

## Тема 3. Въведение в езика HTML. Структурира на HTML документа

### 1. История на HTML

**HTML (Hyper Text Markup Language)** е метаезик, който чрез маркиране разделя логическото от семантичното съдържание на текстов документ. Терминът “метаезик” произлиза от терминологията употребявана в математиката и логиката и означава: „формализиран език, със средствата на който се изследват свойствата на съответните предметни (или обектни) теории, разграничаващ равнището на самите описвани предмети и n-тото равнище на тяхното описание“.

HTML е пряк наследник на езика **SGML (Standard Generalized Markup Language)**. SGML е въведен през 1980 г, независим е от операционната система и хардуерното устройство, което означава, че може да бъде използван практически навсякъде. Приликата между двата езика се изразява в способността на HTML да използва дефиниции на типове данни (**DTD – Document Type Definition**).

HTML е независим от платформите, налични в хетерогенната компютърна мрежа, благодарение на протокола HTTP (Hyper Text Transfer Protocol). Той поддържа парадигми за създаване на хипервръзки между отделни документи. Тези връзки са базирани на URL (**Uniform Resource Locators**) адреси, които са уникални за всеки един документ в интернет пространството. Документите са достъпни чрез технологията клиент-сървър.

Средата за интерпретирането и визуализирането на HTML документи се нарича **уеб клиент** или **уеб браузър** (web browser).

Идеята за HTML се родила в European Laboratory for High-Energy Physics (CERN) в Женева, Швеция, в далечната 1989. За негов създател се посочва Тим Бърнс-Лий, като в края на 1991 г. той публикува в Интернет първият документ, наречен “HTML Tags”.

Към настоящия момент основната грижа за развитието и стандартизиране на езика имат работната група известна под името **W3C (World Wide Web Consortium)**: [www.w3.org](http://www.w3.org).

Пълното описание на препоръчителния стандарт за езика HTML е 5.1, може да се намери на линка: <https://www.w3.org/TR/html51/>.

От началото на своето създаване, езика HTML е еволюирал непрекъснато. Към момента съществуват няколко негови стандарта:

- **HTML 1.0:** Първата стандартизирана версия на езика е създадена през 8 ноември 1990 г.
- **HTML 2.0:** Появява се на 25 ноември 1995 г.
- **HTML 3.2:** Спецификацията на този стандарт се появява през 14 януари 1997 г.
- **HTML 4.0:** Появява се в края на 1997 г, без особени промени.
- **HTML 4.01:** Спецификацията на този стандарт се появява през 24 януари 1999 г.
- **HTML 5:** През 25.10.2012 Започват да се предлагат различни нови допълнения към стандартните тагове на HTML 4.01 и да се говори за нов стандарт.
- **HTML 5.1:** Стандартизиран на 29.10.2013 г. През 1 ноември 2016 година W3C пуснаха неговата окончателна версия.
- **HTML 5.2:** Стандартът е въведен на 8 август 2017 г. В тази пета втора ревизия на пета основна версия на езика HTML продължават да се добавят нови функции за по-лесното създаване на уеб приложения и осигуряване на тяхната сигурност в уеб пространството. Очаква се от този стандарт да отпаднат елементи като: <dialog>, <menu>, <menuitem> и

др. Към момента последният документ публикуван на сайта на W3C (<https://www.w3.org/TR/2017/CR-html52-20170808/>) е в процес на обсъждане.

- **HTML 5.3:** Последната актуализация на стандарта е предложена на 14 октомври 2018 г. Направени са някои доуточнения за начина на представяне на текст, изображения, абзаци и т.н.

## 2. Принципи на HTML 5

W3C дефинира няколко принципи, въз основа на които се изграждат новите актуализации на 5-тата основна версия на езика HTML5.

- **Осигуряване на поддръжка на съществуващото съдържание в Интернет:** Уеб страниците написани на по-стари версии на езика, трябва да се визуализират в уеб браузърите по същия начин, както и тези написани на HTML5.
- **Адекватно представяне на новата функционалност на езика HTML5 на по-старите браузъри:** Уеб страниците написани на HTML5, трябва да съдържат алтернативни елементи, които да гарантират съвместимостта им и с по-стари уеб клиенти.
- **Не е необходимо да се правят глобални нововъведения:** Има се предвид, че ако нещо вече е работило и е широко прието, не е нужно да бъде заменено с новите елементи, които 5-тата версия на езика HTML предлага.
- **Не асфалтирайте добре отпъкания път:** Ако определен начин на работа се е развил по естествен път и е широко приет, то по-добре е той да продължи да бъде използван, отколкото да бъде заменен с нов.
- **Еволюция, а не революция:** Ако един дизайнер е достатъчно добър, в това което прави, не е необходимо той непрекъснато да учи най-новите стандарти на HTML, освен ако те не са много по-добри и лесни за използване от тези, които вече знае и използва.
- **Осигуряване на универсален достъп:** Уеб разработките, трябва да могат да бъдат стартирани от различни видове уеб клиенти, независимо от платформата, на която те работят.

