

# Създаване на уеб сайт

## Въведение

Интернет – е навсякъде! Толкова е всеобхватен, че за оправдание при ненаписано домашно съвременните ученици по-скоро биха казали, че даден уеб сайт е бил извън линия, вместо да настояват, че кучето е изяло домашното им. Глобалната мрежа разкри детайли за многообразието на общия свят, в който живеем, направи достъпна за всички възможността да откриват и изследват нови неща. Интернет предложи нови начини за образование, забавление, бизнес и социални контакти. Но кой създава всички страници в Интернет? Все по-често, отговорът е ВНЕ. В тази глава, вие ще научите основите на езика за публикуване в Интернет — HTML (Хипертекстов маркиращ език), ще създадете страници с помощта на този език и ще научите правилата, които управляват Мрежата. Повечето текстови редактори са идеални за създаване на уеб страници с HTML. Дори ако мислите да използвате WYSIWYG (това, което виждаш е това, което получаващ) уеб дизайн среда, солидната база знания за HTML е много ценна.

## HTML в преход

Съдържанието на Мрежата, кодът, който създава това, което виждате, и правилата, които определят начина на създаване на уеб съдържанието, се променят бързо. По-ранните HTML версии бяха много ясни, но и много статични – това не позволяваше по-голямо взаимодействие с потребителя, освен през хипервръзки към други страници и уеб сайтове. С течение на годините HTML еволюира в динамичната среда, която познаваме днес. Модерните уеб страници и Expression Web са написани с помощта на XHTML.

За да се запази гладката и последователна работа на Интернет, са създадени редица организации, някои от които бяха разгледани. Международната организация, която установява стандартите и правилата за кодиране в HTML езика се казва World Wide Web Consortium (Консорциум на световната глобална мрежа) и често се назовава в съкратен вид като W3C. Нейната цел е “оперативна съвместимост на мрежата”. W3C е създадена през 1994 г. от Тим Бърнарс-Лий, който е изобретил Световната глобална мрежа през 1989 г. Тъй като технологиите непрекъснато се променят, уеб стандартите трябва редовно да се актуализират, за да бъдем сигурни, че уеб технологиите, хардуера и софтуера остават съвместими. Най-новият проект от предложени правила определя стандартите за HTML 5.

В този последен проект за уеб стандарти, cascading style sheets (CSS) (каскадните стилови таблици) са признати като най-добрия начин за създаване на единен дизайн за уеб страници. Много етикети (tags) от старите версии на HTML и стандарти не са включени в най-новия проект за HTML 5 или в XHTML, защото

ефектът им е най-вече специфична визуализация и по-ефективно се поддържа от CSS (каскадните стилкови таблици) и други функции на HTML 5 и XHTML.

Най-актуалните промени в уеб стандартите са също така резултат от развитието на устройства като мобилни телефони, които са в състояние за достъп до интернет, но поради малкия екран са ограничени във визуализацията на информация. В следващите раздели ще научите много повече за каскадните стилкови таблици (CSS). Можете да научите повече за W3C и HTML стандарти на адрес <http://www.w3.org/Consortium/>.

**Бележка:** За целите на упражняване на концепцията за тагове чрез създаване на няколко уеб страници за начинаещи, този текст включва няколко елемента, които често са се използвали в по-стари версии на HTML, но вече не отговарят на най-новите HTML стандарти. Елементи, които вече не се поддържат от текущата версия на HTML се наричат "остарели". Ще отбележим тези остарели елементи, там където се появяват. Други елементи не са остарели, защото работят по-добре с каскадни стилкови таблици. Някои елементи в тази категория са представени отделно в този текст и по-късно включени в уроци за CSS, след като учениците са ги разгледали в по-проста настройка.

## Структура на уеб страница

HTML е лесен за научаване и можете да използвате много различни редактори, за да напишете кода. Най-основните страници могат да се напишат в Notepad, а по-графични или интерактивни страници могат лесно да бъдат проектирани с програми като Expression Web. Независимо какви инструменти използвате за създаване на уеб страници, всички страници трябва да бъдат написани, следвайки синтаксиса или езиковите правила на HTML кода.

Важно е да създадете папка, в която да съхраните всички файлове на страници и изображения, които ще са част от уеб страницата, която създавате.

Командите в HTML се състоят от елементи, които включват един или два етикета (наричани tags), които са отделени от обикновения текста чрез ъглови скоби (символите "по-голям" и "по-малък"). Тези етикети са сърцето на HTML езика. Повечето елементи включват начален и затварящ етикет (означен с наклонена напред черта, преди текста). Тези начален и затварящ етикети обгражда текст, или друга информация:

**<tag> information </tag>**

Въпреки, че преобладават двойките етикети - начален и затварящ, съществуват и самостоятелни етикети (без затварящ етикет), като например <p>, който е етикет за параграф. Той създава ефекта от натискане на клавиша "Enter" два пъти. Този етикет няма нужда от затварящ – т.е. от </p> таг.

**Атрибутите** (attributes) са команди, които могат да променят изобразяването на информацията, включена между началния и затварящ етикет:

```
<tag attribute=xx> information </tag>
```

Някои HTML етикети изискват задължително задаване на детайли (чрез атрибути), за пълнота на инструкциите към брауъра при възпроизвеждане на страницата. При други HTML етикети задаването на атрибути не е задължително, а се прави с цел по-добро оформление на уеб страницата, например задават се цвят и разположение на съответния елемент. Атрибутите винаги са част от началния етикет и HTML елементите могат да бъдат вложени в други елементи, но елементите не могат да се припокриват (т.е. етикетите трябва да се затварят в обратен ред на този, в който са отворени).

```
<tag1><tag2> information </tag2></tag1>
```

В предходния пример, забележете, че елементът TAG2 е вложен изцяло във вътрешната част на елемент TAG1. Не е възможна следната последователност, т.к. елементите се припокриват:

```
<tag1><tag2> information </tag1></tag2>
```

Етикетите се използват за задаване на формат и място на компонентите от една уеб страница, като параграфи, изображения и линии.

## Блоково и инлайн форматиране (Block and Inline Formatting)

Оформлението на съдържанието на една уеб страница може да бъде описано като “блоково” (block) или “инлайн” (вградено). Инлайн елементите са текст и други елементи като изображения. Инлайн елементите се появяват с потока на страницата (т.е. визуално са част от даден текстов сегмент) . Това изображение



е “инлайн” елемент от този параграф. Когато се гледат през уеб браузър, вградените елементи обикновено не започват на нов ред.

