

## Tema: Elementele HTML utilizate în antetul documentului web

După această lecție:



- vei ști ce elemente HTML pot utilizate în antetul documentului web și care este rolul lor;
- vei înțelege care este importanța elementului META pentru un document web;
- vei afla prin ce diferă o "adresă absolută" de o "adresă relativă".

Am reușit deja să vedem care este structura de bază, recomandată, pentru un document web și, sper, să fi reținut că un document web este alcătuit din două părți: *antet* și *conținutul de bază* (sau corp).

La această lecție vom analiza care sunt elementele web, care pot fi utilizate în antetul documentului web și pentru ce ar putea servi ele. Să începem...

### Elementul HEAD

Elementul "**head**" este un container pentru alte elemente HTML și este plasat între eticheta `<html>` și eticheta `<body>`. Acest element este recunoscut de toate browserele web:



```
<!DOCTYPE html>
<html>
  <head>
    ...
  </head>
  <body>
    ...
```

Până la versiunea HTML5 era obligatorie folosirea elementului *HEAD*, dar în HTML5 acesta poate și să nu fie specificat. Adică, elementele din antet vor fi specificate până la elementul BODY, fără a fi încadrate în interiorul elementului HEAD. Trebuie să știi, că persoanele care dezvoltă site-uri web îl utilizează în continuare din obișnuință. Alteori, însă, îl folosesc și pentru a realiza cât mai corect delimitarea dintre antetul și conținutul documentului web. Tu deja decizi – să îl utilizezi sau nu...

Dacă vei folosi elementul HEAD, trebuie să respecți sintaxa acestuia – elementul are *tag* de deschidere și *tag* de închidere, pentru a specifica începutul și sfârșitul antetului

documentului. Conținutul elementelor, specificate în *HEAD* nu va fi afișat în fereastra browserului.

Elementele specificate în interiorul elementului *HEAD* pot include: titlul documentului, setul de caractere utilizat, stilurile, script-urile sau instrucțiunile pentru browser - în cazul în care trebuie găsite foile de stil, să furnizeze informații de tip „metadate” (metadatele sunt ”date despre date”) etc.

Elementele, care pot fi inserate în secțiunea *HEAD* sunt: **<title>**, **<style>**, **<meta>**, **<link>**, **<script>** și **<base>**.

Să analizăm aceste elementele...

### Elementul TITLE

Elementul *TITLE* definește titlul documentului web și trebuie specificat pentru orice document web. Elementul are conținut și posedă atât *tag* de deschidere, cât și *tag* de închidere.

Sintaxa: **<title>Conținut</title>**

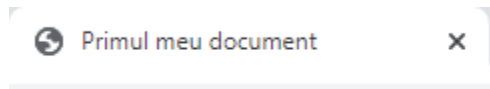
Titlul documentului va apărea în *tab*-ul paginii web, din browser.

Să creăm documentul **index.html** și să înscriem în el următoarele linii:



```
<!DOCTYPE html>
<html>
  <head>
    <title>Primul meu document</title>
  </head>
  <body>
    ...
  </body>
</html>
```

Salvează-l, iar apoi deschide-l în browser. Vei observa în browser, că în fereastră apar doar cele trei puncte, specificate în *BODY*, dar mai sus, în *tab*, apare textul pe care l-am înscris drept conținut, în elementul *TITLE*.



### Elementul LINK

Elementul LINK este utilizat pentru a face referire din documentul curent la unele resurse externe, care vor fi utilizate în cadrul documentului, ca de exemplu: se poate face referire la imaginea-icoană de lângă titlul documentului, la stilurile externe etc. Acest element, LINK, este susținut de toate browserele web și nu are conținut, dar are câteva atribute. Sintaxa elementului:

```
<link rel="tip_relație" type="tip_resursă" href="nume_fisier.extensie" />
```

unde **rel**, **type**, **href** – sunt cele mai uzuale atribute:

- **href** – specifică locația fișierului legat cu documentul curent;
- **rel** – este atributul obligatoriu și specifică relația cu resursa legată. Poate avea mai multe valori: *icon*, *stylesheet* etc.;
- **type** – specifică tipul documentului legat.

