



WEB ACADEMY

Front-end за Начинаещи



HTML

За HTML

4

- HTML – съкр. от *Hyper Text Markup Language*
- HTML е програмен език за маркиране, който уеб браузърите интерпретират за да визуализират текст, изображения и други материали.
- Незададената характеристика за всяка точка се определя в браузъра, като могат да бъдат променяни или засилвани с допълнителни дизайнерски страници, наречени Cascading Style Sheets.
- HTML поддържа многоредови коментари във формат
<!-- Това е коментар, който няма да се интерпретира от браузъра -->



Основна информация



- Етикетът е специален текст заграден от символите < и >
Например <BODY> е етикет (таг) с име BODY
- Крайният (затварящ) етикет включва '/' след '<'.
Например </BODY> е краен етикет.
- Имената на етикетите (таговете) не са чувствителни към големината (регистъра) на символите, т.е изразите <BODY> и <BoDY> имат едно и също значение.
- Съвкупността от начален етикет, съдържание и краен етикет се нарича елемент.

Основна информация



- Някои елементи нямат затварящ елемент и се наричат единични. Например ``
- Елементите не могат да се припокриват, т.е. таговете се влагат един в друг, като първият отворен таг се затваря последен. Например `<i><u>...</u></i>`
- Таговете притежават свои атрибути, като например цвят, дължина, ширина, и други
- HTML таговете са най-малката съставна част на един HTML документ.

- ☐ <area>
- ☐ <base>
- ☐

- ☐ <col>
- ☐ <hr>
- ☐
- ☐ <input>
- ☐ <link>
- ☐ <meta>
- ☐ <param>
- ☐ <source>
- ☐ <track>
- ☐ <wbr>

Както вече споменахме, единичните елементи, нямат съответстващ на тях затварящ елемент.

Посочения тук списък съдържа използваните в тази презентация единични тагове

Основна информация

8

- Празните редове в HTML кода се игнорират от браузъра
- Повече от един последователен празен интервал се игнорира от браузъра
- Символи като <, >, &, могат да създадат проблеми, ето защо трябва да бъдат "ескейпнати"



HTML entities

9

Result	Description	Entity Name	Entity Number
	non-breaking space	 	
<	less than	<	<
>	greater than	>	>
&	ampersand	&	&
¢	cent	¢	¢
£	pound	£	£
¥	yen	¥	¥
€	euro	€	€
©	copyright	©	©
®	registered trademark	®	®




Стойноста на Entity name е чувствителна към регистъра на буквите!



Глобални атрибути




Най-често използваните глобални атрибути са:

- `id` - задава **уникален** идентификатор на елемента
- `class` - задава обръщение към един или повече CSS класа
- `title` - задава допълнителна информация за елемента, която се визуализира когато курсора е върху елемента
- `style` - позволява директно задаване на CSS за елемента
- `data-*` - позволява добавянето на допълнителна информация за елемента.
- `tabindex` - задава `tab` индекса на елемента, т.е. в каква последователност ще се селектират елементите при натискане на бутона  от клавиатурата

Глобални атрибути



 **Запомнете:**
Глобалните атрибути не напразно се наричат глобални - те могат да бъдат използвани с всеки един HTML елемент!

```
<!-- Пример за използване на глобални атрибути -->  
<html>  
  <body>  
      
    </body>  
</html>
```



Глобални атрибути



Таблица с всички глобални атрибути

Attribute	
<u>accesskey</u>	Specifies a shortcut key to activate/focus an element
<u>class</u>	Specifies one or more classnames for an element (refers to a class in a style sheet)
<u>contenteditable</u>	 Specifies whether the content of an element is editable or not
<u>contextmenu</u>	 Specifies a context menu for an element. The context menu appears when a user right-clicks on the element
<u>data-*</u>	 Used to store custom data private to the page or application
<u>dir</u>	Specifies the text direction for the content in an element
<u>draggable</u>	 Specifies whether an element is draggable or not
<u>dropzone</u>	 Specifies whether the dragged data is copied, moved, or linked, when dropped
<u>hidden</u>	 Specifies that an element is not yet, or is no longer, relevant
<u>id</u>	Specifies a unique id for an element
<u>lang</u>	Specifies the language of the element's content
<u>spellcheck</u>	 Specifies whether the element is to have its spelling and grammar checked or not
<u>style</u>	Specifies an inline CSS style for an element
<u>tabindex</u>	Specifies the tabbing order of an element
<u>title</u>	Specifies extra information about an element
<u>translate</u>	 Specifies whether the content of an element should be translated or not



