимият списък от атрибути за командата е следния:

```
<LAYER ID=label TOP=Ypx LEFT=Xpx PAGEX=Xpx PAGEY=Ypx SRC="include.htm"
WIDTH=Npx|N% HEIGHT=Npx|N% clip="left, top, right, bottom" Z-INDEX=N ABOVE=layer-
ID BELOW=layer-ID VISIBILITY=SHOW|HIDE|INHERIT BGCOLOR=color BACKGROUND="image-
file" OnMouseOver="function();" OnMouseOut="function();" OnFocus="function();"
OnBlur="function();" OnLoad=function();">
.....
HTML елементи, формиращи съдържанието на слоя
.....
</LAYER>
```

- **ID=label** - идентификатор, име на слоя. Началният символ на идентификатора трябва да бъде буква и играе роля на етикет за обръщение към слоя от HTML команда или JavaScript функция.
- **PAGEX=Xpx PAGEY=Ypx** - задава вертикалната и хоризонтална позиция в пиксели на горния ляв ъгъл на слоя спрямо включения документ. Атрибутите нямат еквивалент в CSS-P.
- **On...** - атрибутите с префикс **On** формират списък от функции за обработка на събития (*event handler*). Събитието възниква при преминаване на курсора на мишката през полето на слоя, при итеракции с клавиатурата или при зареждане на съдържанието на слоя в полето на документа. Обработката на възникнало събитие се извършва от предварително зададена функция с връщане на резултат. Функцията може да бъде външна или вмъкнат код на JavaScript.

Алтернативно представяне на съдържанието на слой за Web клиенти, които не включват интерпретация на команда `<LAYER>` може да се осигури с `<NOLAYER>`. Командата е от празен вид и има следния синтаксис:

```
<NOLAYER> ... Съдържание на слоя ... </NOLAYER>
```

В документацията ["Dynamic HTML in Netscape Communicator"](#) е представен пример за използване на команда `<LAYER>` с различна комбинация от атрибути. Примерът се намира във файл [layertgl.htm](#) и може да бъде изпълнен локално. За преглед на HTML кода след активиране на страницата от менюто на браузъра може да се избере *View|Page Source*.

Включване на графика (in-line graphic commands)

Силата на езикът HTML се състои в създаването и интегрираното представяне на мултимедийни документи, в които освен текст има графични изображения, икони, видеоклипове, звук и анимирани елементи. Понятието *in-line graphic* предполага, че Web клиентът не се нуждае от допълнителни специализирани програмни средства за визуализация на стандартизирани графични формати.

Командата `` позволява директно включване на графика (*in-line graphics*) в полето на документа. Елементът `` е команда от празен тип и не изисква затварящ елемент.

Командата `` има следния формат:

```
<IMG SRC=URL
[ALIGN=left|right|center|middle|top|texttop|baseline|bottom|textbottom]
[...]
```

[HSPACE=n] [VSPACE=n] [ISMAP] [USEMAP=URL] >

Например, командата **** извлича графиката от файла **"picture.gif"** по относителен адрес (текущата директория) и я извежда в полето на документа. Параметърът **"SRC"** е задължителен и трябва да сочи абсолютен или относителен URL адрес.

Командата **** има избираеми параметри, които специфицират начина на включване на графиката в документа.

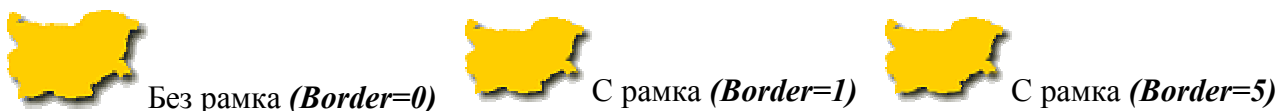
- **ALIGN** - е параметър, който определя как изображението се подравнява спрямо базовата линия на съседния на графиката текст или спрямо ограничителните линии на страницата (екрана) в рамките на текущия параграф. Ако параметърът е изпуснат подравняващото се значение е **ALIGN=bottom**.
 - **ALIGN=left** - графичният обект се подравнява в ляво, а текстът се разполага в дясно от него;
 - **ALIGN=right** - графичният обект се подравнява в дясно, а текстът в параграфа се разполага в ляво от него;
 - **ALIGN=center** - графичният обект се подравнява спрямо центъра на екрана, а текстът заема пространството около графиката;
 - **ALIGN=middle** - хоризонталната ос на обекта се подравнява вертикално спрямо базовата линия на текста;
 - **ALIGN=top** - горният край на обекта се подравнява вертикално спрямо базовата линия на текста;
 - **ALIGN=bottom** - долният край на обекта се подравнява вертикално спрямо базовата линия на текста;

Текст(*Align=top*)Текст(*Align=middle*)Текст(*Align=bottom*)

Екран 7.1 - Подравняване на текст и графика с помощта на параметър **Align**

- **ALIGN=texttop** - горните краища на обекта и текстовата линия се подравняват вертикално;
- **ALIGN=textbottom** - долните краища на обекта и текстовата линия се подравняват вертикално;
- **ALIGN=baseline** - долният край на обекта се подравнява вертикално спрямо базовата линия на текста.
- **ALT="Text"** е параметър, чрез който се задава заместващ графиката текст. Текстът се извежда на местото на графиката в полето на документа. Използването на този параметър е препоръчително в случаите, при които WWW клиентът няма графични възможности или тези му възможности временно са ограничени (чрез параметрите за настройка на Web Browser). Заместващият текст ("**Text**") трябва да съдържа кратко описание на графиката.
- **HEIGHT=n**, **WIDTH=n** са два параметъра, които спомагат за уточняване на размера в пиксели на изобразяваната графика в полето на документа. С помощта на параметрите се постига модифициране на размерите на графичните обекти по височина и/или широчина. Двата параметъра се обслужват напълно само от *Netscape*, където е допустимо и непропорционалното деформиране на графиката.
- **BORDER=n** е параметър за включване на рамка, с която се огражда графиката, като дебелината ѝ се влияе от стойността на **n** зададена в пиксели. При отсъствие на

параметъра подразбиращата се стойност е 0, а при стойност за $n=0$ графиката не се огражда. (Екран 7.2)



Екран 7.2 - Ограждане на графика с рамка

- **LOWSRC=URL** е параметър, приложим само за *Netscape*. Употребата му предизвиква извеждане в полето на документа на една начална графика, изобразяване на целия текст от документа и накрая зареждане на графичния файл от **** върху полето на началната графика. Най-подходящото приложение на атрибута е използване на графично изображение в ниска разрешаваща способност като начална графика, която ще бъде припокрита в края на зареждането на документа с оригинала. Адресът URL сочи абсолютен или относителен адрес на файл с графично изображение.
- **HSPACE=n, VSPACE=n** са два параметъра, които осигуряват възможност за оформяне на празно пространство (вертикално и хоризонтално) около графичния обект. Атрибутът **n** се задава в пиксели. В един HTML документ празните позиции се игнорират, което обективно затруднява подравняването на текст около и до графични елементи. Употребата на параметрите създава условия за добро възприемане на графичен обект заобиколен от текст. (Екран 7.3)



Графиката е подравнена в ляво (**Align=left**), текстът обхожда графиката отдясно. Около графиката е оформено поле в хоризонтална и вертикална посока (**Hspace=10, Vspace=5**)



Графиката е подравнена в дясно (**Align=right**), текстът обхожда графиката отдясно. Около графиката е оформено поле в хоризонтална и вертикална посока (**Hspace=5, Vspace=10**)

Екран 7.3 - Атрибути за подравняване на графика и текст

- **ISMAP, USEMAP=URL** са два параметъра, които се използват при "горещи" точки в полето на графичното изображение. Изобразеният графичен обект в полето на документа се използва като карта, от която са снети координати на точки и са записани във **MAP** файл. Снетите точки се използват като хипервръзки за обръщение към други документи. Елементът **ISMAP** указва, че графиката е избираема. Координатите на избрана точка от нея се изпращат на сървър за обработка, например:

```
<a href="/cgi/coord.map"><img src=pic.gif ismap></a>
```

където, **HREF** сочи адреса на **MAP** файла или **CGI** скрипта, който ще обработи данните от клиента, а в **IMG** се определя изобразената графика като избираема. Елементът **USEMAP** сочи адрес и име на **MAP** елемент в локалната машина. Описание на елемента и начина на неговото използване е дадено в раздела **"Hotspot Images"**.

WWW клиентите визуализират графични изображения, съхранени във файлове с разширения **.gif**, **.xbm** и **.xpm**. Повечето от разпространените версии, имащи графични възможности

(*Netscape, Internet Explorer*), разпознават с успех **.jpeg** и **.png** файлове за визуализацията на цветни графични изображения с високо качество.

Най-често употребяваният тип графичен формат е **.gif**. Форматът на графични файлове **GIF** (**G**raphics **I**nterchange **F**ormat) първоначално е бил разработен за потребителите на частната компютърна мрежа *CompuServe* и използват компресия от вида на LZN. Цветната палитра на графични изображения, записани в GIF формат е ограничена до 256 цвята, което го прави подходящ за рисунки, икони и графични знаци.

Текущия стандарт за *gif* формат е **GIF89a** и поддържа няколко допълнителни ефекта особено подходящи при оформяне на Web страници:

- **Interlacing** - това е свойство, при което Web браузъра изобразява *gif* файла на няколко паса като ефект на преминаване от графика с по-ниска разделителна способност към графика с по-висока разделителна способност. Използването на този ефект осигурява възможност на потребителя да добие предварителна представа за съдържанието на графиката без да дочака пълното ѝ изобразяване в полето на Web клиента. (Екран 7.4)
- **Transparency** - това е свойство, при което един от цветовете в графичното изображение е посочен като прозрачен и основата прозира през него. Използването на това свойство интегрира много добре графичното изображение във Web страницата. Създаването на прозрачен *gif* е възможно със средствата на графични редактори като [Photoshop](#), [Paint Shop Pro](#) и други. С тяхна помощ могат да се създават и *interlaced GIF* файлове. (Екран 7.4)



Екран 7.4 - Изображение във формат GIF със и без свойствата *Interlacing* и *Transparency*

- **Animation** - анимираните *gif* файлове представляват последователност от кадри, записани в един файл. Програмите за създаване на анимирани *gif* изображения, като [Microsoft GIF animator](#), позволяват параметризирането на броя на кадрите, тяхното времетраене, въвеждане на паузи за определено време, броя на цикличните изпълнения на кадрите. Възпроизвеждането на съдържанието на един анимиран *GIF файл* стартира автоматично след неговото зареждане в полето на Web клиента. (Екран 7.5)



Екран 7.5 - Анимиран GIF

Едно пълноцветно изображение записано в компресиран вид се постига с формата **JPEG** (**J**oint **P**icture **E**xperts **G**roup). Този формат е 24 битов и позволява 2^{24} или 16 777 216 цвята. Използването на формата **JPEG** е подходящ при изобразяване на многоцветни картини и цветна фотография (Екран 7.6).



Екран 7.6 - Стенописи в Рилски манастир, заснети с цифрова камера и изобразени в JPEG формат

Висококачествените представителни възможности на формата **JPEG** формират от друга страна файлове с по-голям обем в сравнение с файловете записани във формат **GIF**. Свойствата **Transparency** и **Animation** не са възможни при формата **JPEG**. Сходство със свойството **Interlacing** при **GIF** формата се постига с помощта на модифициран от скоро вариант на **JPEG** - **p-JPEG** (*progressive JPEG*).

Бъдещето на Web графичните формати е свързано с обявения нов стандарт от *World Wide Web Consortium (W3C)* от пролетта на 1996 година. Новият графичен формат **PNG** (*Portable Network Graphic*) има за цел да замести **GIF** и **JPEG** форматите. Проектът предвижда обединяване на качествата на двата графични формата, като заедно с това предоставя допълнителни възможности, свързани с независимост на формата от машинната и програмната платформа. Версиите 4.x на *Netscape* и *Internet Explorer* интерпретират този формат. Основните достоинства на формата **PNG** включват:

- многослойност на свойството **Transparency**;
- възможност за използване на свойството **Interlacing**;
- поддържа **повече от 256 цвята**;
- подобро качество на **изображението**;
- по-ниско ниво на **компресия** в сравнение с това при *jpeg*, поради прилагане на компресиращ метод без загуби;
- независимост от платформата.

Работният документ на консорциума *W3C*, описващ свойствата на графичния формат **PNG** се намира на адрес:

<http://www.w3.org/pub/WWW/TR/REC-png.html>



Екран 7.6.1 - Изображение във формат PNG

Ако на мястото на изображението от Екран 7.6.1 се появява служебната икона заместител на графика, това означава, че браузърът не поддържа PNG формат, или липсва съответният plug-in.

В [таблица 2](#) е посочен списък на графични файлове, разпознавани от болшинството WWW клиенти.

Карта на България

На тази карта на България са нанесени точки с имената на някои от по-големите градове.



Примерът има за цел да илюстрира комбинирането на графика и текст в една HTML страница. Графичното изображение е подравнено в ляво ("Align=left"), без рамка ("Border=0"), в капе с размери Height=180, Width=300 и празно пространство около карето на графиката с размери - във вертикална посока Hspace=10 и в хоризонтална посока Vspace=10.

Координатите на отбелязаните на картата градове са снети със средствата на графичен редактор и са записани в MAP файл с цел да бъдат използвани в един следващ пример.

Екран 7.7 - Смесване на графика с текст

Много често при визуализация на HTML документ, WWW клиентът показва полето на дадена графика като икона с общо предназначение, което означава, че не може да бъде намерен графичен файл със специфицирания URL адрес. Възможни са грешки поради промяна в името или в съдържанието на графичния файл след създаване на HTML документа или грешен URL адрес. Възможна причина може да се търси и в настройките на WWW клиента в секцията режими на визуализация на графични обекти.

Известно е, че графичните файлове са с голям обем. Например, графично изображение с 256 цвята, заемащо целия екран, изисква над 100К байта, докато няколко килобайта HTML текст се разполага в няколко страници на екрана. Следователно честото използване на графика в даден документ забавя визуализацията, особено ако връзката към Internet е бавна.

Първият широко разпространен WWW клиент, *Mosaic 1.0.3* не позволяваше на потребителя да прави нещо друго до пълното получаване на графиката. WWW клиентите от следващото поколение, като *Netscape Navigator* и *Microsoft Internet Explorer*, осъществяват извличането на графичните файлове във "фонов" режим, като позволяват на потребителя да се движи по документа или да осъществява преход към друг документ преди пълното получаване на графичния файл.

Списъчните структури, описани в [List Commands](#), се оформят с специален водещ символ. Този символ, генериран от съответният Web браузър може да бъде заменен от графичен по избор на потребителя. Автоматичното оформяне на списъка в този случай не може да се използва, т.е. подреждането на елементите в списъка, както и тяхното вмъкване трябва да бъде управлявано "речно". Следният HTML код е интерпретиран на Екран 7.8.

```
<HTML>
  <HEAD>
    <TITLE>bullet</TITLE>
  </HEAD>
  <BODY>
    <P>
      <IMG SRC="bullet.gif" ALT="" HEIGHT=14 WIDTH=31><B>Списъчен елемент 1</B><BR>
      <IMG SRC="bullet.gif" ALT="" HEIGHT=14 WIDTH=31><B>Списъчен елемент 2</B><BR>
      <IMG SRC="bullet.gif" ALT="" HEIGHT=14 WIDTH=31><B>Списъчен елемент 3</B>
    </P>
  </BODY>
</HTML>
```

"Ръчното" оформяне на списъка касае самостоятелното изписване на всеки ред в рамките на един параграф (`<P>`), предхождан от команда `` за поставяне на графичния списъчен разделител. Преминаването на следващ ред става с команда `
`. Текстът на всеки списъчен елемент в примера е оформен в получер стил ``. За Web браузъри без графични възможности алтернативно (`ALT`) списъчните елементи се предхождат от символа "*".

- Списъчен елемент 1
- Списъчен елемент 2
- Списъчен елемент 3

Екран 7.8 - Списък с водещ графичен елемент

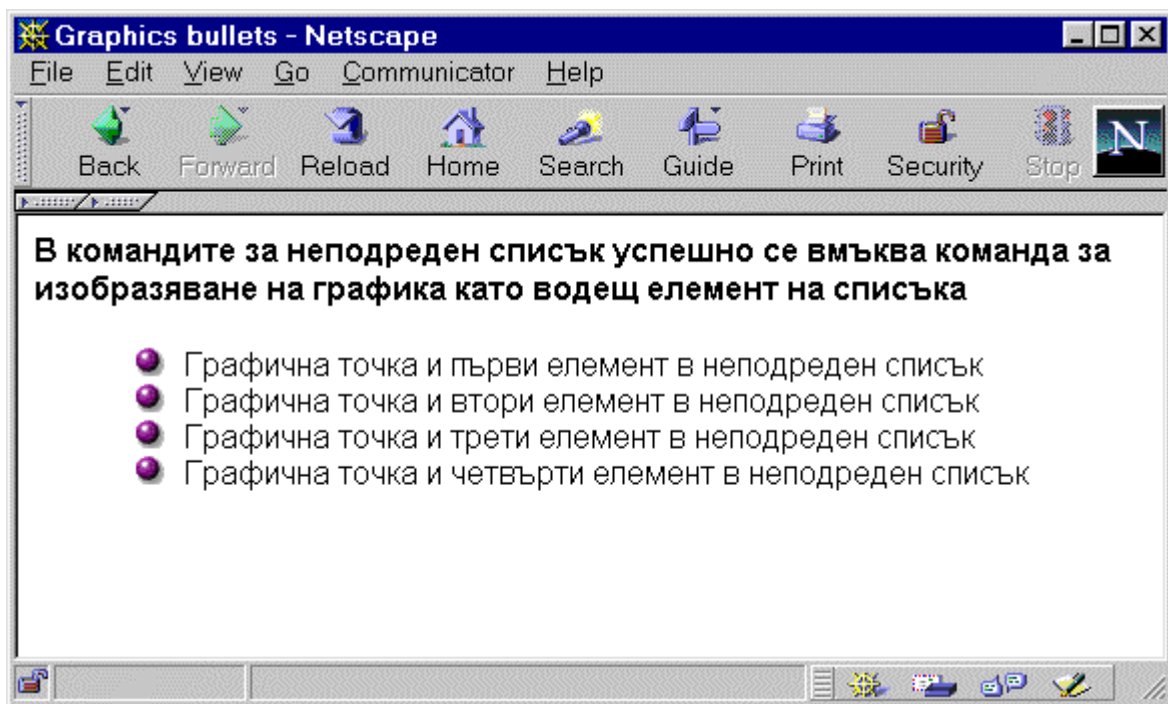
Списък от дефиниции, който наподобява неподреден списък, се оформя чрез последователност от команди `<DL>` `<DT>` `<DD>`. Ако в тялото на елемента се вмъкне команда за изобразяване на графичен компонент, то водещите символи на списъка се заменят с този компонент. Този случай е един от най-често срещаните при приложението на графични елементи във Web страниците. Следният HTML код илюстрира това приложение.


```

<HTML>
  <HEAD>
    <TITLE>Graphics bullets</TITLE>
  </HEAD>
  <BODY>
    <P><B>В командите за неопореден списък успешно се вмъква команда за изобразяване на графика като водещ елемент на списъка</B></P>
    <DL>
      <DT>
        <DD><IMG SRC="bullet.gif" ALT="-" ALIGN=BOTTOM> Графична точка и първи елемент в неопореден списък
        <DD><IMG SRC="bullet.gif" ALT="-" ALIGN=BOTTOM> Графична точка и втори елемент в неопореден списък
        <DD><IMG SRC="bullet.gif" ALT="-" ALIGN=BOTTOM> Графична точка и трети елемент в неопореден списък
        <DD><IMG SRC="bullet.gif" ALT="-" ALIGN=BOTTOM> Графична точка и четвърти елемент в неопореден списък
      </DD>
    </DL>
  </BODY>
</HTML>

```

Интерпретацията на кода е показана на Екран 7.9.



Екран 7.9 - Неопореден списък, реализиран чрез списък от дефиниции

Хоризонтални линии в една HTML страница се изтеглят чрез команда `<HR>`, като дебелината, дължината и ориентацията на линията се задава с помощта на параметрите на командата. Графично оформени линии и записани във формат на файлове *gif* или *jpg* могат успешно да заменят използването на командата. Следният фрагмент от HTML код при интерпретация извежда в полето на документа (Екран 7.10) графична линия от *gif* файл (*blueline.gif*) чрез командата ``.

```

<P><CENTER><IMG SRC="blueline.gif" ALT="- - - " HEIGHT=35
WIDTH=640></CENTER></P>

```

Изображението се извежда в самостоятелен параграф, центрирано спрямо средата на документа с указани значения за височина и дължина. За Web браузъри без графични възможности алтернативно (*ALT*) на мястото на графичната линия се извежда "- - -".

Екран 7.10 - Поставяне на графична линия в полето на документа



Графични изображения могат да бъдат използвани за създаване на хипервръзки (*<A>*) към части от документа или към други HTML документи или изобщо към друг Web site. "Горещата дума" в този случай е графичен обект изобразен чрез команда **. Когато графичен обект сочи хипервръзка, той се огражда в рамка с цветовата палитра на връзките дефинирани от атрибутите (*LINK*, *VLINK* и *ALINK*) в командата *<BODY>*. Ограждането се дължи на факта, че по подразбиране "горещата дума" в смисъла на хипервръзка се извежда с подчертаване. За да се премахне рамката трябва изрично да се укаже *BORDER=0* в командата **. Посоченият по-долу HTML код дефинира хипервръзка чрез графичен обект (книги) от началото на този параграф и сочи към раздела "Литература".

```
<A HREF="../../liter/liter.htm"><IMG SRC="books.gif" ALT="Books" HSPACE=5 VSPACE=2  
HEIGHT=51 WIDTH=51 ALIGN=LEFT></A>
```

Създаването на собствени графики изисква умения за работа с **графични редактори** като *Photoshop*, *CorelDRAW* или *Paint Shop Pro*. Естетичното, добре направеното и цветово балансирано графично изображение е изкуство, което не се отдава в еднаква степен на всеки. Дори направата на един графичен надпис не е чак толкова проста работа. Проблемът за мнозина може да се разреши, ако се копират графични елементи от достъпните по Internet Web сървъри с богат набор от **свободно предоставяни графични компоненти** Сървърът на CERN (<http://www.w3.org/hypertext/WWW/Icons>) и този на Randy Bazaar (<http://www.infi.net/~rdralph/icons/>) предоставят свободно значителен набор от най-разнообрази елементи. По амбициозни решения могат да се постигнат чрез **сканиране на фотоснимки**, постери, диапозитиви. В този случай често се налага допълнителна обработка на сканираното изображение със средствата на познатите графични редактори. Създаването на изцяло оригинални висококачествени изображения се постига най-лесно с помощта на **цифрова камера**. Заснетите кадри се прехвърлят директно в компютър в избран графичен формат. Наличието на CDRom в компютърната конфигурация позволява използването и на PhotoCD format от който лесно цифровите кадри могат да бъдат презаписани в познатите *GIF*, *JPEG* или *PNG*.

Графики с горещи точки (Hotspot Images)

Ако използваният Уеб браузър притежава графични възможности, то не са редки случаите, при които изобразената HTML страница на някой *Web site* предлага широко графично поле на екрана с надписи и фигури, които са избираеми в смисъла на хипервръзки. Подобни графични изображения се наричат **Image Map**. Техният формат може да бъде *GIF*, *JPEG* или *PNG*.