Основен принцип на дизайнът, ръководещ HTML5, е че той е предназначен за подпомагане на съществуващото вече уеб съдържание, и за гъвкавото и лесно създаване на ново. Тези принципи са изключително важни за осигуряване на съвместимостта на съществуващите вече уеб сайтове и създаването на нови такива.

## 3. Структура на HTML документа

HTML е обикновен текстов файл, с разширение .html или .htm. Трябва да направим едно уточнение, че разширението .htm е стандарт наложен от Microsoft, а разширението .html от W3C.

Всеки HTML документ съдържа текст и специални служебни елементи наречени **тагове**. Тагът представлява дума или фраза, заградена между символите за по-голямо и по-малко, т.е. **<таг>**. Таговете се използват за структуриране и форматиране на съдържанието на документа. Те биват **единични**, наречени още **празни тагове** (<img>, <link>, <br>) и **двойни**, наречени още **непразни тагове** (<p>...</p>, <h1>...</h1>).

Двойни тагове са онези, които имат затварящ таг. За отличаване на затварящия таг от отварящия се използва символа наклонена черта – „/“ (*slash*).

```
<име-на-таг атрибут="стойност-на-атрибут">
    съдържание вътре в тага
</име-на-тага>
```

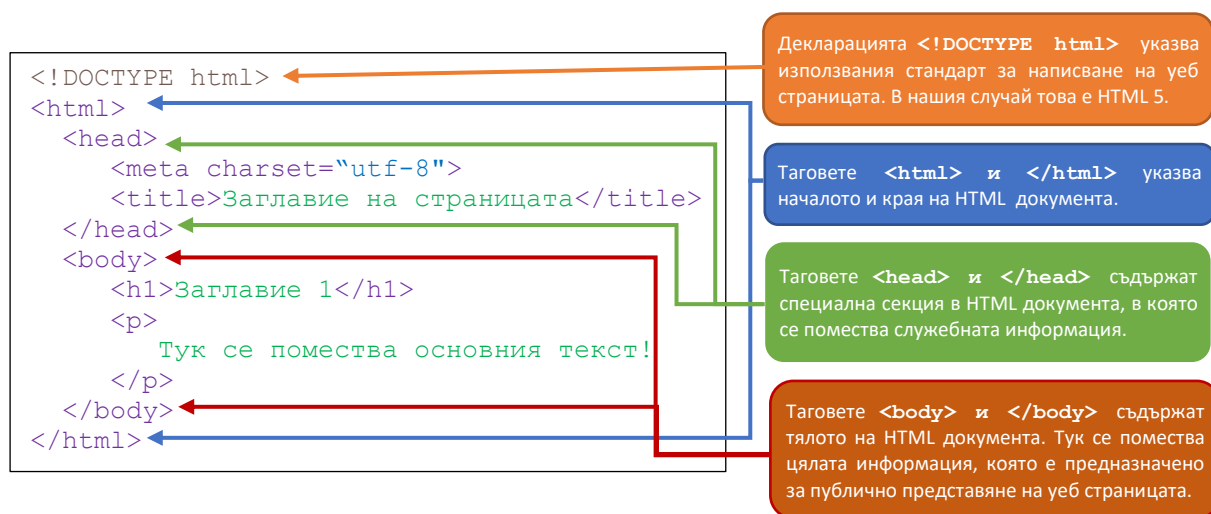
Уеб клиентът игнорира тези стойности (текстове и тагове), които не съдържат валидна команда. Това осигурява преносимостта на уеб страниците и обикновено се определя от стандарта, използван при създаването на страницата. При изписването на таговете може да се използват символи в долния (малки букви) и в горния регистър (главни букви). Смесването на тагове с главни и малки букви не се счита за грешка, но не е добра програмистка практика.

Интервалите, табулаторите или празните редове, които се слагат между таговете се възприемат от интерпретиращия уеб браузър като символ само за един интервал. Въпреки това не трябва да изписвате целия код на HTML страницата само на един ред, без да спазвате стандартите за подредба на кода. Това ще доведе до последващото трудно откриване и коригиране на евентуални грешки.

