

## Лекция 5

### 1. Таблици със стилове в HTML

HTML браузерите притежават имплицитно дефинирана таблица със стилове за HTML таговете – описание на формата чрез който се изобразява информацията в тага. С други думи, HTML браузерът трябва да осигури вградена карта, която да съпостави всеки таг със съответстващото му правило, описващо начина за неговото изобразяване. Това правило съдържа информация за елемента, характеристики за шрифта на текста в тага, дали съдържанието на тага се изобразява в отделна правоъгълна област или да се вгради в текущият параграф и т.н. В по-ранните версии на HTML тези таблици бяха вградени по принцип и трудно можеха да се променят характеристики на таговете. В сегашните HTML браузъри, таблиците със стилове могат да се променят, което дава големи възможности за редакция на текста във Web страниците.

HTML започва своето развитие като език за описване на резюмета по физика – документи, обобщаващи съдържанието на статии, написани от физика. В първоначалната спецификация не е имало нищо за описание на шрифт или цвят, както и някои други елементи, които за сегашните Web страници се смята за задължителни. Тъй като в основната структура, са липсвали много елементи, езикът започва да се допълва с нови тагове, но това е правено по много случаен начин. Така информацията за представяне на съдържанието на една страница става доста сложна и заплетена, и едва с появата на XML се започна основно преосмисляне на начина на структуриране на документите.

Появи се обща парадигма в програмирането, съгласно която процесът на извличане на данните се отеля от процеса на тяхното показване (което е в основата на новата технология XML). Това доведе до изискването за влагане на по-голямо внимание към структурата на данните за сметка на стила. Проблемът с разработване на таблици със стилове датира доста отдавна (60-те години на миналия век), но той придоби ново значение сега при бурното развитие на Web.

Стиловете и таблиците със стилове (**steel sheets**) служат да предоставят информация за изобразяване на данните, а таблиците с каскадни стилове (**Cascading Style Sheets – CSS**) се използват за задаване на някои медийни характеристики на документите (размер и вид на шрифта, гласови характеристики за говорим текст и други). Определението каскадни стилове се отнася до възможността да се използват множество нива на стилови набори в документа, като едно ниво на стил може да бъде използвано за дефиниране на друго.

Дефинирани са две нива на стилове в HTML. CSS1 е прост механизъм за набор от стилове, който дава възможност за асоцииране на основни стилове (шрифтове, цветове, размери на текст) към HTML елементите. CSS1 е въведен в HTML 3.2 версия и се поддържа от Internet Explorer 3 и следващите версии. Navigator 4 и следващите версии и другите браузъри създадени след тях. CSS2 е построен върху CSS1 и добавя поддръжка на набори от стилове, специфични за различните медийни обекти, като позициониране на съдържанието, подредба на таблици, автоматични броячи и други възможности.

Един стил се състои от набор от свойства със зададени стойности във формата **свойство::стойност**. По този начин могат да се подредят неограничен брой стилове описващи даден елемент от езика. В HTML, стиловете могат да се присвоят по един от следните три начина: чрез атрибута **style** на елемент от HTML, чрез елемент **<Style>** или чрез външна таблица със стилове.

Атрибутът **style** се използва по стандартният начин на използване на атрибути в HTML. Ето как изглежда задаването на стил посредством атрибут:

```
<H1 style="background-color:red;color:white">
```

```
This text will be shown with white text on a red background. </H1>
```

В този случай, текстът в елемента **<H1>** ще бъде представен върху червен фон с бял цвят. Това е добър метод за промяна на подразбиращите се стилове в даден елемент от документ.

В много случаи се изисква смяна на стила във всички елементи от даден тип. Атрибутите за стил се прилагат само за елемента в който са зададени. За да се промени стилът във всички елементи от даден тип се използва елементът <STYLE>. Този елемент позволява даден стил да се асоциира с даден тип елементи. Елементът <STYLE> се използва по следният начин:

```
<STYLE>
H1{background-color:red;color:white}
P{background-color:red;color:white}
<STYLE>
<H1>
This text will be shown with white text on a red background.
</H1>
```

Тук текстът в елемента <H1> ще бъде представен по същият начин както в горният пример, но сега този стил се отнася за всички елементи от типа <H1>, както и за елементите от типа <P>. В случая, същинската декларация на за всеки стил се нарича правило (rule), а сборът от тези правила в рамките на тага <STYLE> се нарича таблица на стиловете (style sheet). Името на тага, за който се отнася даден стил се нарича селектор на правилото. Така P{background-color:red;color:white} е правило, P е селектор на правилото, а блокът

```
H1{background-color:red;color:white}
P{background-color:red;color:white}
```

като цяло представлява таблица със стилове.