Основни елементи

13

<!DOCTYPE> елемент

- Декларира се първи, още преди <html> тага.
- Валидира документа.
- Не е HTML таг - той е инструкция за уеб браузъра - указва HTML версията, на която е написана страницата

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">  
<html>  
  <head>  
    <meta charset="UTF-8">  
    <title>Заглавие на документа</title>  
  </head>  
  <body>Текст на документа</body>  
</html>
```



Основни елементи

14

<html> елемент

- Указва на браузъра, че това е HTML документ.
- Отбелязва началото и края на документа
- Съдържа всички други негови елементи (с изключение на <!DOCTYPE> елемента)

<head> елемент

- Съдържа заглавието на документа
- Разполага се преди <body> елемента
- Може да съдържа стилове, скриптове, енкодинг и др.



Основни елементи

15

<title> елемент

- Задава заглавието на документа
- Визуализира се в заглавната лента на браузъра
- Винаги се намира вътре в **<head>** елемента

<body> елемент

- Описва основното съдържание на HTML документа. (основно това, което се визуализира от браузерите)
- Поставя се веднага след крайния етикет **</head>**
- Винаги се намира вътре в **<html>** елемента



Вмъкване на CSS



- чрез деклариране на външен файл вътре в `<head>` тага
`<link rel="stylesheet" type="text/css" href="file.css" />`
- чрез използване на `<style>` таг вътре в `<head>` тага
`<style>`
`<!--`
`.content { width:425px; }`
`---->`
`</style>`
- чрез директно задаване на стил на ниво елемент
`<i class="content" style="color:red;">Text</i>`

Запомнете:

1. Добрата практика е да се използва първия начин
2. Стилите на ниво елемент препокриват предишните дефиниции



Вмъкване на JavaScript



- чрез деклариране на външен файл вътре в `<head>` тага
`<script src="file.js"></script>`
- чрез използване на `<script>` таг
`<script>`
`<!--`
 `alert('Здравейте');`
`-->`
`</script>`
- чрез директно задаване на JS код на ниво елемент
`<i onclick="alert('Здрасти');">Кликнете тук</i>`

Добрата практика изисква:

1. Функциите да се декларират чрез метод 1 или 2
2. На ниво елемент, да се прави извикване на вече дефинираните функции



URL Адреси



URL (*Uniform Resource Locator*) - Унифициран локатор на ресурси представлява универсален начин за указване на адреса на даден ресурс в интернет.

URL адресите биват 2 вида:

- абсолютни (пълни) - указват адреса на произволен ресурс в интернет. Използват се предимно за указване адреса на чужди (външни) сайтове.