Таговете може да имат **атрибути**. Атрибутите са параметри, които позволяват да се опише някаква специфична функция на съответния таг. Стойността, която се присвоява на атрибута е строго определена константа (с изключение на URL адресите), която се огражда в символи за кавички ”стойност-на-атрибут”. Например атрибута `charset` на тага `<meta>`, позволява да се посочи от коя кодова таблица са символите, използвани в съответната уеб страница (За да се посочи, че те са Unicode, на атрибута `charset` се присвоява стойността `utf-8`).

В HTML документа могат да се съдържат и специални символи, наречени още **резервирани** или **запазени символи**, които служат за визуализирането на специални елементи. Например `&copy`; е запазена дума за символа ©, `&amp`; е запазена дума за символа & и т.н. Те ще бъдат подробно разгледани по-нататък.

Цялостният синтаксис на HTML документа е представен на фиг. 3.1.



**Фигура 3.1.** Структура на HTML документа

### Декларацията `<!DOCTYPE>`

Всеки HTML документ започва с декларацията `<!DOCTYPE стандарт-на-HTML>`. Виж фиг. 3.1 и фиг. 3.2.

Декларацията `<!DOCTYPE>` не е HTML таг. Това е инструкция към уеб браузъра, която указва коя версия на HTML стандарт е използвана за написването на документа.

При предишните версии на HTML, т.е. всички преди HTML 5, тази декларацията се отнасяше до определена DTD (Document Type Definition). DTD определя правилата за езика за маркиране, така че браузърите да могат да визуализират правилно съдържанието.

Пълният стандартният синтаксис на `<!DOCTYPE>` изглежда по следния начин:

```
<!DOCTYPE [елемент от най-горно ниво] [публичност]
           "[регистрация"]//[организация"]//[тип] [име"]//[език]"
           "[URL]">
```

Атрибутите на декларацията `<!DOCTYPE>` са:

- **[елемент от най-горно ниво]**: Указва стандарта, на който отговаря документа.
- **[публичност]**: Указва дали елементът е публичен (PUBLIC) или е системен (SYSTEM).
- **[регистрация]**: Информация за това дали **DTD (Document Type Definition)** е регистрирана (означава се със знак „+“) или е нерегистрирана (означава се със знак „-“) в международната организация за стандартизация ISO. Ако стандарта е определен от W3C се използва „-“. DTD сега се използва основно в XML документите.
- **[организация]**: Посочва уникалното име на организацията, която е разработила DTD използвана в този документ. Официално стандарта HTML принадлежи на W3C.
- **[тип]**: За HTML документи, преди стандарта HTML5, трябваше да се укаже типа на DTD.
- **[име]**: Името на стил на документа.
- **[език]**: Езикът, на който са написани описанията на документите. За HTML се използва EN.
- **[URL]**: Посочва се URL адреса на DTD документ.

Пример за различните типове `<!DOCTYPE>`:

DOCTYPE	Описание
<b>HTML 4.01</b>	
<code>&lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd"&gt;</code>	Документът е написан на строг HTML.
<code>&lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"&gt;</code>	Документът е написан на преходен HTML.
<code>&lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd"&gt;</code>	В HTML документа се използват фреймове.
<b>HTML 5</b>	
<code>&lt;!DOCTYPE HTML&gt;</code>	За всички документи написани на HTML 5.

Новият стандарт на езика HTML, а именно HTML5, вече не се основава на SGML и затова не изисква използването на DTD. Ето защо винаги добавяйте декларацията `<!DOCTYPE HTML>`, за да знае уеб браузъра как да рендерира съдържанието на страницата.

След това следва същинските HTML тагове, които се заграждат в двойният таг `<html> ... </html>`. Този таг определя началото и край на HTML документа. Всичко, което е извън него, няма да бъде четено и визуализирано от уеб браузъра.

```
<!DOCTYPE html>

<html>

    ...

</html>
```