По-разширените възможности на HTML 4.0 и вече на 4.01 осигуряват създаването на графика с горещи точки - т.нар. **Hotspot Images (Clickable Images)**. От всяка точка има възможност да се осъществи хипервръзка към друг документ чрез указване на неговия [URL](#)

адрес. Предварително е необходимо да се създаде *файл-карта* - т.нар. *map-file*, в който са описани координатите на горещите точки от графичното изображение и връзките към документите, техните *URL* адреси, които трябва да се извлекат при избор на съответната гореща точка. Картата с координатите на точките може да се дефинира и с HTML елемента [<AREA>](#).

Схемата за управление на връзките зависи от това, къде се взема решение за изпълнение на хипервръзка от определена гореща точка. Ако избрана от потребител връзка зарежда свързания с нея документ чрез Web сървър, то това е управление от страна на сървъра и се нарича - *server-side image maps*. В обратния случай, управлението се осъществява от страна на Web клиента, защото избраната от потребителя гореща точка предизвиква зареждане на съответния документ непосредствено и се нарича - *client-side image maps*.

Управление от *страна на Web сървър* притежава определена специфика:

- Клиентската програма предава на сървъра координатите на избраната от потребителя точка;
- Съответният софтуер на сървъра трябва да обработва постъпващите от клиента координати като входни данни и да бъде в състояние на извежда съответстващите им *URL* адреси на изход. Съответствието между входни координати и *URL* адреси се установява от наличния в сървъра *map* файл. Тази техника се реализира чрез механизмите на *CGI*.

В *map* файл, който има текстов формат, се записва информация за "горещите зони" от избраното графично изображение. Дефинираните горещи зони могат да бъдат под формата на правоъгълник, окръжност, многоъгълник или просто точка. Координатите на изброените геометрични форми се задават в пиксели спрямо горния ляв ъгъл (0,0) на третираното графично изображение.

- **rect** - правоъгълна зона, чийто координати се определят от стойностите в пиксели на горния ляв ъгъл и долния десен спрямо нулевата координата на графиката. Активната зона е вътрешността на правоъгълника;
- **circle** - кръгова зона, чийто координати се определят от двойка точки. Първата определя центъра на окръжността, втората точка лежи на линията на окръжността. Активната зона е вътрешността на определената окръжност;
- **poly** - многоъгълна зона, зададена чрез списък от координати на върховете на многоъгълника. Многоъгълната зона може да бъде дефинирана с не повече от 100 двойки координати. Активната зона е вътрешността на многоъгълника;
- **point** - активната зона се дефинира от координатите на една точка, зададени спрямо горния ляв ъгъл на изобразената графика;
- **default** - е атрибут с помощта на който се означават областите в графичното изображение, които не са били описани като активни зони от други дефиниции.

Във файл (*map file*) се записва вида на активната зона, координатите на точките, които формират тази зона и свързания с нея *URL* адрес. В практиката са се наложили два различни начина на подреждане на параметрите във файла-карта, както следва:

- **CERN Map File Format** - Вид на активната зона координати на точки *URL* адрес (*region_type coordinates URL*);
- **NCSA Map File Format** - Вид на активната зона *URL* адрес координати на точки (*region_type URL coordinates*).

Елементът **<MAP>** осигурява механизъм за локална реализацията на графика с горещи точки съвместно с елемента ****, от който се задава името на графичния файл. Елементът **<MAP>** е празен, т.е. има завършващ елемент **</MAP>**. В съдържанието на **<MAP>** се включват елементи **<AREA>**. Елементът **<AREA>** задава координатите и вида на горещите точки от графиката и **URL** адресите на асоциираните им хипервръзки. Основната форма на елемента е следната:

```
<MAP NAME=cdata> AREA elements </MAP>
```

- **NAME=cdata** е параметър, указващ името на **MAP** секцията. Секцията става достъпна от параметъра **USEMAP=URL** на елемента ****. URL адресът сочи и името, като етикет, на **MAP** секцията.

Елементът **<AREA>** определя вида и координатите на горещите точки в една избираема графика. Елементът е от празен тип. Общия му вид е следния:

```
<AREA [SHAPE=rect|circle|poly] COORDS="x1,y1,x2,y2,..." HREF=URL [NOHREF] ALT="Text"
```

- **Shape** е параметър по избор, подразбиращото се значение е **rect**, и определя вида на зоната, оформяща гореща точка в графичното изображение. Допустимите видове зони са:
 - **rect** - правоъгълна зона, определена с двойка координати **x** и **y** от горния ляв и долния десен ъгъл на правоъгълника;
 - **circle** - кръгова зона, определена от координати **x** и **y** на центъра на окръжността и радиус, зададен в пиксели;
 - **poly** - многоъгълна зона, определена в списък от координати **x** и **y** за всеки връх.
- **Coords** е задължителен параметър за задаване на координатите на точки, формиращи гореща (избираема) зона в графичното изображение. Стойностите за **x** и **y** координатите се измерват в пиксели спрямо горния ляв ъгъл на графиката. Координатите могат да се задават и в проценти като отместване спрямо една начална точка, зададена в пикселни координати.
- **HREF** е задължителен параметър, указващ **URL** адрес на документ;
- **NOHREF** е параметър по избор, указващ че избираемата зона е без последствие;
- **ALT** е задължителен параметър с двойно действие, указващ името на хипервръзката (**Text**). Първо, при WWW клиенти с графични възможности посочва в статус линията на браузъра името на хипервръзката при поставяне на курсора на мишката върху горещата точка. Второ, при WWW клиенти с ограничени или изобщо без графични възможности е ориентир за това, къде в полето на документа се намират горещите точки, които се представят чрез техните етикети (**Text**).

На [Екран 7](#) е изобразена HTML страница с карта на България. От картата са снети координатите на отбелязаните градове с помощта на графичен редактор. Следният HTML код показва начина на формиране и обработка на горещи точки, с използване на елементите ****, **<MAP>**, **<AREA>**.

```
<IMG SRC="mapbg.gif" HSPACE=10 VSPACE=10 HEIGHT=180 WIDTH=296 ALIGN=LEFT
BORDER=0 USEMAP="#map1">
<MAP NAME="map1">
<AREA HREF="sofia.html" ALT="Sofia" SHAPE=RECT COORDS="33 66 83 86">
```

```
<AREA HREF="plovdiv.html" ALT="Plovdiv" SHAPE=RECT COORDS="97,96,153,113">
<AREA HREF="pleven.html" ALT="Pleven" SHAPE=RECT COORDS="89,35,138,48">
<AREA HREF="russe.html" ALT="Russe" SHAPE=CIRCLE COORDS="153,19,10">
<AREA HREF="varna.html" ALT="Varna" SHAPE=CIRCLE COORDS="232,52,10">
<AREA HREF="burgas.html" ALT="Burgas" SHAPE=POLY
COORDS="171,86,217,77,217,100">
</MAP>
```

Добра практика е наред с възможността за избор на горещи точки в една графика да се осигури паралелен вариант за достъп до същата информация посредством обикновени хипервръзки (чрез избираем текст). Последното се прави, за да се осигури случаят, в който нито клиентът, нито сървърът могат да обработват горещите точки.

Снемането на координатите на точки или области от графично изображение може да стане ръчно с помощта на графичен редактор като *PhotoShop*, *CorelDraw* или *Paint Shop Pro*. Съществуват специализирани редактори за тази цел, при които снемането на координати и определяне на вида на областите се извършва автоматизирано и на края се генерира необходимия HTML код ([CuteMap](#)). Подходящ за такава цел е редакторът *MapEdit*. Версия *shareware* с тридесет дневен срок на използване и информация за продукта може да се намери на адрес:

<http://www.boutell.com/mapedit>

Пример:



Екран 7.11 - При преминаване на курсора на мишката през картата има активни зони на хипервръзки, всяка от които сочи към друга страница за избран град.

Примерът е реализиран с помощта на следния код:

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
```

```

<meta name="GENERATOR" content="Mozilla/4.7 [en] (WinNT; I) [Netscape]">
<title>Bulgarian map</title>
</head>
<body>

<center><b><font color="#666600"><font size=+1>Административно делене на
България</font></font></b>
<br><img SRC="bul_map.gif" VSPACE=18 height=266 width=400 usemap="#bul_map"
border="0">
<br><i><font color="#666600">Екран 7.11 - При преминаване на курсора на
мишката през картата има активни <br>зони на хипервръзки, всяка от които сочи
към друга страница за избран град.</font></i><font
color="#666600"></font></center>

<map name="bul_map">
<area shape="circle" alt="София" coords="69,130,18" href="maps/sofia.html">
<area shape="circle" alt="Варна" coords="348,87,22" href="maps/varna.html">
<area shape="circle" alt="Бургас" coords="324,147,21" href="maps/burgas.html">
<area shape="circle" alt="Пловдив" coords="161,185,17" href="maps/plovdiv.html">
<area shape="circle" alt="Видин" coords="34,25,18" href="maps/vidin.html">
<area shape="circle" alt="Русе" coords="238,38,17" href="maps/russe.html">
<area shape="circle" alt="Габрово" coords="186,117,17" href="maps/gabrovo.html">
<area shape="circle" alt="Благоевград" coords="52,199,17"
href="maps/blagoevgrad.html">
<area shape="default" nohref>
</map>
</body>
</html>

```

Възпроизвеждане на звук (Playing Audio)

Аудио формати

Възпроизвеждането на качествен звук чрез персоналния компютър изисква допълнителен хардуер. Съществуващият малък говорител във всеки компютър не може да пресъздаде заложените в съществуващите звукови файлови формати качествени звукови компоненти. Допълнителната звукова карта (*16-bit stereo Sound Blaster Pro*) се монтира лесно на първия свободен слот на дънната платка на компютъра, и с помощта на придружаващото я програмно осигуряване може да възпроизведе чрез включени към нея тон колони всички познати звукови файлови формати. Различните видове звукови файлови формати се събират в три категории:

- Конвертиране на текстови файлове в нещо подобно на говор чрез замяна на буквите с фонети (*Text-to-speech*). Тази техника се използва при компютрите Macintosh.
- Музикални файлове, съдържащи музикални парчета като последователност от ноти и инструмент за тяхното възпроизвеждане. *MIDI* музикалните файлове са най-популярният представител в тази категория.
- Цифровото аудио представлява звук преминал през аналогово-цифров конвертор (*WAV*).

Съвременните Интернет Уеб клиенти разполагат с допълнително вградени възможности за интерпретация на най-разпространените звукови формати чрез т.н. *LiveAudio plug-in* програмни средства. *LiveAudio* техниката разпознава и автоматично възпроизвежда следните звукови формати:

- **AIFF** - е Macintosh/SGI звуков формат;
- **.AU** - е Sun/NeXT звуков формат;
- **MIDI** - Musical Instrument Digital Interface музикален формат;
- **.WAV** - е Microsoft Windows звуков формат.

Най-популярните Web клиенти - *Netscape* и *Internet Explorer* интерпретират тези звукови формати. При възникване на проблеми е нужно правилно да се попълни таблицата на допълнителните програмни средства (*Plug-Ins*) за съответните MIME видове и техните разширения дадени в следната таблица:

MIME вид	Разширение
audio/basic	.AU
audio/x-aiff	.AIF, .AIFF
audio/aiff	.AIF, .AIFF
audio/x-wav	.WAV
audio/wav	.WAV
audio/x-midi	.MID, .MIDI
audio/midi	.MID, .MIDI

Елемент <EMBED>

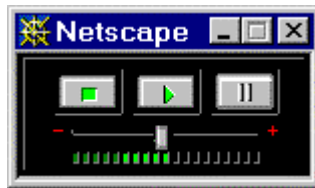
Непосредственото изпозване на техниката **LiveAudio** в една HTML страница е възможно чрез командата <EMBED>. Общият вид на командата е следния:

```
<EMBED SRC="Filename" WIDTH=Integer HEIGHT=Integer [AUTOSTART=False | True]
[VOLUME=Percentage] [AUTOLOAD=True|False] [STARTTIME="mm:ss"] [ENDTIME="mm:ss"]
[ALIGN="value"] [CONTROLS="value"] >
```

където, атрибутите на командата задават параметрите на контролния панел за управление на възпроизвеждането на звуковия файл (Екран 7.12):

- **SRC="Filename"** - име на звуков файл с неговото разширение от допустимите MIME видове. Параметърът е задължителен;
- **WIDTH=Integer** - размер по широчина на контролния панел в пиксели. Параметърът е задължителен;
- **HEIGHT=Integer** - размер по височина на контролния панел в пиксели. Параметърът е задължителен;
- **AUTOSTART=False | True** - подразбиращото се значение е False. При стойност True, свързания към командата звуков файл се стартира автоматично;
- **AUTOLOAD=True|False** - подразбиращото се значение е True, с което звуковият клип се зарежда автоматично;
- **VOLUME=Percentage** - задава в проценти величината на изходния звук;
- **STARTTIME="mm:ss"** - начално време на клипа, зададено в минути и секунди, спрямо самото му начало;
- **ENDTIME="mm:ss"** крайното време на клипа, зададено в минути и секунди, спрямо неговото начало;
- **ALIGN="value"** - подравняване на контролния панел спрямо основната линия на текста (*CENTER, BASELINE, TOP, LEFT, RIGHT*);

- **CONTROLS="value"** - атрибут за управление на контролния панел. Допустими са следните значения:
 - **CONSOLE** - извежда пълен набор от управляващи функции за контролния панел (*Play, Pause, Stop, Volume*);
 - **SMALLCONSOLE** - редуциран набор от контролни функции за контролния панел (*Play, Stop, Volume*). **AUTOSTART** добива значение **True**;
 - **PLAYBUTTON** - извежда само бутон *Play*;
 - **PAUSEBUTTON** - извежда само бутон *Pause*;
 - **STOPBUTTON** - извежда само бутон *Stop*;
 - **VOLUMELEVER** - извежда само бутон за управление на *Volume*;
 - **CONSOLE="name"** - комбинация от контроли за включване на повече клипове към една страница.



Екран 7.12 - LiveAudio контролен панел

Контролният панел, показан на Екран 7.12 съдържа бутони за управление на процеса на възпроизвеждане на указания в командата `<EMBED SRC="Filename">` звуков файл. Следният HTML код илюстрира използването на командата:

```
<HTML>
<HEAD>
  <TITLE>christmas</TITLE>
</HEAD>
<BODY>
  <CENTER><EMBED SRC="msc11.wav" WIDTH=144 HEIGHT=60 AUTOSTART=True VOLUME=100
CONTROLS=Console></CENTER>
</BODY>
</HTML>
```

Интерпретацията на кода може да бъде проследена, ако клиентската машина притежава възможностите на един мултимедийен компютър, т.е. има инсталиран подходящ звуков контролер и тонколони. За възпроизвеждане на звуковия файл натиснете [тук](#)!

Елемент <A>

Друга възможност за възпроизвеждане на звуков файл е асоциирането на [хипервръзка](#) към него. За целта например, може да се използва [графично изображение](#). Следният HTML код илюстрира тази възможност:

```
<HTML>
<HEAD>
  <TITLE>Christmas</TITLE>
</HEAD>
<BODY>
  <CENTER><HR SIZE=3 WIDTH="100%"></CENTER>
  <CENTER><A HREF="msc11.wav"><IMG SRC="tree.gif" ALT="Click Here to play the
  </A></CENTER>
```



```
<CENTER><HR SIZE=3 WIDTH="100%"></CENTER>
</BODY>
</HTML>
```

Интерпретацията на кода извежда в полето на документа съдържанието на графичния файл "tree.gif" и при избор на асоциираната към него връзка се стартира възпроизвеждането на звуковия файл "msc11.wav" (Екран 7.13).



Екран 7.13 - Коледно дърво и звънчета

Елемент <META>

Възпроизвеждане на звуков файл при зареждане на една HTML страница е възможно да се постигне с използване на [META](#) елемент в секцията [HEAD](#):

```
<META HTTP-EQUIV="Refresh" CONTENT="10; URL=media/tada.wav">
```

Елементът **META** с атрибут *Refresh* презарежда текущата страница през интервал указан в *Content* (10 секунди) или зарежда съдържанието по зададен *URL* адрес. При проследяване на [тази хипервръзка](#) примерът може да бъде проигран в реално време.

Създаване на таблици (Table commands)

Табличните функции в HTML са подходящо средство за представяне на информация в таблична форма. Използването на таблична структура улеснява в значителна степен представянето в полето на HTML документа и на не чисто таблична информация. Всичко зависи от предпочитанията на автора, характера и вида на информационните елементи. Таблица може да бъде изобразена и чрез командата **<PRE>**. Създадена с обикновен текстов редактор в ASCII формат таблица с тази команда се извежда в полето на документа във вида на нейния оригинал.

Табличните функции в езика HTML имат множество параметри и при тяхното комбиниране се получава по-атрактивно представяне на таблични структури. Елементът **<TABLE>** е непразен и основен за създаване на таблица. За описание на цялостна таблична структура се използват в комбинация и други HTML елементи.

Съкратен запис:

```
<TABLE> ... </TABLE>
```

Пълен запис с параметри:

```
<TABLE [ALIGN=left|center|right] [BORDER=n] [CELLPADDING=n] [CELLSPACING=n]
[WIDTH=n] [HEIGHT=n] [FRAME=cAttr] [RULES=cAttr] [BACKGROUND="url"]> ...
</TABLE>
```

- **ALIGN** - Определя хоризонталното подравняване на таблицата като цяло при следните възможности:
 - **LEFT** - подравняване към левия ограничител на страницата;
 - **CENTER** - центрирано;
 - **RIGHT** - подравняване към десния ограничител на страницата;
- **BORDER=n** - Определя наличието и дебелината в пиксели на рамка за ограждане на таблицата и всичките ѝ клетки. Отсъствието на параметъра изключва изобразяването на рамка, но необходимото празно пространство около таблицата и елементите ѝ се запазва;
- **CELLPADDING=n** - Определя широчината на празното пространство в пиксели между рамката и съдържанието на клетка;
- **CELLSPACING=n** - Определя разстоянието в пиксели между отделните клетки в таблицата;
- **WIDTH=n|%n** - Желаната широчина на таблицата се задава в пиксели или в проценти от широчината на документа;
- **HEIGHT=n|%n** - Минимално желаната височина на таблицата се задава в пиксели или в проценти;
- **FRAME=cAttr** - свързва се наличието на рамка (**BORDER**) в таблицата и определя кои страни от кадъра да се изобразяват в зависимост от установения [атрибут](#);
- **RULES=cAttr** - свързва се наличието на рамка (**BORDER**) в таблицата и определя допълнителни правила в таблицата в зависимост от установения [атрибут](#);
- **BACKGROUND="url"** - задава адрес и име на графичен файл за фон (*mapet*) на таблицата.

Елементът **<CAPTION>** се използва съвместно с **<TABLE>** или **** за поставяне на заглавие или коментар на таблица или графика.

Съкратен запис:

```
<CAPTION> Text </CAPTION>
```

Пълен запис:

```
<CAPTION [ALIGN=top|bottom|left|right|center] [VALIGN=top|bottom]> Text
</CAPTION>
```

- **ALIGN** - Определя разположението на текста на заглавието по отношение на таблицата или фигурата. Изравняване на текста в ляво (**Left**), дясно (**Right**) или центрирано (**Center**) не може да се използва съвместно с горно (**Top**) и долно (**Bottom**) позициониране. При тези случаи се използва допълнителен параметър **VALIGN**.
- **VALIGN** - Определя месторазположението на заглавието спрямо таблицата или фигурата. **Top** поставя заглавието преди таблицата, **Bottom** - след таблицата. Този параметър се използва когато в **ALIGN** е зададено хоризонтално подравняване на текста.

Елементът **<TR>** е от празен тип и задава нов *ред* в една таблица. Използването му е възможно с или без параметри.

Съкратен запис:

```
<TR>
```

Пълен запис:

```
<TR [ALIGN=left|center|right] [VALIGN=top|middle|bottom|baseline]>
```

- **ALIGN** - Задава хоризонталното подравняване на съдържанието в клетките на реда - ляво (*Left*), дясно (*Right*), центрирано (*Center*);
- **VALIGN** - Задава вертикалното подравняване на съдържанието в клетките на реда - към горния край (*Top*), в средата (*Middle*), към долния край (*Bottom*), спрямо базовата линия на текста (*Baseline*).

Елементът **<TD>** служи за създаване на една *клетка* за данни в таблицата. Той може да унаследи всички параметри зададени чрез елемента **<TR>**. Елементът **<TD>** се задава като празен елемент.

Съкратен запис:

```
<TD>
```

Пълен запис:

```
<TD [ALIGN=left|center|right] [COLSPAN=NUMBER] [NOWRAP] [ROWSPAN=NUMBER]
[VALIGN=top|middle|bottom|baseline] [WIDTH=value|percent]>
```

- **ALIGN** - Задава хоризонталното подравняване на съдържанието на клетката - в ляво (*Left*), в дясно (*Right*), центрирано (*Center*);
- **COLSPAN** - Задава броят на колоните, обхванати от тази клетка;
- **ROWSPAN** - Задава броят на редовете, обхванати от тази клетка;
- **NOWRAP** - Предотвратява пренасяне на текст на следващ ред в рамките на клетката;
- **VALIGN** - Задава вертикалното подравняване на съдържанието на клетката - към горния край (*Top*), в средата (*Middle*), към долния край (*Bottom*), спрямо базовата линия на текста (*Baseline*);
- **WIDTH** - Задава предпочитания размер на клетката в пиксели или в проценти от размера на таблицата.

Обикновено таблично представената информация съдържа клетки с *наименование* на колоните или редовете. Тази необходимост се реализира с помощта на елемента **<TH>**.

Съкратен запис:

```
<TH>
```

Пълен запис:

```
<TH [ALIGN=left|center|right] [COLSPAN=NUMBER [NOWRAP] [ROWSPAN=NUMBER]
[VALIGN=top|middle|bottom|baseline] [WIDTH=value|percent]>
```

Сходството на атрибутите с тези на елемента **<TD>** е пълно, само функцията на елемента го различава и обстоятелството, че подразбиращите се значения за основните атрибути са насочени към центрирано представяне на съдържанието на клетката в получер шрифт (*Bold*).

Разполагането на една таблица в HTML документ често пъти създава проблеми, свързани с обема изобразявана информация. Възможно най-компактно представената таблица изисква установяване на параметрите ***border***, ***cellspacing*** и ***cellpadding*** на елемента **<TABLE>** със стойност 0. Съдържанието на отделните клетки не се различава от допустимото съдържание за един HTML документ, т.е. текст, графика, хипервръзки, списъци, друга таблица. Включването на графични обекти като съдържание на клетка от таблица е възможно за новите версии на WWW клиентите като *Netscape Navigator 3.x* и *Microsoft Internet Explorer 3.x*. Оцветяване на клетките, рамките и текстовите компоненти в клетките е достъпно в момента за *Microsoft Internet Explorer 3.x*. *Netscape 3x* ги интерпретира частично. Следващите атрибути могат да се включват по избор към елементите за формиране на таблица.

- ***b bgcolor="#rrbbgg"|"colour"*** - Използва се за определяне на основния цвят в клетка, зададен като шестнадесетично число или по наименованието на цвета (виж Таблица 5);
- ***bordercolor="#rrbbgg"|"colour"*** - Задава цвета на рамката за клетка (виж Таблица 5);
- ***bordercolorlight="#rrbbgg"|"colour"*** - Задава цвета на светлото поле за рамка на таблица, изобразена в 3D-формат (виж Таблица 5);
- ***bordercolordark="#rrbbgg"|"colour"*** - Задава цвета на тъмното поле за рамка на таблица, изобразена в 3D-формат (виж Таблица 5).

Следващите примери представят комбинираната употреба на команди за управление на таблици и техните параметри. Пример за таблица - HTML код:

```
<TABLE>
<TR><TD>1</TD><TD>2</TD><TD>3</TD><TD>4</TD></TR>
<TR><TD>5</TD><TD>6</TD><TD>7</TD><TD>8</TD></TR>
</TABLE>
```

Визуализация след интерпретация на кода (Екран 8). Примерът може да бъде изпълнен и директно при проследяване на [тази хипервръзка](#).