Малко по-сложно стои въпросът, когато в един тип елементи трябва да се използват различни стилове. Тогава, би могло да се използват атрибути за задаване на стила за всеки конкретен елемент, но при разработване на големи документи това би довело до доста затруднения. За преодоляване на този проблем в HTML са въведени класове при задаване на стиловете. Ето един пример на използване на класове при задаване на стиловете:

```
<STYLE>
.typ{background-color:blue;color:white}
.resource{background-color:red;color:white}
.select{background-color:yellow;color:black}
</STYLE>
```

```
<H1 class="typ">
```

```
This text will be shown with white text on a blue background. </H1>
```

```
<P class="select"> This text will be shown with white text on a red background.
```

```
</P>
```

В този пример са създадени три различни стила с помоща на атрибута class, които се използват в различни елементи на HTML документа. Отделните стилове се дефинират като им се задава наименование, което в елемента <STYLE> се предхожда от ‘.’. В дефиницията на стиловете може да се използва и селектор на елемент, при което се дефинират някои специфични правила отнасящи се само до отделни елементи. Например в горният пример в секцията <STYLE> отделните стилове могат да се именуват като P.typ, P.resource и P.select и тогава те ще се отнасят само за елементите от типа <P>.

Създаването на таблици със стилове като **външни** документи, а не като вътрешни блокове с код се превърна в широко разпространена практика. Може да се създаде цялостен изглед на един сайт посредством една единствена таблица със стилове, към която се обръща всеки HTML документ. В HTML този механизъм се поддържа по един от следните два начина: посредством елемента <LINK> или чрез CSS директива @import.

Тагът <LINK> е рядко използван елемент в HTML, въпреки че неговото основно предназначение е създаването на връзка на текущия документ с други документи. За този

елемент съществува ограничение, че се поддържа само в тага <HEAD>, но той може да се използва за зареждане на таблица със стилове, например посредством израза:

```
<LINK type="text/css" rel="stylesheet" href="mystylesheet.css">
```

Тази конструкция асоциира таблица със стилове **mystylesheet.css** с текущия документ. Атрибутът `type` указва типа на свързаният документ (**text/css**), а `rel` информира HTML, че връзката е от типа `stylesheet` и трябва да използва съответният начин за импортиране. Накрая, `href` дава хипертекстовата препратка към URL адреса, където се намира файла с таблицата със стилове.

Директивата **@import** може да се използва за зареждане на стил от даден <STYLE> блок или външна таблица със стилове. Например в HTML документ може да бъде включен следният фрагмент:

```
<STYLE>
  @import url(notes.css)
</STYLE>
```

като в случая ще се приложат зададените в **notes.css** правила към текущият документ. Конструкцията `url()` често се среща в CSS и указва, че съдържанието може да бъде намерено на указания URL адрес.

## 2. Изображения и мултимедия в Internet

Една от главните причини за успеха на Web е възможността за интегриране на графика, текст и звук. Най-сложният аспект от включването на графика във Web страниците е свързан с проблемите по дизайна на страниците, които съдържат графика. Създаването на ефективни изображения и правилното им разполагане в пространството на Web страниците е свързано с доста трудности. Един от проблемите на Web е скоростта на трансфер на графичните елементи във сайтовете. Времето за показване на една Web страница зависи от съдържанието на графични елементи. Скоростта на трансфер на данните в Internet зависи от редица фактори:

- **Скорост на достъп.** Различните потребители достигат да Web посредством връзки с различна скорост. Един и същ сървър може да обслужва една Web страница с различна скорост, в зависимост от това колко е натоварен. Когато се изпробва една нова, богата на изображения страница на отдалечен компютър, всичко може да стане много бързо. Но ако се постави същата страница на сървър и се направи опит да разгледа с помощта на модем, работещ например със скорост 28.8 Kbps (килобайта за секунда) и то по време, когато много потребители използват същия сървър, всичко може да се извърши много бавно.

- **Тип на графиката.** Скоростта на трансфер на данните и качеството на страниците зависят и от типа и качеството на графиките. Понякога, не се обръща толкова голямо внимание на забавянето на трансфера на дадено изображение, стига качеството на графиката да е добро. Поради тази причина, графичният дизайн на страницата е особено важен.

Всяка графична програма записва файловете с графика в свой собствен графичен формат – специфично подреждане на данните, използвано от програмата за запис на графичната информация. В HTML се използват два формата за запис на графична информация – GIF и JPEG – формати.