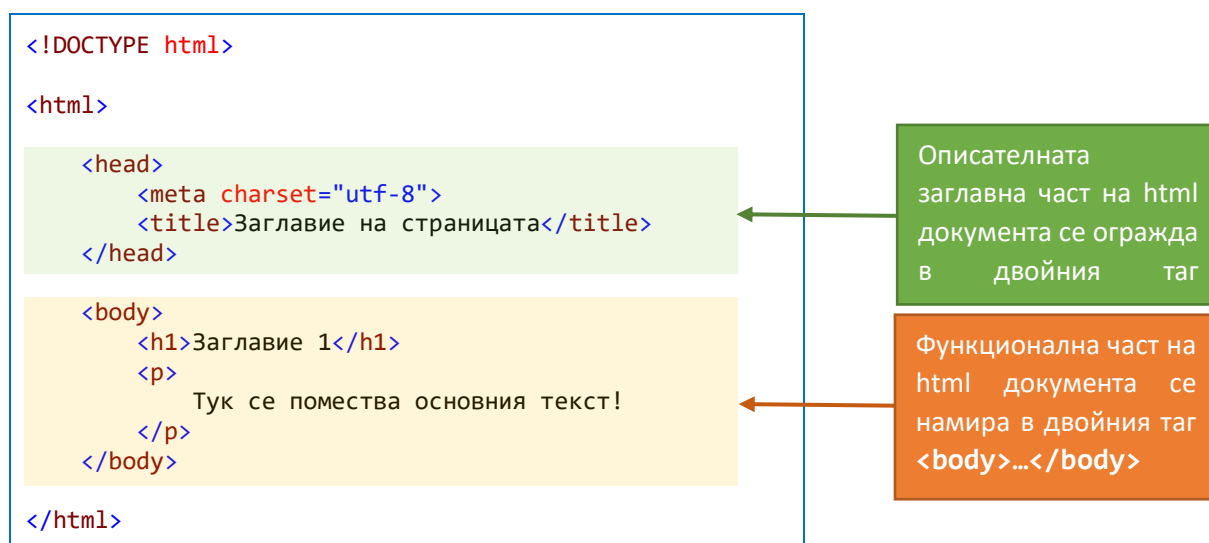
Според стандарта HTML тагът `<html>` няма атрибути. Но когато уеб документа е подчинен на XHTML стандарта, тогава към него се добавя атрибута `xmlns`, на който се присвоява URL адреса `"http://www.w3.org/1999/xhtml"`, сочещ към описанието на текущия стандарт на XHTML.

```
<html xmlns="http://www.w3.org/1999/xhtml">

    ...

</html>
```

Според HTML стандарта съдържанието между двойния таг `<html>` се разделя на две основни части: **описателна** (дескриптивна) и **функционална** (изпълнима). Тази структура е представена на фиг. 3.2.



Фигура 3.2. Основни части в HTML документа

#### Описателна част на HTML документа

**Системната информация**, необходима за идентифицирането на HTML документа се разполага между заглавния таг `<head>...</head>`. Този таг е контейнер, в който се разполагат таговете, съдържащи важна служебна информация за HTML документа, като `<title>` (този елемент е задължителен за HTML документа), `<meta>`, `<style>`, `<base>`, `<link>`, `<script>`, `<noscript>`.

Някои от тези тагове ще бъдат описани по подробно в настоящата тема. Повече информация може да бъде потърсена в посочените в края на темата литературни източници.

### Тагът <title>

В този таг се определя заглавието на HTML документа. Неговото съдържание се показва в заглавната лента на браузъра, като заглавие на страница, когато тя бъде добавена към любими (favorite) и като заглавие на страница, при визуализирането в списъка с резултати от търсещите машини. Елементът е задължителен за HTML/XHTML документите.

```
<head>
  <title>Заглавие на страницата</title>
</head>
```

### Тагът <meta>

В този таг се задават метаданните за страницата, например кодировката, ключови думи, последната дата на актуализиране и т.н. Тагът не е задължителен, но не бива да се изпуска. Често системите за търсене (Search Engine) използват тези елементи за формиране на общите индекси в техните бази от данни. Информацията в този таг указва и начина на възпроизвеждане на страницата от браузъра. В HTML5 към възможностите на <meta> тага беше добавена възможност за определяне на видимата потребителска област ("viewport").

```
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="refresh" content="n; [URL=url]">
<meta name="author" content="Автор на документа">
<meta name="keywords" content="HTML, CSS, Web Design">
```

Тагът <meta> може да съдържа следните атрибути:

Атрибут	Стойност	Описание
charset	<i>character_set</i>	Определя кодовата таблица свързана с HTML документа.
content	<i>text</i>	Определя съдържанието асоциирано с <b>http-equiv</b> или с <b>name</b> .
http-equiv	content-type default-style refresh	Определя заглавната част за протокола http или определя кога да се презареди страницата.
name	application-name author description generator keywords viewport	Определя вида на използваните метаданни

### Примери за използване на тага <meta>

За да се укаже автора на съответната страница, тогава на атрибута name се присвоява стойността "author" и в атрибута content се изписва името на автора:

```
<meta name="author" content="Име Фамилия / Name Surname">
```

За да се даде кратко описание на страницата, тогава на атрибута name се присвоява стойността "description" и в атрибута content се изписва описанието:

```
<meta name="description" content="Тук се съдържа кратко описание на страницата">
```

За да се презарежда текущата страница през 5 секунди, трябва да се използва следния код:

```
<meta http-equiv="refresh" content="5">
```

За да се зареди нова страница (<http://www.swu.bg>) 5 секунди, след като сме заредили текущата е необходимо да се използва следния код:

```
<meta http-equiv="refresh" content="5; http://www.swu.bg/">
```