```
1 2 3 4
5 6 7 8
```

Екран 8 - Таблица с два реда и четири колони

Пример с обособена рамка и отделени клетки - HTML код:

```
<table border=10 cellpadding=10 cellspacing=10 width=75%>
<tr><th>Първо</th><th>Второ</th><th>Трето</th><th>Четвърто</th></tr>
```

```
<tr><th>Четири<th>Пет<th>Шест
</table>
```

Визуализация след интерпретация на кода (Екран 9). Примерът може да бъде изпълнен и директно при проследяване на [тази хипервръзка](#).



Едно	Две	Три
Четири	Пет	Шест

Екран 9 - Нестандартно приложение на елемент <TH>

По-сложен е примерът с използване на графика вложена в клетка, общи клетки за серия колони и редове, празна клетка в таблица - HTML код:

```
<CENTER>
<TABLE border=2 width=70%>
<tr><th colspan=2 align=center><IMG SRC="computer.gif"> </th>
<th colspan=2><FONT SIZE="+2">Pentium Pro</FONT SIZE="+2"></th></tr>
<tr><td></td><td></td><th>CPU-180 Mhz</th><th>CPU-200 Mhz</th></tr>
<tr><th rowspan=6> Components </th><th>RAM</th><td
align=center>32 MB</td><td align=center>64 MB</td></tr>
<tr><th>Hard Disk</th><td align=center>1600 MB</td><td
align=center>2100 MB</td></tr>
<tr><th>Monitor</th><td align=center>15" Color SVGA 0.28 MPR-II</td>
<td align=center>17" Color SVGA 0.28 MPR-II</td></tr>
<tr><th>CD-ROM</th><td align=center>8xSpeed EIDE</td><td
align=center>12xSpeed EIDE</td></tr>
<tr><th>Case</th><td align=center>MINI Tower 200W</td><td
align=center>MIDI Tower 200W</td></tr>
<tr><th>Mouse</th><td
align=center>3-Button Mouse+Pad</td><td align=center>3-Button
Mouse+Pad</td></tr>
<tr><td></td><td colspan=3 align=right>
<A HREF="mailto://master@muled.vmei.acad.bg">For more, please ask!</A><IMG
SRC="smile.gif"></td></tr>
</TABLE>
</CENTER>
```

Визуализация след интерпретация на кода (Екран 10). Примерът може да бъде изпълнен и директно при проследяване на [тази хипервръзка](#).

		Pentium Pro	
		CPU-180 Mhz	CPU-200 Mhz
Components	RAM	32 MB	64 MB
	Hard Disk	1600 MB	2100 MB
	Monitor	15" Color SVGA 0.28 MPR-II	17" Color SVGA 0.28 MPR-II
	CD-ROM	8xSpeed EIDE	12xSpeed EIDE
	Case	MINI Tower 200W	MIDI Tower 200W
	Mouse	3-Button Mouse + Pad	3-Button Mouse + Pad
		For more, please ask! 	

Екран 10 - Таблица с вмъкнати компоненти

Атрибут FRAME

Атрибутът се използва единствено в елемента **<TABLE>** съвместно с атрибута **BORDER**. Приложението му е възможно за навигатори от по-ново поколение (*Internet Explorer 4.x, 5.x, Netscape Communicator 4.x*), тъй като е въведен от HTML 4.0. Предназначението на атрибута е да определя кои страни от рамките в таблицата да бъдат изобразени с помощта на следните атрибути:

- **ABOVE** - изобразява външните рамки от горната страна на таблицата..
- **BELOW** - изобразява външните рамки от долната страна на таблицата.
- **BORDER** - изобразява рамка от всички страни на таблицата.
- **BOX** - изобразява рамка около таблицата.
- **INSIDES** - изобразява рамка само от горе и отдолу в таблицата.
- **HSIDES** - изобразява рамка на външните хоризонтални страни на таблицата.
- **LHS** - изобразява рамка на външната лява страна на таблицата.
- **RHS** - изобразява рамка на външната дясна страна на таблицата.
- **VOID** - премахва всички външни страни от рамката в таблицата.
- **VSIDES** - изобразява рамка на външната дясна и лява страна на таблицата.

Атрибут RULES.

Атрибутът се използва единствено в елемента **<TABLE>** съвместно с атрибута **BORDER**. Приложението му е възможно за навигатори от по-ново поколение (*Internet Explorer 4.x, 5.x, Netscape Communicator 4.x*), тъй като е въведен от HTML 4.0. Предназначението на атрибута е да определя и променя правилата за изобразяване на рамка вътре в таблицата, чрез следните атрибути:

- **NONE** - премахва всички рамки вътре в таблицата..
- **GROUPS** - извежда хоризонтална рамка между секции.
- **ROWS** - извежда хоризонтална рамка между всички линии в таблицата.
- **COLS** - извежда вертикална рамка между всички колони в таблицата.
- **ALL** - разчертава напълно таблицата.

Следните атрибути се използват с основния елемент на таблицата.

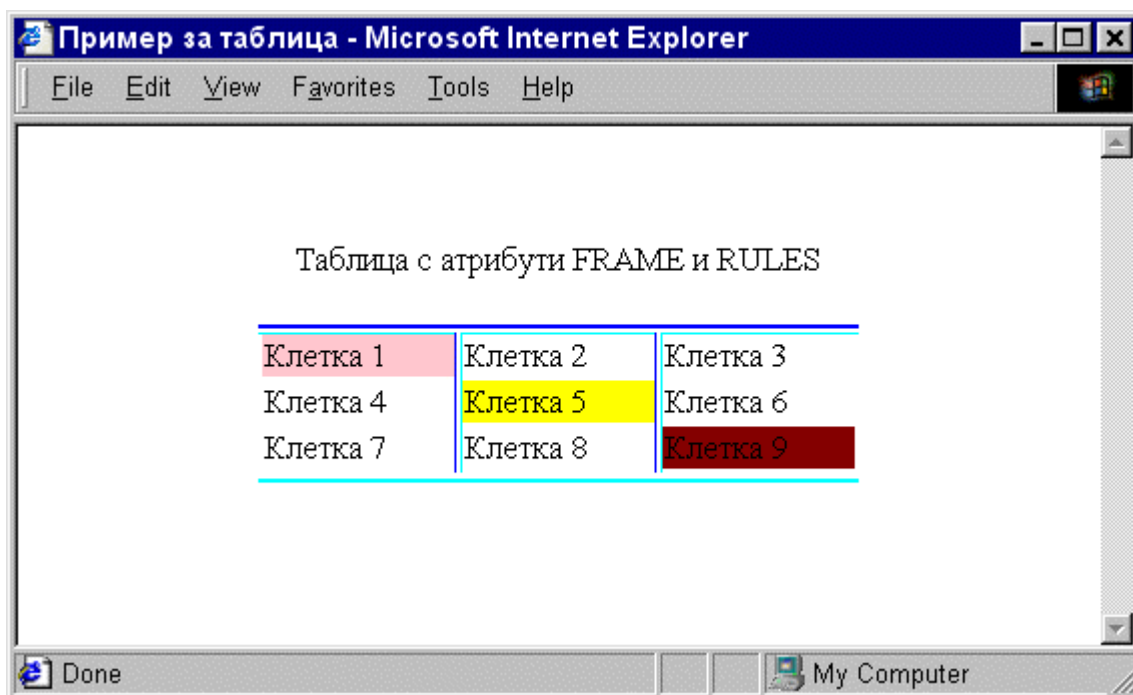
- **THEAD** - формира антетка в таблицата.
- **TBODY** - задава тялото на таблицата.
- **TFOOT** - дефинира коментарна линия под таблицата.

Тези атрибути не се възприемат еднакво от *IE* и *Netscape* и съответно не се интерпретират по един и същи начин. Например, **BORDERCOLORDARK** е атрибут, използван съвместно с **BORDERCOLOR** и определя цвета на сянката на рамката в една таблица. Той се интерпретира правилно само от *Internet Explorer 4.x, 5.x*. Комбинацията от атрибути **FRAME=HSIDES, RULES=COLS** в дефиницията на таблица се игнорират от *Netscape Communicator 4.x*, а от *Internet Explorer 4.x, 5.x* се интерпретират коректно.

Следния пример показва възможностите на разгледаните атрибути.

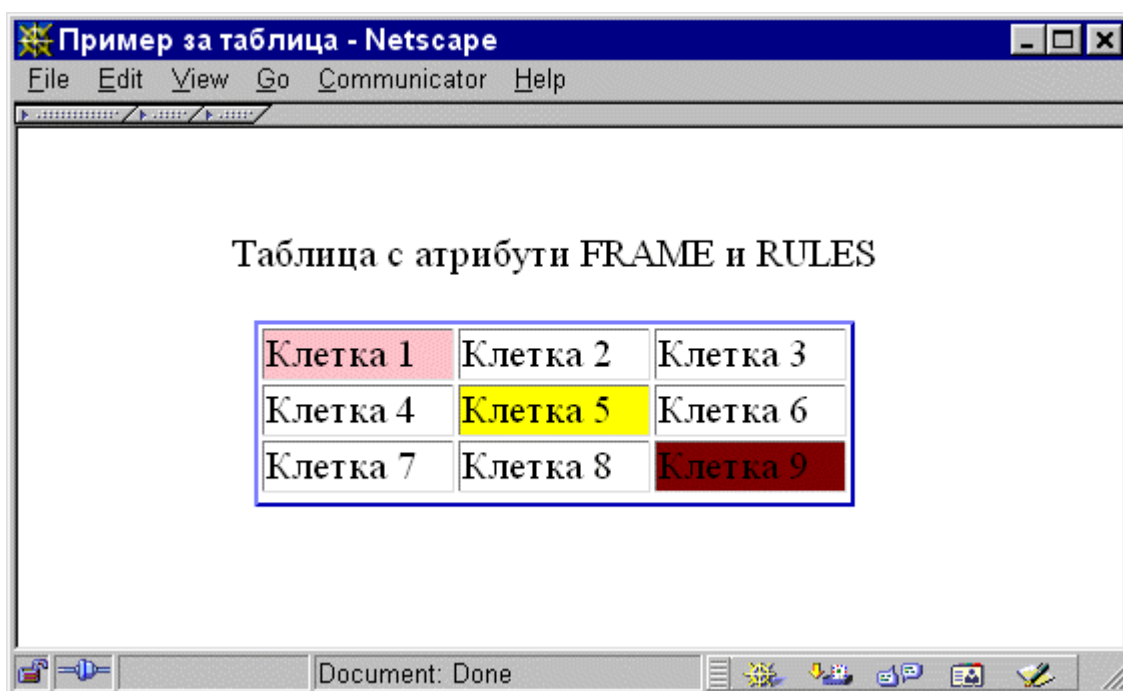
```
<HTML>
<HEAD><TITLE>Пример за таблица</TITLE></HEAD>
<BODY>
<p>&nbsp;</p>
<p ALIGN=CENTER>Таблица с атрибути FRAME и RULES</p>
<TABLE ALIGN=CENTER BORDER=2 WIDTH=300 FRAME=HSIDES RULES=COLS
BORDERCOLOR="BLUE" BORDERCOLORDARK="AQUA">
<THEAD>
<COLGROUP SPAN=3 ALIGN=LEFT VALIGN=TOP>
</THEAD>
<TBODY>
<TR>
<TD BGCOLOR="PINK">Клетка 1</TD>
<TD>Клетка 2</TD>
<TD>Клетка 3</TD>
</TR>
<TD>Клетка 4</TD>
<TD BGCOLOR="YELLOW">Клетка 5</TD>
<TD>Клетка 6</TD>
</TR>
<TR>
<TD>Клетка 7</TD>
<TD>Клетка 8</TD>
<TD BGCOLOR="MAROON">Клетка 9</TD>
</TR>
</TBODY>
</TABLE>
</BODY>
</HTML>
```

Примерът може да бъде изпълнен и директно при проследяване на [тази хипервръзка](#).
Визуализация след интерпретация на кода с *Internet Explorer 5.0* - Екран 10.1.



Екран 10.1 - Атрибути в таблица от HTML 4.0

Визуализация след интерпретация на кода с Netscape Communicator 4.72 - Екран 10.2.



Екран 10.2 - Атрибути в таблица от HTML 4.0

Включване на формуляри (Form commands)

Интерактивният формуляр, включен в HTML документ е прекрасна възможност за установяване на директен контакт с потребителите на информация по Internet мрежата. Командата **<FORM>** създава именувани полета в документа, в които желаещият може да въведе исканата от него информация или да зададе въпрос, на който да получи съответен

отговор. Един WWW клиент извлича документа в машината на потребителя, а съдържанието от полетата на формуляра се връщат в сървъра на автора. В сървъра предварително замислен механизъм обработва тази информация и при необходимост връща резултат на клиента. Пълната реализация на този модел не се поддържа от езиковите възможности на HTML. За целта са необходими допълнителни програмни средства, най-често реализирани на базата на (*Common Gateway Interface*) **CGI** интерфейс (*cgi-script*). Най-подходящият език за създаване на **CGI** процедури е **PERL** (*Practical Extraction and Report Language*), възможни са реализации, разбира се, и на **C++**, **PASCAL**. С появата в последно време на езика *Java* тези средства се обогатяват. Елементът **<FORM>** е от непразен тип и се употребява с или без атрибути в зависимост от приложението.

Съкратен запис:

```
<FORM> ... </FORM>
```

Разширен запис:

```
<FORM [ACTION=URL] [ENCTYPE=cdata] [METHOD=GET|POST] >
съдържание на формата с възможно участие на елементите
INPUT, TEXTAREA, SELECT </FORM>
```

- **ACTION** - URL адрес на сървър и/или услуга, където ще бъде прехвърлено съдържанието на формата;
- **ENCTYPE** - алгоритъм за кодиране/декодиране на съдържанието (подразбира се **MIME**);
- **METHOD** - определя метода на предаване на съдържанието на формата към сървъра (подразбира се **GET**);
 - **GET** - информацията се предава посредством променлива;
 - **POST** - информацията се предава посредством програмата **STDIN**.

Съвместно с **<FORM>** могат да се използват и следните HTML елементи:

Елементът **<INPUT>** се използва в секцията на командата **<FORM>** за осигуряване на механизъм, по който потребителят може да въвежда информация. Елементът се употребява като празен или непразен.

Съкратен запис:

```
<INPUT>
```

Пълен запис:

```
<INPUT [ALIGN=top|middle|bottom|left|right] [CHECKED] [MAXLENGTH=NUMBER]
[NAME=cdata] [SIZE=NUMBER] [SRC=URL]
[TYPE=TEXT|PASSWORD|CHECKBOX|RADIO|SUBMIT|RESET|FILE|HIDDEN|
IMAGE|RANGE] [VALUE=cdata] >
```

- **ALIGN** - контролира подравняването на полетата от формата - в ляво (**Left**), в дясно (**Right**), спрямо базовата линия на текста (**Top, Middle, Bottom**);
- **CHECKED** - при активни атрибути **RADIO** или **CHECKBOX** инициализира полето за избор;
- **MAXLENGTH** - определя максималната дължина на текста в атрибутите **TEXT** и **PASSWORD**;
- **NAME** - идентификатор на полето при предаване на съдържанието на формата към сървъра;
- **SIZE** - определя броят на видимите символи за **TEXT** и **PASSWORD**;
- **SRC - URL** адрес на графичното изображение, използвано съвместно с атрибутите **SCRIBBLE, IMAGE, SUBMIT, RESET**;
- **TYPE** - определя вида на полето със следните атрибути:
 - **TEXT** - определя, че в полето от формуляра ще се въвежда свободен текст;
 - **PASSWORD** - въведените символи се маскират с '*' или '.';
 - **CHECKBOX** - поле за избор, изисква атрибути **NAME** и **VALUE**;
 - **RADIO** - радио-бутон, изисква атрибути **NAME** и **VALUE**;
 - **SUBMIT** - предизвиква предаване на съдържанието на формуляра посредством указаните действия в **ACTION**;
 - **RESET** - предизвиква анулиране на въведената в полето на формата информация;
 - **FILE** - присъединени файлове за прехвърляне;
 - **RANGE** - контрол на данни в определен обхват, зададен в **VALUE**;
 - **HIDDEN** - подтиска се изобразяването на текст;
 - **IMAGE** - действа подобно на **SUBMIT**;
- **VALUE** - използва се за инициализация на **TEXT, RANGE** и **HIDDEN**.

Командата **<INPUT>** е най-често използваната. Тя дава възможност да се създават различни типове управляващи елементи. Управляващите елементи включват различни видове бутони, полета и прозорци за въвеждане на текст.

Командата има много *параметри*. Най-важният от тях е **TYPE**. Той специфицира вида на очаквания вход.

Параметърът **NAME** е задължителен при всички изброени значения на атрибутите на **TYPE** с изключение на **SUBMIT** и **RESET**. Той създава етикет, който се свързва с отговора на потребителя. След прехвърляне на съдържанието на формуляра в сървъра по този параметър може да се определи към кой елемент от формуляра се отнася даден отговор. Думата, която се задава след **NAME** не се извежда в полето на документа, например:

```
NAME="text_about"
NAME="variable1"
NAME="what"
```

Атрибутът **CHECKBOX** на параметъра **TYPE** позволява да се създаде бутон, заемащ две състояния - включено или изключено. Той се използва, когато се задават прости въпроси, имащи един от два възможни отговора (включено/ изключено, да/не, има/няма и т.н.). В един документ може да има произволен брой такива елементи. Те са независими помежду си. В действителност към сървъра се предават само маркираните от потребителя бутони. Всички останали се приемат за немаркирани. При необходимост даден бутон да се счита по подразбиране за маркиран (избран), в елемента се поставя допълнителен параметър **CHECKED**. Общият формат за създаване на бутони от типа **CHECKBOX** е следния:

```
<INPUT TYPE="checkbox" NAME="име" VALUE="стойност" CHECKED>
```

Параметърът **VALUE** се употребява в комбинация с параметъра **NAME** и се използва за задаване на символен низ, който се изпраща на сървъра при състояние *включено (on)* на бутона. Например, ако **NAME="setting"** и **VALUE="on"**, към сървъра ще се изпрати **"setting=on"**. В случай, че повече бутони използват едно и също име в **NAME**, всеки избран от потребителя бутон, ще бъде изпратен към сървъра. Например, ако един бутон има име **NAME="device"** и **VALUE="printer"** и друг бутон има име **NAME="device"** и **VALUE="scanner"**, към сървъра ще бъдат изпратени **"device=printer"** и **"device=scanner"**. Не е допустимо използването на бутони с еднакви значения на параметрите **NAME** и **VALUE**. Следният пример илюстрира употребата на елемента **TYPE** с атрибут **CHECKBOX**:

```
<INPUT TYPE="checkbox" NAME="dev" VALUE="printer" CHECKED> Принтер
<INPUT TYPE="checkbox" NAME="dev" VALUE="scanner"> Скенер
<INPUT TYPE="checkbox" NAME="dev" VALUE="monitor" CHECKED> Монитор
```

В полето на документа се изобразяват три бутона с пояснителен текст, два от тях са маркирани по подразбиране. За всеки маркиран бутон към сървъра се предава символен низ, състоящ се от **NAME** и **VALUE** (от примера **"dev=printer"** и **"dev=monitor"**). (Екран 10-1)

☒ Принтер ☐ Скенер ☒ Монитор

Екран 10-1 - Въвеждане на списък чрез маркиране на избора

Атрибутът **RADIO** на параметъра **TYPE** позволява да се създаде бутон, заемащ две състояния - включено или изключено. Той наподобява функцията на атрибута **CHECKBOX**. Разликата се състои в това, че в една група от бутони в даден момент може да бъде избран само един. Изборът на даден бутон предизвиква освобождаването на изборния до момента. Общият формат за създаване на бутони от типа **RADIO** е следния:

```
<INPUT TYPE="radio" NAME="име" VALUE="стойност" CHECKED>
```

Радио бутони с едно и също име принадлежат към една и съща група. Ако някой от бутоните в дадена група има параметър **CHECKED**, той автоматично се показва в натиснато (избрано) състояние. Следният пример илюстрира употребата на елемента **TYPE** с атрибут **RADIO**:

```
Какъв вид е вашата кредитна карта?
<INPUT TYPE="radio" NAME="credit" VALUE="visa" CHECKED> Visa<br>
<INPUT TYPE="radio" NAME="credit" VALUE="mc"> Mastercard<br>
<INPUT TYPE="radio" NAME="credit" VALUE="ae"> American Express
```

След символния низ **"Какъв вид е вашата кредитна карта?"** се изведат 3 радио бутона, всички от групата **"credit"**. Първият бутон е избран по подразбиране - **CHECKED**. (Екран 10-2)

Какъв вид е вашата кредитна карта?

- ☒ Visa
- ☐ Mastercard
- ☐ American Express

Екран 10-2 - Въвеждане на единичен избор от списък

Атрибутът **TEXT** на параметъра **TYPE** позволява да се въвежда свободно текст от клавиатурата. Във формулярите често се налага потребителят да въведе отделни редове - адреси за електронна поща, телефонни номера, номера на кредитни карти и друга информация. Общият формат за създаване на текстово поле е следния:

```
<INPUT TYPE="text" NAME="име" VALUE="стойност" SIZE=Number MAXLENGTH=Number>
```

Подобно на другите входни елементи и този има параметър **NAME**, което позволява именуването му. Обикновено се поставя име, което е свързано с предназначението на елемента. Например, ако **NAME="email"** към сървъра ще се изпрати низът **"email="**, последван от въведения от потребителя текст. Параметърът **VALUE** задава символния низ, който ще се изведе в полето за текст в първия момент. Ако такъв параметър не е зададен, полето за текст ще бъде празно при първоначалната му поява. Параметърът **SIZE** позволява да се зададе размер на текстовото поле. Например, при **SIZE="40"**, този размер е 40 символа. Много WWW клиенти поддържат възможност за повече от един ред текст и възприемат параметър **SIZE=ширина, височина**. Например, **SIZE="40,10"** задава поле за текст с ширина 40 символа и височина 10 реда, което дублира функцията на елемента **TEXTAREA**. Стойността по подразбиране на параметъра **SIZE** е 20 символа. Параметърът **MAXLENGTH** установява максималния брой символи, които могат да бъдат въведени в текстовото поле. Ако такъв брой не е указан, полето е с неограничена дължина. Например, **MAXLENGTH="100"** означава, че на един ред могат да се въведат до 100 символа. Следният пример илюстрира употребата на елемента **TYPE** с атрибут **TEXT**:

Въведете вашия e-mail адрес: <INPUT TYPE="text" NAME="email">

Тази команда ще предизвика разполагането на поле за текст след низа **"Въведете вашия e-mail адрес:"**. Текстът, който се въведе, ще се предаде към сървъра във вида **"email=въведеният текст"**. (Екран 10-3)

Въведете вашия e-mail адрес:

Екран 10-3 - Въвеждане на текст

Атрибутът **PASSWORD** на параметъра **TYPE** е текстово поле за въвеждане на информация от вида на "парола за достъп". Общият формат за създаване на маскирано текстово поле е следния:

```
<INPUT TYPE="password" NAME="име" VALUE="стойност" SIZE=Number MAXLENGTH=Number>
```

Използва се когато е необходимо въвеждането на информация, която не трябва да се визуализира върху дисплея (пароли, номера на кредитни карти и др.) Въвежданите символи

се показват във вид на звездички или други символи в зависимост от интерпретацията на WWW клиента, т.е. въвежданите символи се маскират. Параметрите **NAME**, **VALUE**, **SIZE** и **MAXLENGTH** имат същото действие и значение както при атрибута **TEXT**. Следният пример илюстрира употребата на елемента **TYPE** с атрибут **PASSWORD**:

```
Please enter your password:
<INPUT TYPE="password" NAME="pass" SIZE=10>
```

Размерът на полето е 10 символа, но могат да се въвеждат низове с произволна дължина, тъй като не е зададен параметър **MAXLENGTH**. Към сървъра се изпраща "pass=", последвано от въведения в полето текст. (Екран 10-4)

Please enter your password:

Екран 10-4 - Въвеждане на парола за достъп

Атрибутът **RESET** на параметъра **TYPE** се използва в следния общ формат:

```
<INPUT TYPE="reset" VALUE="стойност">
```

Този атрибут създава бутон за установяване в начално състояние на полетата във формуляр. Това начално състояние се определя от стойностите по подразбиране, зададени в отделните елементи. Параметър **NAME** отсъства, тъй като не се предава информация към сървъра. Инициализацията се извършва автоматично в WWW клиента при натискането на бутона. Символният низ след **VALUE** се използва като етикет на бутона. Следният пример илюстрира употребата на елемента **TYPE** с атрибут **RESET**:

```
<INPUT TYPE="reset" VALUE="Click here to reset the form">
```

Този ред създава бутон с надпис "Click here to reset the form". (Екран 10-5)

Click here to reset the form

Екран 10-5 - Изчистване на попълнена форма

Атрибутът **SUBMIT** на параметъра **TYPE** се използва за създаване на бутон за изпращане на съдържанието на формуляр на сървър за обработка

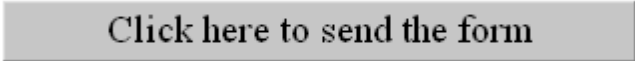
```
<INPUT TYPE="submit" VALUE="стойност">
```

Този елемент създава бутон, при активирането на който става изпращане на формуляра към сървъра, специфициран чрез **ACTION** в командата **<FORM>**, посредством зададения в същата команда параметър **METHOD**. Параметър **NAME** отсъства, тъй като самата функция не изпраща към сървъра информация. Символният низ след **VALUE** се използва като етикет на бутона. Следният пример илюстрира употребата на елемента **TYPE** с атрибут

SUBMIT:

```
<INPUT TYPE="submit" VALUE="Click here to send the form">
```

Този ред създава бутон с надпис *"Click here to send the form"*. При натискане на бутона всички попълнени полета във формуляра ще се изпратят към сървъра. Полета и бутони, които не са установени не се изпращат. (Екран 10-6)



Екран 10-6 - Изпращане на попълнена форма

Елементът **<SELECT>** се използва за създаване на многозначен избор, подобно на меню. Отделните елементи, формиращи списъка за многозначен избор се задават с елемент **<OPTION>** в тялото на **<SELECT>**. Елементът е от празен тип и има съкратен и разширен запис.

Съкратен запис:

```
<SELECT> ... </SELECT>
```

Разширен запис:

```
<SELECT [MULTIPLE] [NAME=cdata] [SIZE=NUMBER]> ... </SELECT>
```

- **MULTIPLE** - определя менюто като многозначно, подразбира се единствен избор;
- **NAME** - променлива параметър, на която се присвоява стойност от направения преди прехвърлянето ѝ в сървъра избор;
- **SIZE** - определя броят на видимите айтъми при **MULTIPLE**.

Командата **<OPTION>** задава един елемент за избор от списъка за многозначен избор, оформен като меню и управляван от командата **<SELECT>**. Един пълен списък за многозначен избор се съставя от съответния брой елементи **<OPTION>**. Употребява се като празен елемент със или без атрибути:

Съкратен запис:

```
<OPTION>
```

Разширен запис:

```
<OPTION SELECTED VALUE=cdata>
```

- **SELECTED** - задава елемента, който се избира по подразбиране;
- **VALUE** - задава стойност или значение, което ще се прехвърля към сървъра при предаване на съдържанието на формуляра.

Езикът HTML позволява да се създават списъци, които да съдържат един или повече елементи. Потребителят може да избере от даден списък един или повече елементи. Елементът *INPUT* с атрибут *CHECKBOX* е подходящ за малки списъци. Елементът **SELECT** позволява създаване на списъци с произволен размер. Списъците се визуализират във вмъкнати прозорци (*inset windows*) или падащи прозорци (*popup windows*).

Елементът **<SELECT>** започва даден списък. Чрез параметъра *NAME* се задава име на списъка. Например, **<SELECT NAME="list1">** ще изпрати към сървъра за всеки избран елемент от списъка *list1* името на съответния елемент. Ако списъкът съдържа "One", "Two" и "Three" и са избрани "One" и "Three" към сървъра ще бъдат изпратени "list1=One" и "list1=Three".

Елементът **SELECT** има параметър за размер *SIZE*. Размерът по подразбиране е 1. Размер на списък 1 означава, че списъкът се показва в прозорец "*popup*", като потребителят може да избира само един от елементите в списъка (подобно на радио бутоните). При размер на списъка по-голям от 1, списъкът се визуализира в прозорец "*inset*" и посоченият размер показва максималния брой на видимите елементи от прозореца. Например, при *SIZE=5* в прозореца ще са видими 5 елемента от списъка. Елементите, които остават невидими могат да се визуализират, като се използва намиращият се вдясно от списъка плъзгач (*slider*).

Друг по избор параметър на **<SELECT>** е *MULTIPLE*. Той не приема стойност. Използването му предизвиква появата на списъка в прозорец "*inset*", за избор на повече от един елемент от него. Това става чрез задържане на бутон *CTRL* докато се избират желаните елементи от списъка.

Между елементите **<SELECT>** и **</SELECT>** се поставя елементът **<OPTION>**. Всеки избираем елемент от списъка се описва с **<OPTION>**. Допълнителната опция **SELECTED** задава елемента от списъка, избран по подразбиране. Следният пример илюстрира употребата на елементите **SELECT** и **OPTION**:

```
Какъв е вашият домашен любимец: <BR>
<SELECT NAME="list1" SIZE=3 MULTIPLE>
<OPTION SELECTED> Рибки
<OPTION> Котка
<OPTION> Куче
<OPTION> Хамстер
<OPTION> Папагал
</SELECT>
```

В примера е дефиниран списък от 5 елемента, от които 3 са видими (*SIZE=3*). Потребителят може да посочи любимото си домашно животно. Списъкът е дефиниран като *MULTIPLE*, т.е. любимците могат да бъдат повече от един. Всеки избран елемент се изпраща към сървъра като *list1=*, следвано от съдържанието на избрания елемент. По подразбиране е избран първият елемент. (Екран 10-7)

Какъв е вашият домашен любимец:



Екран 10-7 - Избор от дефиниран списък

<TEXTAREA> е елемент за въвеждане на повече от един ред текст в поле от формуляр. Елементът е непразен. Полето за текст се оформя като каре от атрибутите на елемента. Невидимата част от карето е достъпна след придвижване на плъзгачите за скролиране на прозореца на екрана.

Съкратен запис:

```
<TEXTAREA> ... </TEXTAREA>
```

Разширен запис:

```
<TEXTAREA [COLS=NUMBER] [NAME=cdata] [ROWS=NUMBER] > ... </TEXTAREA>
```

- **COLS** - определя броят на видимите колонии;
- **NAME** - име на полето, на което се присвоява въведеният текст за прехвърлянето му на сървъра;
- **ROWS** - определя броят на видимите редове.

Този елемент служи за въвеждане на блок от текст от потребителя. Това дава възможност цели документи да се вмъкват в HTML страници.

Параметър **NAME** служи за именуване на въведения текстов блок. Например, при задаване на **NAME="body"** към сървъра ще се изпрати **"body="**, следвано от въведения текст. Параметрите **ROWS** и **COLS** задават размера на прозореца за текст, изобразяван на дисплея. Тези параметри определят размера на видимия прозорец и не ограничават размера на текста, въвеждан от потребителя. При **ROWS=10** и **COLS=50** се задава поле с ширина 10 символа и височина 50 реда. Затвореният текст между началния и крайния елемент на командата **TEXTAREA** се появява на екрана при първоначално изобразяване на формата. При отсъствие на такъв текст, текстовото поле е празно при първоначално показване. Следният пример илюстрира употребата на **TEXTAREA**:

Въведете Вашите предпочитания:

```
<TEXTAREA NAME="requirements" ROWS=10 COLS=15>Нямам специални  
изисквания</TEXTAREA>
```

Инструкцията оформя текстово каре в прозорец с размери 10 реда и 15 колонии. Карето се предхожда от коментарен текст - *"Въведете Вашите предпочитания"*. В полето на прозореца се изписва текст по подразбиране - *"Нямам специални изисквания"*. В текстовото каре

потребителят може да въведе желания от него текст, който ще бъде изпратен на сървъра при изпращане на цялата форма. (Екран 10-8)

Въведете Вашите предпочитания:

Нямам специални изисквания

Екран 10-8 - Въвеждане на текстово поле

Обобщен пример за всички разисквани до тук компоненти е достъпен директно при проследяване на [тази хипервръзка](#).

Създаване на формуляри

Езикът HTML предоставя възможност за взаимодействие с потребителя по начин, който не може да се осигури от обикновените хипервръзки. Примери за такова взаимодействие са потребителска инспекция, критерии за търсене в база данни, бутони за избиране на опции, въвеждане на команди, интерактивни съобщения, компютърни игри и др.

Интерактивен формуляр в полето на един HTML документ може да бъде създаден на две стъпки.

- Описание на полетата от формуляра в HTML документ;
- Определяне на механизъм за извличане на въведената информацията.

Механизми за обработка на формуляри

Формулярите са средство за въвеждане на информация от потребител в един HTML документ, която ще бъде предадена и обработвана на ниво сървър. В общия случай формулярите се използват за представяне на въпросници, форми за регистрация или бележки, адресирани към автора на документа. Проблемите тук са в средствата за обработка на прехвърлената от формуляра към сървъра информация. Често пъти това изисква допълнителен софтуер, инсталиран на сървъра, закупен или специално разработен за тази цел. Обикновено това са програмни средства на *cgi-script* или *java*.

Работата с формуляри изисква взаимодействие с WWW сървър, известен като **HTTP Daemon**. Сървърът **HTTPD** осигурява обработката на данните от формуляр, когато те бъдат подадени. Данните от формуляра се попълват от клиента в неговата машина и се подават с помощта на съответна команда на указан в нея URL адрес. В адреса може да се съдържа и името на обработващата програма. Върху HTTPD сървъра се изпълнява програмата, която определя типа на формуляра, обработва данните и възпроизвежда съответен отговор. Отговорът се връща на клиента.

В елемента **<FORM>** се използват два параметъра за реализация на съответния механизъм за предаване на данните от формуляра към сървъра.

Параметърът **ACTION** специфицира валиден абсолютен или относителен **URL** адрес. Данните от формуляра се изпращат на адреса, зададен в **ACTION**.

Параметърът **METHOD** специфицира метода, използван от сървъра за изпращане на данните на формуляра до програмата, зададена в **ACTION**. Съществуват два основни метода - **GET** и **POST**.

Методът **GET** се прилага по подразбиране. Когато формуляр, използващ метода **GET**, се получи в HTTPD сървър, елементите му се преобразуват в команден ред и се предават като параметър на програмата, специфицирана в **ACTION**.

Методът **POST** е с по-големи възможности. Данните от формуляра се предават на сървъра посредством стандартния вход (STDIN). Пример за команда **<FORM>** може да изглежда по следния начин:

```
<FORM ACTION="http://www.act.com/cgi-bin/prg1" METHOD="POST">
```

В него параметърът **ACTION** указва, че данните от формуляра ще се изпратят по протокол *http* на адрес *www.act.com*. Сървърът ще осъществи достъп до програмата (скрипта) с име *prg1*, намираща се в директорията на сървъра - *cgi-bin*. Параметърът **METHOD** показва, че формулярът ще се предаде на програмата *prg1* по метода **POST**.

Най-простият начин за обработка на формуляри е чрез *електронна поща* да се изпратят на определен адрес, където да се разглеждат по желание. Това решение е известно като **FORM REMAILING** и голяма част от сървърите вече предлагат тази услуга. *Обработващите програми (remailers)* изискват от потребителя да посочи *e-mail* адрес, на който да се изпращат всички формуляри. След това програмата *remailer* събира всички получени формуляри, форматира ги във вид на читаеми отговори и ги изпраща на указания адрес. Следващият пример показва начина на обръщение към програма *remailer*

```
<FORM ACTION="http://www.act.com/cgi-bin/remailer?iv@www.act.com" METHOD="POST">
```

В този формуляр значението за **ACTION** е **URL** адрес на сървър с име *www.act.com*. Сървърът намира програмата с име *remailer* в директория *cgi-bin*. Указан е метод **POST** за връзка с програмата *remailer*. Ако има информация, която трябва да се подаде към програмата посредством командния ред, то тя се отделя от името на програмата със знака " ?".

Методът с препращане на формуляри чрез електронна поща е най-достъпен за ползване, тъй като не изисква наличието на директен достъп до сървъра.

При наличие на достъп до сървъра потребителят може да състави собствена програма за обработка на формуляри или да използва съществуващи програми. Много програмни системи за обработка на формуляри могат да се получат чрез Internet. Те могат директно да се използват или да се модифицират според конкретните нужди.

Описание на формуляри

В един HTML документ може да се съдържат повече от един формуляр, но не е възможно формулярите да се съдържат един в друг. Интерактивните формуляри се създават на основата на следните елементи:

- Радио бутони (Radio buttons);
- Бутони за проверка (Check boxes);
- Списъци (Lists);
- Области за въвеждане на текст (Text entry areas).

Всеки формуляр съдържа един или повече такива елементи. Няма ограничение върху броя на използваните елементи.

Езиковите възможности на HTML 3.2 са достатъчни за реализиране на най-разнообразни формуляри. Алтернативно могат да се разработват съвместно с *java-script*. Общите правила за създаване на един формуляр са следните:

- Обособяване на определена зона в документа за реализиране на формуляр;
- Определяне на метода за прехвърляне на набраните от полетата на формуляра данни към сървъра;
- По възможност, определяне на месторазположението върху сървъра на програмата, предназначена за обработка на прехвърлените данни;
- При необходимост, задаване на списък от аргументи към обработващата програма.

Създаването на формуляри в един HTML документ е много практичен начин за установяване на директен контакт с потребител, но от казаното до тук става ясно, че това не винаги е реализуемо. С един пример могат да бъдат представени възможностите за редактиране на формуляри, но неговото цялостно функциониране зависи от конкретния WWW клиент. В примера *Action* атрибута се свързва с команда за изпращане като електронна поща на попълнените данни във формуляра. Понастоящем по Internet в множество сървъри могат да се намерят и изтеглят най-разнообразни *cgi* приложения, удовлетворяващи обработката на попълнените във формуляри данни. Пример за формуляр - регистрационна форма:

```
<h2>Пример за формуляр</h2>
<form action="mailto:user@provider.bg" method="POST">
<input type="hidden" name="subject" value="Example Form">
<select name="age">
<option value="under 18">under 18 </option>
<option selected value="18 to 25">18 to 25 </option>
<option value="25 to 30">25 to 30 </option>
<option value="30 to 40">30 to 40 </option>
<option value="over 40">over 40 </option>
</select> Възраст<br>
<input type="text" name="name" size=40> Име<br>
<input type="text" name="email" size=40> Електронен адрес
<p><textarea name="details" cols=50 rows=7 wrap=physical>Something about
yourself</textarea>
<BR>
Как преценявате това описание
<input type="radio" name="opinion" value="great" checked> Отлично
<input type="radio" name="opinion" value="middling"> Добро
<input type="radio" name="opinion" value="very poor"> Лошо
<p>
<input type="submit" value="Send">
<input type="reset" value="Clear">
</form>
```

При опит за прехвърляне на данни по този пример в реални условия ще предизвика съобщение за грешка (след активиране на бутон *SEND*), иначе примерът на ниво клиент е работоспособен. Част от причините за това бяха изяснени по-горе.

Пример за формуляр

Възраст

Име

Електронен адрес

Something about yourself

Как преценявате това описание
 ☒ Отлично
 ☐ Добро
 ☐ Лошо

Екран 11 - Формуляр за регистрация

Изпълнението на примера може да се ограничи в свободното попълване на текстовото поле, активиране на меню-избор, промяна в състоянието на радио-бутони. Визуализация - резултат от интерпретацията на кода - Екран 11. Примерът може да бъде изпълнен и директно при проследяване на [тази хипервръзка](#).

Следващият пример реализира интерактивен формуляр от вида въпросник.

```
<FORM ACTION="http://www.comp.com/cgi-bin/sellcomp" METHOD=POST>
Enter your name: <INPUT TYPE="text" NAME="name" SIZE=40> and phone:
<INPUT TYPE="text" NAME="phone" SIZE=+00> <P>
Enter your address: <TEXTAREA NAME="address" ROWS=5 COLS=45>
Please put your address here</TEXTAREA> <P>
Choose a type for your personal computer<P>
<INPUT TYPE="radio" NAME="type" VALUE="pentium" CHECKED> Pentium
<INPUT TYPE="radio" NAME="type" VALUE="pc_486"> PC-486
<INPUT TYPE="radio" NAME="type" VALUE="pc_386"> PC-386
<INPUT TYPE="radio" NAME="type" VALUE="apple"> Apple
Select your computer configuration options:
<SELECT NAME="options" MULTIPLE SIZE=4>
<OPTION> CD Drive
<OPTION> Sound Blaster
<OPTION> Video Blaster
<OPTION> Fax Card
<OPTION> Modem
<OPTION> Laser Jet
</SELECT>
```

```

Select what you want us to do now: <P>
<INPUT TYPE="checkbox" NAME="dispose" VALUE="info"> Send more information about
my choices <BR>
<INPUT TYPE="checkbox" NAME="dispose" VALUE="prices"> Send a price list <BR>
<INPUT TYPE="checkbox" NAME="dispose" VALUE="make"> Make me a computer and call
me when it is done <P>
<INPUT TYPE="submit" VALUE="Press this button to submit the form">
</FORM>

```

Визуализация след интерпретация на кода - Екран 12. Примерът може да бъде изпълнен и директно при проследяване на [тази хипервръзка](#).

Екран 12 - Въпросник

Структуриране на секции (Frame commands)

Понятието **FRAME** има широко приложение в компютърната и комуникационната терминология. В изграждането на един HTML документ това понятие се въвежда като технология за разделяне на екрана при визуализация от WWW клиента на обособени, функционално самостоятелни секции, в които съвместно или поотделно могат да се изпълняват различни приложения.

Технологията **FRAME** разделя екрана на един или повече прозорци, като във всеки един от тях могат да се визуализират различни HTML документи. Прозорците могат да бъдат съседни, различни по размер, цвят и фон или да се припокриват и при това запазват общите характеристики на прозорец в HTML документ. Тази технология се поддържа от по-новите модификации на WWW клиенти и е един атрактивен и практичен начин за представяне на структурирана информация. От версия на *Netscape 2.x* и нагоре, и *Microsoft Internet Explorer 3.x* поддържат и интерпретират правилно фреймове.

Фреймовете се създават и контролират от фреймови дефиниции, където се задава техния брой, размер и позициониране, както и *URL* адресите на тяхното съдържание. Съдържанието на фрейма се задава в отделен HTML документ, а не в дефиниционната му част.

Структура на HTML файл.

Структурата на HTML файл с фреймови дефиниции не се различава от общоприетата. Разликата е в описанието на функционалната част, където командата **BODY** е заменена с команда за дефиниране на фреймово множество **FRAMESET**. В тялото се разполагат командите **FRAME** с необходимия списък от параметри, описващи характеристиките на всяка част поотделно. Общият вид на един HTML файл с фреймови дефиниции е следния:

```
<HTML>
  <HEAD>
    <TITLE>Заглавие на страницата ...</TITLE>
  </HEAD>
  <FRAMESET>
    Описание на прозорците по секции ...
  </FRAMESET>
</HTML>
```

Дефиниция на фреймово множество