Елементите с блоково форматиране са отделени от потока на съдържанието. Елементите с такова форматиране се забелязват лесно защото те стоят сами на страницата и са отделени от потока на съдържанието. Например следното изображение е с блоково форматиране:



Тези елементи обикновено започват на нов ред, когато се гледат в браузър. Заглавията (headings) и таблиците (tables) са два примера за блокови елементи, които ще разгледаме в тази глава. Заглавието ще се появи като блок елемент в брауъра.

## Запознаване с етикети (tags) и атрибути (attributes)

Когато за пръв път видите код на страница, може да ви се стори много объркващо. В тази глава ще разбием кода на малки части и ще обясним редовете на кода.

Първият елемент, който ще видите в най-горната част на повечето уеб страници, указва на браузъра какъв набор от "правила" да използва. Ето един пример:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Това изглежда много сложно, но всъщност е доста ясно. Ще научите повече за това по-късно в тази глава. За сега, мислете за това като ред, който казва на браузъра чрез какъв набор от HTML (или в този случай, XHTML) правила е създадена тази страница.

HTML файловете обикновено се записват с помощта на .html или .htm разширение. Обикновено, входната страница за даден уеб сайт се наименува **index.html**. Ако в адресното поле на браузъра се въведе <http://mysite.org>, той ще се опита да отвори страница, намираща се на сървъра, с изписаното домейн име и казваща се index.html.

HTML кода на една страница се състои от две основни части - заглавна (head) и тяло (body). Така както заглавната част на едно писмо съдържа специфична информация, отнасяща се до това читателя да разбере дали съответното писмо е служебно, е от приятел или е търговско предложение, така и заглавната част на всички HTML документи съдържат информация, отнасяща се за тълкуване на кода от браузъра.

В едно добре оформено писмо, обикновено новите параграфи са отместени, някои заглавия или съкращения може да са изписани само с главни букви, спазени са граматичните правила на съответния език. Най-вероятно бихте могли да прочетете, писмото и ако е написано **сбитозаедно със стра нни паузи**.

Разбира се, никой читател не би бил доволен от такъв стил! По същия начин, HTML има собствен стил и е добре той да се спазва, за да бъдат уеб страниците лесни за четене.

Най-доброто правило за стил е да поставяте интервали на подходящите места и да организирате HTML кода, така че този код да може лесно да разбира от читателя. Имайте предвид, че браузърът игнорира допълнителните интервали и празни редове в рамките на кода.

HTML кода не е чувствителен към малки и големи букви, но най-добре е винаги да използвате малки букви за етикетите. Бъдещите XHTML стандарти ще изискват малки букви, така че използването на малки букви е добър навик.

Следващите три примера показват различни стилове на един и същ HTML код, при създаване на една и съща страница. По-късно в тази глава ще научите какво означава всеки един от етикетите, използвани в примерите. Помислете си кой примерен код от предложените е най-лесен за четене?

### Примерен стил 1

```
<html>
<head><title>My first web page</title></head>
<body>all the important details</body>
</html>
```

### Примерен стил 2

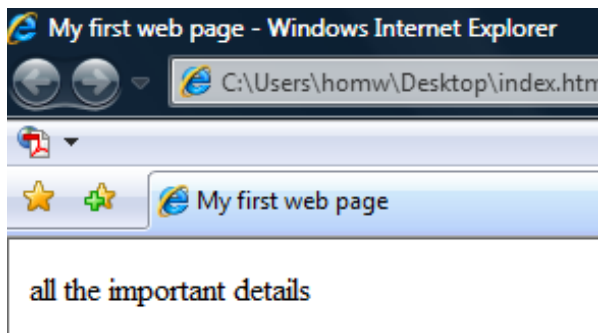
```
<html>
<head>
<title>My first web page </title>
</head>
<body>
    all the important details
</body>
</html>
```

### Примерен стил 3

```
<html><head><title>My first web page</title></head><body>all the important
details</body></html>
```

Дори и да не знаете все още смисъла на етикетите, използвани в горните примери, най-вероятно можете да прецените, че „Примерен стил 2” е най-лесно читаем. Причината за това е, че етикетите са ясно отделени от текста и лесно могат да се забележат двойките начален и затварящ етикет.

Следната фигура показва какво извежда браузъра, когато визуализира уеб страницата, написана с който и да е от горните 3 стила. Забележете къде се показват думите My first web page, обградени с етикетите за заглавие <title>, Обърнете внимание къде в уеб страницата са изписани думите, обградени с етикетите за тяло <body>.

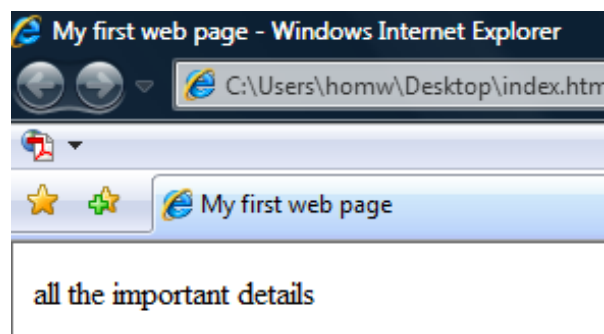


## Използване на етикети (tags) и атрибути (attributes) за създаване на уеб страница

Време е да научите как HTML етикетите се използват за създаване на уеб страница. Като се има предвид голямото количество информация в Интернет, изненадващо е, че уеб страниците се създават от значително малък брой етикети (tags). Ще видите, че те са лесни за научаване и използване.

Кодът за всички HTML страници започва с етикета `<html>` и завършва с етикета `</html>`. Тази двойка етикети определя началото и края на страницата и всички други HTML етикети трябва да се съдържат между тях.

След задължителният `</html>` етикет е елементът HEAD (ГЛАВА) (изобразен чрез етикетите `<head>` и `</head>`). Той служи като контейнер, в който може да се вмъкнат други видове елементи, като например елемента TITLE (ЗАГЛАВИЕ). Елементът TITLE съдържа думите, които се появяват на заглавната лента (title bar) и в заглавния ред на панела за страницата (page tab). В следния пример, “My first web page” (Моята първа уеб страница) е заглавието на страницата.



Заглавието – “My first web page” – което се появява на 2 места – в заглавния ред на панела за страницата (browser tab) и в заглавната лента на прозореца на браузъра. Това изображение е направено с помощта на следните редове код.

```
<html>  
<head>
```

```
<title>My first web page</title>
</head>
...
</html>
```

На мястото на многоточието може да има други етикети, които не се отнасят към изобразяването на заглавието.

Повечето от елементите, които произвеждат това, което виждате на една уеб страница, се намират в елемента BODY (ТЯЛО). Този елемент се огражда от двойката етикети <body> и </body>. Етикетът <body> трябва да следва етикета </head>.

