

HTML valoda

Pēc Kaspara Antoneviča materiāliem

SATURS

"Mana pirmā web lapa" - dažādi padomi	2
Ievads HTML	2
Sintakse	2
HTML pamatelementi	3
Virsraksti (headings)	5
Rindkopas (paragraph)	6
Saraksti (lists)	6
Saites (hyperlinks)	9
Attēli (image)	11
Tabulas (tables)	12
Meta dati	16
Dokumenta veids (Doctype)	18

"Mana pirmā web lapa" - daži padomi

Lai veidotu savas web (tīkla) lapas, būs nepieciešamas vismaz trīs lietas:

- izstrādes rīki;
- web pārlūkprogrammas;
- web serveris, uz kura gatavo lapu izvietot.

Vēl daži ieteikumi:

- web lapa parasti sastāv no vairākiem failiem (datnēm), kas savā starpā savienoti ar saitēm (hyperlinks);
- galveno jeb pirmo failu, kuram jāatveras, parasti nosauc "index", piemēram, "index.html" (bez pēdiņām);
- pirms sākat veidot web lapu, izveidojiet mapi (folder), kurā saglabājat visus jaunus lapas failus, tajā skaitā "index" failu, attēlus un dokumentus, t.i., viss vienuviet! Protams, šajā mapē varat veidot apakšmapes, taču "index" failu dziļāk neliekat. Kad lapa gatava, tad šīs galvenās mapes visu saturu (bet ne pašu galveno mapi) iekopējiet uz izvēlētajā web servera, taču nemainiet iekopējamo failu savstarpējo izkārtojumu, jo tad saites vairs nedarbosies.

Ievads HTML

HTML (Hyper Text Markup Language)

Sintakse

HTML valodā galvenie elementi ir tagi. Tagu raksturo šādi norobežojošie elementi: $\langle \rangle$. Piemēram, lai pateiktu, ka attēlojamais teksts jāuztver kā rindkopa (paragraph), lieto rindkopas tagu: $\langle p \rangle$, t.i., starp norobežojošajiem elementiem iesprauž attēlojamā elementa (virsraksta - "h1", rindkopas - "p", tabulas - "table" u.c.) apzīmējumus, piemēram:

```
<p>Es pašlaik mācos HTML!</p>
```

Interneta pārlūkprogrammā šāds kods tiktu attēlots kā sekojoša rindkopa:

Es pašlaik mācos HTML!

Vēl ievērosim, ka mums arī bija jāpasaka, kur rindkopa beidzas. To izdara ar noslēdzošo tagu ($\langle /p \rangle$), kurš no atverošā, jeb rindkopu iesākošā ($\langle p \rangle$) atšķiras tikai ar slīpsvītru pirms attēlojamā elementa apzīmējuma! Šādi noslēdzošie tagi jālieto gandrīz visiem HTML tagiem. Un vēl - katram atverošajam tagam pienākas savš noslēdzošais tags! Piemēram, nedrīkst noslēgt vairākas rindkopas ar vienu noslēdzošo tagu.

Lai elementus noformētu, tiem var piešķirt vienu vai vairākas īpašības jeb atribūtus (attributes). Atribūtus raksta konkrētajā atverošajā tagā uzreiz aiz elementa un starp tiem nekādas atdalošas pieturzīmes nelieto, piemēram, $\langle table \ align="center" \rangle$. Šajā gadījumā mēs izveidojam tabulu ("table") un tai piešķīrām atribūtu jeb īpašību - nedefinējam tās novietojumu ("align"), kam piešķīrām vērtību "nocentrēta" ("center"). Ievērosim, ka definējot atribūta vērtību, lietojama "=" zīme un vērtība liekama pēdiņās!

Tātad, runājot par HTML sintaksi nākas saskarties ar trim lietām:

- tags (tag) - <>;
- elements (element) - konkrēts taga veids, piemēram, <table>;
- atribūti jeb īpašības (attribute) - noformē, jeb piešķir kādu īpašību attēlojamajam elementam, piemēram, tabula tiek nocentrēta <table align="center">.

Lūk visvienkāršākās web lapas piemērs. Iekopējiet šo kodu "Notepad-ā", saglabājiet, piemēram, ar nosaukumu "index.html" (bez pēdiņām) un tad atveriet failu ar interneta pārlūkprogrammu (konkrēts lietoto tagu apskats meklējams jau nākamajās stundās):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
</head>

<body>
<p align="center">Mana web lapa!</p>
</body>

</html>
```

Interneta pārlūkprogrammā vajadzētu tikt attēlotam šādam rezultātam:

Mana web lapa!

HTML pamatelementi

Vēlreiz aplūkosim iepriekšējo piemēru:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
</head>
<body>
</body>
</html>
```

No piemēra varam secināt, ka ir trīs pamatelementi:

1. Izmantotās HTML versijas definējums (vairāk par to sadaļā "Doctype"), kam seko <html> atverošais tags - šis tags iesāk un arī noslēdz web lapu. Tā uzdevums ir pateikt interneta pārlūkprogrammai, ka starp šiem tagiem tiek lietots HTML kods;
2. <head> - dokumenta "galvene" :). Šajā sadaļā parasti iekļauj vismaz divas lietas:
 - <title> jeb virsraksts. Teksts, kas atrodas starp "title" tagiem tiks attēlots interneta pārlūkprogrammas virsraksta joslā. Pie tam, saskaņā ar W3C rekomendācijām, TITLE elements ir obligāts un tam vajadzētu būt ar kontekstuālu saturu, piemēram, nevis "HTML", bet "HTML pamatelementi";



- lapas aprakstu un valodu, t.i., lai interneta pārlūkprogramma korekti attēlotu latviešu valodas īpašos simbolus ("charset= windows-1257") (vairāk par valodas kodējumu sadaļā "Kodējums"). Pie tam ievērosim, ka šajā gadījumā nav noslēdzošā taga. Visu šo garo aprakstu un valodu ir vērts vienkārši iegaumēt un iekopēt jaunveidojamā web lapā:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

<head>
<title>Mana lapa</title>
<meta http-equiv="content-type" content="text/html; charset= windows-1257">
</head>

<body>
</body>

</html>
```

3. <body> - HTML dokumenta ķermenis. Šeit ievieto visu lapas saturu: virsrakstus (<h1>), teksta rindkopas (<p>) utt.

Lūk kopējs piemērs:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

<head>
<title>Mana lapa</title>
<meta http-equiv="content-type" content="text/html; charset= windows-1257">
</head>

<body>
<h3>Manas lapas virsraksts</h3>
<p>Čau tauta! Te atkal es..</p>
</body>

</html>
```

Piezīme!

Ļoti svarīgi ir ievērot elementu tagu atvēršanas un aizvēršanas secību! Secība ir tāda pati, kā spēlējoties ar slavenajām krievu "matrjoškām" (lielajā koka lellē ir cita - nedaudz mazāka. Tajā atkal nedaudz mazāka. Šajā mazākajā atkal mazāka utt.). Tad nu iedomāsimies, ka esam salikuši 7 matrjoškas vienu otrā, bet beigās atklājam, ka esam aizmiršusi ielikt vismazāko. Neko darīt - noņemam virspusi ārējai (lielākajai) lellei un noliekam to sānis, tad tāpat rīkojamies ar 2. lielāko, tad 3. lielāko utt., līdz noņemta virspuse 7. lellei. Tad ieliekam tajā aizmirsto 8. lelli. Nu sākam krāmēt atpakaļ noņemtās leļļu virspuses - sākam ar 7., tad 6. utt. Tātad atveram no lielākās uz mazāko, bet aizveram no mazākās uz lielāko. Lūk arī tagu princips - ja <x> tags ievietots tagā <y>, tad vispirms jāaizver tags </y> un tikai tad tags </x>!

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

<head>
<title>Mana lapa</title>
<meta http-equiv="content-type" content="text/html; charset= windows-1257">
</head>

<body>
<h3>Manas lapas virsraksts
<p>Čau tauta! Te atkal es..</p></h3>
</html>
</body>
```

- Virsraksts </h3> aizvērts nepareizi, jo virsraksts un rindkopa <p> ir divi neatkarīgi elementi un nav padoti viens otram.
- </html> aizvērts pa ātru, jo <body> tags ir pakļauts <html> tagam un tāvad </body> tags bija jāaizver vispirms.

Virsraksti (headings)

HTML var definēt sešu veidu virsrakstus, kas savā starpā atšķiras pēc attēlojamo burtu lieluma. To apzīmējums ir "h", kam pievieno kārtas numuru no 1 līdz 6, t.i., no lielākā uz mazāko:

- <h1>Mans virsraksts</h1>
- <h2>Mans virsraksts</h2>
- ...
- <h6>Mans virsraksts</h6>

Protams, burtu lielumus, krāsas, fontu u.c. var katrs mainīt un nodefinēt kā viņam labpatīk (skatīt CSS mācību materiālu).

HTML kods:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

<head>
<title>Mana lapa</title>
<meta http-equiv="content-type" content="text/html; charset= windows-1257">
</head>

<body>
<h1>Mans 1. virsraksts</h1>
<h6>Mans 6. virsraksts</h6>
</body>

</html>
```

Mans 1. virsraksts

Mans 6. virsraksts

Rindkopas (paragraph)

Kā jau varēja noprast, tad ja angļiski rindkopa ir "paragraph", tad tās apzīmējums arī būs "p". Ievērosim, ka rindkopas automātiski savā starpā tiek atdalītas ar atstarpī! Aplūkosim kopīgu piemēru ar virsrakstu un pāris rindkopām..