### Тагът <style>

Този таг дефинира информация за **стиловете** използвани в HTML документа. Вътре в двойния таг <style> е необходимо да се укаже как браузърът трябва да рендерира съдържанието на документа.

```
<style type="text/css">
  h1 {
    font-family: Arial;
    font-size: 20pt;
    color: white;
    font-style: normal;
    font-weight: bolder;
    background: blue;
    padding: 5px;
    text-align: center;
    width: 500px;
  }
  body {
    background: aqua;
    width: 500px;
    margin: 0 auto;
  }
</style>
```

### Тагът <base>

Този таг определя базовия абсолютен URL адрес, спрямо който в HTML документа ще се формира относителния.

```
<head>
  <base href="http://www.swu.bg/images" target="_blank">
</head>
```

### Тагът <Link>

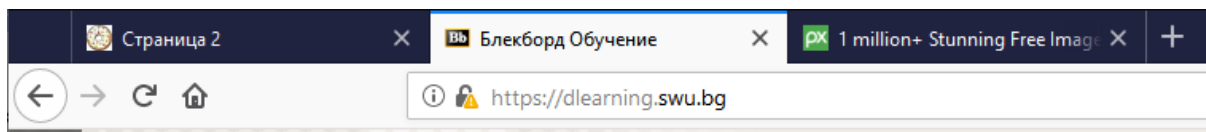
Този таг дефинира връзката между документа и необходимите външни ресурси. Най-често се използва за връзка към външен стилев файл.

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

Тагът <link> може да съдържа следните атрибути:

Атрибут	Стойност	Описание
crossorigin	anonymous use-credentials	Указват начина на обработка на заявката при кръстосано използване
href	URL	Определя локацията на свързания елемент
hreflang	Language_code	Определя езика на текста в свързания документ
rel	alternate author dns-prefetch help icon license next pingback preconnect prefetch preload prerender prev search stylesheet	<b>Задължителен.</b> Определя връзката на настоящия документ с този, към който сочи линка.
media	media_query	Определя вида на устройството, на което ще се покаже съдържанието
size	HeightxWidth any	Определя размера на иконката по височина и ширина и се използва само ако rel="icon".
type	media_type	Определя файловия тип на свързания документ.

Но освен това с помощта на този таг може да се добави иконка на уеб страницата. Иконката е малко графично изображение, което се показва пред името на всяка страница. Стандартният размер е 16x16 pixels и обикновено е с разширение ico. Последните версии на уеб браузърите вече могат да поддържат различни големина (като 16x16, 32x32, 48x48 и т.н. до 512x512) и различни файлови формати (ico, png, svg) на изображенията.



Повече информация за използването на иконки в уеб, можете да намерите на този линк <https://mobiforge.com/design-development/adding-favicons-in-a-multi-browser-multi-platform-world> както и на този линк <https://html.spec.whatwg.org/#rel-icon>.

И така в случаите, при които трябва да използвате тага link за добавяне на иконка, е необходимо да използвате и неговите атрибути type, rel, size и href.

Пример за използване на иконка в уеб страница, можете да видите на следващия линтинг:



```
<head>
  <link type="image/x-icon" rel="icon"
        href="icon/favicon-16x16.png" sizes="16x16" />
  <link type="image/x-icon" rel="icon"
        href="icon/favicon-32x32.png" sizes="32x32" />
  <link type="image/x-icon" rel="icon" href="icon/favicon.ico">
  <link type="image/svg+xml" rel="icon" sizes="any" href="icon/favicon.svg">
</head>
```

### Тагът <script>

Този таг служи за **дефиниране на скрипт**, който ще се изпълни на клиентската машина, написан най-често на езици като JavaScript или VBScript.

### Тагът <noscript>

Това е двоен таг, който служи за **дефиниране на алтернативно съдържание**, което ще бъде използвано при условие, че потребителският браузър не поддържа скриптове или използването на скриптове е деактивирано. Тогава съдържанието, което е поместено в този таг ще се визуализира. Интересното при него е, че може да бъде използван както в пространството на тага <head>, така и в тага <body>. Когато <noscript> се използва в тага <head> в него могат да се вграждат само елементите <link>, <style> и <meta>. Когато се използва в тага <body>, тогава той може да съдържа и текст.