Despre inserarea stilurilor externe vom discuta la o lecție următoare, dar acum aș vrea să inserezi iconița, care stă alături de titlul documentului, în *tab* și care este caracteristică oricărui site web. Cred că ai observat...



Pentru adăugarea iconiței, am accesat un site, ce oferă fotografii și imagini gratuite, de pe care am descărcat o poză, care va servi drept emblemă. Imaginea, ar fi bine să aibă extensia **.ico** sau **.png**. Am salvat-o în mapa "*images*".



Să realizăm acum legătura cu ea, cu ajutorul elementului LINK. Pentru aceasta voi adăuga încă o linie în documentul web:



```
<head>
  <title>Primul meu document</title>
  <link rel="icon" href="images/favicon.png" type="image/x-icon" />
</head>
```

Deja, când vei accesa pagina web, observă că s-a realizat modificarea, adăugându-se și imaginea lângă titlul documentului...



Pagina a devenit mai personalizată...

## Elementul META

Să continuăm cu elementul META, care are foarte multe atribute, dintre care vom analiza câteva. În primul rând elementul META nu are conținut. Sintaxa acestui element va fi:

```
<meta atribut="valoare" />
```

Elementul META este utilizat, de obicei, pentru a furniza *metadate* despre un anumit document HTML. Este suportat de toate browserele web. Metadatele nu vor fi afișate pe pagina. Elementul **META** specifică: setul de caractere utilizat în document, cuvintele cheie, descrierea paginii, autorul documentului etc. Toate aceste date sunt utilizate de browser, motoarele de căutare sau alte servicii web.

Suplimentar, începând cu HTML5 a fost introdusă, prin elementul META, o metodă ce permite proiectanților web să preia controlul asupra viewport-ului dispozitivului care prezintă conținutul web. **Viewport**-ul este zona vizibilă utilizatorului, la accesarea unei pagini web.

Vom adăuga în documentul nostru web unele metadate:



```
<head>
  <title>Primul meu document</title>
  <link rel="icon" href="images/favicon.png" type="image/x-icon" />
  <meta charset="UTF-8" />
  <meta name="description" content="Prima pagina HTML" />
  <meta name="keywords" content="Cafea, produse din cafea" />
  <meta name="author" content="Kity Coffee" />
  <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
</head>
```

Deci, să le analizăm succesiv...

`<meta charset="UTF-8" />` - atributul **charset**, al elementului META, este utilizat pentru a specifica codificarea caracterelor, folosite într-o pagină web. Cea mai uzuală azi codificare este **utf-8**. Se recomandă să fie specificată în toate proiectele web, pentru ca browser-ul să prezinte corect conținutul paginii. Atunci când browser-ul "nu va ghici" corect codificarea, iar tu vei utiliza în text litere românești precum: ă, î, â, ț, ș sau litere rusești, pe ecran vor fi reflectate, uneori, ieroglife, neclare ochiului uman.

În continuare observă că am utilizat în câteva rânduri, pentru elementul META, același atribut "**name**" și "**content**" cu câteva valori diferite. Valorile pentru atributul **name** sunt standarde, predefinite, iar valorile pentru atributul **content** – le specificăm noi, în funcție de context:



```
<meta name="description" content="Prima pagina HTML" />
<meta name="keywords" content="Cafea, produse din cafea" />
<meta name="author" content="Kity Coffee" />
```

După cum am menționat deja, în HTML5 a apărut posibilitatea definirii viewport-ului. "Fereastra de vizibilitate" sau **VIEWPORT**-ul, utilizatorului, se schimbă în funcție de dispozitivul utilizat și va fi mult mai mic pe un telefon, decât pe un ecran de calculator. Viewport-ul anunță browserul cum trebuie acesta să controleze dimensiunile și scara paginii web. Încercând să îmbunătățească lucrul cu site-urile web, browserele dispozitivelor mobile prezintă pagina cu aceeași lățime ca și pe un ecran de calculator. De obicei, această lățime este de aproximativ 980 pixeli, dar totuși, la diferite dispozitive poate să difere puțin. După aceasta browserul mărește dimensiunea fontului și modifică dimensiunea conținutului, astfel încât să poată amplasa pagina pe ecran. În rezultat utilizatorii văd diferite dimensiuni ale fontului, iar atunci când le trebuie să vadă ce este pe site, aceștia trebuie să mărească scara site-ului.