**GIF (Graphic Interchange Format)** е формат който се използва често за обмен на графична информация. Той е ефективен формат, когато изображението се състои от малко на брой цвята (до няколкостотин). Такива са най-често създаваните изображения с графични програми – хистограми, икони и други. Ако изображението има повече от 256 цвята (нюанси на цветовете), известна част от информацията се губи, когато се преобразува в GIF формат. Трябва да се сравни изображението преди конвертирането и след това за да се прецени дали загубите на данни са допустими.

GIF форматът дава някои допълнителни възможности за графичен дизайн, които не съществуват при някои други формати. В GIF изображенията, съществува възможност да се представят цветовете прозрачни и по този начин картината да се използва като фон на страница. Освен това тези изображения могат да се записват в *презредов (interlaced)* формат. Презредовите изображения са записани по такъв начин, че когато браузърът започне да прехвърля изображението, то най-напред се вижда с много ниска разделителна способност (грубо) и след това става все по ясно, докато се покаже в най-добър вид. Това прави GIF изображенията предпочитани за бързо извеждане на графични елементи от Web страници.

**JPEG (Joint Photographic Experts Group)**, е формат създаден за компресиране на сложни изображения. Повечето съвременни браузъри поддържат JPEG по същият начин, по който поддържат GIF, извеждайки JPEG изображението направо в Web страницата. JPEG се използва, когато трябва да се изобрази картина с много цветови нюанси, включително фотографии. Той е проектиран да може ефективно да компресира сложни изображения с най-малка загуба на информация и качество на картината.

*Прозрачните изображения* съдържат прозрачна област около съществената част на даден обект. Например, ако се представя снимка на ръчен часовник и не е необходимо наличието на фонов цвят около него, а трябва да се вижда само часовникът върху страницата, то е необходимо прозрачно изображение. За целта се използва прозрачно GIF изображение (фоновият цвят върху който е изобразен предметът е прозрачен). Напоследък се разработват и JPEG формати, които осигуряват прозрачност на цветовете.

**PNG (Portable Network Graphics)** – произнася се “пинг”. Този формат е създаден като заместител на GIF формата, поради използваната от GIF патентована схема на компресия. PNG поддържа прозрачност и анимация.

Освен изображения, Web страниците могат да съдържат и звук и видеоинформация. Преди няколко години, файловете със звук и видео се изтегляха чрез Web и след това се стартираха със специални приложни програми. Сега мултимедийните файлове са вградени във Web страниците посредством така наречените plug-in програми.

Използването на мултимедия, поражда същите проблеми, както и използването на графичните изображения – сомо че в още по-голям мащаб. Много потребители нямат подходящи plug-in програми за използване на мултимедийни и графични файлове. Производителността също е проблем. Да се чака с минути или часове за изтегляне на един клип или песен е доста дразнещо. Мултимедийните файлове могат да влошат цялостната производителност на Internet. Едни от най-популярните plug-in програми са RealAudio, QuickTime, ShockWave и други.

### **3. Създаване на Web страница или Web сайт**

Създаването на Web страница или Web сайт не е сложна задача. Разбира се, разработката на сложни, комплексни сайтове е трудна задача, както всяка задача свързана със създаването на сериозни програмни приложения. Съществуването на безплатни Web базирани услуги за публикуване, позволява само за няколко часа да се създаде и публикува собствена страница в Internet. Един от най-популярните Web сайтове, които предоставят възможност за публикуване на лични страници е GeoCities. Това е сайт, който предлага безплатно публикуване на персонални Web страници. От създаването си GeoCities е дал възможност за публикуване на милиони персонални страници. Сега той е притежание на Internet гиганта Yahoo. Пълният адрес на GeoCities е <http://www.geocities.yahoo.com/>.

Първият опит за създаването на Web приложение, обикновено е създаването на лична (персонална) Web страница. Много Web страници първоначално са били замислени като експериментални лични страници, и след това са се првъръщали в големи и популярни сайтове с много посетители. Публикуването (качване на Web сървър) на Web страниците се извършва съгласно инструкциите на инструмента с който се създава страницата (FrontPage Express, Dreamweaver и други) или инструкции от сайта, където се публикува.

Публикуването на Web страници може да бъде елементарно, ако се разработва персонален или тематичен сайт, който трябва да бъде видян само от няколко приятели или колеги. Но, когато се създава бизнес сайт или се търси широко разпространение на информацията в сайта се изискват малко повече операции. Първата стъпка е да се получи пространство на Web сървър. Тук може да се избира. Може да се получи безплатно пространство или да се откупи пространство за публикуване. Когат трябва да се създаде сайт за фирма или организация, повечето от услугите не се предоставят безплатно.