### Функционална част на HTML документа

Функционалната (описателната) част на HTML, т.е. информацията, която ще бъде визуализирана в уеб браузъра, се разполага между отварящия таг <body> и затварящия таг </body>. За разлика от предишната версия HTML 4.01, при която тагът <body> имаше няколко атрибута, като background, bgcolor, link, alink, vlink, text, то в новата версия HTML5 те са премахнати и вече не се използват.

```
<body>
  <h1>Заглавие 1</h1>
  <p>
    Тук се помества основния текст!
  </p>
  
</body>
```

Тагът <body> е контейнер, в който могат да се поставят различни тагове като тагът за параграф (<p>...</p>), таговете за заглавия (<h1> ... </h1>, ..., <h6> ... </h6>), тагът за добавяне на изображение (<img>) и т.н. Всички те ще бъдат подробно разгледани в някои от следващите теми.

Във функционалната част на HTML документа, според новия стандарт на езика HTML5, са добавени 18 нови тага, които ще бъдат разгледани на по-късен етап, а тук само ще споменем, някои от тях.

article	dialog	meter
aside	header	nav
audio	figure	progress
command	footer	section
datalist	m (mark)	time

details

menu

video

Всички тези тагове се помествам в тага `<body>`.

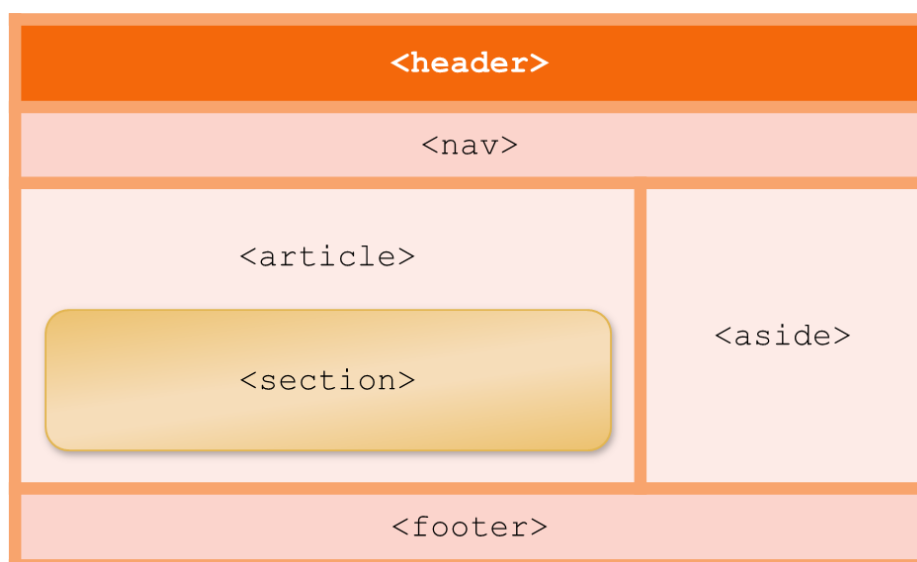
Премахнати са някои от остарелите тагове, като `font`, `center`, в това число и фреймовата структура на HTML документа

В HTML5 са въведени тагове за изграждане на логическата структура на HTML документа. Характерното за тези елементи е че, ако те са асоциирани с някаква стилови правила написани с CSS, ще бъдат визуализирани на клиентския браузър като отделни абзаци.

За обособяване на логически структура на документ написан на HTML5 се използват следните тагове:

- **`<header>`**: Този таг обособява логическата заглавна част в тялото на документа. Допустимо е използването на повече от един таг `<header>` в един HTML документ.
- **`<nav>`**: Този таг обособява секция за навигация. Най-често с негова помощ се организират менютата в новите HTML документи.
- **`<article>`**: Този таг указва независима секция от съдържанието – статии, новини, коментари и др.
- **`<aside>`**: Този таг се използва за показване на страничния панел, например блок с връзки.
- **`<section>`**: Тагът `<section>` дефинира секция в документа.
- **`<footer>`**: Тагът `<footer>` дефинира долния колонтитул на HTML документа. Тук обикновено се помества информация за автора на документа, данни за контакти, условия на ползване и т.н.

Пример за възможно използване на тези тагове при изграждане на една структура на документ, може да се види на фиг. 3.3



**Фигура 3.3.** Примерна организация на HTML документа

```
<body>
  <header>...</header>
```

```

<nav>...</nav>
<article>
  <section>
    ...
  </section>
</article>
<aside>...</aside>
<footer>...</footer>
</body>

```

## 5. XHTML

**Extensible Hypertext Markup Language (XHTML)** е **XML (Extensible Markup Language)** базиран език, който може да се използва като алтернатива на HTML.

Документите написани на XHTML трябва да нямат никакви грешки, в противен случай няма да бъде разпознаваемо от уеб клиентите.

Документите, в които има код написан на XHTML и XML, се третира от браузърите като XML документ и се анализира чрез XML процесор. Но при тяхното рендване се използва XML парсер, вместо SGML.