- относителни - указва адреса на ресурс, спрямо текущия файл. Използва се за достъп до ресурси от същия сайт.

URL Адреси



Абсолютните (наричани още пълни) URL адреси се задават във формат:

протокол : //домейн[:порт]/път/до/файла/име.разширение

Използването предимно на абсолютни URL адреси води до определени неудобства, тъй като:

- Необходимо е да се изписват сравнително дълги URL адреси
- Ако се промени местоположението на уеб сайта на друг адрес трябва всички абсолютни адреси използвани в него да бъдат заменени с нови.

Ето защо е значително по-удобно използването на относителни адреси.



URL Адреси



Относителните URL адреси се указват по няколко начина в зависимост от местоположението на файла :

- файл от същата директория
 - указва се с *"име.разширение"* или с *"/име.разширение"*
- файл в под директория
 - указва се с *"под директория/име.разширение"*
- файл в по-горна директория
 - указва се с *"../.. /име.разширение"*
 - всяко двуточие представлява едно ниво по-горна директория.
- произволен файл на същия компютър
 - указва се с *"/директория/под директория/име.разширение"*

Относителните адреси се изчисляват, като за основа се използва базовия адрес.

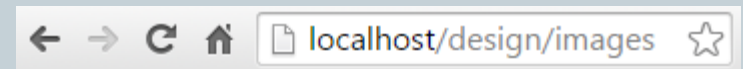
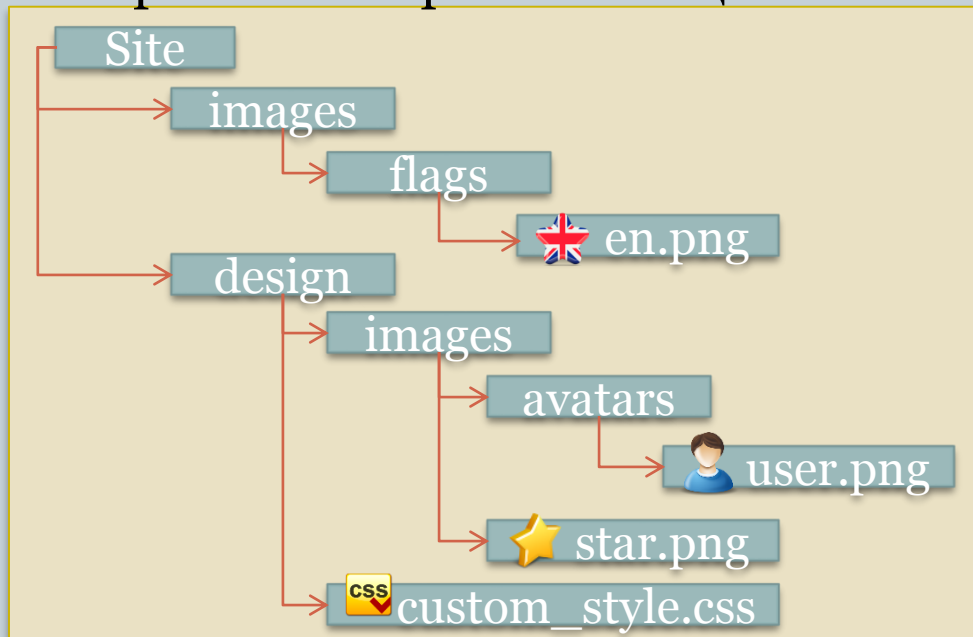


URL Адреси

21

Примери за относителни адреси:

- файл от същата директория - `"star.png"` или `"./star.png"`
- файл в под директория - `"avatars/user.png"`
- файл в по-горна директория - `"../custom_styles.css"`
- произволен файл на същия компютър - `"/images/flags/en.png"`



URL Адреси



<base> - задава стойността по подразбиране на базовия (основния) адрес. Този адрес ще се използва като базов, относно който ще се изчислят относителните адреси.

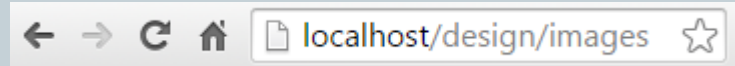
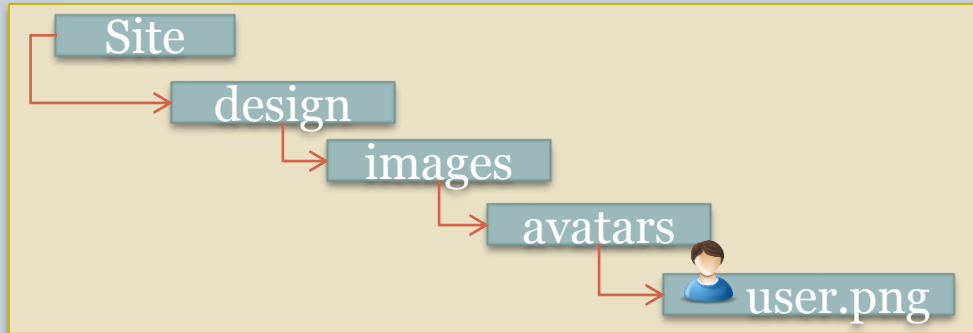
 Имайте в предвид, че:

- Може да има само един **<base>** таг и той трябва да бъде разположен като първи елемент в **<head>** тага.
- Може да има само атрибутите *href* или *target* или и двата заедно
- Стойността на атрибута *href* трябва да бъде пълният URI адрес до директорията, в която се намират файловете на уеб сайта
- Неправилно използване на **<base>** може да създаде проблем със зареждането на хипервръзките, и респективно SEO проблеми.
- **НЕ** използвайте този таг, ако не сте сигурни, какво правите! :)



URL Адреси

23



Двата примера са еквивалентни!