Ето и пълния код за уеб страницата в предходното изображение, илюстриращ три HTML етикетита, както и елементите HEAD и BODY.

```
<html>
<head>
<title>My first web page</title>
</head>
<body>
all the important details
</body>
</html>
```

## Увеличаване на разбираемостта на HTML кода с коментари

Добре е авторът на HTML кода да поставя коментари. Даден коментар обяснява целта на съответната секция от изходния код, за която той е поставен. Коментарите не се появяват на действителната уеб страница, но са много полезни за всеки, който преглежда кода, като например член на уеб екип или любопитен зрител на уеб сайта. Коментарите могат да включват подробности относно HTML версията, планове за актуализиране или подробности за процеса на развитие. Ето един пример за коментар:

```
<!-- This is a comment to tell the reader of the code something -->
```

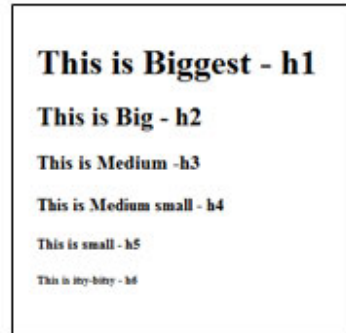
Както се вижда от горния пример, за да се разпознае един текст като коментар, той трябва да се огради в специфичните етикети <!-- и -->.

## Подобряване на облика на съдържанието на уеб страница

В основния текст на уеб страницата (т.е. този в елемента BODY) често е нужно дадени думи да се открият като заглавие. Това става чрез така наречените **заглавни етикети (Heading tags)**. Те се използват за подобряване на стилового

оформление на текста в една страница, чрез промяна размера на шрифта и удебеляване на текста. Има шест размера на текста, които се задават със заглавните етикети H1 (за най-големия шрифт) до H6 (за най-малкия шрифт). Следващият пример демонстрира използването на всички шест заглавни етикета:

```
<body>
<h1>This is Biggest - h1</h1>
<h2>This is Big - h2</h2>
<h3>This is Medium - h3</h3>
<h4>This is Medium small - h4</h4>
<h5>This is small - h5</h5>
<h6>This is itsy-bitsy - h6</h6>
</body>
```



Понякога се налага открояването на елементи от уеб страницата да става не само чрез промяна на размера и дебелината на шрифта, а и чрез поставяне на специфична позиция. За изпълнението на тази задача се използва атрибута **align** (**подравняване**), който може да се зададе за всеки от заглавните етикети (както и на някои други етикети). На атрибута се присвоява желаната позиция: align = position, където позицията (position) може да бъде right (дясно), center (център), или left (ляво). В примера даден по-долу се използва и етикет <br>. Този етикет е самостоятелен (т.е. няма съответен затварящ) и функцията му е да зададе, че текста на съответното място в уеб страницата трябва да продължи на нов ред – т.е. той задава еквивалента на “enter” при възпроизвеждане на страницата.

```
<h1 align=left>Hickory</h1><br>
<h1 align=center>Dickory</h1><br>
<h1 align=right>Dock</h1><br>
```

Резултатът при визуализацията на горния код като уеб страница е следния:

**Hickory**

**Dickory**

**Dock**

В следващата таблица са изброени някои често срещани HTML етикети. Ще намерите по-подробен списък в края на тази глава.

Елемент	Значение	Използва се за
<p>text</p>	Параграф	Нов параграф (нов абзац). Автоматично се вмъква допълнителен празен ред между параграфите.



		<i>Бележка:</i> Повечето браузъри приемат <p> като нов абзац, без съответен етикет за край, но етикета за край се препоръчва от W3C.
<b>text</b>	Удебелен (остарял)	Форматира текста между таговете като удебелен.
<strong>text</strong>	Удебелен	Форматира текста между таговете като удебелен.
<u>text</u>	Подчертан	Подчертава текста между началния и краен етикет.
<i>text</i>	Курсив (остарял)	Форматира текста между таговете като курсив.
<em>text</em>	Курсив	Форматира текста между таговете като курсив.
   	Нов ред	Нов ред (по подразбиране, не се вмъква допълнителен нов ред след   етикет, за разлика от етикета за абзац <p>.) Бележка:   е алтернативна форма на  ; не е затварящ етикет.
<hr> <hr width = x% size = n color = #0000FF>	Хоризонтална линия	Създава хоризонтална линия за разделяне на секции. Допълнителни атрибути включват ширина (като процент), размер (дебелина на линията - като цяло число) и цвят (#0000FF задава син цвят).
<sub>text</sub>	Долен индекс (Subscript)	Форматира текста между етикети като долен индекс.
<sup>text</sup>	Горен индекс (Superscript)	Форматира текста между етикети като горен индекс

## Специални символи

Има много символи, които се използват в уеб страници, но не се въвеждат по обикновения начин – чрез натискане на клавиш от клавиатурата, т.к. за тях не са предвидени означени клавиши. Два примера от тези символи са ñ и é. За да създадете такива символи в HTML, се използват комбинации от няколко клавиша, чрез които въвеждате съответния код на желанния символ.

Въвеждането на кода започва с амперсанд (&), следван от символа диез #. След тях се изписва номера на специалния символ, следван от точка и запетая.

Например, &#169 ще произведе ‘©’. Тази техника трябва да се използва за показване на символи ‘<’ или ‘>’ на уеб страница, защото ако просто добавите ‘<’ или ‘>’ в низ от символи, това ще накара браузъра да мисли, че въвеждате етикет и той ще очаква останалата част от синтаксиса на етикета.

По-долу е списъка с кодовете на някои от по-често срещаните специални символи, както и тези, които може да объркат браузъра или интерпретиращата програма.

Unicode Number (Unicode номер)	Символ
34	"
38	&
60	<
62	>
169	©
174	®
162	¢
177	±
191	¿
232	Ё
233	Ђ
241	Њ

За да видите пълен списък със специални символи, посетете <http://www.utexas.edu/learn/html/spchar.html>.

## Създаване на цветове в HTML

Има няколко начина за задаване на цвета на различни елементи на уеб страница. Предходните примери използват един хексадецимален номер за черно и едно име за цвят за жълто, за задаване на атрибутите за цвят на етикета на тялото. Също така можете да използвате RGB (Червено Зелено Синьо) стойности за определяне на цветовете.