```
<FRAMESET> [ROWS="x1,x2,...|x1%,x2%,...|x1*,x2*,..."]
[COLS="x1,x2,...|x1%,x2%,...|*"] [FRAMEBORDER="yes|no|0"] [BORDER="value"]
[BORDERCOLOR="#hexcolor"] [FRAMESPACING=n] ... </FRAMESET>
```

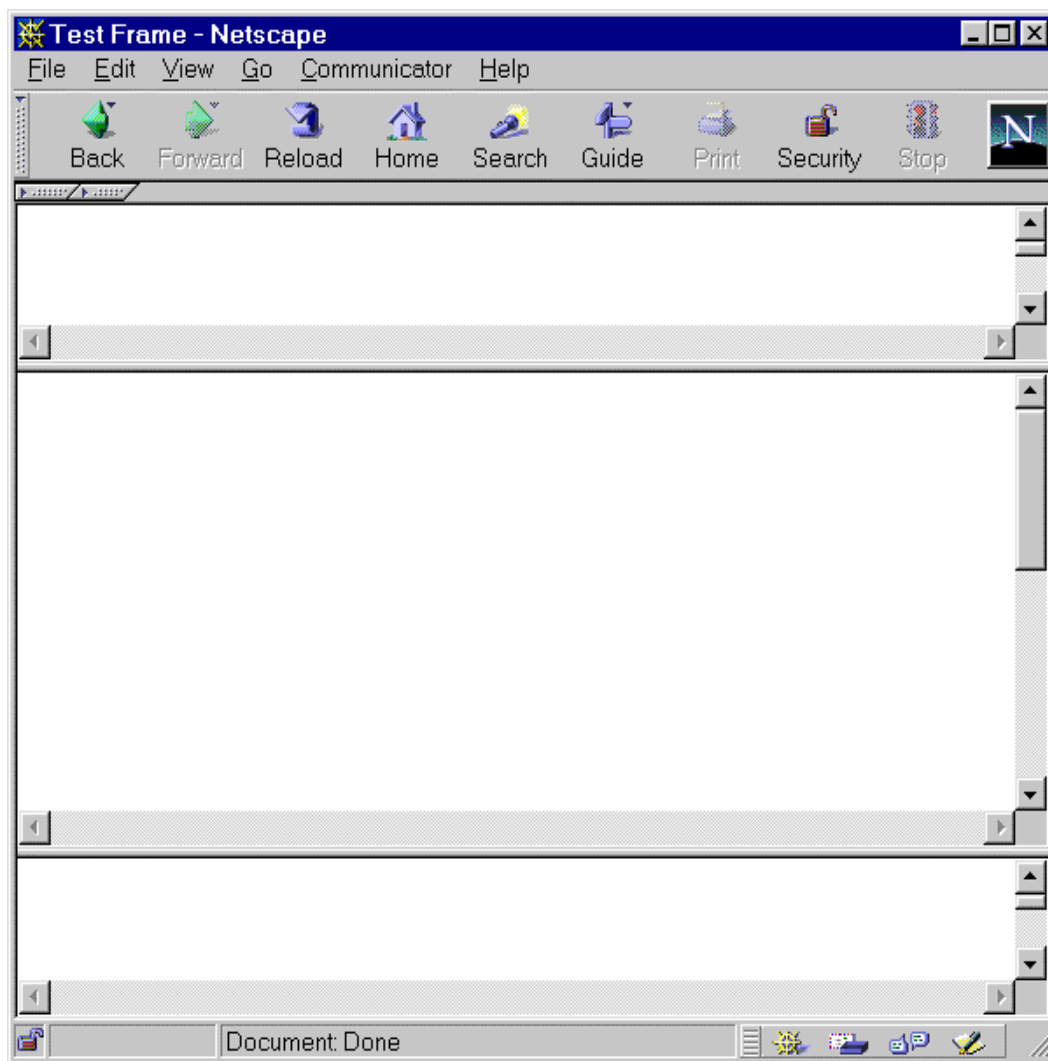
където:

- **ROWS** е атрибут за определяне на размерите на фрейм във вертикална посока. Редовете се задават като списък от елементи, разделени със запетаи в три различни формата. Абсолютни единици - в пиксели (x1, x2, ...); в проценти спрямо общия размер (x1%, x2%, ...); в относителни единици (x1*, x2*, ...). Например:

<FRAMESET ROWS="2*,*"> - хоризонтално разделяне на екрана на две части, горната част е с двойно по-голям размер спрямо долната;

<FRAMESET ROWS="50*,50"> - хоризонтално разделяне на екрана на три части, с фиксиране на размерите на горната и долната част, останалото място се предоставя на средната част;

<FRAMESET ROWS="20%,60%,20%"> - визуализира голям среден фрейм, а горният и долният са с по-малки размери процентно разделени спрямо размера на общия прозорец (*Екран 13-1*).



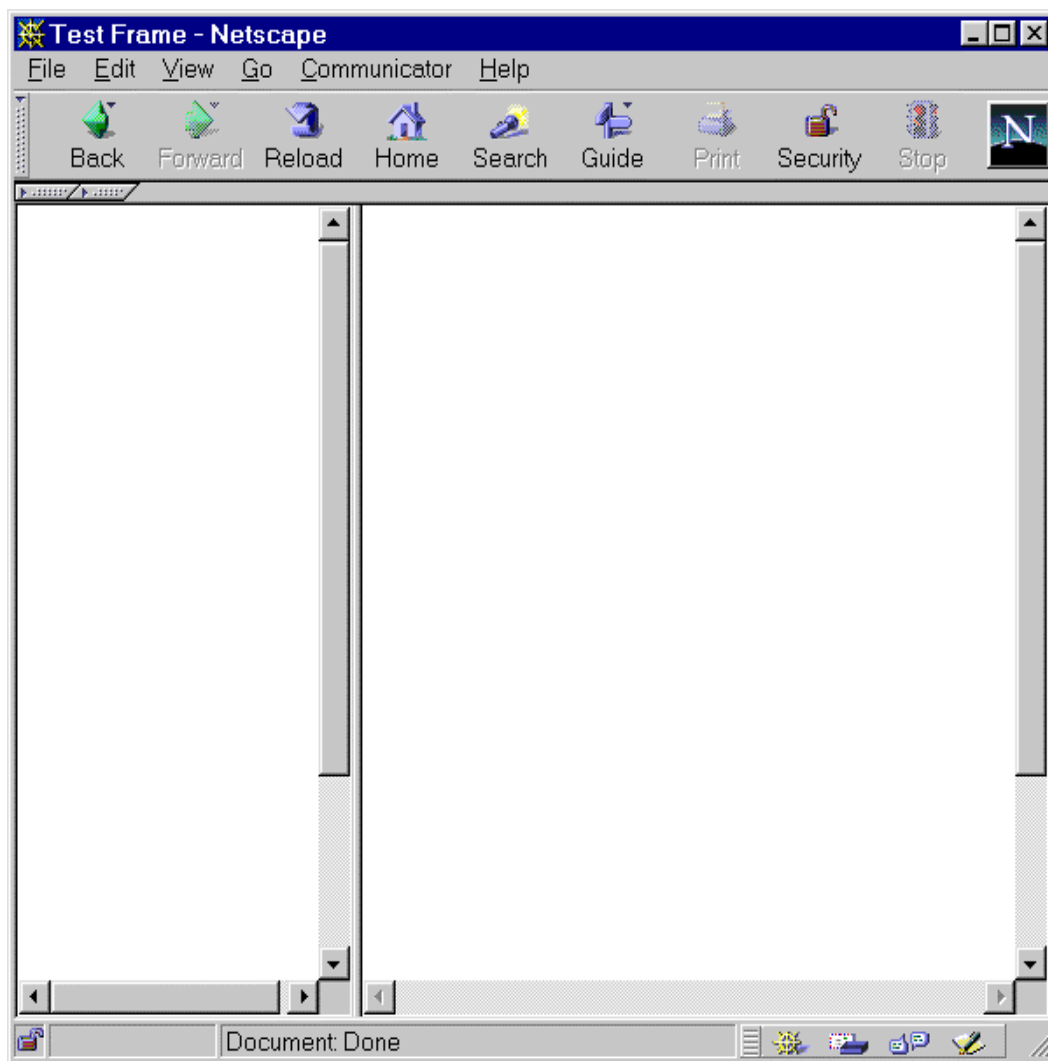
Екран 13-1 Фрайм с три хоризонтални секции

- **COLS** е атрибут за определяне на размерите на фрейм във хоризонтална посока. Колоните се задават като списък от елементи, разделени със запетаи в три различни формата. Абсолютни единици - в пиксели (x1, x2, ...); в проценти спрямо общия размер (x1%, x2%, ...); в относителни единици (x1*, x2*, ...). Примерите за **ROWS** са приложими и за **COLS**.

<FRAMESET COLS="2*,*"> - вертикално разделяне на екрана на две части, горната част е с двойно по-голям размер спрямо долната; (Екран 13-2)

<FRAMESET COLS="50*,50"> - вертикално разделяне на екрана на три части, с фиксиране на размерите на горната и долната част, останалото място се предоставя на средната част;

<FRAMESET COLS="20%,60%,20%"> - визуализира голям среден фрейм, а левият и десният са с по-малки размери процентно разделени спрямо размера на общия прозорец.



Екран 13-2 Фрайм с две вертикални секции

- **FRAMEBORDER="yes|no|0"** - атрибутът включва или изключва рамка около фрейм. Правилната интерпретация зависи от WWW клиента, *Netscape Navigator 3.x* и *Netscape Communicator 4.x* използват "Yes|No", докато *MS Internet Explorer* подразбира "0".
- **BORDER="value"** - атрибутът задава дебелината на рамката на фрейм в пиксели, приложим е понастоящем за *Netscape 3.x*.
- **BORDERCOLOR="#rrggbb"** - атрибутът задава цвета на рамката на фрейм в RGB шестнадесетичен формат, приложим е понастоящем за *Netscape 3.x* или по-нова версия.
- **FRAMESPACING=n** - атрибутът задава разстояние между отделните кадри в пиксели (n).

Дефиниция на фрейм

```
<FRAME> [SRC="URL"] [NAME="WINDOW_NAME"] [NORESIZE] [SCROLLING="YES|NO|AUTO"]
[MARGINWIDTH="VALUE"] [MARGINHEIGHT="VALUE"] [BORDERCOLOR="#HEXCOLOR"]
[FRAMESPACING="VALUE"] [FRAMEBORDER="YES|NO|"] ... </FRAME>
```

където:

FRAME е команда, употребявана в структурата на **FRAMESET** и дефинира един фрейм. Командата е от непразен тип. Атрибутите му са незадължителни и могат да се употребяват в произволен ред.

- **SRC="URL"** - атрибутът указва URL адрес на документ, чието съдържание ще се изобразява във фрейма. Отсъствието на атрибута изобразява празен фрейм;
- **NAME="WINDOW_NAME"** е атрибут за именуване на фрейма. Името може да се ползва като хипервръзка в други документи или като значение във фреймови атрибути;
- **SCROLLING="YES|NO|AUTO"** - атрибутът позволява включване или изключване на хоризонталната и вертикална линия за преместване (*ScrollBar*). Подразбиращото се значение е **AUTO**, при което *ScrollBar* се появява в зависимост от нуждите;
- **NORESIZE** - употребата на атрибута забранява промяната на размерите на дефиниран фрейм. По подразбиране размерите на един фрейм могат да се променят по стандартния за всеки прозорец в Windows начин;
- **MARGINWIDTH="VALUE", MARGINHEIGHT="VALUE"** - са два атрибута, които задават широчината и височината в пиксели на разделителното поле във фрейма;
- **BORDERCOLOR="#HEXCOLOR"** - позволява задаване на цвят за рамката на фрейма (виж Таблица 5), интерпретира се за сега само от *Netscape 3.x* или по-нова версия;
- **FRAMESPACING="VALUE"** - атрибутът позволява оформяне на празно пространство около фрейм, зададено в пиксели. Използва се за сега само от *MS Internet Explorer 3.x*;
- **FRAMEBORDER="YES|NO|"** - атрибутът включва или изключва рамка около фрейм. *Netscape 3.x* използва значенията "Yes/No", а *MS Internet Explorer* използва стойност "0".

Свободното комбиниране на фреймовите атрибути винаги е възможно, но за различните Web браузъри дава различни резултати. Например, при **FRAMEBORDER="No"** в **FRAMESET** и отсъствие на атрибут **NORESIZE** в **FRAME** предполага липса на рамка около фрейма и възможност за промяна на първоначално установените му размери от потребителя в диалогов режим. Това задание се изпълнява изцяло от *MS Internet Explorer 4.x*. Браузърите на *Netscape* не поддържат промяна в размерите на прозореца на фрейма при липса на рамка, т.е. в този случай **FRAMEBORDER** трябва да бъде установен в "Yes" за да стане достъпна промяната в размерите на прозореца.

Алтернатива на фрейм

```
<NOFRAMES> ... </NOFRAME>
```

Използването на фрейми е нововъведение в HTML документите и се поддържа от елементите на езика HTML след версия 3.0. Не всички WWW клиенти (Web Browser) интерпретират фреймовите структури. С цел да не бъде притесняван клиентът, един документ структурират на базата на фреймови елементи, може да се изобразява алтернативно в обичаен формат, който трябва да бъде допълнително описан. Това става с помощта на командата **NOFRAMES**, която е от непразен тип и затваря алтернативното описание.

Приложение

Използването на фрейми е много подходящ начин за представяне на сложна в структурно отношение информация. Един фрейм може да се разглежда като функционално

самостоятелен прозорец, с всички характеристики на този обект, в чието поле се изобразява информацията от определен HTML документ.

Във видимото поле на WWW клиента се разполагат произволен брой фрейми. Едновременно на един екран се разполага информация от множество HTML документи, която може да бъде обвързана по определен начин. Например, представяне на кратки анотации на съдържанието на една книга по глави.

Секционирването на екрана в един HTML документ става с помощта на два елемента:

- Първият е елемента **FRAMESET**, чрез който се определя размера и разположението на отделните секции;
- Вторият е елемента **FRAME**, чрез който се задава съдържанието на всяка отделна секция.

Структурата на една HTML страница за дефиниране на секции ще изглежда така:

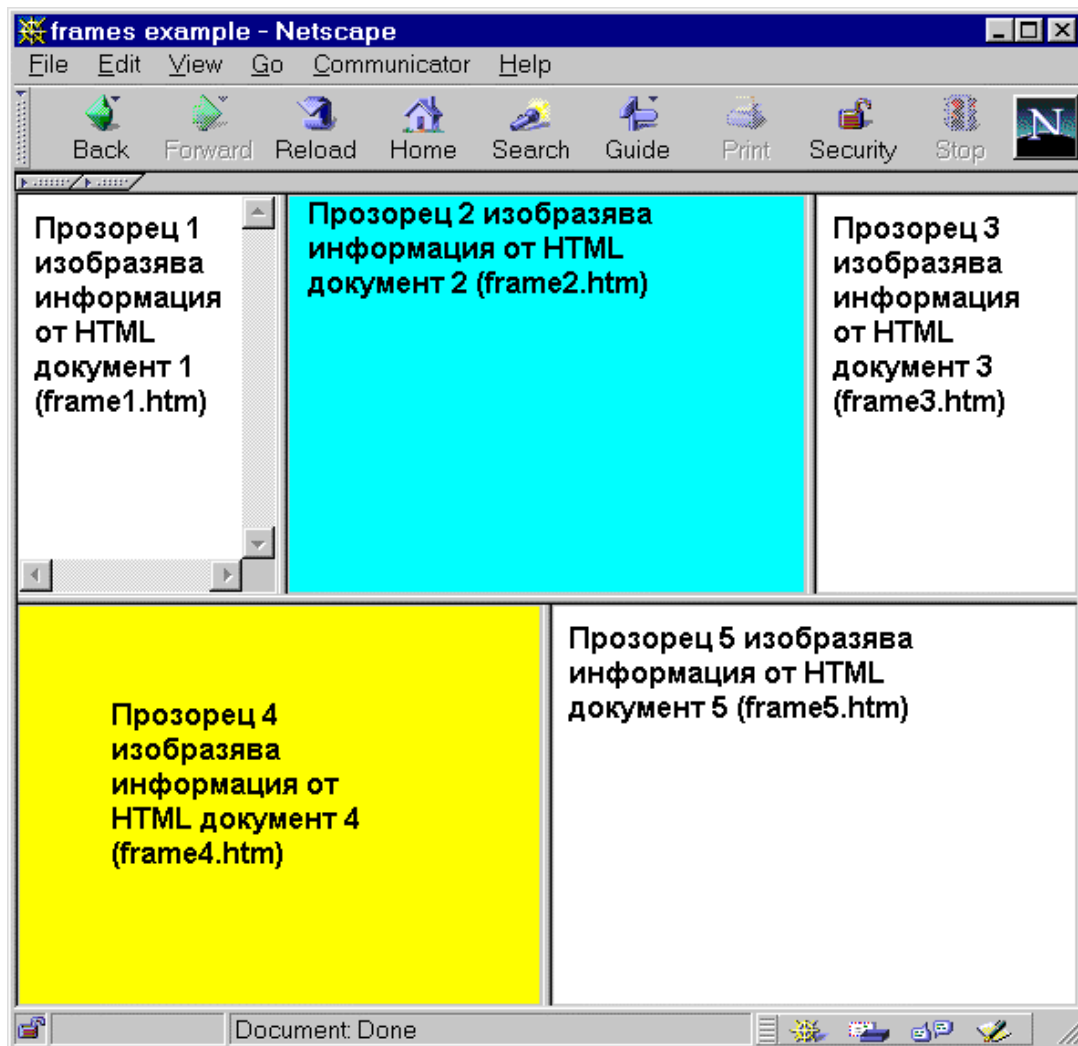
```
<FRAMESET за две хоризонтални секции>
  <FRAMESET за три вертикални секции>
    <FRAME> съдържание на първи прозорец
    <FRAME> съдържание на втори прозорец
    <FRAME> съдържание на трети прозорец
  </FRAMESET>
  <FRAME съдържание на втората хоризонтална секция>
</FRAMESET>
```

Предложеният по-долу пример има за цел да демонстрира по-скоро езиковите възможности за описание на фрейми, отколкото напълването им със съдържание. Интерпретацията на следния код е представена на *Екран 13-3*.

```
<HTML>
<HEAD>
  <TITLE>FRAMES EXAMPLE</TITLE>
</HEAD>
<FRAMESET ROWS="50%,50%">
  <FRAMESET COLS="150,*,150">
    <FRAME SRC="frame1.htm" SCROLLING="yes">
    <FRAME SRC="frame2.htm" NAME="frame2" SCROLLING="auto" MARGINWIDTH="10"
NORESIZE>
    <FRAME SRC="frame3.htm" NAME="frame3">
  </FRAMESET>
  <FRAMESET COLS="50%,50%">
    <FRAME SRC="frame4.htm" MARGINHEIGHT=50 MARGINWIDTH=50>
    <FRAME SRC="frame5.htm">
  </FRAMESET>
</FRAMESET>
<NOFRAMES>
  <A HREF="noframes.htm">
</NOFRAMES>
</HTML>
```

Екранът е разделен хоризонтално на две равни части (**<FRAMESET ROWS="50%,50%">**), горната част е разделена вертикално на три части, като две са с фиксиран размер в пиксели, а средната част получава остатъка - (**<FRAMESET COLS="150,*,150">**). Последователността от три команди **FRAME** с атрибути дефинират трите фрейма от горната част на екрана.

Левият фрейм `<FRAME SRC="frame1.htm" SCROLLING="yes">` изобразява съдържанието на файла `"frame1.htm"` при активирани линии за хоризонтално и вертикално преместване. Средният фрейм - `<FRAME SRC="frame2.htm" NAME="frame2" SCROLLING="auto" MARGINWIDTH="10" NORESIZE>` е наименован `"frame2"` и изобразява съдържанието на `"frame2.htm"`. При него *Scrolling* се включва при необходимост, не се допуска преоразмеряване на фрейма, и има височина на ограничителна линия 10 пиксела. Долната част се разделя на две равни части (`<FRAMESET COLS="50%,50%">`), описани съответно от две команди **FRAME**. Левият фрейм се отличава по това, че изобразеното съдържание на файла `"frame4.htm"` отстоява на 50 пиксела от ограничителните линии на фрейма - `<FRAME SRC="frame4.htm" MARGINHEIGHT=50 MARGINWIDTH=50>`.



Екран 13-3 Фреймови компоненти

Съдържанието на HTML документите, изобразени в отделните прозорци се оформя съгласно с общите изисквания, като може да съдържа разнообразни компоненти включващи текст, графика, звук и др. HTML кодът на един от изобразените документи е показан по-долу и е подобен на останалите.

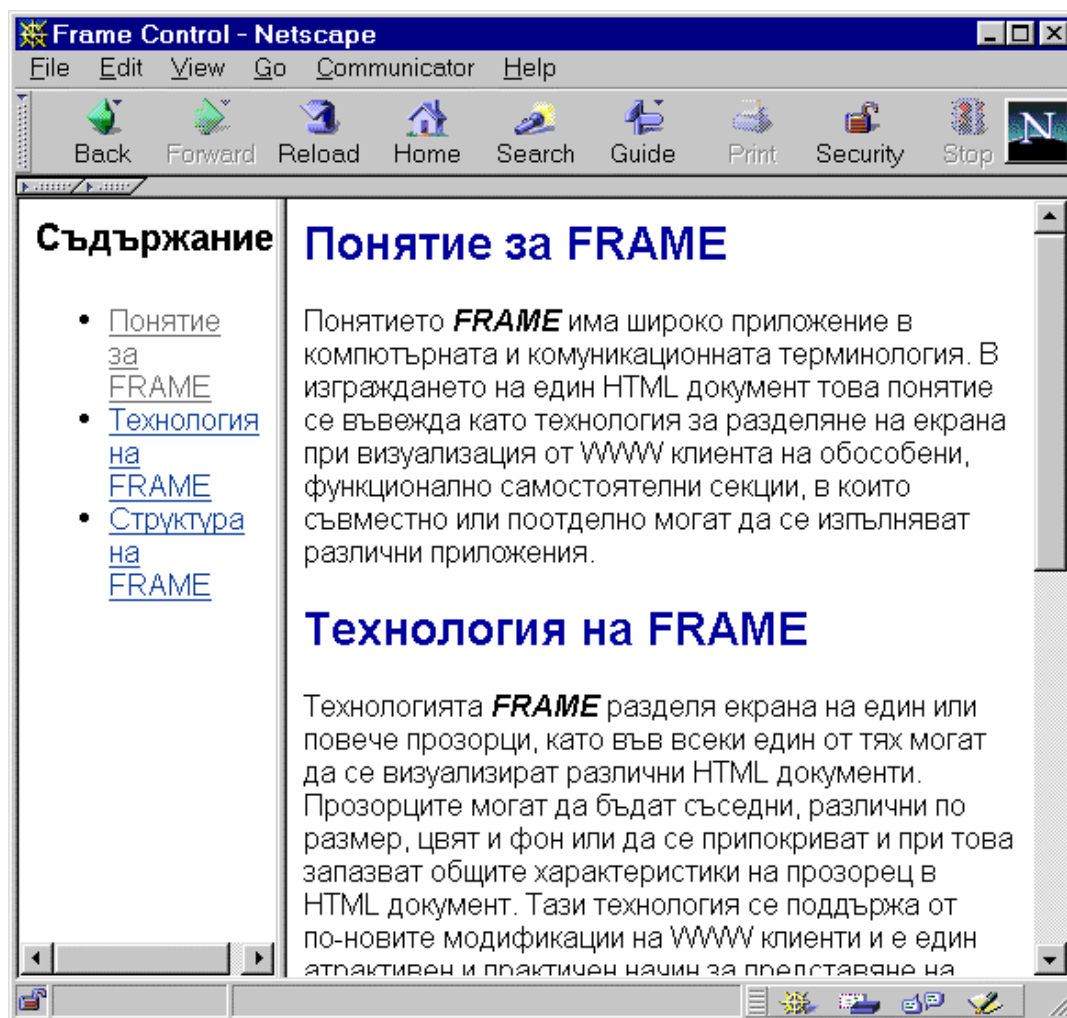
```
<HTML>
  <HEAD>
    <TITLE>frames example 4</TITLE>
  </HEAD>
```

```
<BODY bgcolor=yellow text=black link=red vlink=maroon alink=fuschia>
  <P><B> Прозорец 4 изобразява информация от HTML документ 4
(frame4.htm)</B></P>
</BODY>
</HTML>
```

Връзка между отделните фрейми може да бъде установена чрез команда **TARGET="Име на прозорец"**. Името на прозореца се задава чрез атрибут **NAME** на команда **FRAME**. В съдържанието на документ, изобразен в един фрейм може да се укаже хипервръзка към друг документ, който да бъде изобразен в съседен фрейм. В дефиницията на хипервръзката (****) се задава името на прозореца, в който ще се визуализира съдържанието на документа, определен в описанието на фреймовата секция. Подобна техника се използва за оформяне на HTML страници, които представят съдържанието на книги, фирмена документация, речници или като средство за управление на менюта.

Следващият пример (*Екран 13-4*) представлява по-пълно и функционално завършено представяне на възможностите на фреймовите компоненти. Дефинирани са два вертикални прозореца в съотношение 25% към 75% в полза на десния прозорец. Левият прозорец е описан чрез команда **FRAME**, с име **"Window-L"** и изобразява съдържанието на документ с име **"frame-idx.htm"**. Десният прозорец е с име **"Window-R"** и изобразява съдържанието на документ с име **"doc1.htm"**. Командата **NOFRAME** осигурява алтернативно изобразяване на съдържанието на документа **"frame-idx.htm"** при работа с браузъри, които не интерпретират фрейми. Интерпретацията на показаният по-долу HTML код е изобразена на *Екран 13-4*.