HTML kods:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Mana lapa</title>
<meta http-equiv="content-type" content="text/html; charset= windows-1257">
</head>
<body>
<h1>Mans 1. virsraksts</h1>
<p>Kaut kas daudz sarakstīts. Kaut kas daudz sarakstīts. Kaut kas daudz sarakstīts. Kaut kas
daudz sarakstīts. Kaut kas daudz sarakstīts.</p>
<p>Atkal kaut kas daudz sarakstīts. Atkal kaut kas daudz sarakstīts. Atkal kaut kas daudz
sarakstīts. Atkal kaut kas daudz sarakstīts. Atkal kaut kas daudz sarakstīts.</p>
</body>
</html>
```

Rezultāts:

Mans 1. virsraksts

Kaut kas daudz sarakstīts. Kaut kas daudz sarakstīts. Kaut kas daudz sarakstīts. Kaut kas daudz sarakstīts. Kaut kas daudz sarakstīts.

Atkal kaut kas daudz sarakstīts. Atkal kaut kas daudz sarakstīts. Atkal kaut kas daudz sarakstīts. Atkal kaut kas daudz sarakstīts. Atkal kaut kas daudz sarakstīts.

Saraksti (lists)

Ikdienā mēs bieži veidojam dažādus sarakstus, piemēram, ejot uz veikalu, mēs pierakstām: "man jānopērk: maize, sviests, siers, piens utt." HTML ir trīs veidu saraksti:

- sakārtotais (ordered list) - uzskaitījumam lieto skaitļus vai burtus;
- nesakārtotais (unordered list) - uzskaitījumam lieto simbolus;
- definīciju <dl> (definition list) - pielieto, lai veidotu definīcijas, skaidrojumus un tml.

Šie tagi tikai iesāk un noslēdz saraksta veidošanu, bet tiem pakļauti (t.i., tajos atrodas) paši saraksta elementu tagi. Sakārtotajam un nesakārtotajam sarakstam tie ir , bet definīciju sarakstam <dt> (definējamajam terminam) un <dd> (termina skaidrojumam):

HTML kods:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Mana lapa</title>
<meta http-equiv="content-type" content="text/html; charset= windows-1257">
</head>

<body>
<h5>Sakārtotais saraksts</h5>
<p>Šodien jānopērk:</p>
<ol>
<li>maize</li>
<li>sviests</li>
<li>siers</li>
</ol>

<h5>Nesakārtotais saraksts</h5>
<p>Šodien jānopērk:</p>
<ul>
<li>maize</li>
<li>sviests</li>
<li>siers</li>
</ul>

<h5>Definīciju saraksts</h5>
<p>Produktu definīcijas:</p>
<dl>
<dt>Maize:</dt>
<dd>produkts, ko iegūst no rudzu u.c. miltiem, to sajaucot ar raugu ..</dd>
<dt>Sviests:</dt>
<dd>produkts, ko iegūst no krējuma, to sakuļot un pievienojot ..</dd>
<dt>Siers:</dt>
<dd>produkts, ko iegūst no biezpiena ..</dd>
</dl>
</body>
</html>
```

Rezultāts:

Sakārtotais saraksts

Šodien jānopērk:

1. maize
2. sviests
3. siers

Nesakārtotais saraksts**Šodien jānopērk:**

- maize
- sviests
- siers

Definīciju saraksts**Produktu definīcijas:****Maize:**

produkts, ko iegūst no rudzu u.c. miltiem, to sajaucot ar raugu ..

Sviests:

produkts, ko iegūst no krējuma, to sakuļot un pievienojot ..

Siers:

produkts, ko iegūst no biezpiena ..

Piezīme:

dažkārt sakārtoto sarakstu nepieciešams sākt nevis ar skaitli "1", bet gan turpināt no kāda lielāka skaitļa, piemēram, "4". Šajā gadījumā nāksies pielietot atribūtu "start" ar konkrēto vērtību (start="4"). Atceramies, ka atribūtus raksta iekšā atverošajā tagā un to vērtības liek pēdējās!