```
<html>
<head>
  <base href="http://localhost/design/">
</head>
<body>
  
</body>
</html>
```

```
<html>
  <!-- Няма <BASE> таг, по този начин
  базовият адрес си остава по
  подразбиране, т.е. базовият адрес е:
  http://localhost/design/images -->
  <body>
    
  </body>
</html>
```



xHTML



Какво е xHTML?!

xHTML (*eXtensible HyperText Markup Language*) представлява нов разширен формат HTML, който е съвместим с XML и по тази причина е по-стриктен и изисква точност при писането на кода.

Какво изисква xHTML?!

1. пълно изписване на атрибутите
2. затваряне на всички тагове (дори единичните)
3. единичните тагове се затварят вътре в самият таг
4. стойностите на атрибутите да се задават винаги с "кавички"
5. имената на тагове и атрибути трябва да се изписват с малки букви

Използването на xHTML е добра практика и се препоръчва от W3C!



xHTML



Примери за изпълняване на изискванията

1. пълно изписване на атрибутите
`<option selected>Text</option>` ⇔ `<option selected="selected">Text</option>`
2. затваряне на всички тагове (дори единичните)
`
` ⇔ `
`
3. единичните тагове се затварят вътре в самият таг
`` ⇔ ``
4. стойностите на атрибутите да се задават винаги с "кавички"
`<option value=29>Text</option>` ⇔ `<option value="29">Text</option>`
5. имената на тагове и атрибути трябва да се изписват с малки букви
`<OPTION Value="10">Text</OPTION>` ⇔ `<option value="10">Text</option>`
6. Всички правила заедно:
`<input type="radio" checked>` ⇔ `<input type="radio" checked="checked" />`

 **Внимание:** преди `"/>"` има интервал. т.е. `
`

МЕТА тагове

26

- Разполагат се вътре в <head> тага
- Използването на повечето метатагове не е задължително, но е препоръчително.
- МЕТА таговете са единични тагове, т.е. те нямат затварящ елемент.
- Те НЕ са предназначени за посетители на сайта
- Те НЕ се визуализират на екрана
- съдържат в себе си двойки стойности от атрибутите **name** и **content** или **http-equiv** и **content**
 - атрибутите **name** и **http-equiv** указват предназначението на тага
 - **content** съдържа стойността на МЕТА тага.



META тагове

27

Според предназначението си META таговете могат да бъдат разделени в няколко групи:

- за комуникация с браузъра
- за комуникация с работи (паяци)
- за индексирание от търсещи машини
- за вътрешна категоризация на страниците

Най-често META таговете съдържащи атрибута HTTP-EQUIV се използват за комуникация с браузъра, а META таговете съдържащи атрибута NAME се използват за да подават определена информация на търсачките.

МЕТА тагове

28

Най-често използваните МЕТА таговете за комуникация с браузъра са:

- `<meta http-equiv="charset" content="utf-8">`
 - задава кодировка на символите
- `<meta http-equiv="refresh" content="5">`
 - задава обновяване на страницата на всеки N секунди
- `<meta http-equiv="refresh" content="5;url=http://abv.bg">`
 - указва пренасочване (редирект) към друга страница след N секунди
- `<meta http-equiv="pragma" content="no-cache">`
 - служи за контрол кеширането на страниците
 - Възможните стойности са **cache/no-cache**, които съответно указват дали съдържанието да се кешира или не. По подразбиране е cache.



МЕТА тагове

29

Най-често използваните МЕТА таговете за комуникация с "паяци" са:

- `<meta name="revisit-after" content="14 days">`