Цветовете на компютърния екран се състоят от комбинации от червено, зелено и синьо (Red, Green, Blue – RGB). Всеки един от тези цветове може да варира по интензитет от 0 до 255. Чрез смесване на тези цветове могат да се създадат 16 777 216 уникални цветове, всеки от които си има присвоен шестнайсетичен номер. Разбира се, очите ви обикновено не могат да различат цветове, които са много близки (т.е. са близки и по номер). Черното се изразява като шеснайсетична стойност #000000 или RGB стойност от (00,00,00), т.е. 0 от червения цвят, 0 – от зеления и 0 – от синия. Чистото червено е с шестнайсетичен номер#FF0000, или RGB (255,0,0).

Има няколко имена на цвят, които могат да се използват в HTML код. Можете да изпробвате няколко имена на цветове, за да видите кои работят (единичните

имена на цветовете работят най-добре). В следния код например е използвана думата за жълто, вместо кода, за да се зададе цвета на текста в тялото на уеб страница, докато цвета на фона е зададен с номер:

```
<body bgcolor="#000000" text="yellow">
```

Можете да използвате следните уеб сайтове, за да откриете шестнайсетичните кодове, RGB стойностите, или използваеми имена на цвят:

[www.html-color-codes.com](http://www.html-color-codes.com)

[www.computerhope.com/htmcolor.htm](http://www.computerhope.com/htmcolor.htm)

[www.htmlcolorcodes.info/](http://www.htmlcolorcodes.info/)

Следва пример на цветова схема, която включва и шестнайсетичните кодове.

Snow FFFAFA	GhostWhite F8F8FF	WhiteSmoke F5F5F5	Gainsboro DCDCDC
FloralWhite FFFAF0	OldLace FDF5E6	Linen FAF0E6	AntiqueWhite FAEBD7
PapayaWhip FFEFD5	BlanchedAlmond FFEBCD	Bisque FFFAF0	PeachPuff FFDAB9
NavajoWhite FFDEAD	Moccasin FFE4B5	Cornsilk FFF8DC	Ivory FFFFF0
LemonChiffon FFFACD	Seashell FFF5EE	Honeydew F0FFF0	MintCream F5FFFA
Azure F0FFFF	AliceBlue F0F8FF	lavender E6E6FA	LavenderBlush FFF0F5
MistyRose FFE4E1	White FFFFFF	Black 000000	DarkSlateGray 2F4F4F
LightSlateGray 778899	Grey BEBEBE	LightGray D3D3D3	MidnightBlue 191970
CornflowerBlue 6495ED	DarkSlateBlue 483D8B	SlateBlue 6A5ACD	MediumSlateBlue 7B68EE
LightSlateBlue 8470FF	NavyBlue 000080	DimGrey 696969	SlateGrey 708090
MediumBlue 0000CD	RoyalBlue 4169E1	Blue 0000FF	DodgerBlue 1E90FF
DeepSkyBlue 00BFFF	SkyBlue 87CEEB	LightSkyBlue 87CEFA	SteelBlue 4682B4
LightSteelBlue B0C4DE	LightBlue ADD8E6	PowderBlue B0E0E6	PaleTurquoise AFEEEE
DarkTurquoise 00CED1	MediumTurquoise 48D1CC	Turquoise 40E0D0	Cyan 00FFFF
LightCyan E0FFFF	CadetBlue 5F9EA0	MediumAquamarine 66CDAA	Aquamarine 7FFFD4
DarkGreen 006400	DarkOliveGreen 556B2F	DarkSeaGreen 8FBC8F	SeaGreen 2E8B57
MediumSeaGreen 3CB371	LightSeaGreen 20B2AA	PaleGreen 98FB98	SpringGreen 00FF7F
LawnGreen 7CFC00	Green 00FF00	Chartreuse 7FFF00	MedSpringGreen 00FA9A
GreenYellow ADFF2F	LimeGreen 32CD32	YellowGreen 9ACD32	ForestGreen 228B22
OliveDrab 6B8E23	DarkKhaki BDB76B	PaleGoldenrod EEE8AA	LtGoldenrodYello FAFAD2
LightYellow FFFFE0	Yellow FFFF00	Gold FFD700	LightGoldenrod EEDD82
Goldenrod DAA520	DarkGoldenrod B8860B	RosyBrown BC8F8F	IndianRed CD5C5C
SaddleBrown 8B4513	Sienna A0522D	Peru CD853F	Burlywood DEB887
Beige F5F5DC	Wheat F5DEB3	SandyBrown F4A460	Tan D2B48C
Chocolate D2691E	Firebrick B22222	Brown A52A2A	DarkSalmon E9967A
Salmon FA8072	LightSalmon FFA07A	Orange FFA500	DarkOrange FF8C00
Coral FF7F50	LightCoral F08080	Tomato FF6347	OrangeRed FF4500
Red FF0000	HotPink FF69B4	DeepPink FF1493	Pink FFC0CB
MediumVioletRed C71585	VioletRed D02090	Magenta FF00FF	Violet EE82EE
Plum DDA0DD	LightPink FFB6C1	PaleVioletRed DB7093	Maroon B03060
Orchid DA70D6	MediumOrchid BA55D3	DarkOrchid 9932CC	DarkViolet 9400D3
BlueViolet 8A2BE2	Purple A020F0	MediumPurple 9370DB	Thistle D8BFD8

## Организиране със списъци

Всеки използва списъци, за да организира информацията си или задачите с приоритет. Списъците ни помагат да мислим по-ясно. Може би сте направили подреден или номериран списък на домашните, които трябва да направите тази седмица. В примерния списък по-долу цифрите биха могли да посочват степента на важност на задачата или реда, в който планирате да ги направите.

Задачи за домашно:

1. Изучаване на правопис
2. Направи оформление на семестриалния реферат
3. Завършване на доклад на книга
4. Учене за изпит по математика

Други списъци могат да бъдат неподредени – това означава, че те не задават приоритет по никакъв начин. Характерното за тези списъци е че техните градивни компоненти споделят нещо общо – обща черта, идея, момент или др.

Например списък с идеи за подарък за брат или сестра може да бъде такъв тип неподреден списък. Неподредените списъци често имат водещ графичен символ (graphic bullet) пред всеки елемент.

Рожден ден на Ким:

- Тенис ракета
- MP3 плеър
- Билети за кино
- Дневник

Списъците са много полезни при дизайна на една уеб страница, т.к. могат да организират информацията в нея и да я направят по-разбираема. Има няколко вида списъци с много специфична употреба. Неподредените списъци обикновено имат водещ символ (bullet), за да се различават елементите в списъка. Подредените списъци имат номерирани или буквени елементи.

Списъците с дефиниции използват отместени абзаци, за да създадат йерархия от елементи. Всеки елемент в списъка се нарича “елемент от списък”.

Ето обобщение от етикети, използвани за различни списъци.