Web сървърът е компютър, включен във Web, със специален софтуер, който му позволява да предоставя информация на потребителите на Web. В Internet са включени стотици хиляди такива сървъри. Само чрез създаването на собствен Web сървър с натоварени на него файлове на Web сайт, или чрез натоварването на файловете на друг сървър, може даден сайт да стане част от Web пространството.

Много фирми предоставят **Web хостинг**, което означава, че те осигуряват Web пространство за Web сайтове на потребители. Тези фирми осигуряват определени услуги (консултации и поддръжка), като същевременно определят името на домейна за потребителският сайт. Намирането на подходящ Web хостинг е важна и отговорна стъпка при разработване на Web сайтове.

Една от полезните възможности на Web публикуването е, че може да се създаде, тества и модифицира Web страница или сайт на собствен компютър, несвързан директно с Web сървъра. След окончателната редакция на страниците, файловете могат да се прехвърлят върху Web сървъра. В някои случаи, това е свързано с известни трудности.

Най-сериозните трудности при прехвърляне на файловете на даден сайт върху Web сървъра са свързана със структурата на директориите. Проблемът е, че една връзка, например от HTML документ към изображение, трябва да се указва в коя поддиректория се намира графичният файл. Когато се прехвърлят файлове на друг компютър (Web сървър), поддиректориите се променят, което е причина връзката на документа с графичното изображение да се прекъсне. Ако сайта се състои от малко на брой файлове, проблемът се решава, като всичките файлове се запишат в една директория и се прехвърлят също в една директория.

За сайтове с много файлове, трябва да се организира проста структура на поддиректориите и да се използват връзки с относително адресиране. При относителното адресиране не се указва целият път до файла, а само относителния – от файла в който се намира връзката до файла към който сочи тя. Това позволява да се преместват файлове от един компютър на друг без да се променят връзките.

Един от начините за прехвърляне на файлове между компютри, свързани в Internet е FTP (File Transfer Protocol) – протокол за трансфер на файлове. FTP направи популарен Internet още преди появата на World Wide Web. Това е относително лесен начин за прехвърляне на файлове в глобалната мрежа. Много потребители използват FTP, за да свалят файлове от FTP сървър на своя компютър. Използването на Web интерфейса скрива от потребителите подробностите на FTP. Но за да се публикува Web страница, може да се наложи използването на услуга за прехвърляне на файловете чрез FTP. За голяма част от потребителите, това е непознато действие, но то не е сложно. Съществуват множество FTP програми за Windows, Macintosh или UNIX.

### **3.1 FrontPage и FrontPage Express**

Microsoft, признатият лидер в софтуерната индустрия за персонални компютри, разработва голяма част и от софтуера за Internet. Компанията е водеща в установяването на техническите стандарти, при разработване на по-голяма част от софтуерните приложения. Тя контролира голяма част от пазара на браузери (Internet Explorer) и притежава водещата програмна среда за разработване на Web страници (Microsoft FrontPage 98 и 2000). Накрая

Microsoft играе водеща роля в процеса на установяване на стандартите за Internet и новите версии на HTML.

FrontPage Express е леко орязан вариант на FrontPage, който се разпространява безплатно. Той може да бъде свален чрез Internet от адрес <http://www.microsoft.com/sitebuilder/>. Въпреки, че има малки възможности от FrontPage2000, FrontPage Express притежава “drag-and-drop” (издърпай и пусни) възможности, и поддръжка на Java, ActiveX и Dynamic HTML – т.е всички съвременни стандарти за програмиране и дизайн на Web страници.

FrontPage Express е инструмент за разработване на Web страници, който използва средствата на визуалното програмиране. Той осигурява, проектирането на страницата да става пред погледа на програмиста, като скрива подробностите за форматиращия текст на HTML. Текстовият документ (HTML документа), може да бъде разглеждан и редактиран по всяко време само със избор от главното меню. Най-важните достойнства на този програмен инструмент за разработване на Web страници са:

- Създаване и редактиране на Web страници без необходимостта да пише HTML код.
- Изтегляне и пускане с мишката на елементи и връзки към други места във Web, без да е необходимо въвеждане от клавиатурата на ULR-адреси или пътища до файловете
- Прехвърляне на графика през клипборда (системният буфер) във разработваната Web страница, промяна на размера на графиката и добавяне на текст към нея.
- Създаване и редактиране на таблици.
- Създаване и редактиране на формуляри – полета за интерактивно въвеждане на информация.

Осигурена е възможност за включване на мултимедийни файлове и компютърни програми.

FrontPage Express поддържа формуляри, но не може да осигури CGI – скриптове, които са необходими за работа с формулярите.