 - задава колко често паяците да посещават и индексират сайта
- `<meta name="robots" content="noindex">`
 - задава дали страницата да се индексира
 - Възможните стойности са **index/noindex**
- `<meta name="robots" content="follow">`
 - да се проследяват ли линковете
 - Възможните стойности са **follow/nofollow**
- Последните 2 тага често се комбинират в един общ таг
Например: `<meta name="robots" content="index,nofollow">`



МЕТА тагове

30

Най-често използваните МЕТА таговете за комуникация с търсещите машини при индексирането на сайта са:

- `<meta name="title" content="Заглавие">`
 - задава заглавието на страницата, което търсачките, ще извеждат
- `<meta name="keywords" content="дума1, дума2, пример">`
 - указва кои са ключовите думи за страницата, разделени със ","
- `<meta name="description" content="Примерно описание">`
 - предава на търсачката описание на страницата

В последните години, търсещите машини умаловажават МЕТА таговете, придавайки тежест на съдържанието на сайта, но въпреки това употребата им е препоръчителна.



МЕТА тагове

31

Най-често използваните МЕТА таговете за вътрешна категоризация на страниците на сайта са:

- `<meta name="author" content="Йордан Енев">`
 - указва кой е автора на документа
- `<meta name="copyright" content="BGO Media">`
 - указва кой държи авторските права
- `<meta name="last edited" content="17 януари 2017 г.">`
 - указва дата на последно обновяване
- `<meta http-equiv="expires" content="30 февруари 2015 :) ">`
 - указва срока на актуалност на страницата
- `<meta name="pubdate" content="2015-02-08">`
 - указва датата на публикуване на страницата



Параграфи и Раздели



- **<p>** - дефинира параграф. Параграфът е block елемент.
- **<div>** - дефинира раздел. Разделът е block елемент.
- **** - групира няколко inline елемента.
- **
** - задава „насилствено“ преминаване на нов ред
- **<hr>** - задава разделител
 - HTML 4.01 - задава хоризонтален разделител (хоризонтална линия)
 - HTML 5 - задава тематичен разделител, и има семантично значение

Заглавия



- Таговете от `<h1>` до `<h6>` дефинират заглавия в HTML документа.
- Важността на заглавието е обратно пропорционална на номера му, т.е. `<h1>` дефинира най-важното заглавие, докато `<h6>` дефинира най-маловажното заглавие.
- Пример:
 - `<h1>Заглавие 1</h1>`
 - `<h2>Подзаглавие 2</h2>`
 - `<h3>Подзаглавие 3</h3>`
 - `<h4>Подзаглавие 4</h4>`
 - `<h5>Подзаглавие 5</h5>`
 - `<h6>Подзаглавие 6</h6>`

Цитати

34

<q> елемент

- задава кратък цитат

<blockquote> елемент

- задава секция, която е цитирана от друг източник

<cite> елемент

- задава заглавие на творбата

<abbr> елемент

- задава абривиатура или акроним
- <address>

Форматиране на текст

35

 елемент

- Задава **удебелен** текст

<i> елемент

- Задава *наклонен* текст

<u> елемент

- Задава подчертаване на текста

<center> елемент

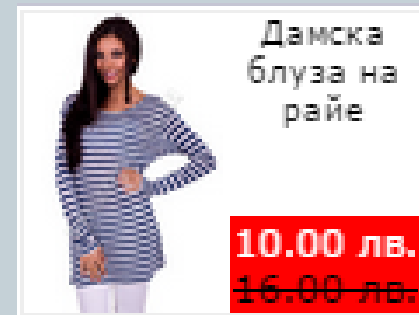
- Задава хоризонтално центриране на текста

<sub> елемент

- Задава текста в _{долен} индекс

<sup> елемент

- Задава текста в ^{горен} индекс



Форматиране на текст

36

 елемент

- Задава шрифт, размер и/или цвят на текста
- Не се поддържа от HTML5
- Атрибути:
 - face - задава шрифт.
 - color - задава цвят.
 - size - задава размер, който може да бъде относителен.