```
<HTML>
  <HEAD>
    <TITLE>Frame Control</TITLE>
  </HEAD>
  <FRAMESET COLS="25%,75%">
    <FRAME SRC="frame-idx.htm" SCROLLING="auto" NAME="Window-L">
    <FRAME SRC="doc1.htm" SCROLLING="auto" NORESIZE NAME="Window-R">
  </FRAMESET>
  <NOFRAMES>
    <A HREF="frame-idx.htm">Click here</A> to Choose No Frames page.
  </NOFRAMES>
</HTML>
```

Екран 13-4 Едновременно представяне на съдържание и свързания с него текст

Левият прозорец изобразява съдържанието на документ, в който се дефинират хипервръзки към секции от съдържанието на документа изобразен в десния прозорец. HTML кодът на документа от левия прозорец е следния:

```
<HTML>
<HEAD>
  <TITLE>Frame - Index</TITLE>
</HEAD>
<BODY TEXT="#000000" BGCOLOR="#FFFFFF" LINK="#0A3990" VLINK="#606060"
ALINK="#FF0000" COLOR="WHITE">
<P><B><FONT SIZE=+1>Съдържание</FONT></B></P>
<UL>
<LI>
<A HREF="doc1.htm#frame1" TARGET="Window-R">Понятие за FRAME</A></LI>
<LI>
<A HREF="doc1.htm#frame2" TARGET="Window-R">Технология на FRAME</A></LI>
<LI>
<A HREF="doc1.htm#frame3" TARGET="Window-R">Структура на FRAME</A></LI>
</UL>
</BODY>
</HTML>
```

Съдържанието на документа изобразен в десния прозорец се зарежда от HTML файл с име "doc1.htm". Описаните хипервръзките сочат към отделни секции в него (<A HREF="doc1.htm#frame2"...), а пренасочване на изображението от активния прозорец (левия) в друг (десния) се осъществява чрез команда **TARGET="Window-R"**, където *Window-R* е името на десния прозорец, зададено в описанието на фрейма.

Локални кадри <IFRAME>

Техниката на вмъкнати кадри се постига с елемента <IFRAME>. Той е стандартизиран в *HTML 4.0*, но се интерпретира за сега само от *Internet Explorer 4.x*, *5.x*. Навигаторите на *Netscape* го игнорират.

Елементът <IFRAME> позволява изобразяване на прозорец на произволно място в HTML документа. В полето на този прозорец се вмъква съдържанието на документ, адресиран от атрибутите на елемента. За разлика от <FRAME> елементът <IFRAME> се вгражда в съдържанието на основния документ, т.е. между елементите <BODY> и </BODY>. Атрибутите му са същите, както при елемента <FRAME>, с изключение на два нови.

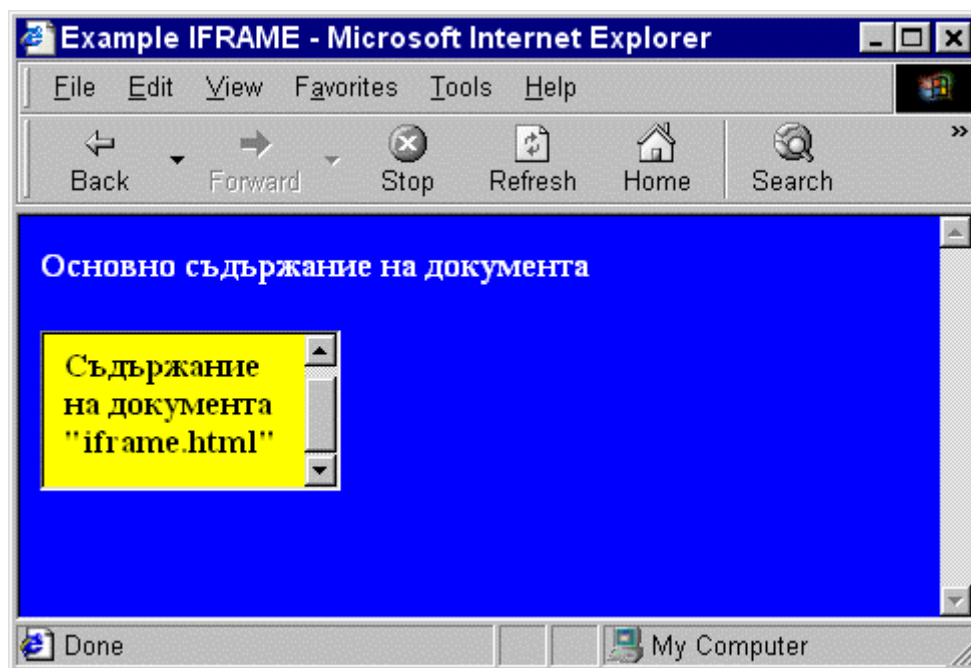
```
<IFRAME> [SRC="URL"] [NAME="WINDOW_NAME"] [NORESIZE] [SCROLLING="YES|NO|AUTO"]
[MARGINWIDTH="VALUE"] [MARGINHEIGHT="VALUE"] [BORDERCOLOR="#HEXCOLOR"]
[FRAMESPACING="VALUE"] [FRAMEBORDER="YES|NO|"] [HEIGHT=n] [WIDTH=n] ...
</IFRAME>
```

Новите атрибути **HEIGHT=n** и **WIDTH=n** задават съответно височината и широчината в пиксели на вмъкнатия в документа прозорец. В съдържанието на елемента трябва да се вмъква алтернативен текст, който ще бъде изобразен от навигатор на *Netscape*.

Следният пример илюстрира използването на елемента <IFRAME>.

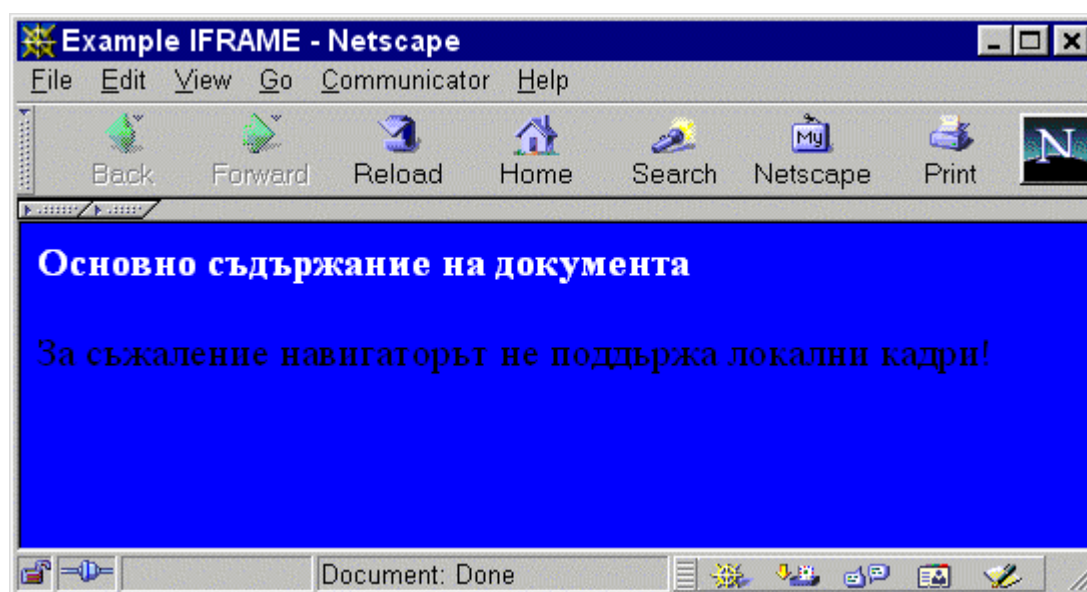
```
<html>
<head><title>Example IFRAME</title></head>
<body bgcolor="blue">
<p><b><font color="white">Основно съдържание на документа</font></b></p>
<IFRAME SRC="iframe.html" NAME="IF11" HEIGHT="80" WIDTH="150"> За съжаление
навигаторът не поддържа локални кадри! </IFRAME>
</body>
</html>
```

Интерпретация на кода с *Internet Explorer 5.0*:



Екран 13-5 Вмъкване на локален кадър в документ

Интерпретация на кода с *Netscape Communicator 4.7*:



Екран 13-6 Вмъкване на локален кадър в документ

Връзка между фрейми

В една обикновена HTML страница с хипервръзка към друга страница, предизвиква изобразяването ѝ в същия прозорец. При използване на фрейми с атрибутите **TARGET** и **NAME** се определя в кой прозорец да се изобразява съдържанието на определен HTML документ. Ако в една хипервръзка за атрибутът **TARGET** се зададе име на прозорец, който не е дефиниран, то документът се изобразява в нов празен прозорец. За управление на връзките между фрейми са изведени резервирани имена на прозорци със следното значение:

Служебно	Значение
----------	----------

<i>име</i>	
_self	Документа от хипервръзката ще бъде изобразен в същия фрейм.
_parent	Документа от хипервръзката ще бъде изобразен в предишната структура.
_blank	Документа от хипервръзката ще бъде изобразен в нов прозорец.
_top	Документа от хипервръзката ще бъде изобразен в същия прозорец с премахване на фреймите.

Начална страница (home page)

Предложен е пример за начална страница на HTML, чието съдържание се отнася до областта на приложение на мултимедията в образованието. Чрез промяна в съдържанието и връзките страницата лесно може да се преобразува за обслужване на произволна друга област.

Даденият по-долу програмен код съдържа всички основни компоненти на езика **HTML**. Използвани са графични изображения във формат **GIF**, **JPEG** и хипервръзки.

[illegible]

```
links</A> </P>
</UL>
<P><HR></P>
<H2><B>Comments or Problems</B></H2>
<P><IMG SRC="www.gif" HEIGHT=48 WIDTH=56 ALIGN=LEFT> Thanks for visiting
our WWW page. We hope to hear from you soon. This WWW page will be
periodically updated.</P>
<P><IMG SRC="temail.gif" HEIGHT=39 WIDTH=35> <I>For information including
questions about courses and services please send e-mail to <A
HREF="mailto:webmaster@multimedia.group.bg">
webmaster@multimedia.group.bg</A>, phone us at +359-2-636-2441, or FAX
your request to +359-2-636-2441.</I></P>
</BODY>
</HTML>
```




MULTIMEDIA SYSTEMS FOR HIGHER EDUCATION

MULTIMEDIA GROUP



AIM

- Multimedia systems should enable the promotion of a new quality of education in the university



OBJECTIVES

- Technology and know-how transfer of multimedia systems
- Establishment of a centre of excellence of multimedia technologies



Project Activities and Output

[Development of curricula, courses and syllabi](#) [New teaching materials developed or adapted](#) [Courses](#)
[Training of university staff](#) [University links](#) [Industry links](#)

Comments or Problems



Thanks for visiting our WWW page. We hope to hear from you soon. This WWW page will be periodically updated.



For information including questions about courses and services please send e-mail to webmaster@multimedia.group.bg, phone us at +359-2-636-2441, or FAX your request to +359-2-636-2441.

Екран 14 - Начална HTML страница

Това е само един пример (Екран 14) без претенции да бъде най-доброто възможно. Тук са включени базови елементи, употребявани в почти всички HTML страници. Основните съображения при създаване на страници са да не се претрупва документа с твърде много информация и информацията да се структурира на части. Подходящото смесване на текст с

графика и цветовата окраска на страницата може да се разглежда и като произведение на изкуството, разбира се, много трябва да се внимава да не се прекали с някои елементи.

Строга рецепта за създаване на HTML страници не може да се даде. Нещата са взаимосвързани, отделните компоненти трябва да се комбинират за четливо, прегледно и изразително представяне на определен вид информация. Чисто текстовите страници са скучни, колкото и съдържателни да са, а тях обикновено никой не ги чете. Запълването на страницата с много графични изображения, икони и аплети удължава времето за изобразяването ѝ на екрана и с нищо не подобрява съдържателната част. Обемисти документи е добре да се структурират на повече HTML страници с подходящи препратки между тях.

Забележка: При възникване на проблеми в процеса на създаване на собствени страници е възможно да се получи помощ в реално време - WWW Help през цялото денонощие посредством групата #WWW от IRC (*Internet Relay Chat*). Помощ не в реално време може да се получи чрез Net News групата **comp.infosystems.www**.



Редактори и филтри за HTML

За потребители, за които е трудно да запомнят командите на езика HTML или искат да избегнат преправянето на вече създадени и форматираны с помощта на Word документи, са разработени специализирани редактори и филтри. Най-простият, но и най-ефективният начин за създаване на HTML страници е използването на текстов редактор за съставяне на кода и WWW клиент за неговата интерпретация. Този вариант изисква добро познаване на елементите на езика и е предпочитан от напредналите потребители.

Редакторите са програми, с които лесно могат да се създават HTML документи. Те се делят на текстови и графични. Много са подходящи редакторите, реализиращи режим *WYSIWYG* (*What You See is What You Get*). В такъв режим на работа авторът вижда непосредствено резултата от неговите действия. Някои редактори притежават собствена интерпретираща среда, други асоциират външни WWW клиенти за интерпретация на HTML кода.

Филтрите предоставят шаблони или макроси за различни, широко разпространени текстообработващи програми, които включват възможност за използване на командите на HTML или извършват преобразуване от вътрешния формат на текстообработващата система в HTML файл.

Използването на редактори и филтри може да улесни изучаването на езика HTML. Потребителят въвежда текст и разглежда непосредствено получения HTML код. Трябва да се отбележи, че редакторите и филтрите често не притежават най-новите възможности езика и на разпределената информационна система Web, което налага някои елементи да се въвеждат "ръчно" в съответния документ. Многократно нанесени корекции в един HTML документ, обикновено добавя или изоставя множество ненужни редове в кода на HTML страницата. Това е един страничен ефект при работа с редактиращите системи.

Изборът на редактор или филтър зависи от предпочитанията на потребителя. Всички те имат сходни възможности. За по-обикновени нужди за предпочитане са HTML филтрите. При необходимост от изграждане на по-сложен HTML проект трябва да се използват HTML редактори. Сложността зависи от броя на връзките, наличието на повече графични елементи, таблици, фреймови конструкции. Голяма част от HTML филтрите са *freeware* или *shareware*

и са достъпни за копиране по Internet. Съществуват HTML редактори *freeware* или *shareware*, които също са общодостъпни. Редакторите с по-сериозни възможности обикновено са комерсиални. В текста, който следва е направен опит за кратко представяне на по-известните HTML редактори и филтри. Сравнителен анализ с функционална оценка на възможностите на голямо количество редактори може да се намери на следния адрес:

<http://homepage.interaccess.com/~cdavis/index.htm>

Редактор HTML Assistant

Програмата е създадена от разработчика на WWW - Howard Harawitz. Работи под Microsoft Windows и осигурява създаването на WYSIWYG документи. Използува външен клиент (Mosaic, Cello или Netscape) за разглеждане на създадените документи. Това предопределя малкия й размер при сравнително богати възможности. В горната част на екрана се намира наборът от инструменти - във вид на бутони за достъп до най-употребяваните HTML елементи. Изборът на съответния елемент предизвиква вмъкването му в текущия документ. Ако командата има отварящ и затварящ елемент, става вмъкването и на двата. Например при избор на бутон *H1* в документа автоматично се добавят `<H1></H1>`. Чрез позициониране на курсора между тези два елемента може да се вмъкне текстът, който трябва да се изобрази като заглавие.

При избор на функция *User Tools* се появява прозорец, съдържащ до 10 бутона. На всеки бутон може да бъде присвоен потребителски низ от бутона *Modify* в същия прозорец. След присвояването избирането на съответния бутон предизвиква появата на низа в текущото място на документа.

Много от другите команди, като *LINK*, *ANCHOR*, *IMAGE*, показват прозорци, позволяващи да се въведат данните за тези команди. Командите *IMAGE* и *LINK* дават възможност за съхранение на най-често използваните имена в списък за бърз достъп.

Редакторът HTML Assistant е лесен за използване с прост потребителски интерфейс и пълен набор от възможности.

Копие на редактора може да се намери на следния адрес:

ftp://ftp.cs.dal.ca/htmlasst/*.

HTML Assistant се разпространява във вид на .ZIP архив, придружен от документация. Последната версия е HTML Assistant Pro с вграден браузър.

HoTMetal

Това е HTML редактор, създаден от SoftQuad Inc. Изборът на команди става от списък. Този редактор притежава вградено средство за визуализация, даващо възможност за предварително разглеждане на документите. За разлика от HTML Assistant HoTMetal е твърде рестриктивен редактор. Това е предимство, което не позволява на потребителя да направи грешка в HTML документа. Редакторът е добро средство за изучаване на HTML.

При вмъкване на команда в документа HoTMetaL автоматично изисква от потребителя въвеждане на задължителните елементи (атрибути) и предлага списък от опциите по избор.

Редакторът HoTMetaL контролира всеки елемент, потребителят не може да променя даден елемент просто чрез въвеждане на текст. Модифицирането на елемент става чрез меню *Markup* и избор на *Change* (или *Edit Links and Attributes* за промяна на вътрешни параметри на елементи).

Меню *View* предлага различни начини за визуализиране на HTML документи. Възможно е скриване на HTML елементите, съдържащи се в даден документ. Това позволява потребителят да види документа в почти същия вид, който би се получил при визуализация с WWW клиент. Други опции на същото меню позволяват контекстно или структурно визуализиране на документи.

Една от най-полезните възможности на HoTMetaL е командата *PUBLISH* в меню *File*. Тази функция позволява създаването на документите да става върху система, различна от тази, в която те след това ще се интерпретират. Един HTML документ може да съдържа връзки към други файлове. При преместване на документа върху друг сървър всички тези връзки трябва да бъдат актуализирани. Чрез командата *PUBLISH* всички връзки могат автоматично да се променят, така че да указват коректния адрес. Тази възможност спестява време и намалява опасността от грешки в процеса на промяната.

Редакторът HotMeTaL е подходящо средство за начинаещи и експериментатори, но опитни програмисти на HTML могат да оценят интелигентния му интерфейс като твърде рестриктивен.

Копие на редактора и допълнителна информация за последните му версии и изменения може да се намерят на следния адрес:

<http://www.sq.com/hotmetal.html>
ftp://ftp.ncsa.uiuc.edu/Web/html/hotmetal/*.

Първият ред извежда представителната страница (*Home Page*) на редактора HotMeTaL, а вторият копира всички файлове, свързани с неговата инсталация. Последната известна версия на редактора е HoTMetaL Pro 3.0, която се разпространява като комерсиална.

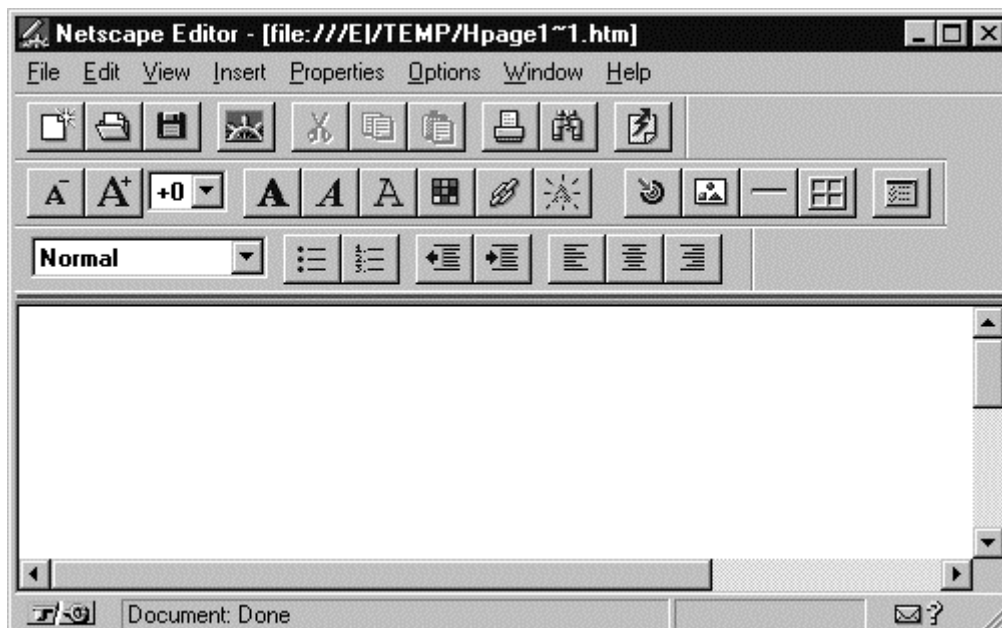
Редактор Claris Home Page

Редакторът *Claris Home Page* е първият *WYSIWYG*, който със своята версия 2.0, освен всички останали функции, е подходящ за създаване на фрейми в HTML страница. Версията е 32-битово приложение подходящо за Macintosh, Power PC, Windows 95/NT за PC. В редактора е интегриран речник на английски и библиотека с множество графични икони и примитиви (*ClipArt*). Редакторът се разпространява само в комерсиална версия.



Редактор в Netscape Gold

Популярният WWW навигатор *Netscape* на Netscape Communications Corporation в своята версия *Gold*, разполага с вграден редактор на HTML страници в режим WYSIWYG (*Екран 14.1*). Версия на навигатора съществува за всички платформи (Windows 3.x, Windows 95/NT, MacOS, Linux, Unix).



Екран 14.1 - HTML редактор в Netscape Navigator Gold

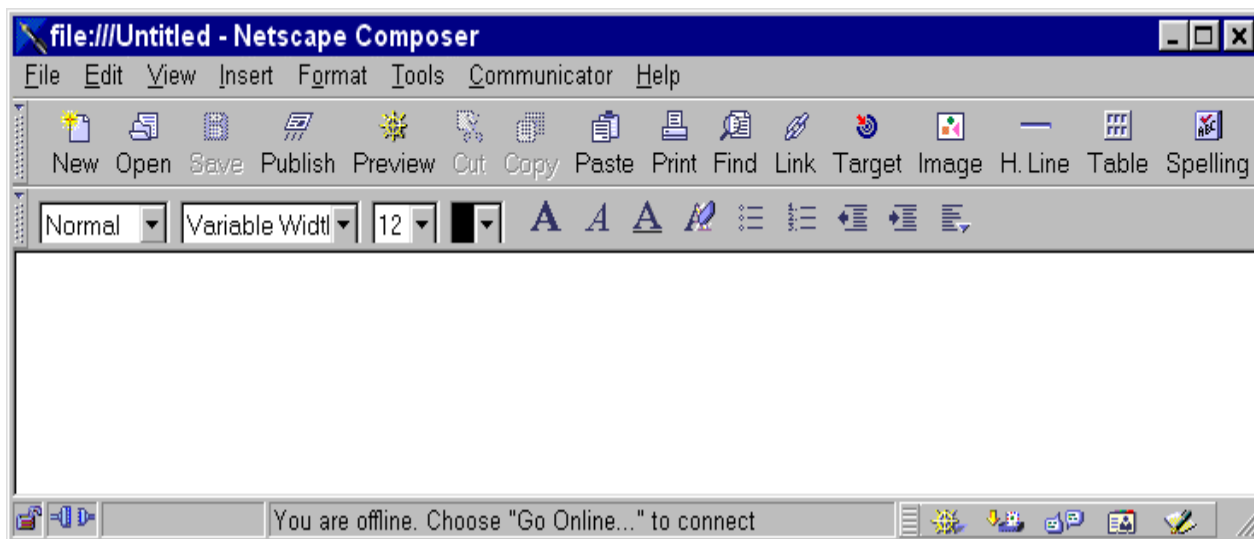
Основните команди са изведени на меню-бутони, така форматиране на текст, изграждане на списъци, таблици, вмъкване на графика и задаване на хипервръзки се извършва с лекота. Атрибутите за основните елементи се избират от диалогови кутии. Достъпното множество от параметри и атрибути за съответните команди е доста ограничени и възможностите за извършване на корекции са затруднени. Съществува възможност за преглеждане на първичния код на HTML страницата, а за редактиране му е необходимо асоцииране на подходящ текстов редактор, например *Notepad* или *EditPad* при липса на по-подходящ. Същата възможност може да се реализира и по отношение на графичните изображения, чрез асоцииране на съответен графичен редактор. Интегрираните редакторски възможности в *Netscape Navigator Gold* го правят предпочитана среда за разработване и поддържане на HTML страници от широк кръг потребители. Той е много подходящ за начинаещи. Актуална информация за навигатора и неговите компоненти може да се намери на адрес:

<http://www.netscape.com/>



Редактор Netscape Composer

Netscape Composer е част от комуникационния център наречен Netscape Communicator 4.x, разработен от Netscape Communications Corporation. Версия на комуникатора съществува за всички платформи (Windows 3.x, Windows 95/NT, MacOS, Linux, Unix). Netscape Composer е WYSIWYG HTML редактор (*Екран 14.2*) със значителни подобрения спрямо своя предшественик Netscape Navigator Gold.



Екран 14.2 - HTML редактор Netscape Composer

Редакторът е подходящ както за начинаещи, така и за напреднали потребители. Възможността за изпълнение на функцията Publish само с един бутон улеснява в значителна степен настройката и управлението на един HTML проект.

Основните функции на редактора са изведени в два реда бутони-инструменти.

Горният ред съдържа списък от бутони за работа с файлове, редактиране на текстови фрагменти, диалогови кутии за управление на основните HTML команди, описание на хипервръзки.

Вторият ред съдържа списъци и бутони за управление на текстовите стилови формати.

Създаването на една HTML страница с този редактор се извършва с лекота. Въвеждането на текст става по обичайния за всеки текстов редактор начин в работното поле. Вмъкването на графични елементи или таблици, както и подравняването им спрямо текста се извършва чрез настройка на съответните атрибути, достъпни от диалогова кутия. Графичните възможности на редактора позволяват промяна в размерите и разположението на графики и таблици в полето на документа директно на екрана със средствата на мишката. Редакторът поддържа едно оптимално множество от команди и техните атрибути на езика HTML 4.0.

Актуална информация за комуникатора и неговите компоненти може да се намери на адрес <http://www.netscape.com/>. Информация за работа с редактора се намира на адрес:

<http://help.netscape.com/docs/client/communicator/Introcomm/Introcom.html>

Редактор FrontPage 98

Редакторът е създаден от Microsoft и е предназначен за работа в среда Windows 95/98 или Windows NT. Версията 98 притежава всички възможности за поддържане и реализация на собствен Web site (WWW точка в Internet). Включени са CGI скриптове за обработка на формуляри и търсене на информация. Предвидена е възможност за достъп до бази данни (IDC - Internet Database Connector). Обработката на графики, таблици и форми се извършва

директно в средата на редактора. С десния бутон на мишката се предизвиква появата на контекстно ориентирани падащи менюта за уточняване на атрибутите и задаване на стойности на необходимите параметри към избраната команда. Със същата лекота се създават и фрейми. За всички основни елементи редакторът предлага на автора шаблони (*Wizards*) за ускоряване на работата. За създаването на горещи точки в графики, редакторът предлага собствени средства за снемане на координатите и автоматично създава *MAP* файла. Затруднения се срещат при редактиране на вече готови HTML страници. Промяна в елементите и техните атрибути не се извършва толкова лесно. Необходимо е да се търси и извежда менюто за всеки отделен атрибут. Редакторът не поддържа всички елементи от версията на езика HTML 4.0. Поддръжката на Java аплети е недостатъчна. Редакторът *FrontPage 98* е със сериозни възможности за създаване на не много големи HTML проекти, което ще го нареди сред най-предпочитаните. Той се разпространява само в комерсиална версия. Допълнителна информация може да се намери на адрес:

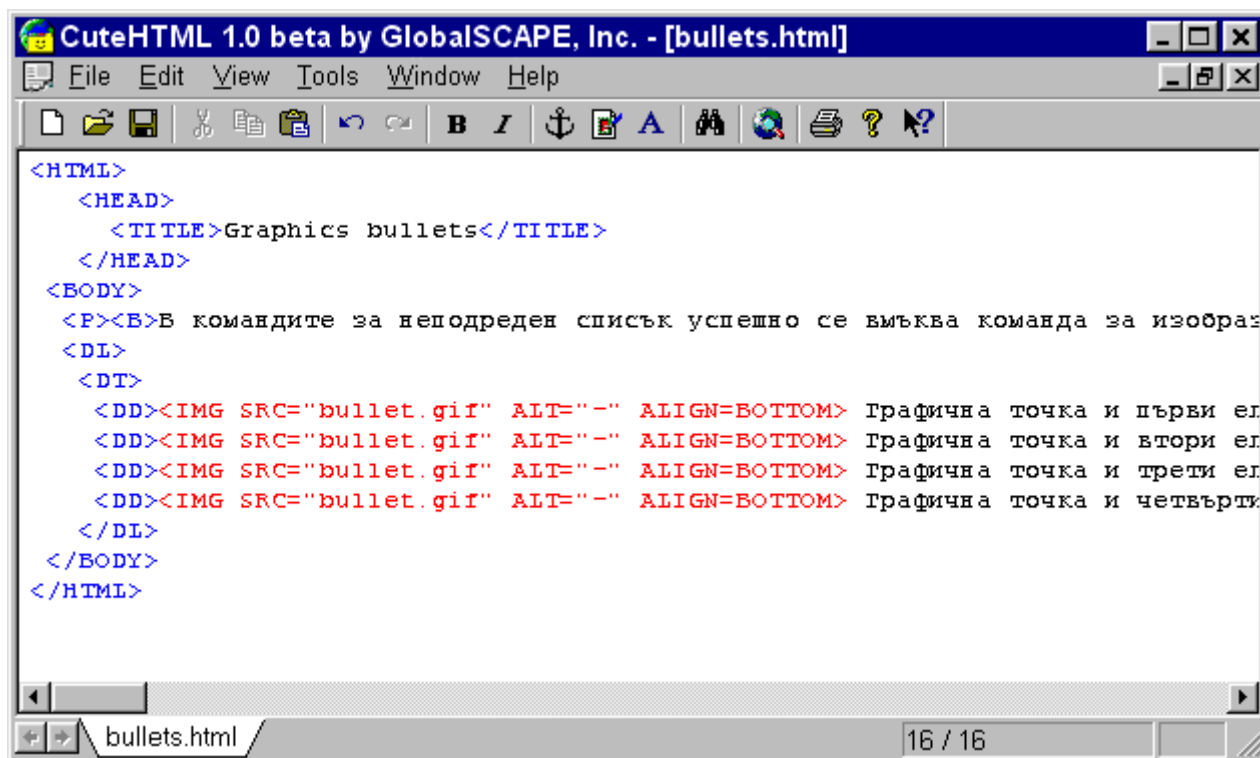
<http://www.microsoft.com/>

Последните модификации на продукта са насочени към администриране на един *Web Site*. *FrontPage* се включва като интегрална част на *MS Office 2000*, а средствата за разработка в *Visual Studio 6* на Microsoft.



Редактор CuteHTML

CuteHTML е текстов редактор, специално ориентиран за разработка и поддръжка на HTML страници, локално или на отдалечен Web сървър. Разпространява се от [Global Scape](#) заедно с FTP-клиента *CuteFTP*. Редакторът е предназначен за работа в среда Windows 95/98/NT (Екран 14.3).



Екран 14.3 - HTML редактор CuteHTML

След инсталация се асоциира с *Windows Explorer* за редактиране на локални HTML файлове, и с *CuteFTP* за редактиране на файлове в отдалечена машина. По FTP протокол избраният файл се извлича в локалната машина и след редактиране по същия протокол се връща в отдалечената машина. За нормалното използване на тази технология потребителят трябва да притежава необходимите права за достъп до отдалечената машина.

Програмата е предназначена за средно и по-напреднали потребители. Директното манипулиране на HTML кода е улеснено от контекстно-ориентирани менюта, но познаването на отделните му елементи и техните атрибути е препоръчително.

CuteHTML поддържа HTML 4.0 и JavaScript 1.1. Командите и техните атрибути се оцветяват различно, синтактично неправилно изписан ред се сигнализира. За предварителна визуализация на кода в параметрите за настройка на редактора се избира използваният Web клиент (*Netscape* или *Internet Explorer*).

Macromedia Dreamweaver

Редакторът може да работи в режим обработка на първичен код и в режим WYSIWYG. Сред професионалистите на Web дизайна се е наложил като лидер и е предпочитан при работа в екип. Класацията на HTML редакторите го поставят на първо място. Във възможностите на редактора влизат всички HTML компоненти, DHTML, CSS1, CSSP, Javascript 1.2. Потребителският интерфейс е построен ергономично, с много улеснения и контекстно ориентирани менюта и помощни фрагменти. Редакторът е предназначен предимно за професионалисти. Начинаещите ще срещнат повече затруднения и трудно биха оценили великолепните му възможности. Версия на редактора от типа *Trial* за Windows 98/NT е достъпна за копиране от Web сайта на [Macromedia](#).

Adobe PageMill

Редактор на HTML страници, разработен в стила на продуктите на Adobe. Нещото, което го отличава от останалите редактори е, че притежата отлични възможности за създаване и обработка на фрейми. Демонстрационна версия може да се копира по Интернет от Web сайта на [Adobe](#).

Филтър CU_HTML

Филтърът е предназначен за работа с Microsoft Word for Windows и е създаден от Kenneth Wong и Anton Lam от университета в Хонг Конг. Той представлява разширение (*Template*) за Word 2.0 и Word 6.0. След зареждането му потребителят може да установи, че към основните стилкови формати са добавени и елементи от HTML. Изборът на елемент предизвиква вмъкването му в текущата позиция на курсора. Филтърът работи в средата на текстообработващата система, което го прави WYSIWYG - документът се изобразява по начин, много близък до този, получен при използване на WWW клиент. В горната част на прозореца се появява меню HTML. В допълнение към него на разположение е група бутони, осигуряващи същите функции като менюто. Изборът на *Insert GIF* от меню HTML позволява търсене на графика във формат *GIF*, нейното избиране и вмъкване в текущата позиция на документа. Опцията *Insert Link* позволява да се създаде връзка към друг документ или обект отново чрез диалогова кутия. Първо е необходимо да се избере думата или фразата, които ще станат избираеми. След уточняване на параметрите на връзката, Word визуализира хипертекста в синьо.

След приключване на работата по създаване на документа от меню HTML се избира опцията *Write HTML*. Word преобразува документа в HTML формат и го съхранява с разширение *.htm*.

Този филтър е удобен, лесен за използване и дава възможност за преобразуване на Word и други типове документи в HTML. Особено атрактивен е за потребители на Word.

Филтър GT_HTML

Филтърът GT_HTML е набор от макроси за преобразуване на един документ от Microsoft Word 6.0 за Windows 3.x, Microsoft Word 7.0 за Windows 95/NT или Microsoft Word 6.0 за Macintosh в HTML страница. Макросът се разпространява като *freeware* продукт по Internet в *.zip* архив с документация. Инсталирането се извършва след дезархивиране и копиране на файла **gt_html.dot** в директория \WINWORD\TEMPLATE или в папката TEMPLATES при Macintosh. Достъп до макроса се осигурява след зареждането му от меню *File - Templates - Add*. Функциите на макроса се ползват от меню-бутони в стила на Word. Активирането им става от *View - Toolbars - Toolbar1(gt_html.dot)*, *Toolbar2 (gt_html.dot)*. Предоставените от макросите възможности покриват основните езикови елементи на HTML. Те са подходящи за бързо преобразуване предимно на текстови документи в HTML страници и за потребители с невъзможност да боравят директно с код.

Списък на най-препоръчваните HTML редактори

Редакторите за създаване на HTML страници и за изграждане на цели сайтове условно се делят на:

- редактори за начинаещи (*Netscape Composer*);
- редактори за професионалисти (*Dreamweaver*).

Редакторите с по-разширени възможности включват създаване на фрейми (*Home Site*) и формуляри (*GoLive*), както и скриптов компоненти на *Javascript* и *VBScript* (*Dreamweaver*, *FrontPage*) във визуална среда за програмиране (*WYSIWYG*). Така наречената визуална среда за програмиране за начинаещите се оказва доста измамна. Те препологат, че с помощните средства на средата ще бъдат в състояние да решат всеки проблем. Разочарованието идва много бързо и тогава разбират, че средата на редактора със своите компоненти създава една добре структурирана база върху която може да се изгражда едно по-сериозно приложение със средствата на програмен код изписан ръчно!

В множеството от най-различни редактори за създаване на Web страници с най-висока оценка са *Dreamweaver* и *FrontPage*.

Редактор	WYSIWYG	URL адрес на сървър
Arachnophilia	не	http://www.arachnoid.com
Macromedia Dreamweaver 3.0	да	http://www.macromedia.com
Microsoft FrontPage 2000	да	http://www.microsoft.com
Microsoft FrontPage Express	да	http://www.microsoft.com
Adobe GoLive 1.0	да	http://www.adobe.com
Home Page	да	http://www.clarishomepage.com

Allaire Home Site	не	http://www.allaire.com
Namo WebEditor	не	http://www.namo.com
Netscape Composer 4.7	да	http://www.netscape.com
Adobe PageMill 3.0	да	http://www.adobe.com
Visual Page	да	http://www.symantec.com
Web Construction Kit	не	http://www.pierresoft.com
WebExpert	не	http://www.visic.com

Динамични прозорци

WWW клиентите *Netscape (Navigator и Communicator)* и *Internet Explorer* на *Microsoft* дават възможност за отваряне на повече от един прозорец в рамките на една сесия, в които да се визуализират различни HTML документи. В средата на *Netscape Communicator* това става чрез меню *File|New|Navigator Window*. Новият прозорец притежава всички възможности и характеристики на отворените преди него.

Този процес може да бъде управляван и с помощта на HTML елементите в един документ. Отваряне на нов прозорец се предизвиква от следната команда:

```
<A HREF="URL" TARGET="Име_на_прозорец"> Текст </A>
```

Свързващото звено в случая е атрибутът *Name="име"*, който се употребява за наименоване на:

- определена област от документ;
- цял документ;
- прозорец, създаден като секция (*Frameset*).

Ефективното използване на тази техника позволява контрол и управление на изобразяваните документи в различните прозорци отворени на екрана. От един управляващ прозорец може да се разпределя визуализацията на различни HTML документи в точно определени прозорци.

Обръщението към определен прозорец се осъществява чрез атрибута *Target*:

- в дефиниция за [хипервръзка](#):

```
<A HREF="URL" TARGET="Име_на_прозорец">Свързващ текст</A>
```

- в команда [Base](#) от заглавната секция *Head*:

```
<BASE TARGET="Име_на_прозорец">
```

- в команда [Area](#) за дефиниране на област:

```
<AREA SHAPE="shape_type" COORDS="x,y,..." HREF="URL" TARGET =
"Име_на_прозорец">
```

- в команда [Form](#) за описание на формуляр:

```
<FORM ACTION="URL" TARGET="Име_на_прозорец">
```

На атрибута *Target* трябва да се присвои символен низ, започващ с буква (a-z, A-Z). В други комбинации Web клиентът (*User Agent*) може да не интерпретира правилно името. При особени случаи, когато се използват служебни наименования, присвоеното му значение започва с "_". Следният списък от [Target](#) имена са резервирани и имат специално приложение:

- **TARGET="_blank"** - предизвиква отваряне на нова (не наименована) страница;
- **TARGET="_self"** - отваря страница в същия прозорец, спрямо точката на извикване, без да ги размножава;
- **TARGET="_parent"** - извикване на кореновата страница или средство за преминаване в единен прозорец. При отсъствие на изявен корен ефектът е същия както при **"_self"**;
- **TARGET="_top"** - извикване на първата страница от хронологичния списък.

Тази техника е използвана в настоящия Web Site при представянето на този текст. Бутонът **Index** в края на всеки екран е асоцииран към следната команда:

```
<A HREF=" ../book-2.htm" Target="_parent">
```

Това предизвиква връщане към първия екран със съдържанието на отделните части в същото копие на браузъра. Бутонът **Home** в края на всеки екран е асоцииран към следната команда:

```
<A HREF=" ../backhome.htm" TARGET="Back_To_Home">
```

което, предизвиква стартирането на ново копие на браузъра за изобразяване на съдържанието на документа *backhome.htm*. В неговата заглавна част (*<Head> ... </Head>*) е употребена командата *<Base>* в следния вид:

```
<BASE TARGET="Back_To_Home">
```

Динамични шрифтове - OpenType & TrueDoc

Интегрирането на шрифтове в HTML страница е изключително удобство за всички създатели на Web страници. В практиката често се налагаше текстови фрагменти да се представят в графичен вид в стремежа на автора да създаде желана от него визия. Системата на динамични или внедрени шрифтове (*Embedded fonts*) предполага HTML страницата да бъде изобразена при клиента с определените от автора шрифтове, независимо от навигатора, защото шрифта се предоставя заедно с прехвърлянето на съдържанието на страницата при клиента.

Основни шрифтови формати.

- **PostScript Type 1** - формат създаден от *Adobe* (един от първите).
- **TrueType** - първоначално създаден от *Apple*, възприет след това и от *PC*.
- **OpenType** - нов формат, създаден от *Microsoft* и *Adobe* в стремеж да се съчетят предимствата на *PostScript* и *TrueType* форматите.

За целите на Web технологията се оказва невъзможно шрифтове да бъдат внедрявани в техния първоначален формат. *Netscape* и *Microsoft* създават за технологията **Embedding Fonts** два нови преносими формата, които се получават след преобразуване с програмни средства от първоначалните шрифтови формати.

- **Embedded OpenType** (.eot файл) - предоставя се от *Microsoft*.
- **TrueDoc** (.pfr файл) - предоставя се от *Netscape* и *Bitstream*.

Тези два съперничаещи си Web ориентирани шрифтови формата имат множество ограничения. Първо, не работят с всички видове шрифтове. Второ, от различните Web браузъри се интерпретират по различен начин.

Внедрен формат	Шрифтов формат	Браузъри
Embedded OpenType	TrueType, OpenType	От Internet Explorer 4.x
TrueDoc	TrueType, PostScript Type 1	От Netscape Navigator 4.03, Internet Explorer 4.x (само за Windows)

От таблицата е видно, че *TrueType* шрифтовете се явяват общи за внедрените шрифтови формати и Web браузърите.

CSS1 и динамичните шрифтове

Системата на динамични шрифтове е дефинирана като разширение на шрифтовите описатели в [CSS1](http://www.w3.org/TR/CSS1). Това включва едно множество от шрифтови дескриптори, чрез които използваният шрифт се описва с неговите индивидуални характеристики и адрес от който може да бъде изтеглен. Работната спецификация на *W3C*, описваща динамичните шрифтове се намира на адрес:

<http://www.w3.org/TR/WD-font>

Шрифтът се описва в секцията за стилова спецификация чрез своите дескриптори, които се унаследяват при използване на съответните форматни спецификации за изобразяване на текстови елементи. Общият формат за описание на шрифтова спецификация е следният:

```
@font-face { <font description> }
```

където ** съдържа списък от дескриптори:

```
descriptor: value;
descriptor: value;
[...]
descriptor: value;
```

Системата на динамични шрифтове се представя при *Internet Explorer* като интегрирани шрифтове - **OpenType**, докато при *Netscape Navigator* като динамични шрифтове - **TrueDoc**. Двете технологии се различават и съвместяването им е затруднено. Това в някаква степен компрометира една много практична идея.

Интегрирането на шрифт към HTML страница се осъществява посредством команда **@font-face** в секцията за стилов формат:

```
<STYLE TYPE="text/css">
<!--
@font-face
{
  font-family: font-family-name;
  font-style: font-style-name;
  font-variant: font-variant-name;
  font-weight: value;
  font-size: value
  src: local(font-name); url(URL);
  format: (Font Type);
}
//-->
</STYLE>
```

Стиловата дефиниция за интегриране на шрифта се указва с секцията **<HEAD> ... </HEAD>** на HTML документа.

- **font-family:** *Arial, Verdana, Hebar, Times*; - указва наименованието на фамилията на шрифта.
- **font-style:** *normal, italic, oblique*; - задава стила на шрифта.
- **font-variant:** *normal, small-caps*; - определя нормално изписване на шрифта или главни букви в намален размер.
- **font-weight:** *all, normal, bold, 100 & 800*; - определя дебелината на шрифта.
- **font-size:** *all, length*; - определя размера на шрифта.
- **src:** *local; url*; - задава местонахождението на шрифта. При **local** шрифта се търси в системата на клиента. При **url** шрифта се изтегля от сървъра за да бъде препратен на клиента. Едновременната употреба на двата атрибута отменя необходимостта всеки път един и същи шрифт да бъде прехвърлян в клиентската машина, ако той вече се намира инсталиран там.
- **format:** *opentype|truedoc*; - задава типа на шрифта.