Uneori se obține ceva de genul:



...ceea ce nu e chiar frumos ☹️

Pentru a controla dimensiunea spațiului de vizibilitate în browser și pentru ca pagina să se adapteze după lățimea ecranului dispozitivului de prezentare, se recomandă utilizarea

tag-ului **meta**, cu valoarea **viewport** pentru atributul **"name"**. Cu ajutorul lui, dimensiunea și amplasarea conținutului se va schimba, iar site-ul va arăta bine pe orice dispozitiv. Adăugați în acest element, în atributul **"content"**, drept valoare **width=device-width**, pentru a adapta lățimea ferestrei de vizibilitate la ecranul dispozitivului. De asemenea, inserează și valoarea **initial-scale=1**, pentru a asigura scara de 1:1 între pixelii CSS și pixelii independenți ai dispozitivului de prezentare.

Astfel, este necesară adăugarea liniei:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

## Elementul STYLE

Elementul STYLE este utilizat pentru a defini informațiile despre stilurile elementelor din documentul web curent. Sintaxa recomandată este:

```
<style>stiluri_CSS</style>
```

De exemplu, dacă voi adăuga în documentul web textul:



```
<body>
    Aici voi descrie produsele din cafea
</body>
```

În fereastra browserului acesta va apărea așa:

---

Aici voi descrie produsele din cafea

...Iar dacă în elementul STYLE din HEAD voi adăuga:



```
<style>
    body {
        background-color: rgb(106, 117, 86);
        color: orange;
        font-size: 2em;
        text-align: center;
    }
</style>
```

...deja acest text "își schimbă înfățișarea", adică, "își schimbă stilul" și textul va arăta astfel:

## Aici voi descrie produsele din cafea

Mai multe detalii referitoare la stiluri – vei afla la lecțiile următoare.

### Elementul SCRIPT

Elementul SCRIPT poate fi specificat pentru a adăuga JavaScript-uri, ce vor fi interpretate de browser și implementate în documentul curent. Sintaxa recomandată este:

```
<script> JavaScript-uri </script>
```

Să adăugăm și un mic scriptuleț în documentul nostru web:



```
<script>
    function daData(){
        var data_all = new Date();
        var rezultat = data_all.getDate()+"/"+data_all.getMonth() +
"/" +data_all.getFullYear();
        document.getElementById("rez").innerHTML = rezultat;
    }
</script>
```

În acest script am cerut să fie afișată data curentă, atunci când voi face click pe un buton din pagina web. Pentru aceasta voi adăuga elementele necesare în conținutul documentului web.



```
<body>
    <button onclick="daData();">Afiseaza data</button>
    <p id="rez"></p>
    Aici voi descrie produsele din cafea
</body>
```

Așa arată pagina până la activarea JavaScript-ului:

Afiseaza data

Aici voi descrie produsele din cafea

Și așa va arăta pagina web, după ce va fi "apelat" JavaScriptul, definit pentru afișarea datei curente:



Mai multe detalii despre JavaScripturi, vei afla la lecțiile următoare, în care vom descrie mai detaliat limbajul de programare JavaScript.

## Elementul BASE

Elementul BASE - specifică URL-ul sau calea de bază pentru alte URL-uri relative din pagina curentă.

Sintaxa recomandată este: `<base href="calea" />`

Pentru fișierul nostru eu am specificat:

```
<base href="D:\exemple\" />
```

Dar să vedem, în acest context, unele informații utile despre căi. Calea specifică locația sau amplasarea unui fișier. Fișierul se poate afla pe un server web sau poate fi plasat pe un disc local. În exemplul de mai sus am specificat că fișierul nostru **index.html** se află în mapa **exemple**, iar mapa este plasată pe discul **D** a calculatorului meu. Observă că între denumiri a fost utilizat drept separator simbolul "back-slash". În sistemul de operare Windows poate fi utilizat drept separator și "slash"-ul.

Pentru site-urile web, care sunt păstrate pe servere web, drept cale de bază poate servi numele de domeniu al site-ului.

Căile spre resursele web pot fi: *relative* sau *absolute*.