```
<body>                                <!-- Примери за работа с <font> елемента -->
  <font size="+1">Text</font>
  <font color="red">Text</font>
  <font face="Verdana">Text</font>
  <font face="Arial" color="#FFFFFF" size="13">Text</font>
</body>
```



Форматиране на текст

37

** елемент**

- Указва текст, който е заместен или изтрит от документа
- Използва се най-често в комбинация с **<ins>** елемент

<ins> елемент

- Указва текст, който е вмъкнат в документа
- Използва се най-често в комбинация с **** елемент

<s> елемент

- Указва текст, който вече не е коректен
- За заместен или изтрит текст се използва **** и **<ins>**
- В HTML4.01 се използва за указване на ~~зачеркнат~~ текст
- В HTML5 е предефиниран и указва некоректен текст



Фразеологични тагове

38

** елемент**

- Указва открояващ се текст. Да се различава от <i> тага!

** елемент**

- Указва важен текст. Да се различава от тага!

<code> елемент

- Форматира текста, като компютърен код

<samp> елемент

- Форматира текста като изходен от конзолна програма

<kbd> елемент

- Форматира текста като входен от клавиатурата

<var> елемент

- Форматира текста като променлива



Хипервръзки



Хипервръзката (*наричана още хиперлинк, или на кратко само линк*) представлява препратка към друг документ, или към конкретно място в същия документ

- задава се чрез <a> таг
- често използвани атрибути са:
 - href - указва URI на страницата към която сочи хипервръзката
 - target - задава как да се отвори връзката. Възможни стойности:
 - ✦ _self - отваря връзката в същия прозорец. Стойност по подразбиране.
 - ✦ _parent - отваря връзката в родителския прозорец.
 - ✦ _blank - отваря връзката в нов прозорец
 - ✦ _top - отваря връзката на най-високо ниво в брауъра.Използва се най-често при фрейми, когато линка се намира във фрейм.
 - name - задава името на "котвата", т.е. името на текущата позиция



Хипервръзки

41

<!-- Пример за работа със хипервръзки -->

```
<html>
<head><meta charset="UTF-8" /></head>
<body>
  <!-- Задаваме име на позицията чрез атрибута name -->
  <a name="top_place"><!-- Текста тук НЕ е задължителен --></a>
  Текст
  <!-- Линк към външна страница -->
  <a href="http://abv.bg" title="Отвори пощата">Поща</a>
  <!-- При линк към позиция, пред името на позицията се поставя # -->
  <a href="#top_place" name="footer">Върни ме горе</a>
  <!-- Атрибутите name и href могат да се комбинират в един таг -->
</body>
</html>
```



Изображения



**** - вмъква изображение вътре в HTML документ.

Често използвани атрибути:

- **src** - **задължителен атрибут**
 - задава URL адреса на изображението, което да се зареди.
- **alt** - **задължителен атрибут**
 - задава алтернативен текст, което се визуализира, ако евентуално изображението не може да се зареди.
- **border**
 - задава рамка около изображението.
- **width**
 - задава широчина на изображението.
 - стойността да бъде в пиксели или проценти.
- **height** - задава височина. Аналогично на width.



WEB ACADEMY