Етикет (Tag)	Значение	Използва се за
<ul>	Неподреден списък	Начален етикет за списък с водещ символ (bullet)
<ol>	Подреден списък	Начален етикет за номериран списък
<li>	Елемент от списък	Начален етикет за елемент с водещ

		символ / номериран (отстъп)
<dl>	Списък с дефиниции	Начален етикет на списък с дефиниции.
<dt>	Елемент от дефиниция	Начален етикет за дефиниране на дума (т.е. дума от първото ниво от йерархията на списъка с дефиниции)
<dd>	Описание на дефиниция	Начален етикет за Дефиниция (т.е. за текста от второто ниво от йерархията на в списъка с дефиниции)

Обърнете внимание, че всеки от горните начални етикети има и съответен затварящ етикет.

По-долу са дадени три примера за използването на етикетите за списък. В дясно от всеки от тях е показана визуализацията в уеб страница.

### Неподреден списък с неща за вършене / An unordered "To-do" list /

To do list (Списък с неща за вършене)

```
<ul>
<li>pick up dry cleaning</li>
<li>go to grocery</li>
<li>get new CD</li>
</ul>
```

To do list

- ◆ pick up dry cleaning
- ◆ go to grocery
- ◆ get new CD

### Подреден списък с десерти

Favorite desserts (любими десерти)

```
<ol>
<li>strawberry pie</li>
<li>chocolate cake</li>
<li>apple crisp</li>
</ol>
```

Favorite desserts

1. strawberry pie
2. chocolate cake
3. apple crisp

### Списък с дефиниции на "Неща, които трябва да знам"

Things to know (Неща, които трябва да знам)

```
<dl>
<dt>HTML</dt>
<dd>Hypertext Markup Language</dd>
<dt>browser</dt>
<dd>a program used to interpret HTML</dd>
</dl>
```

Things to know

HTML  
Hypertext Markup Language  
browser  
a program used to interpret HTML

## Добавяне на интереси чрез изображения

Изображенията добавят интерес и стойност към уеб страницата. Те се използват, за да илюстрират информация, да добавят акцент и като цяло, да направят уеб страницата по-привлекателна. Файловете с изображения могат да се записват в много формати – вероятно сте чували, а може би и сте работили с файлове с разширения .bmp, .jpg, и .tif. Разширението .jpg е най-популярният избор за изображения, предназначени за уеб страници. Файлове с разширение .bmp не може да се използват. В следващите глави ще научите повече за видове файлове с изображения и ще експериментирате със създаването на изображения за уеб страници.

Изображенията се включват в HTML кода с етикета <img>, който не включва затварящ етикет, но има атрибут src, който указва местоположението на файла. Буквите “src” са съкращение от source, което означава източник. Информацията между кавичките е местоположението на изображението на сървъра.

****

Понякога потребителите изключват възможността на техните браузъри да показват изображения, защото това ускорява зареждането на уеб страницата. Друг път изображението не се зарежда както трябва, което може да е резултат от липсващ или неправилно поставен файл на сървъра.

Това е, което ще се покаже на уеб страницата, ако файлът с изображение не може да се възпроизведе.



Използвайте атрибута **alt = “description”** за да кажете на потребителя какво липсва, ако снимката не може да бъде показана. В горния пример на този атрибут е направено следното присвояване: **alt = “cool picture”**. Слепите хора имат компютри и браузъри, които “изговарят” думите на уеб страниците. Съдържанието на **alt** атрибута се изговаря на глас от такива браузъри и е необходим за тези хора, за да разберат напълно съдържанието на страницата.

Към етикета <img> могат да се добавят атрибути за височина (height) и ширина (width), за да се гарантира правилното поставяне на изображението на страницата.

Ето пример за етикет на изображение с атрибути за размер и описателен **alt** атрибут.

```

```

Когато изображенията не се намират в същата папка като HTML файла, който е предназначен за показване на изображението, трябва да се зададе "път" до изображението.

```

```

В този пример файлът се намира в поддиректория pictures на директорията, в която е файла с HTML кода. Понякога пътят показва, че местоположението на изображението не е в поддиректория, а в директория "над" местоположението на HTML файла или на същото ниво в йерархията от директории, но в различна директория. Следният път се отнася за изображение, което е в директория, която е на едно и също ниво с директорията на HTML файла. Двете точки (..) казват на компютъра да "се върне" една директория по-нагоре в йерархията и да погледне в папката "picture" (снимка) на това ниво, за да открие изображението pic1.jpg.

```

```

Пътищата към изображения и други файлове могат да станат много сложни. От решаващо значение за дизайнера е да планира внимателно файла и структурата на директорииите за уеб страници, за да гарантира лесен и удобен процес на работа.

## Хипервръзки (Hyperlinks)

Вълнуващ аспект от Интернет е възможността да се прави връзка към други уеб страници в рамките на сайта или други уеб сайтове. Това е същността на концепцията за интернет – преминаване от един сайт към друг чрез така наречените хипервръзки. Кодът, който се използва за създаването им, е много лесен.

Хипервръзките могат да се състоят от текст или изображения и се определят от етикета `<a>` (от ANCHOR) , заедно с **href** атрибут.

Това е основният синтаксис за създаване на хипервръзка:

```
<a href = "url and/or file name"> words or picture the user clicks on </a>
```

Ето някои примери:



Свързване към друга страница в същия сайт, използвайки текста "My Favorite Movies" (Моите любими филми) като връзка:

```
<a href= "myfavoritemovies.html">My Favorite Movies</a>
```

Свързване към отдалечена страница, използвайки текст като връзка:

```
<a href= "http://myfavoritemoviestarsite.com">My Favorite Movie Star</a>
```

Свързване към отдалечен сайт, използвайки изображение в папка "picture" на този сайт.

```
<a href="http://anothersite.com"></a>
```

Понякога е желателно да отворите страница или сайт в нов прозорец. Към anchor етикета може да се добави target атрибут \_blank-, за да се каже на брауъра да отвори страницата, към която се прави връзка в нов прозорец.

Следният код произвежда хипервръзка, която отваря целевия сайт в нов прозорец, вместо да замества съдържанието на текущия прозорец.

```
<a href="http://www.microsoft.com" target="_blank">Microsoft</a>
```

## Добавяне на цвят към хипервръзки

За посетителите на уеб страници е полезно да имат визуален помощник, чрез който да следят дали са посещавали вече или не дадена хипервръзка. За целта обикновено използваните (посетените) хипервръзки се оцветяват в различен цвят. Цветовете за непосетени връзки, посетени връзки и активни връзки могат да бъдат зададени като част от елемента BODY. Цвета за активна връзка се използва, когато кликнете върху нея.

```
<body link = "#FF0000> задава червен цвят за неизползваните връзки
```

```
<body vlink = "#00FF00> задава зелен цвят за използваните връзки
```

```
<body alink = "#0000FF> задава син цвят за активните връзки
```

## Каскадни стилони таблици (Cascading Style Sheets)

Сега, когато разгледахме основните HTML елементи, може би ще бъдете развълнувани да научите колко много от тях могат да бъдат включени в каскадните стилони таблици (CSS). CSS дават невероятна гъвкавост в създаването на "външния вид", който искате за дадена уеб страница и улесняват създаването на този "специален външен вид".