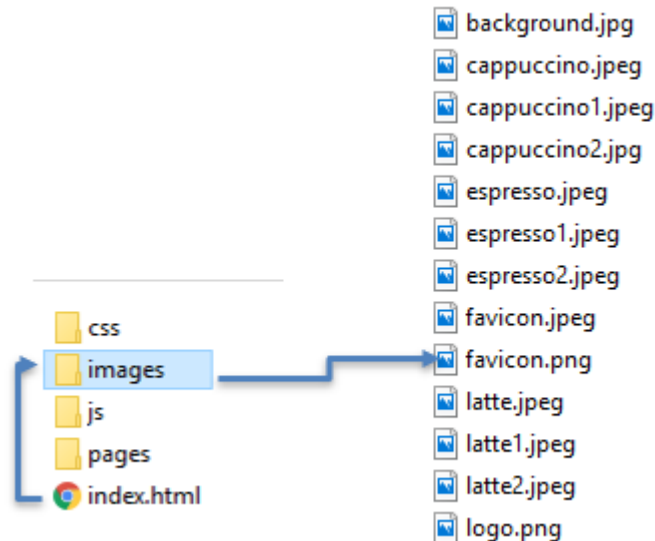
Atunci când nu este specificată complet locația unui fișier, browserul îl va căuta în aceeași mapă cu fișierul *.html* curent. În acest caz se spune că **folosim o adresă relativă**. Adică se va face căutarea *în raport* sau *relativ* cu fișierul curent. Iar atunci când fișierul este stocat



într-un subdirector (sau o mapă inclusă în mapa curentă), trebuie specificată calea relativă până la acel fișier. De exemplu, aici:

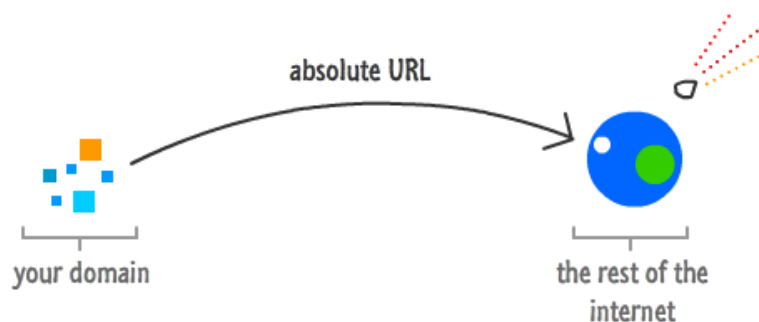
```
<link rel="icon" href="images/favicon.png" type="image/x-icon" />
```

a fost specificat că fișierul **favicon.png** trebuie preluat din subdirectorul **images**, care se află în mapa curentă.



Alteori, însă, este necesar accesul la un fișier sau o imagine stocată pe un server web sau pe un alt disc. Ca de exemplu:

<https://www.pexels.com/photo/selective-focus-photo-of-white-ceramic-mug-on-saucer-filled-with-espresso-1170659/>. În acest caz vom specifica calea completă spre fișier și vom spune că **folosim o adresă absolută**.



Să revedem ideile de bază ale acestei lecții:



- Orice document web este format din antet și conținut. Antetul este definit de elementul HEAD;
- TITLE – este un element, utilizat în antet pentru a defini titlul documentului web;

- LINK – este un element, utilizat în antet pentru a realiza legături între documentul web curent cu resurse externe documentului;
- Elementul META, utilizat tot în antet, furnizează *metadata* browserului, motoarelor de căutare etc., despre documentul web curent;
- Elementul STYLE – este utilizat în antet pentru a specifica stilurile interne ale documentului web;
- Elementul SCRIPT – este utilizat în antetul documentului, pentru a defini scripturi interne, sau pentru a realiza legătura cu un script extern;
- BASE este un element HTML, utilizat în antet pentru a specifica calea de bază pentru căile *relative*.

Punem punct acestei lecții. Sper că ai înțeles care sunt elementele HTML, ce ar putea fi specificate în antetul documentului web. La lecția următoare începem să studiem elementele HTML ce pot fi specificate în BODY și interpretarea cărora se va reflecta în pagina web, încărcată în fereastra browserului.