Като цяло синтаксиса на XHTML до голяма степен прилича на HTML5, но не трябва да се забравя, че до XHTML е базиран на XML. Или с други думи HTML и XHTML имат идентична лексика, но различни правила при анализ. За XHTML е характерно строгият синтаксис, което означава че дори при една дребна синтактична грешка ще се визуализира жълт екран („Yellow Screen of Death“) с информация за грешката. Докато при HTML дори и да има някои грешки по таговете или техните атрибути, уеб браузърите ще продължат да показват съдържанието на такъв вид уеб страница. В XHTML има тагове, които не се използват в HTML.

Документите написани на XHTML се съхраняват с разширение **.xhtml**. При този тип документи задължително трябва да бъде включен един от двата MIME типа `application/xhtml+xml` или `application/xml`. Някои от браузърите не разпознават този формат и затова програмистите предпочитат да съхранят XHTML документите като HTML, което не е правилно.

MIME типа е механизъм, който казва на клиентската машина какво е разширението на съответния документ. Това позволява на съответния клиент да намери подходящото приложение, с което да отвори този документ.

В XHTML5 не е задължително използването на декларацията **DOCTYPE**.

В XHTML всички тагове са двойни. Ето защо дори и единичните трябва да бъдат затваряни.

```
<единичен-таг />
```

Пример за документ, отговарящ на стандарта XHTML е:

```

<?xml version="1.0" encoding="utf-8" ?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <!--
    За определяне на MIME типа на документа и кодовата таблица, може да се
    използва и следното

```

```

    <meta charset="UTF-8" />
    <meta http-equiv="Content-Type" content="application/xhtml+xml" />
  -->
  <meta http-equiv="Content-Type" content="application/xhtml+xml; charset=utf-8"
/>
  <title>Уеб страница на XHTML5</title>
</head>
<body>
  ...
</body>
</html>

```

## Задачи за лабораторни занятия към тема 3

### Задача 1

Да се създадат три документа отговарящи на стандартите за HTML 5, XHTML5 и HTML4.01. Целта на задачата е да се представи основната структура на тези три стандарта, чрез използване на езика HTML.

#### Решение:

За решаване на тази задача е необходимо използването на Notepad++.

*Уеб страница написана на XHTML5:* Създайте файл с име **pageXHTML.xhtml**

```

<?xml version="1.0" encoding="utf-8" ?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="application/xhtml+xml; charset=utf-8" />
  <title>Уеб страница на XHTML5</title>
</head>
<body>
  <header>
    <h1>Заглавие</h1>
  </header>
  <section>
    <article>
      <h2>Подзаглавие 1</h2>
      Текст към първото подзаглавие
    </article>
    <article>
      <h2>Подзаглавие 2</h2>
      Текст към второто подзаглавие.
    </article>
  </section>
  <footer>
    &#169; 2020 Име Фамилия. Всички права запазени.
  </footer>
</body>
</html>

```

Уеб страница написана на HTML5: Създайте файл с име **pageHTML.html**

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Уеб страница на HTML5</title>
</head>
<body>
  <header>
    <h1>Заглавие</h1>
  </header>
  <section>
    <article>
      <h2>Подзаглавие 1</h2>
      Текст към първото подзаглавие
    </article>
    <article>
      <h2>Подзаглавие 2</h2>
      Текст към второто подзаглавие.
    </article>
  </section>
  <footer>
    &copy; 2019 Име Фамилия
  </footer>
</body>
</html>
```

Уеб страница написана на HTML4.01: Създайте файл с име **pageOldHTML.html**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  <title>Уеб страница на HTML 4.01</title>
</head>
<body>
  <div class="ttl">
    <h1>Заглавие</h1>
  </div>
  <div class="ss">
    <div class="art">
      <h2>Подзаглавие 1</h2>
      Текст към първото подзаглавие
    </div>
    <div class="art">
      <h2>Подзаглавие 2</h2>
      Текст към второто подзаглавие.
    </div>
  </div>
  <div class="ftr">
    &copy; 2019 Име Фамилия
  </div>
</body>
</html>
```

## Задача 2

Да се създадат три документа с имена **page1.html**, **page2.html**, **page3.html**, които да са взаимно свързани, но не с хиперлинкове, а посредством тага **meta**. Връзката да бъде такава, че при зареждане на страница **page1.html** след **10 sec** да се зареди страница **page2.html**. След като изминат **7 sec**, да се зареди страница **page3.html**, а след като тя престои на екрана **4 sec**, да се зареди отново **page1.html**.

Задайте на всяка от страниците иконка, като използвате изображения с различни файлови формати и различни големина. За целта използвайте файловете от архивния файл **icon.zip**, качен на страницата на темата.