```

```



Кarti



<map> елемент - Създава карта, която се свързва с дадено изображение. Картата съдържа списък с области. Изображението се свързва с картата посредством `usemap` атрибут, чиято стойност трябва да съответства на `name` атр.






Често използвани атрибути:

- `name` - **задължителен атрибут**
 - трябва да съответства на атрибута `usemap` на тага ``
- `id` - глобален незадължителен атрибут
 - неговата стойност трябва да съответства на ст-та на атрибута `name`

<area> елемент - задава кликаема област в картата. Разполага се винаги вътре в `<map>` тага. Вижте атрибутите!



Таблица с атрибути на <area> тага

Attribute	Value	Description
<u>alt</u>	<i>text</i>	Specifies an alternate text for the area. Required if the href attribute is present
<u>coords</u>	<i>coordinates</i>	Specifies the coordinates of the area
<u>download</u>	 <i>filename</i>	Specifies that the target will be downloaded when a user clicks on the hyperlink
<u>href</u>	<i>URL</i>	Specifies the hyperlink target for the area
<u>hreflang</u>	 <i>language_code</i>	Specifies the language of the target URL
<u>media</u>	 <i>media query</i>	Specifies what media/device the target URL is optimized for
<u>nohref</u>	<i>value</i>	Not supported in HTML5. Specifies that an area has no associated link
<u>rel</u>	alternate author bookmark help license  next nofollow noreferrer prefetch prev search tag	Specifies the relationship between the current document and the target URL
<u>shape</u>	default rect circle poly	Specifies the shape of the area
<u>target</u>	_blank _parent _self _top <i>framename</i>	Specifies where to open the target URL
<u>type</u>	 <i>media_type</i>	Specifies the media type of the target URL



Изображения++



Чрез влагане на тага `` вътре в `<a>` можем да създадем изображение, което е хипервръзка към дадена страница

```
<a href="http://webacademy.bg/">  
    
</a>                                <!-- Пример за изображение-хипервръзка -->
```

Чрез таговете ``, `<map>` и `<area>` можем да зададем само област от дадено изображение, да бъде хипервръзка

```
                                <!-- Пример за изображение-карта -->  
  
<map name="planetmap" id="planetmap">  
  <area shape="rect" coords="0,0,82,126" alt="Sun" href="sun.html">  
  <area shape="circle" coords="90,58,3" alt="Mercury" href="mercur.html">  
  <area shape="circle" coords="124,58,8" alt="Venus" href="venus.html">  
</map>
```



Видове списъци



Списъците биват 3 вида:

- Подредени - дефинират се с `` тага
- Неподредени - дефинират се с `` тага
- Описателни - дефинират се с `<dl>` тага


Всеки списък без значение от типа си, се състои от отделни елементи, дефинирани със съответния таг

- Елементите на подредените и неподредените списъци се дефинират чрез `` таг
- Елементите на описателните списъци се дефинират чрез комбинация от `<dt>` и `<dd>`



Подредени списъци



Attribute	Value	Description
<u>compact</u>	compact	Not supported in HTML5. Specifies that the list should render smaller than normal
<u>reversed</u> 	reversed	Specifies that the list order should be descending (9,8,7...)
<u>start</u>	number	Specifies the start value of an ordered list
<u>type</u>	1 A a I i	Specifies the kind of marker to use in the list

Атрибути:

- type - задава типа на използвания маркер
- reversed - задава намаляваща стойност
- start - задава началната стойност на елементите на списъка

```
<html>
<body>

...
Меню за довечера
<ol start="0">
  <li>Боза</li>
  <li>Супа</li>
  <li>Баница</li>
</ol>

...
</body>
</html>
```



Неподредени списъци



Attribute	Value	Description
<u>compact</u>	compact	Not supported in HTML5. Specifies that the list should render smaller than normal
<u>type</u>	disc square circle	Not supported in HTML5. Specifies the kind of marker to use in the list

За задаване на типа на иконките на неподредения списък преди HTML5 може да се използва атрибута "type".

След HTML5 за задаване на типа на иконките следва да се използва CSS свойството list-style-type

```
<html>
<body>
...
Какво пия сутрин
<ul type="circle">
  <li>Кафе</li>
  <li>Боза</li>
  <li>Айрян</li>
</ul>
...
</body>
</html>
```

Описателни списъци



Създават се, чрез `<dl>` тага, а елементите се задават чрез комбинация от `<dt>` и `<dd>` таговете.

Внимание:

В HTML 4.01 `<dl>` тага задава списък с дефиниции

В HTML 5 `<dl>` тага задава описателен списък

```
<!-- Описателен списък -->  
<html>  
<body>  
<dl>  
  <dt>Биричка</dt>  
  <dd>  
    - студена, животоспасяваща  
    напитка  
  </dd>  
</dl>  
</body>  
</html>
```