Първоначалната цел на HTML е била определянето на съдържанието на уеб документ. Тъй като основните браузъри добавят нови етикети, повечето от които определят стила, стана много трудно да се разделя съдържанието от елементите на дизайн на страницата. Стилните етикети като `<font>` и `<color>` остават заплетени в съдържанието на параграфа `<p>`. Промяната на атрибутите на всеки елемент на един голям уеб сайт отнема много време!

Стиловете са създадени от W3C, които станаха част от HTML 4.0. Това позволи на дизайнерите да разделят съдържание от стил, използвайки каскадни стилни таблици (CSS). Един сайт може да придобие напълно нов външен вид в рамките на няколко минути и няколко промени в документа за стилни таблици.

Стиловата таблица върви ръка за ръка с HTML страницата. Това е просто списък, който казва на HTML страницата как да се справя с форматирането на визуалните аспекти от страниците. CSS файлът се създава отделно и се записва със **.css** разширение. Вътре в HTML страницата ще има връзка към CSS страницата.

В допълнение към единството и лекота за стил, каскадните стилни таблици подобряват времето за зареждане на страницата, защото има по-малко данни за прехвърляне от сървъра към компютъра на потребителя. Когато уеб браузърът изтегля първата страница от даден сайт, той кешира (съхранява в паметта) стилната таблица, за да я използва към всяка допълнителна страница от същия сайт. Това помага на всички следващи страници да се зареждат по-бързо. Допълнително преимущество е това, че CSS дава на сайтовете по-характерен вид, тъй като стилът е последователен от страница на страница.

Стилните таблици могат да определят стилните елементи по няколко начина. Стиловете могат да се определят в рамките на отделни HTML елементи, в рамките на елемента HEAD (Глава) `<head>`, или в един или няколко външни CSS файла—всички по едно и също време! Въпреки, че това може да стане доста объркващо за начинаещ дизайнер, важно е да разберете как взаимодействат множество стилове в рамките на една уеб страница или сайт. Но не се тревожете, този модул стъпка по стъпка ще изгради вашите умения и знания в употребата на CSS.

Терминът “cascading” (каскадни) е основата на силата, която CSS предлага на дизайнерите. Тъй като стиловете могат да бъдат определени в много места (браузър, етикети и външни документи), се създава ред от приоритетни правила. Действителният стил, приложен към съдържанието, следва следните правила, където правило номер четири има най-голям приоритет:

1. Браузър по подразбиране
2. Външна стилна таблица
3. Вътрешна стилна таблица (вътре в етикета `<head>`)
4. Инлайн (inline) стил (вътре в HTML етикет)

Инлайн стилът има най-голям приоритет, което означава, че има предимство пред всеки стил, определен в което и да е ниво. Тази "cascading" (каскада) дава на дизайнерите удобство и лекота при задаване на стилове във външен CSS документ или в рамките на етикета <head> и силата да отменят тези стилове в етикета на HTML елемент както желаят.

## Синтаксис на CSS

Синтаксисът на CSS има три части: селектор (selector), свойство (property) и стойност (value), които се задават в този ред

### Селектор {property: value} (свойство: стойност)

Селекторът е HTML елементът, който трябва да бъде определен, свойство (property) е атрибутът на този елемент, който трябва да бъде променен, а стойността (value) е определеното качество на този елемент.

*Ето няколко примера:*

#### **body {color: red}**

Следният пример създава отстъп на всеки параграф три пъти ширината на символа главна буква "M" (вижте "em" в речника). За да добавите повече свойства (properties) към параграфа, просто ги добавете на следващия ред. Уверете се, че крайната скоба (}) е включена, за да се затвори селектора.

#### **p {text-indent: 3em}**

Един селектор може да има няколко стила. Те са включени в рамките на скобите и са разделени от точка и запетая. Ето една дефиниция за стил за елемента HEADER. Свойствата са поставени на отделни редове, за подобряване на четливостта, но това не е необходимо.

```
h1  
{  
text-align: center;  
color: red;  
font-family: arial  
}
```

Същият стил може да бъде определен за множество елементи чрез групиране на селектори. В този пример, всички HEADER елементи (разделени със запетая) са определени в единна стилова дефиниция:

**h1,h2,h3,h4,h5,h6**

```
{
text-align: center;
color: red;
font-family: arial
}
```

## Селекторни (selector) категории

Припомнете си, че първата дума в CSS синтаксиса (selector {property: value}) е “selector”(селектор). Селекторът е HTML елемент, който трябва да бъде определен в съответния HTML код. Селекторите (selectors) попадат в 3 категории:

### 1. Селектор на HTML етикет – променя стила на съществуващия HTML етикет.

*Синтаксис:*

```
HTMLtagSelector
{
property:value;
property:value;
}
```

Всички примери до сега бяха селектори на HTML етикети – HTML етикети <body>, <p> и <h1>. В допълнение към определяне на стойността на различни свойства на HTML етикети, вие можете всъщност да промените начина, по който браузъра интерпретира стандартни етикети, като този за удебеляване на текст. В следващият пример ефектът от <b> HTML етикетите се променя в цялата страница – когато баузъра прочете и CSS файла, текстът се изобразява като червен.