```
<STYLE TYPE="text/css">
<!--
@font-face
{
  font-family: "CableCyr"; src: url(fonts/cablecyr); format: (opentype);
}
```

```
//-->
</STYLE>
```

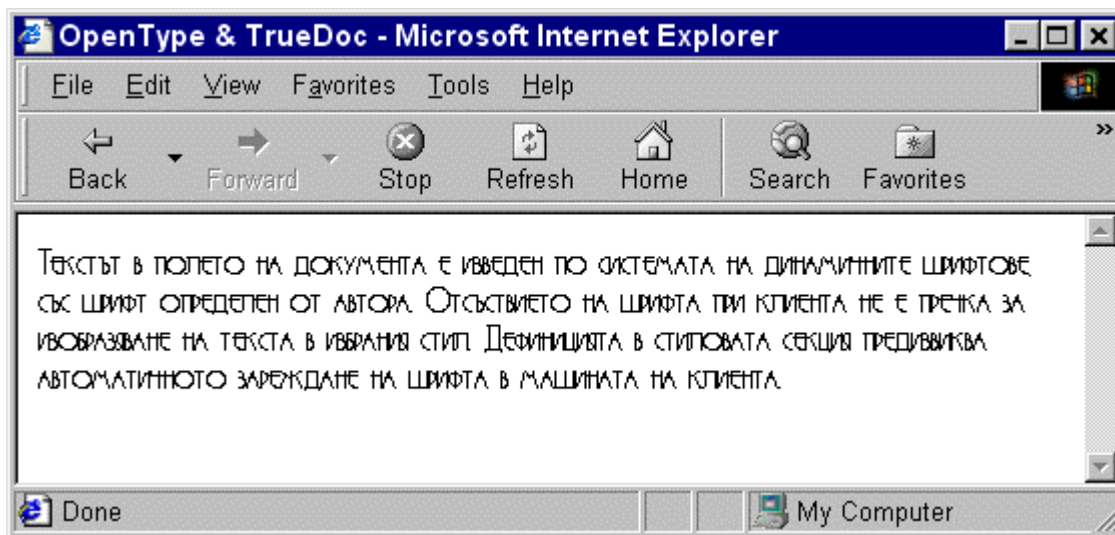
В този пример се прехвърля шрифт с име *CableCyr* от директория *fonts* и шрифта е от тип *OpenType (Microsoft)*.

Връзката със стиловата дефиниция се извършва посредством HTML командата ** ... ** при извеждане на конкретен текст в полето на документа.

Следващият пример илюстрира визуализацията на текст в полето на документа с избран от автора шрифт. Атрибутът *local* в параметъра *src*: осигурява използването на локалното копие на шрифта, ако той е налице при клиента, в противен случай се зарежда от указания URL адрес:

```
<html>
<head>
<title>OpenType &amp; TrueDoc</title>
<style TYPE="text/css">
<!--
@font-face
{
font-family: "cabelcyr"; src: local (cablecyr); url(fonts/cabelcyr); format:
(opentype);
}
//-->
</style>
</head>
<body>
<font face="CabelCyr">Текстът в полето на документа е изведен по системата на
динамичните шрифтове, със шрифт определен от автора. Отсъствието на шрифта при
клиента не е пречка за изобразяване на текста в избрания стил. Дефиницията в
стиловата секция предизвиква автоматичното зареждане на шрифта в машината на
клиента.</font>
</body>
</html>
```

Визуализация на кода след интерпретация: (Екран 15)



Екран 15 - Система на динамични и интегрирани шрифтове.

Проектантите и авторите на Web страници трябва да прибегват към тази техника само в онези случаи, в които това действително се налага. Големината на използваният шрифт влияе на времето на зареждане на HTML страницата при клиента. Действително, "ефектните" шрифтове обикновено са с по-голям обем от "стандартните".

Средства за създаване на Web Embedding Fonts.

За преобразуване на шрифтове в *Embedded OpenType format* фирмата Microsoft предоставя програмно средство свободно за използване (freeware) **WEFT** (*Web Embedding Fonts Tool*). Продуктът е лесен за инсталиране и ползване. С негова помощ се създава своеобразна база данни на наличните в системата инсталирани шрифтове с индикация за всяка фамилия дали е подходяща за преобразуване в *Embedded OpenType*. Резултатът от работата на продукта е файл с разширение **.eot** и автоматично обновяване на HTML страницата, използваща този шрифт, с необходимите скриптові елементи. За всеки използван шрифт в страницата се създава отделен файл, който трябва да бъде прехвърлен от сървъра към клиента заедно със съдържанието ѝ. Тук липсва ефекта на прогресивно изобразяване, което се отразява на времето за зареждане на страницата. Фактът, че не се използва оригиналният шрифт с неговите пълни достойнства, а негово преобразувано отражение влошава в известна степен качеството на изобразявания текст. По този начин, от друга страна, се защитава шрифта, т.е. потребителят не може в последствие да го използва за свои цели. Нещо повече, продуктът позволява записване в **.eot** файла само на използваните в страницата символи.

Средства за създаване на TrueDoc Fonts.

За създаване на динамични *TrueDoc* шрифтове, съвместими с навигаторите на *Netscape*, на пазара на програмни продукти се предлагат две помощни средства.

- **WebFont Maker** заедно с около 200 *TrueType* шрифта за *Windows* и *Mac* и обявена цена по Интернет около 200\$.
- **HexWeb Typography** за *Windows* и *Mac* и *BEdit plug-in*, с обявена цена на Интернет 140\$.

Двата продукта предлагат демонстрационни версии, достъпни за копиране след регистрация на потребителя на техните сайтове.

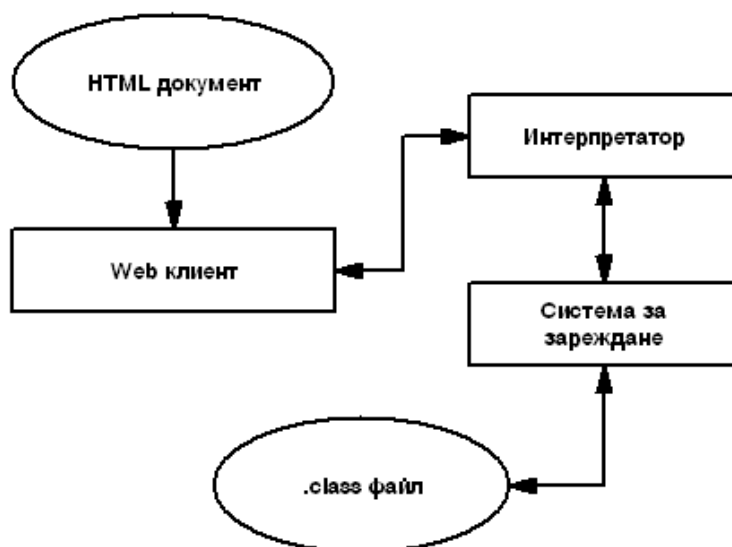
Двата продукта работят по подобен начин при създаване на *TrueDoc format*. За всеки символ от шрифта се създава образ, който се записва във файл с разширение *.pfr* (*portable font resource*) и се свързва със съответния HTML документ. При потребителя браузърът, със съответен декодер на символните образи, превръща всеки образ в елемент от динамичен шрифт за изобразяване на текста. За разлика от технологията *Embedded OpenType*, където по мрежата се прехвърля преобразуван шрифт, за *TrueDoc* се предава информация за образа на символа.

Включване на Java аплети

Елементът **<APPLET>** се използва за включване в HTML документ на външна програма, наречена аplet и написана на обектно-ориентирания език за програмиране **Java**. Външното приложение се зарежда в машината на клиента от Web сървъра. Изпълнението на програмния код става в средата и под контрола на Web браузъра. За тази цел клиентският навигатор трябва да притежава възможността за стартиране на *виртуална Java машина* (*Java VM*). Версиите на *Netscape Navigator* след 2.02 и *Internet Explorer* след 3.02 са в състояние да интерпретират Java аплети.

Езикът за програмиране **Java** притежава множество достойнства, характерни за модерните програмни езици. Той е обектно-ориентиран и машинно независим. Едно от различията между *Java* и *JavaScript* се състои в това, че докато кодът на *JavaScript* е вграден в HTML кода на документа и се интерпретира непосредствено от Web браузъра, то кодът на *Java* предварително е компилиран до **псевдо-код**, след което може да бъде интерпретиран на произволна платформа чрез съответния **Java интерпретатор** след неговото зареждане.

Аpletът е по-особен вид Java програма, която е състояние да използва ресурсите на Web браузъра и да обогати представянето на информацията с анимация, звук и други специални ефекти. Аplet технологията е представена схематично на *Фигура 16-1*.

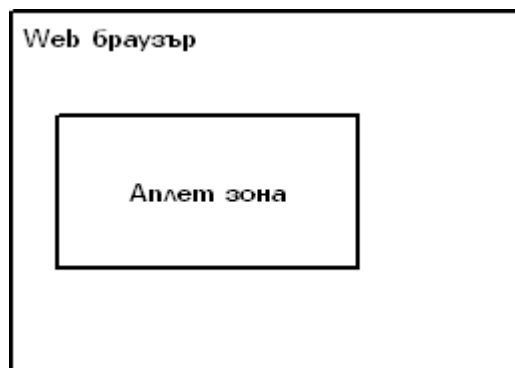


Фигура 16-1 - Аplet технология

От показаното на фигурата става ясно, че след зареждане на HTML документа, Web браузърът активира Java интерпретатора. Системата за зареждане извлича аплета и започва неговото изпълнение в полето на документа. Аpletът се изпълнява от *Java Runtime Interpreter*,

свързан с брауъра. Web брауърът действа като преходна среда между *Java Virtual Machine* и потребителския интерфейс, с което се гарантира и сигурността за клиента.

При наличие на аплет в една Web страница, Web брауърът изобразява неговото съдържание. Изходите от аплета се изобразяват в предварително дефинирана зона, част от полето на документа (Фигура 16-2).



Фигура 16-2 - Извеждане на аплет в зоната на Web брауър

Създаването на собствени аплети предполага познания и умения по съставяне, прилагане и използване на алгоритми и познания за програмиране на обектно-ориентиран език. Без тези предпоставки, потребителят може да използва готови аплети, достъпни за копиране от публичното Web пространство. Използването на готови или собствени аплети дава възможност да се внесе интерактивност и динамика в иначе статичните по принцип HTML страници. Динамичното обновяване на съдържанието на една HTML страница е технология, която все повече се налага от потребностите на клиента да разплага в реално време с актуална информация, бърз достъп до бази данни, експониране на мултимедийни сцени. Елементите на езика **Java** ще бъдат разгледани в отделна глава.

Готови аплети (*javacode.class*) могат да бъдат копирани за използване от публичното Web пространство на следните адреси:

http://dir.yahoo.com/Computers_and_Internet/Programming_Languages/Java/Applets/

<http://www.javasoft.com/applets/>

<http://www.electrinet.com/>

Интегрирането на един компилиран аплет в HTML документа става посредством елемента **<APPLET>**, въведен от стандарта HTML 3.2. В стандарта HTML 4.0 се въвежда друга възможност за интегриране на аплети чрез елемента **<OBJECT>**.

Елементът **<APPLET>** е от непразен вид и има завършващ елемент **</APPLET>**.

```
<APPLET CODE="JavaCode.class" [CODEBASE="Path"] WIDTH=n HEIGHT=n [ALT="Text"]
[ALIGN=TOP|BOTTOM|MIDDLE|RIGHT|LEFT] [HSPACE=n] [VSPACE=n] [BORDER=n] >
[<PARAM NAME=cName VALUE="String">]
```

```
[<PARAM NAME=cName VALUE="String">]
....
</APPLET>
```

- **CODE="JavaCode.class"** - името на аплета за интегриране;
- **CODEBASE="Path"** - определя пътя до аплета (*файла .class*), подразбира се текущата директория на HTML файла;
- **WIDTH=n** - задава широчината на прозореца за изобразяване на аплета в пиксели;
- **HEIGHT=n** - задава височината на прозореца за изобразяване на аплета в пиксели;
- **ALT="Text"** - възможност за задаване на заместващ текст при браузъри, неподдържащи Java (*"Вашият Web навигатор не поддържа Java, снабдете се с подходящ"*);
- **ALIGN=Align_value** - определя подравняването на аплета в страницата, подразбира се *align=left*;
- **HSPACE=n** - задава в пиксели хоризонталното отместване на съдържанието на аплета в дефинираната графична зона;
- **VSPACE=n** - задава в пиксели вертикалното отместване на съдържанието на аплета в дефинираната графична зона;
- **BORDER=n** - задава дебелината в пиксели на рамката, ограждаща графичното поле за изобразяване на аплета.

На аplet могат да се предават параметри, особено в случаите, в които неговата функция подлежи на параметризация. Параметрите и техните стойности се задават във вид на списък, формиран от елемента <PARAM>, в отделна секция на елемента **<APPLET>**, където:

- **NAME** - наименование на параметъра;
- **VALUE** - стойност за параметъра.

Включването на елемента **<APPLET>** в кода на **HTML** страница се извършва по следната обща схема:

```
<HTML>
<HEAD>
  <TITLE> ... </TITLE>
</HEAD>
<BODY>
  ...
  <APPLET> ... </APPLET>
  ...
</BODY>
</HTML>
```

Следният пример илюстрира използването на предварително подготвен аplet (*fillarc.class*) за извеждане в полето на HTML документ на кръгова диаграма с оцветени сегменти.

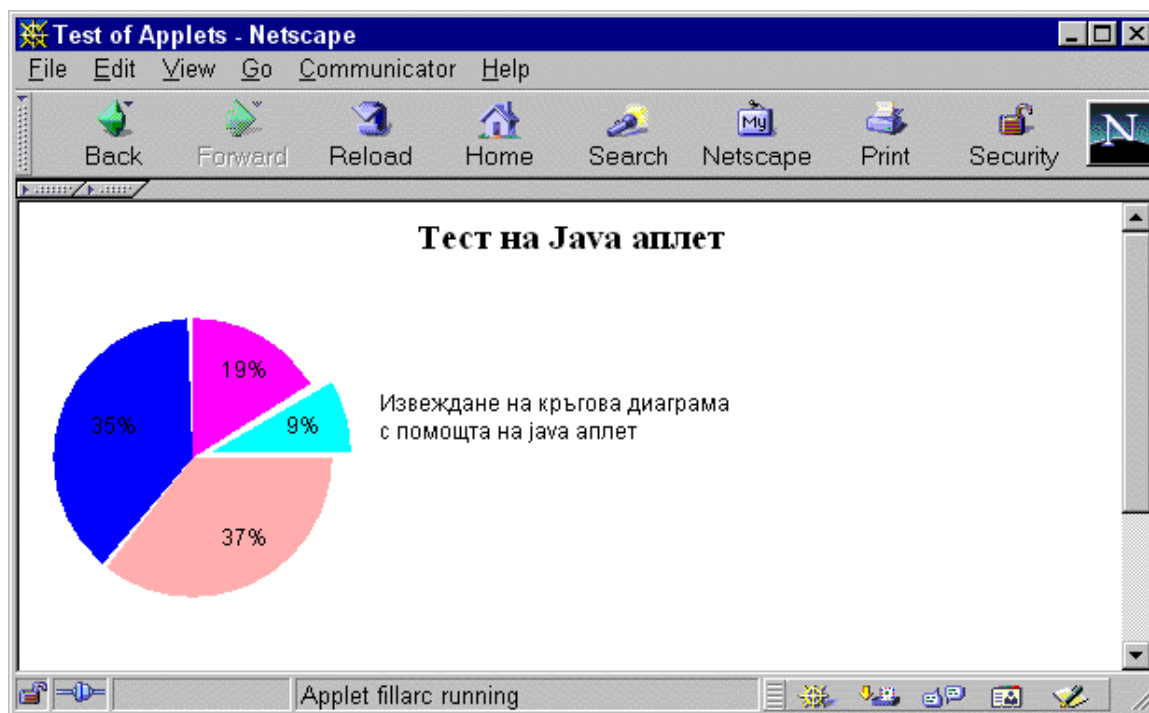
```
<HTML>
<HEAD>
  <TITLE>
    Test of Applets
  </TITLE>
  ....
```

```

<center><p><b>Тест на Java аplet</b></p></center>
<APPLET CODE="fillarc.class" WIDTH = 540, HEIGHT=300>
  Вашият навигатор не е съвместим с Java
</APPLET>
</BODY>
</HTML>

```

Интерпретация на кода в полето на браузър - Екран 17.



Екран 17 - Използване на Java аplet в HTML страница

Указател на HTML елементи

В таблица е представен указател на по-голямата част от HTML елементите, които влизат в състава на стандарта HTML 4.0 с кратко описание на приложението им.

Азбучен каталог

Елемент	Описание
<ABBR>	Съкращение
<ACRONYM>	Акроним
<ADDRESS>	Задава URL в текстов вид
<A HREF>	Задава хипервръзка
<APPLET>	Вмъкване на аplet в HTML документ
<AREA>	Дефинира зона за многозначен избор
	Текстов атрибут - получер

<BASE>	Базов URL адрес за текущата HTML страница
<BASEFONT>	Базов размер на шрифт
<BDO>	Забрана за смяна на посоката при изписване на текст
<BIG>	Задава шрифт с по-голям размер
<BLOCKQUOTE>	Форматира текст като цитат отместен в дясно
<BODY>	Тяло на HTML документ
 	Преминава на нова линия
<BUTTON>	Дефинира бутон във формуляр
<CAPTION>	Дефинира област за заглавие на таблица
<CENTER>	Атрибут за центриране на текст, графика, таблица
<CITE>	Текст, представен като цитат
<CODE>	Текст, представен като първичен код на програма
<COL>	Дефинира колона в таблица
<COLGROUP>	Прегрупиране на колони в таблица
<DD>	Задава списък като дефиниция
	Информация за документ
<DFN>	Логически атрибут за извеждане на текст в курсив
<DIR>	Дефинира списък от множество размерности
<DIV>	Структурира елементи в блок
<DL>	Задава списък от дефиниции
<DT>	Елемент от списък от дефиниции
	Задава текст в курсив
<FIELDSET>	Прегрупиране на зони във формуляр
	Задава атрибути на текст
<FORM>	Дефинира формуляр
<FRAME>	Дефинира съдържание на секция
<FRAMESET>	Дефинира секция
<H1>	Заглавие ниво 1
<H2>	Заглавие ниво 2
<H3>	Заглавие ниво 3
<H4>	Заглавие ниво 4
<H5>	Заглавие ниво 5
<H6>	Заглавие ниво 6
<HEAD>	Дефинира заглавната част на HTML документ
<HR>	Извежда хоризонтална линия

<HTML>	Заглавен елемент на HTML документ
<I>	Атрибут за извеждане на текст в курсив
<IFRAME>	Вмъква прозорец в текст
	Вмъква графика
<INPUT>	Въвежда информация от клавиатура във формуляр
<INS>	Отделя измененията в един документ
<ISINDEX>	Дефиниране на средство за търсене
<KBD>	Логически атрибут за извеждане на текст с шрифт куриер
<LABEL>	Въвежда етикет във форма
<LEGEND>	Легенда за колоните от таблица
	Задава елемент от списък
<LINK>	Задава връзки за документа в секцията <HEAD>
<MAP>	Въвежда описание на графична зона за многозначен избор
<META>	Мета данни за активния HTML документ
<NEXTID>	Сочи следващ документ
<NOFRAMES>	Извежда алтернативно съдържание на елемент секция
<NOSCRIPT>	Извежда алтернативно съдържание на скрипт
<OBJECT>	Вмъква обект (мултимедия, аплет, скрипт)
	Формира подреден списък
<P>	Задава начало на параграф
<PARAM>	Дефинира параметър за вмъкнат обект (аплет)
<PRE>	Преформатиран текст
<Q>	Дефинира блок като цитат
<S>	Атрибут за подчертаване на текст
<SAMP>	Атрибут за извеждане на текст с шрифт с фиксирана широчина
<SCRIPT>	Вмъкване на скрипт (JavaScript)
<SELECT>	Дефиниране на списък във формуляр
<SMALL>	Логически атрибут за извеждане на текст с по-малък шрифт
	Структуриране на елементи в една линия
<STRIKE>	Логически атрибут за извеждане на зачертан текст
<STYLE>	Дефиниция на стилев формат
<SUB>	Логически атрибут за извеждане на долен индекс
<SUP>	Логически атрибут за извеждане на горен индекс
<TABLE>	Дефинира структурата на таблица
<TBODY>	Прегрупиране на таблици

<TD>	Задава клетка от таблица
<TEXTAREA>	Дефинира поле за въвеждане на текст
<TFOOT>	Прегрупиране на таблици
<TH>	Дефинира клетка за заглавие на таблица
<THEAD>	Прегрупиране на таблици
<TR>	Дефинира ред в таблица
<TT>	Texte sous forme de type machine a écrire
<U>	Извежда текст с подчертаване
	Дефинира неподреден списък
<VAR>	Текст под формата на променлива

Предметен каталог

Общи елементи, съдържат се във всички HTML страници	
<HTML></HTML>	Маркира начало и край на файл
<HEAD></HEAD>	Описателна заглавна част
<TITLE></TITLE>	Заглавие на страница
<BODY></BODY>	Съдържателна част на страницата
Структурни дефиниции, изобразяването им се контролира от Web браузъра	
<H? ALIGN=LEFT CENTER RIGHT> </H?>	Заглавие, където "?" определя размера на буквите от 1 до 6
<DIV ALIGN=LEFT RIGHT CENTER JUSTIFY> </DIV>	Дефинира секция с атрибути за подравняване
<BLOCKQUOTE></BLOCKQUOTE>	Форматира текст като цитат отместен в дясно
	Задава текст в курсив
	Задава текст в получер
<CITE></CITE>	Задава текст в курсив като цитат
<CODE></CODE>	Извежда първичен текст на програма
<SAMP></SAMP>	Извежда текст с шрифт с фиксирана широчина
<KBD></KBD>	Извежда текст с шрифт куриер
<VAR></VAR>	Текст под формата на променлива
<DFN></DFN>	Извежда текст в курсив (<i>лог. атрибут</i>)

<ADDRESS></ADDRESS>	Задава URL в текстов вид и курсив
<BIG></BIG>	Задава шрифт с по-голям размер
<SMALL></SMALL>	Задава шрифт с по-малък размер
Стилови формат, контролират стилово представяния текст	
	Извежда текст в получер
<I></I>	Извежда текст в курсив
<U></U>	Извежда текст с подчертаване
<STRIKE></STRIKE> <S></S>	Извежда зачертан текст
	Извежда долен индекс
	Извежда горен индекс
<TT></TT>	Извежда текст в шрифт с фиксирана широчина
<PRE WIDTH=?> </PRE>	Преформатиран текст, ? - брой символи
<CENTER></CENTER>	Атрибут за центриране на елемент
<BLINK></BLINK>	Извежда мигащ текст
	Задава размер на шрифт в относителни единици (-2, +6)
<BASEFONT SIZE=?>	Задава размер на основен шрифт в относителни единици (1-7)
	Задава цвят на буквите в текста
	Задава име на шрифт
<MULTICOL COLS=? GUTTER=? WIDTH=?> </MULTICOL>	Управление на многоколонен обект
<SPACER TYPE=horizontal vertical block SIZE=? WIDTH=? HEIGHT=? ALIGN=left right center>	Дефинира разположението на празна зона
Хипервръзки и графика	
	Задава URL адрес за хипервръзка
	Задава URL адрес за хипервръзка и етикет в документ
	Хипервръзка в текущ документ
	Задава URL адрес за хипервръзка и име на прозорец за визуализация
	Задава етикет за хипервръзка
	Изобразява графичен файл от URL адрес, с уточняване на атрибути за управление

LOWSRC="URL" ISMAP USEMAP="URL">	размери и отстояние, карта и рамка
<MAP NAME="***"></MAP>	Въвежда описание на графична зона за многозначен избор
<AREA SHAPE="RECT" COORDS=",,, " HREF="URL" NOHREF>	Дефинира графична зона за избор
<META HTTP-EQUIV="Refresh" CONTENT="?"; URL=URL">	Дефинира презареждане на страница от адрес
<EMBED SRC="URL" WIDTH=? HEIGHT=?>	Вмъкване на външен обект в страницата
Разделители	
<P [ALIGN=LEFT CENTER RIGHT]></P>	Параграф
<BR [CLEAR=LEFT RIGHT ALL]>	Нов ред в параграф
<HR [ALIGN=LEFT RIGHT CENTER SIZE=? WIDTH=? WIDTH="%" NOSHADE]>	Хоризонтална разделителна линия и параметри
<NOBR></NOBR>	Възпрепятства прекъсването на линия
<WBR>	Указва къде да бъде прекъснат реда при нужда
Списъчни структури	
<UL [COMPACT TYPE=DISC CIRCLE SQUARE]> 	Формира неподреден списък
<LI [TYPE=DISC CIRCLE SQUARE]>	Задава елемент от неподреден списък
<OL [COMPACT TYPE=A a I i]1 START=?]> 	Формира подреден списък
<LI [TYPE=A a I i]1 VALUE=?]>	Задава елемент от подреден списък
<DL COMPACT> <DT> <DD> </DL>	Формира дефиниция, където <DT>=термин, <DD>=дефиниция
<MENU COMPACT> </MENU>	Списък тип меню, където задава всеки елемент
<DIR COMPACT></DIR>	Списък тип секция, където задава всеки елемент
Управление цветността на страница	
<BODY BACKGROUND=" URL" BGCOLOR="#\$\$\$\$\$\$" TEXT="#\$\$\$\$\$\$" LINK="#\$\$\$\$\$\$" VLINK="#\$\$\$\$\$\$" ALINK="#\$\$\$\$\$\$">	Дефинира цветовете компоненти на една страница, с адрес на графичен тапет, основен цвят на фона, цвят на текста и цвят на хипервръзки
Специални символи	
&lt;	Извежда символ "<"
&gt;	Извежда символ ">"

&amp;	Извежда символ "&"
&quot;	Извежда символ ""
&#174; или &reg;	Извежда символ за запазена марка "TM"
&#169; или &copy;	Извежда символ за запазени права "Copyright"
&nbsp;	Извежда празна позиция
Формуляр, въвежда данни при клиента за обработка на сървър	
<FORM ACTION=" URL" METHOD=GET POST ENCTYPE=" multipart/form-data> </FORM>	Дефиниция на формуляр, адрес на скрипт за обработка, метод, кодиране
<INPUT TYPE=" TEXT PASSWORD CHECKBOX RADIO IMAGE HIDDEN SUBMIT RESET" NAME="****" VALUE="****" CHECKED SIZE=? MAXLENGTH=?>	Дефиниция на поле във формуляр за въвеждане на информация
<SELECT NAME="****" SIZE=? MULTIPLE> </SELECT>	Дефиниране на списък във формуляр
<OPTION SELECTED>	Избор от списък
<TEXTAREA NAME="****" ROWS=? COLS=? WRAP=OFF VIRTUAL PHYSICAL> </TEXTAREA>	Дефиниция на текстово каре във формуляр, въвежда описание на повече от един ред