Задайте на всяка страница име, автор, описание и ключови думи.

Определете използваната кодова таблица на страницата да е UNICODE, а използваният език на текст в страницата да бъде български.

За успешната реализация на тази задача, моля в тага **<body>** на съответните страници да използвате, кода представен в блоковете по-долу.

### page1.html

```
<body>
  <h1 style="font: bold 60pt arial; color: green; text-align: center;">
    Здравей, HTML свят!
  </h1>
  <p style="font-size:150px; text-align: center;">&#10024;</p>
</body>
```

### Page2.html

```
<body>
  <h1 style="font: bold 60pt arial; color: blue; text-align: center;">
    Здравей, HTML свят!
  </h1>
  <p style="font-size:150px; text-align: center;">&#129299;</p>
</body>
```

### Page3.html

```
<body>
  <h1 style="font: bold 60pt arial; color: red; text-align: center;">
    Здравей, HTML свят!
  </h1>
  <p style="font-size:150px; text-align: center;">&#9925;</p>
</body>
```

## Решение:

За решаване на тази задача е необходимо използването, на който и да било текстов редактор. Реализацията на страница **page1.html** е представен на следващото изображение. За вас остава да реализирате сами останалите две страници.

```

page1.html → ×
1  <!DOCTYPE html>
2
3  <html lang="bg">
4  <head>
5      <meta charset="utf-8" />
6      <meta http-equiv="content-language" content="bg" />
7
8
9      <meta http-equiv="refresh" content="10; page2.html" />
10     <meta name="author" content="Radoslava Krалева" />
11     <meta name="description" content="Втора задача към тема 3" />
12     <meta name="keywords" content="тара мета, заглавен таг, уеб системи и технологии" />
13
14     <!-- Icons & Platform Specific Settings
15     <!-- favicon-16x16.png - The classic favicon, displayed in the tabs. -->
16     <link type="image/x-icon" rel="icon" href="icon/favicon-16x16.png" sizes="16x16" />
17     <!-- favicon-32x32.png - For Safari on Mac OS. -->
18     <link type="image/x-icon" rel="icon" href="icon/favicon-32x32.png" sizes="32x32" />
19     <!-- favicon-48x48.png - For Google TV and Google Chrome -->
20     <link type="image/x-icon" rel="icon" href="icon/favicon-48x48.png" sizes="96x96" />
21
22     <link type="image/x-icon" rel="icon" href="icon/favicon-128x128.png" sizes="128x128" />
23     <link type="image/x-icon" rel="icon" href="icon/favicon-196x196.png" sizes="196x196" />
24
25     <!-- SVG може да се използва като формат за иконка само във Firefox и Chrome -->
26     <link rel="icon" type="image/x-icon" href="icon/favicon.ico">
27
28     <!-- SVG може да се използва като формат за иконка само във Firefox -->
29     <link rel="icon" type="image/svg+xml" sizes="any" href="icon/favicon.svg">
30
31     <title>Страница 1</title>
32 </head>
33 <body>
34     <h1 style="font: bold 60pt arial; color: green; text-align: center;">Здравей, HTML свят!</h1>
35     <p style="font-size:150px; text-align: center;">&#10024;</p>
36 </body>
37 </html>
38

```

## Литература

- 1 M. J. Taylor (2009) Four methods of adding CSS to HTML - <http://matthewjamestaylor.com/blog/adding-css-to-html-with-link-embed-inline-and-import>
- 2 I. Pouncey, R. York (2011) *Beginning CSS. Cascading Style Sheets for Web Design*, Wiley Publishing
- 3 J. N. Robbins (2012) *Learning Web Design, Fourth Edition, A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics*, O'Reilly Media
- 4 L. F. Sikos (2013) *Web Standards. Mastering HTML5, CSS3, and XML*, Apress
- 5 B. P. Hogan (2009) *Web Design for Developers. A Programmer's Guide to Design Tools and Techniques*, The Pragmatic Bookshelf
- 6 M. Bowers, D. Synodinos, Victor Sumner (2011) *Pro HTML5 and CSS3 Design Patterns*, Apress
- 7 Д. Колисниченко (2012) HTML5 & CSS3, изд. Асеновци
- 8 Ив. Маджаров (2001) Мултимедийни документи в Internet.
- 9 W3Schools.com - HTML 5 – [http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)
- 10 W3C, XHTML™ 2.0 W3C Working Group Note 16 December 2010; <https://www.w3.org/TR/xhtml2/>

- 11 Mozilla: MDN web doc, Incomplete list of MIME types, 2017:  
[https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics\\_of\\_HTTP/MIME\\_types/Complete\\_list\\_of\\_MIME\\_types](https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Complete_list_of_MIME_types)