*Пример:*

CSS файл	HTML файлов сегмент (с препратка към свързания .css файл)
b { font-family:arial; font-size:24px; color:red }	<body> <b><i>This is the redefined bold HTML</i></b> </body>

*Резултат:*

***This is the redefined bold HTML***

**2. Класов селектор (class selector) – определя стила, който да се използва, без да засяга съществуващ HTML етикет. Това означава, че той няма да предефинира изцяло един HTML етикет, както в предния пример използвахме получер (bold), а ще влезе в сила само, когато бъде поставен вътре в определен етикет.**

*Синтаксис:*

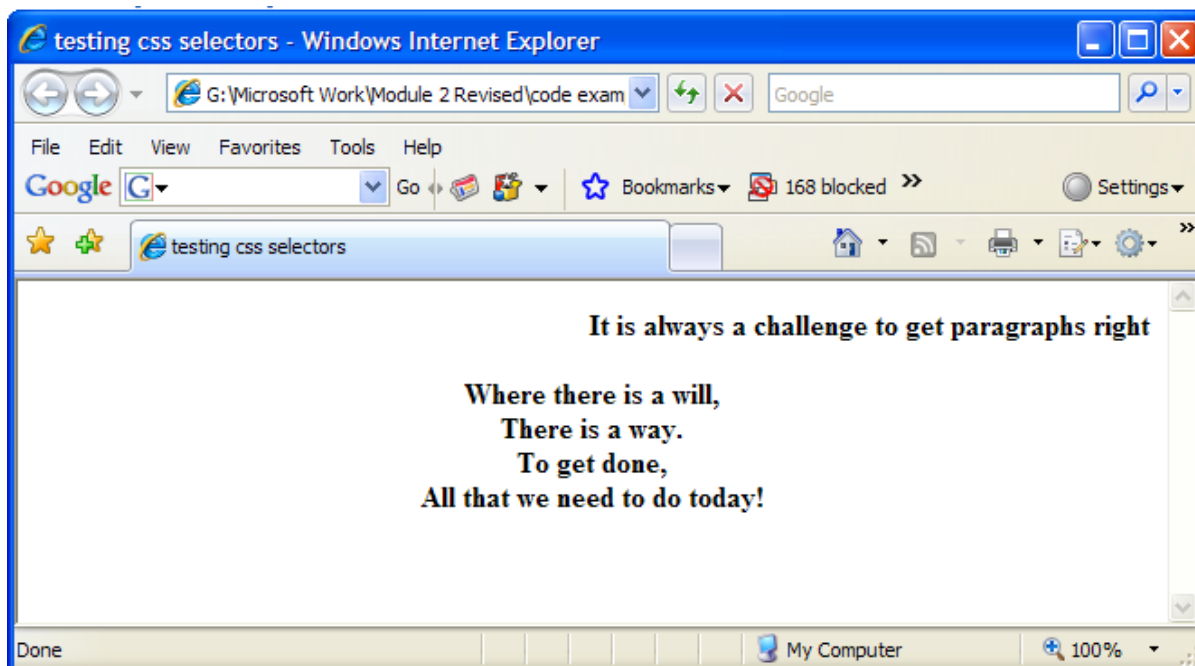
```
tag.classSelector
{
  property: value
}
```

Класовите селектори дават възможност на дизайнера да създава различни уникални стилове за HTML елементи, без те да отменят съществуващите стилове. Класовият селектор дава свободата да се дефинират различни стилове за даден HTML елемент.

Понякога би било полезно да имаме параграф, който е центриран, а понякога би било полезен и дясно подравнен параграф. По-долу има два примера за класови селектори, които създават уникални стилове за подравняване за параграфния етикет.

CSS файл	HTML файлов сегмент (с препратка към свързания .css файл)
p.right { text-align:right }	<body>... <p class="right"> It is always a challenge to get paragraphs right </p>
p.center { text-align:center }	<p class="center"> Where there is a will,  There is a way.  To get done,  All that we need to do today! </p> </body>

*Резултат:*



*Ето още един пример:*

Въпреки, че е желателно шрифтовете на една веб страница да са еднакви, има случаи, когато е целесъобразно да нямаме уникален външен вид. Чрез употребата на класови селектори, CSS позволява използването на различни формати за същия етикет. За по цветна страница, авторът може да иска някои параграфи да са с червени букви, други със син шрифт, а трети с бели надписи.

Ще са необходими три класови селектори; всеки с различна цвятова дефиниция.

CSS файл	HTML файлов сегмент (с препратка към свързания .css файл)
p.blue_text {color:blue}	<p class="blue_text">blue font</p>
p.red_text {color:red}	<p class="red_text">red font!</p>
p.white_text color:white}	<p class="white_text">white font</p>

*Резултат:*

blue font

red font!

white font

Някои стилове могат да се използват с повече от един HTML елемент или етикет. В следващия пример класов селектор се определя без препратка към определен HTML етикет, след което се използва в съюз с <b> елемент в специфично място в

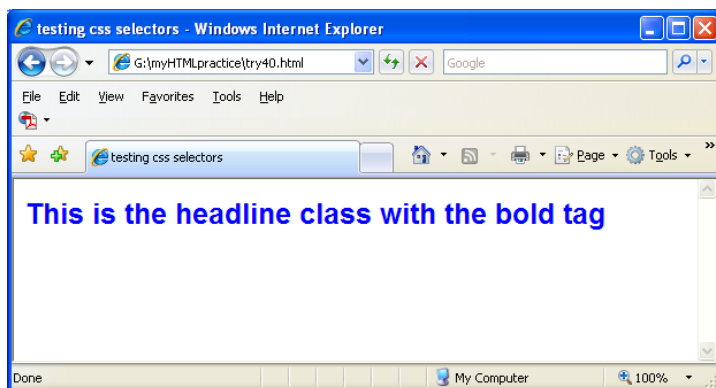
уеб страница. Същият стил може да бъде приложен към етикет на параграф или заглавка (heading).

*Синтаксис:*

```
.classSelector  
{  
  property: value  
}
```

CSS файл	HTML файлов сегмент (с препратка към свързания .css файл)
<pre>.headline { font-family:arial; font-size:25px; color:blue }</pre>	<pre>&lt;body&gt; &lt;b class="headline"&gt;This is the headline class with the bold tag&lt;/b&gt; &lt;/body&gt;</pre>

*Резултат:*



**3. ID селектор – определя стил във връзка с обекти, които имат уникално потребителско ID.**

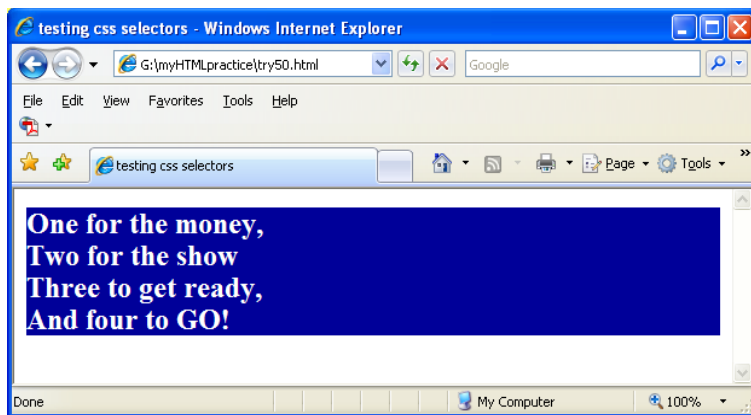
*Синтаксис:*

```
#IDSelector{  
property: value;  
property: value;  
}
```

Правилото за стил съвпада с елемент, който има ID атрибут със стойността, определена в стила. В следващия пример стиловете, определени в ID селектора "top" (горна част) ще бъдат приложени към етикета <div> в HTML кода. Забележете как се препраща стила в HTML кода – id="top".

CSS файл	HTML файлов сегмент (с препратка към свързания .css файл)
<pre>#top {   background-color: #000099;   color: #FFFFFF }</pre>	<pre>&lt;body&gt;    ... &lt;div id="top"&gt;   &lt;h2&gt;     One for the money,&lt;br&gt;     Two for the show&lt;br&gt;     Three to get ready,&lt;br&gt;     And four to GO!  &lt;/h2&gt; &lt;/div&gt; &lt;/body&gt;</pre>

*Резултат:*



## Поставяне на стилови таблици (style sheets)

Когато страницата се възпроизвежда в прозореца на браузъра, браузърът чете всяка свързана стилова таблица и форматира по съответния начин показаното. Има три начина за прилагане на стилова таблица към HTML документ.

### Външни стилови таблици (External Style Sheets)

Когато дизайнер създава сайт с множество страници, външните стилови таблици са идеалното решение. Всички стилови дефиниции се записват в отделен текстов файл и се “свързват” (linked) към всяка една страница в уеб сайта. Стилите могат лесно да бъдат променяни за всички страници, като просто се променят стиловите дефиниции в документа на външната стилова таблица.

Етикетът за връзка, който прилага външната стилова таблица към уеб страницата отива вътре в главния (head) раздел от HTML кода.

```
<link rel="stylesheet" type="text/css" href="myStyle.css">
```



Браузърът прочита стиловата дефиниция от файла mystyle.css и създава съответно веб страницата.

Създайте веб страница, показваща следния текст по начина, по който той изглежда тук в две лесни стъпки.

## CSS makes HTML even easier

1. Определете стила в селектора на HTML h1 етикет в .css файла. Въведете следния код в текстов документ. Запазете като myStyle.css.

```
h1 {
  color: blue;
  font-family: Arial, Helvetica, sans-serif;
  background-color: yellow;
}
```

2. Създайте още един текстов документ и въведете следния HTML код.

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="myStyle.css">
  </head>
  <body>
    <h1>CSS makes HTML even easier</h1>
  </body>
</html>
```

Семейството шрифтове sans serif бе избран, тъй като обикновено е най-лесния за четене. Шрифтовете са или serif или sans (без) serif. Шрифтовете serif имат малки линии в символите в края на всяка черта.

Много нови веб дизайнери често използват модерните шрифтове, които намират в техните компютри. Но ако зрителите нямат тези определени шрифтове на компютрите си, текстът няма да се визуализира коректно.

Съществуват много свойства и стойности, които могат да бъдат използвани за промяна на шрифт; размер на шрифт, стил на шрифт, дебелина на шрифт и вариант на шрифт са само някои от тях. За щастие, повечето от тях са направени така, че лесно да се тълкуват и запомнят.

### Вътрешни стилони таблици (Internal Style Sheets)

Вътрешните стилови таблици се използват за създаване на уникален стил на един документ. Вътрешните стилове са дефинирани в рамките на раздел <head> от HTML файла, използвайки етикета <style>.

*Синтаксис:*

```
<style type="text/css">give all declarations here</style>
```

Създайте същата уеб страница, както в предното упражнение, използвайки инструкциите.

### **1. Създайте друг текстов документ и въведете следния HTML код.**

```
<html>
<head>
<style type="text/css">
  h1 {
    color: blue;
    font-family: Arial, Helvetica, sans-serif;
    background-color: yellow;
  }
</style>
</head>
<body>
<h1>CSS makes HTML even easier</h1>
</body>
</html>
```

### **Инлайн (Inline) стилове**

Инлайн стиловете жертват много от предимствата на външните и вътрешни стилове. Те трябва да се използват пестеливо и само, когато е важно да се замести даден стил, дефиниран в рамките на други стилови таблици. Ще си спомните от по-ранната ни дискусия за каскадирането (cascading), че инлайн стиловете имат най-висок приоритет и ще заместят стилове, дефинирани от други стилови таблици.

Създайте същата уеб страница, както в предишното упражнение, използвайки следните инструкции.

### **1. Създайте друг текстов документ и въведете следния HTML код.**

```
<html>
<body>
<h1 style="color: blue;font-family: Arial, Helvetica, sans-serif; background-color:
yellow">
  CSS makes HTML even easier
```

```
</h1>  
</body>  
</html>
```

Можете да научите повече за CSS, на адрес <http://www.w3schools.com/css/>.

## Стандарти и достъпност

Заявената цел от W3C е “Да водим Световната глобална мрежа към пълния си потенциал чрез разработване на протоколи и насоки, които гарантират дългосрочен растеж на Мрежата.” W3C има отговорността да задава стандарти, да осигури еднаквост и достъпност в цялата Мрежа. Много организации по целия свят са допринесли за работата на W3C.

HTML стандартите, установени от W3C, както и повечето издания на компютърни софтуери, HTML е номериран последователно, като се приемат нови версии и набор от стандарти. HTML 4.0 е настоящият набор от стандарти, а HTML 5.0 е в процес на разработване.

W3C също така установява стандарти за достъпност, за да гарантира, че всеки има възможност за достъп до Мрежата. Стандартите за достъпност са базирани на идеята за осигуряване на ясен, пълен и прост език и дизайн, по такъв начин, че хората, които имат комуникативни увреждания, като слепота или тези, които имат когнитивни увреждания, или за тези за които езика на уеб страницата не е първи език на потребителя, могат да участват пълноценно. Можете да научите повече за стандартите за достъпност на W3C, на адрес <http://www.sitepoint.com/article/w3c-accessibility-guidelines>.

Уеб страницата на W3C, на адрес <http://www.w3.org/> съдържа информация за тази организация и много ресурси за уеб дизайнери.

## Резюме

След като бе представен най-простият HTML код, читателят трябва да е готов да започне създаването на уеб страници. Етикети и атрибути формират основата на уеб страниците. Специални символи позволяват на дизайнерите да включват символи, които не са на клавиатурата или които могат да причинят непредсказуеми резултати при възпроизвеждане в браузъра. Цветът може да направи една страница привлекателна и да помогне в организирането на информацията. Хипервръзките дават на страницата силата, която е самата същност на Мрежата. Таблицы и списъци организират информацията. CSS е мощен инструмент за създаване на привлекателни уеб сайтове с последователен стил.