HTML kods:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Mana lapa</title>
<meta http-equiv="content-type" content="text/html; charset= windows-1257">
</head>
<body>
<p>Šodien jānopērk tas, ko aizmirsu vakar:</p>
<ol start="4">
<li>kola</li>
<li>čipsi</li>
<li>šokolāde</li>
</ol>
</body>
</html>
```

Šodien jānopērk tas, ko aizmirsu vakar:

4. kola
5. čipsi
6. šokolāde

Saites (hyperlinks)

Saites ir neatņemama interneta un weblapu sastāvdaļa. Tā ir iespēja, uzklikšķinot uz atbilstošās saites, atvērt vajadzīgo interneta resursu: mājaslapu, attēlu, lejupielādēt materiālu utt. Ir trīs veidu saites:

1. saites uz citu lapu savā domēnā, piemēram šī mācību materiāla kreisajā pusē esošās "HTML" sadaļas saites "Sintakse", "Pamatelementi" u.c. atver šī paša mācību materiāla jaunu, atbilstošu lapu;
2. saites uz kādu pilnīgi citu web lapu internetā;
3. saites uz kādu mūsu lapā iepriekš ieliktu atzīmi ("anchor" jeb "bookmark", t.i., grāmatzīmi). Šajā gadījumā atvērsies vajadzīgā lapa nevis no šīs lapas sākuma, bet vajadzīgajā vietā.

Tā kā saites veido uz kāda cita elementa bāzes, piemēram, teikuma, attēla, vārda (tā, klikšķinot uz vārda "Sintakse", tiks atvērta lapa, kurā būs materiāls par HTML sintaksi), tad šis elements, vai vārds jāiekļauj saites tagos - `<a></a>` ("anchor" - "enkurs").

Saites uz citu lapu (1., 2. variants)

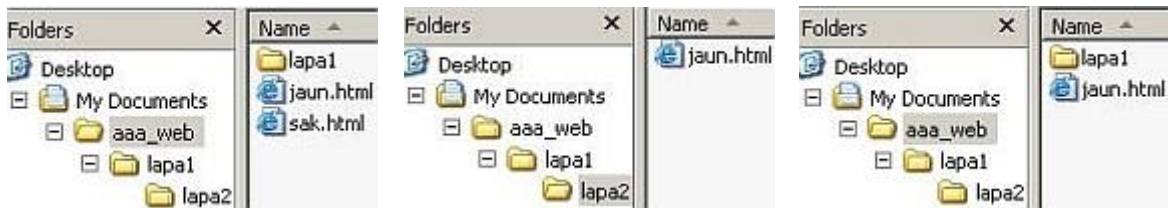
HTML kods (atļaušos turpmāk vairs nerakstīt versijas definējumu, "html" un "head" sadaļas, jo tur nekas nemainās, bet tās joprojām ir obligātas!):

```
<body>
<p>Te būs <a href="/saraksti.html">saite uz lapu "HTML saraksti".</a></p>
<p>Te būs <a href="http://www.w3schools.com">saite uz W3Schools mājaslapu.</a></p>
<p><a href="http://www.w3schools.com"></a></p>
</body>
```

Mazliet "paķidāsim" uzrakstīto:

- saites ("a") tagos var iekļaut atsevišķu rindkopas ("p") teksta daļu;
- kā saiti var izmantot attēlu, iekļaujot "img" attēla tagu (par kuru runāsim vēlāk) saites tagos;
- ar "href=" palīdzību norāda precīzu adresi atveramajai lapai. Pie tam pati adrese liekama pēdējās;
- ja veido saiti uz citu lapu savā domēnā, tad norāda ļoti precīzu atveramā faila nosaukumu un tā paplašinājumu, kā arī ceļu līdz tam. Ceļu norāda sekojoši:

Pieņemsim, ka atveramā lapa būtu "sak.html", bet saite uz šo lapu tiktu izveidota failā (lapā) "jaun.html" (piemēru attēli ņemti izmantojot programmu "Windows Explorer")



atveramā lapa atrodas vienā un tajā pašā mapē ar "jaun.html" (href="/sak.html")	atveramā lapa atrodas divus līmeņus augstāk par "jaun.html", t.i., mapē "aaa_web" (href="../../sak.html")	atveramā lapa atrodas vienu līmeni zemāk kā "jaun.html" (href="lapa1/sak.html")
--	--	--

Taču ir vēl kāda svarīga saišu īpašība - kur tiks atvērta lapa:

- (target="_blank") atver lapu jaunā pārlūka logā;
- (target="_self") atver lapu esošajā pārlūka logā (parasti šī ir noklusētā īpašība);
- (target="_parent") atver lapu augstākstāvošā ietvarā (frame);
- (target="_top") atver lapu esošajā pārlūka logā atceļot visus ietvarus (frames).

HTML kods:

```
<body>
<p>Te būs <a href="/saraksti.html">saite uz lapu "HTML saraksti".</a></p>
<p>Te būs <a href="http://www.w3schools.com" target="_self">saite uz W3Schools mājaslapu.</a></p>
<p><a href="http://www.w3schools.com" target="_blank"></a></p>
</body>
```

Noklusētā adrese (base)

Ieliekot "head" sadaļā noklusēto saišu adresi (base), panāksim, ka visas saites, kurām nebūs norādīta konkrēta adrese vai kuru adrešu pierakstā būsīm kļūdījušies, atvērs šo norādīto lapu:

```
<head>
<base href="/index.html">
</head>
```

E-pasta saite

Ir iespējams izveidot saiti, kura atvērs e-pasta pārlūkprogrammu (piemēram, "MS Outlook Express") uzreiz ierakstot vajadzīgo adresi, tematu (subject=) un aizpildot saturu (&body=). Gan tematu gan saturu var arī nenorādīt:

```
<a href="mailto:kaučkas@inbox.lv?subject=no labvēļa &body=Novēlu jums to labāko, tik to labāko vien!">Nosūtīt vēstuli!</a>
```

Tiesa, šādi, neslēpti ievietojot e-pasta adreses, mēs riskējam iekļūt spama izsūtītāju sarakstos. Ir dažādi veidi, kā slēpt adreses. Visvienkāršākais no tiem ir uzrakstīt savu e-pasta adresi nederīgā veidā, piemēram, "kaučkas[atat]inbox.lv" ar piebildi, ka sūtītājam "[atat]" jāaizvieto ar "@".

Attēli (image)

Veidojot mājaslapu, arī visi paredzamie attēli jāiekopē mājaslapas mapē! Protams, var izveidot speciālu apakšmapi "bilde", kurā tad visi attēli tiks glabāti.

Kā jau iepriekš redzējām, tad ar attēlu ievietošanu nodarbojas tags , savukārt "src" norāda attēla adresi. Adresi jeb atrašanās vietu norāda tā pat kā saitēm. Ievērosim, ka tagu nenoslēdz!

HTML kods:

```
<body>  
  
</body>
```

Rezultāts:



Protams, šo tagu var iekļaut starp rindkopas u.c. tagiem.

Attēla lielums

Ja lapā paredzams ievietot daudz attēlu, jeb tie būs lieli, t.i., ne tik ātri ielādējami pārlūka logā (jārēķinās, ka joprojām ir tauta, kas, pateicoties "Lattelekom" gādībai, joprojām bauda 42Kbps iezvanpieejas priekšrocības), tad būtu vēlams noteikt attēla platumu (width) un augstumu (height). Rezultātā pārlūkprogramma uzreiz rezervēs attiecīgo vietu attēlam un pārējais lapas saturs vairs nemainīs savu atrašanās vietu attēla ielādes gaitā. Ievietojot izmērus, vienmēr būtu vēlams arī norādīt mērvienības ("px" - punkti, "cm" - centimetri un tml.):

```
<body>  
  
</body>
```

Sīktēli (thumbnails)

Parasti sīktēlus pielieto veidojot attēlu (foto) galerijas. Lapā tiek ievietoti nelieli, dažus kilobaitus "viegli" attēliņi, uz kuriem klikšķinot atveras oriģinālattēli. Tādējādi lapa iegūst gan izskata gan ātruma ziņā. Principā tiek pielietots jau iepriekš apskatītais saišu princips:

```
<body>  
<a href="./smuki_liels.jpg" target="_blank"></a>  
</body>
```



Piezīme: attēlu apstrādei var lietot visdažādākās programmas (piemēram, brīvprogrammu "IrfanView"), taču vienmēr nākas balansēt starp kvalitāti un attēla faila lielumu. Iesaku:

- nav ieteicama lielāka izšķirtspēja kā 72 DPI;
- no attēlu standartiem izvēlieties ".jpg", ".gif" vai ".png", bet noteikti ne ".bmp" un tml.

Fona bilde (background)

Lai ieliktu fona bildi visai lapai:

```
<body background="smuki_liels.jpg">
```

```
Bla-bla  
</body>
```

Tabulas (tables)

Tabulu veidošanas izpratne HTML-ā ir ir patiesi svarīga. Nereti visa mājaslapa tiek veidota tabulā, tādējādi piešķirot lapai konkrētību un atvieglojot pašam veidotājam orientēšanos kodā.

Tabulas tags ir <table>, savukārt tabula sastāv no rindām pirmkārt (<tr> - table rows) un kolonnām otrkārt (<td> - table columns). Ievērosim, ka tieši šādā secībā tagi atverami un atbilstoši pēc tam aizverami! Attiecīgi vienā tabulā var būt vairākas rindas, bet katrā rindā vairākas kolonnas. Kolonnās varam rakstīt tekstu, ievietot attēlus utt.

HTML kods:

```
<body>  
<table>  
  <tr>  
    <td>  
      <p>1.rinda 1.kolonna</p>  
    </td>  
    <td>  
      <p>1.rinda 2.kolonna</p>  
    </td>  
  </tr>  
  <tr>  
    <td>  
      <p>2.rinda 1.kolonna</p>  
    </td>  
    <td>  
      <p>2.rinda 2.kolonna</p>  
    </td>  
  </tr>
```

</table>

</body>

Rezultāts:

1.rinda 1.kolonna 1.rinda 2.kolonna

2.rinda 1.kolonna 2.rinda 2.kolonna

Esam izveidojuši tabulu, taču neredzam tabulas līnijas, jo neesam tās nodefinējuši :(

Nedaudz no noformējuma

- (border=) - nosaka līniju biezumu. Jāņem vērā, ka tiek apvilkti katrā šūnā!
- (cellspacing=) - nosaka attālumu starp šūnu līnijām;
- (cellpadding=) - nosaka attālumu no šūnas līnijas līdz šūnas saturam, piemēram, tekstam;
- (rowspan="2") - apvienos šūnas virzienā uz leju, t.i., rindu šūnas - šajā gadījumā divas šūnas (pēdējās norāda apvienojamo šūnu skaitu);
- (colspan="2") - apvienos šūnas virzienā uz priekšu pa horizontāli, t.i., kolonnu šūnas - šajā gadījumā divas šūnas;
- (background=) – nosaka šūnas fona bildi;
- (bgcolor=) – nosaka šūnas fona krāsu.

HTML kods:

```
<body>
<table border="1px" cellspacing="5px" cellpadding="10px">
<tr>
<td colspan="3"><h5>Virsraksts</h5></td>
</tr>
<tr>
<td><p>2.rinda 1.kolonna</p></td>
<td rowspan="2" background="bilde.jpg"><p>2.rinda 2.kolonna</p></td>
<td><p>2.rinda 3.kolonna</p></td>
</tr>
<tr>
<td bgcolor="red"><p>3.rinda 1.kolonna</p></td>
<td><p>3.rinda 3.kolonna</p></td>
</tr>
</table>
</body>
```

Rezultāts:

Virsraksts		
2.rinda 1.kolonna	2.rinda 2.kolonna bilde	2.rinda 3.kolonna
3.rinda 1.kolonna		3.rinda 3.kolonna

Mazliet noformējuma

Kā jau sākumā minēju, tad HTML elementu noformējumam vēlams lietot CSS. Taču arī HTML4.01 ir iekļauti atsevišķi noformējuma tagi.

Un tomēr atgādināšu vēlreiz - pirms nopietni sākat veidot HTML lapas, noteikti apgūstiet CSS!!!

Lietojot zemāk minētos tagus, jāņem vērā, ka tie ne vienmēr korekti tiek attēloti visās pārlūkprogrammās. Es tos notestēju uz "Internet Explorer 6.0.29" (IE), "Opera 9.01" (Op) un "FireFox 1.5.0.6" (FF). Zaļa krāsa nozīmē, ka pārlūkprogramma doto tagu atbalsta, dzeltena - atbalsta, bet, manuprāt, nav līdz galam nostrādāts, sarkana - neatbalsta. Parasti vēlamais teksts tiek vienkārši ierakstīts starp tagiem, bet ar zvaigznīti (*) atzīmēti tie tagi, kuru pierakstā ir kādas īpatnības un to kodu piemēri doti zemāk.

Īpašība vai tags	nozīme	IE FF Op
<abbr></abbr>*	abreviatūras definēšana, piemēram: ASV	
<acronym></acronym>*	akronīma definēšana, piemēram: LOL	
<address></address>	pasta un tml. adreses pieraksts, piemēram: <i>Jānis Tautujānis</i> <i>"Leļķojas"</i> <i>Pākšu pagasts</i>	
	treknraksts	
<bdo></bdo>*	teksta (tabulu) virziena definēšana. Nāksies lietot atribūtu "dir" ar iespējamajām vērtībām "ltr" (left-to-right, noklusētā vērtība) un "rtl" (right-to-left), piemēram: mainīts virziens	
<big></big>	visi burti nu ir lielāki	
<blockquote></blockquote>	Gara citāta definējums, piemēram: Kāds students reiz teica: Citēt vajag mācēt, citēt vajag spēt, beigās savu murgu, citam iesmērēt! Tomēr pasniedzēji ar nav vakarējie.. :(	
<cite></cite>	Definē citātu, piemēram: <i>Šis ir citāts</i>	
<code></code>	teksts atgādina dator kodu	
	uzrāda vēlāk dzēstu tekstu, piemēram: iešu uz veikalu, bet pirms tam ieskriešu kafenē un tad atpakaļ.	

<code><dfn></dfn></code>	Definīcijas definēšana, piemēram: <i>kaķis - četrkājains, pret cilvēkiem iecietīgs zvērs</i>
<code></code>	<i>uzsvērtais teksts (emphasized)</i>
<code><i></i></code>	<i>slīpraksts</i>
<code><ins></ins></code>	uzrāda vēlāk pievienotu tekstu, piemēram: iešu uz veikalu, <u>bet pirms tam ieskriešu kafenē</u> un tad atpakaļ.
<code><kbd></kbd></code>	klaviatūras (keyboard) veida teksts
<code><pre></pre>*</code>	nenoformēts teksts - teksts tiks attēlots <u>tieši</u> tāds, kādu uzrakstīsim, ņemot vērā visas atstarpes, atkāpes, pārejas uz jaunu rindkopu ("Enterus") utml.
<code><q></q></code>	īss citāts (quotation), piemēram: Kāds students reiz teica: Pietiek zubrīt, gribas ēst!
<code><samp></samp></code>	datora paraug-, jeb nolases kods (sample)
<code><small></small></code>	sīks teksts
<code></code>	Izceltāks, treknāks teksts
<code><sub></sub></code>	apakšraksts, piemēram: O ₂
<code><sup></sup></code>	augšraksts, piemēram: x ²
<code><tt></tt></code>	teletaipa veida teksts
<code><var></var></code>	<i>mainīgā (variable) attēlošana</i>
<code>
*</code>	nav atsevišķa noslēdzošā taga. Identisks MS Word taustiņiem "Shift+Enter" - pāreja uz jaunu rindiņu
<code><hr>*</code>	nav atsevišķa noslēdzošā taga. Uzzīmē horizontālu līniju ar atsarpī virs un zem līnijas

Sākumā šķiet, ka daudziem noformējumu tagiem ir līdzīgs rezultāts. Tad kāda ir jēga!? Runa iet par pareizas jeb ērtas lapas struktūras veidošanu. To, šķiet, būs lemts izprast tikai tiem, kas apgūs arī CSS. Piemēram, ja mēs veidotu mācību materiālu par kādu programmēšanas valodu, sniedzot daudz koda piemēru, tad tikai ar pāris rindiņām CSS koda visam tekstam mūsu materiālā, kas iekļauts `<code></code>` tagos, mēs varētu piešķirt zilu krāsu, bet mainīgajiem starp `<var></var>` tagiem - dzeltenu!

Meta dati

Meta dati ļauj autoram sniegt informāciju par dokumentu "mašīnlasāmā" veidā. Tātad, šī informācija nav domāta cilvēkam, bet, piemēram, meklētājrobotiem - programmām, kas ievāc informāciju par tīklā pieejamajiem resursiem (Google-i un tml.)

Dažas piezīmes:

- * "Meta" datus ievieto HTML lapas "head" sadaļā.
- * "Meta" elementiem nav atsevišķi noslēdzošie tagi.
- * "Meta" elementiem jāpiešķir īpašību *name* vai *http-equiv* (t.i., par ko ies runa, piemēram, par autoru) un tās vērtību *content* (tiek nosaukts konkrētais autors, piemēram, Kaspars Antonevičs). Saturu liek pēdiņās.

Piemērs:

```
<head>  
<meta name="author" content="Kaspars Antonevičs">  
</head>
```

Tagad aplūkosim konkrētus *meta* elementus..

Autors (author)

Kā jau minēju, tad ar šīs īpašības palīdzību varam definēt dokumenta autoru. Papildus iespējams lietot arī *lang* (valodas) vai *dir* (teksta virziena) atribūtus, ja tiek lietota kāda specifiska valoda. Tādējādi runas sintezatori spēj piemērot pareizo valodu saturu, šajā gadījumā autora vārda, izrunāšanai.

HTML kods:

```
<head>  
<meta name="author" lang="en" content="Gerald Durel">  
</head>
```

Atslēgvārdi (keywords)

Ir dažādi veidi, kā dot ziņu meklētājprogrammām (Google, Yahoo u.c.) par savas lapas esamību, lai klients, ievadot attiecīgu atslēgasvārdu meklēšanas lauciņā, atbildē saņemtu norādi arī uz mūsu lapu. Var personīgi paziņot, piemēram, Google-i par lapas esamību un tās saturu. Var nedarīt neko un gaidīt, kad interneta meklētāju roboti paši savāks visu informāciju par mūsu lapu. Taču var lietot arī "meta" elementus, norādot šiem robotiem, mūsaprāt, vajadzīgo informāciju. Šajā gadījumā mēs pasakām, ka tiks lietoti atslēgvārdi, kā arī tie nosaukti. Var nosaukt vienu vai vairākus atslēgasvārdus. Ja tie ir vairāki, tad vienu no otra tos atdala ar komatu. Ir jānosauc tikai lapas saturam atbilstoši atslēgvārdi, kā arī tie nav jāatkārto.

HTML kods:

```
<head>  
<meta name="keywords" content="HTML, CSS, tutorials, mācību materiāli, kasis" >  
</head>
```

Apraksts (description)

Bez atslēgvārdiem mēs pāris teikumos meklētājprogrammām varam sniegt arī īsu savas lapas aprakstu, pie tam te būtu jāparādās iepriekš minētajiem atslēgvārdiem.

HTML kods:

```
<head>  
<meta name="keywords" content="HTML, CSS, tutorials, mācību materiāli, kasis" >  
<meta name="description" content="Latviski HTML un CSS mācību materiāli iesācējiem no kasis" >  
</head>
```

Pēdējās izmaiņas (revised)

Varbūt lielas jēgas nav, bet ir arī iespēja paziņot meklētājprogrammām kad tad šajā lapā veiktas beidzamās izmaiņas.

HTML kods:

```
<head>  
<meta name="revised" content="Pēdējās izmaiņas: 10/12/2005" >  
</head>
```

Atjaunošana un pārsūtīšana (refresh & redirect)

Dažkārt gadās, ka mēs esam nomainījuši domēnu, t.i., lapu izvietojusi zem citas adreses. Tad būtu labi, ka atverot mūsu veco adresi, neinformētie apmeklētāji automātiski tiktu pārsūtīti uz mūsu jauno lapas atrašanās vietu. To veic ar "refresh" palīdzību, kura satur (content=) sadaļā norāda pēc cik sekundēm lapas saturs tiks atjaunots un, norādot pilnu jauno adresi, pēc šīm sekundēm apmeklētājs tiks automātiski pārvirzīts uz jauno adresi. Ja mēs atjaunošanas laiku ieliekam vairākas sekundes ilgu, tad vecajā lapas atrašanās vietā būtu vērts ielikt apmēram šādu paziņojumu: "Mūsu lapas atrašanās jaunā vieta ir: www.kasis.com ! Pēc 3 sekundēm Jūs automātiski uz turieni tiksiet pārvirzīts!"

Piebilde! Tomēr W3C standarti neiesaka lietot šo pārsūtīšanas (caur atjaunošanu) metodi, jo tādējādi dažiem lietotājiem lapa var kļūt nepieejama. Tiesa, man nav nācies novērot šādu problēmu.

HTML kods:

```
<head>  
<meta http-equiv="refresh" content="3; url=http://www.kasis.lv" >  
</head>
```

Noklusēto (default) vērtību definēšana

Ar "meta" datu palīdzību var definēt sekojošas noklusētās vērtības:

- [dokumenta rakstāzīmju kodējumu](#)
- koda (scripting) valodu

- [stila lapas valodu](#). Tiesa, arī šī meta elementa vietā var lietot manis sniegtās saites un definēšanas piemēru - vismaz man tā šķiet ērtāk. Starp citu - zemāk piemērā dots variants der, ja tiek lietots lapas iekšējais CSS. Bez tam, W3C standarti nosaka, ka gadījumos, ja stila lapas valoda nav definēta, tad noklusēti tiek pieņemts, ka tā ir CSS (text/css)

```
<head>  
<meta http-equiv="Content-Style-Type" content="text/css">  
</head>
```

Piebilde

- "meta" dati nav obligāti
- atsevišķu informāciju var paust arī ar citiem HTML elementiem vai atribūtiem, nelietojot "meta" datus, piemēram ar "address", "title" un tml.
- dažkārt "meta" elementa "content" vērtības vietā var ievietot saiti uz kādu dokumentu, kur glabājas nepieciešamā informācija, jeb šo "meta" datu vietā vienkārši lietot saiti (link)

Dokumenta veids (Doctype)

Runājot par HTML, laika gaitā ir tikušas izveidotas vairākas šīs valodas versijas. Pēdējo versiju HTML4.01 organizācija W3C "laida tautās" 1999.gada 24.decembrī, kas patiesībā ir versijas HTML4 apakšversija. Katrai no šīm versijām ir savas īpatnības, noteikumi, savi atļautie un aizliegtie (neiekļautie) tagi. Ja šodien lietojam HTML, tad W3C iesaka lietot tieši HTML4.01 versiju. Šīs versijas lielā atšķirība no versijas HTML3.2 ir tāda, ka tagad tiek strikti nodalīts lapas saturs (HTML) un noformējums (CSS).

Tātad, kad Interneta pārlūkprogramma atver mūsu sagatavoto datni, tai, pirmkārt, būtu precīzi jāzin, pēc kādiem kritērijiem dokumentu interpretēt jeb validēt. Veids, kā mēs darām zināmu pārlūkprogrammai mums vēlamās interpretācijas noteikumus, ir pašā dokumenta sākumā pirms <html> taga ievietojot attiecīgu DTD (Document Type Definition) definīciju. HTML4.01 tās ir sekojošas:

1. Striktais definējums - šajā gadījumā tiks atbalstīti tikai HTML4.01 versijā iekļautie tagi, bet netiks atbalstīti tagi, kas tomēr bija iekļauti iepriekšējās versijās, bet tagad vairs netiek atbalstīti. Tāpat netiks atbalstīti rāmji (framesets). Šķiet, šo definējumu tomēr lieto diezgan reti tā strikto noteikumu dēļ.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

2. Pārejas definējums - šajā gadījumā tiks atbalstīti gan HTML4.01 versijas tagi, gan šajā versijā vairs neatbalstītie tagi un īpašības. Tomēr netiks atbalstīti rāmji (framesets). Šis, šķiet, būs visbiežāk lietotais definējums.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

3. Rāmju definējums - šajā gadījumā tiks atbalstīts viss pārejas definējumā iekļautais, kā arī rāmji (frameset).

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"  
"http://www.w3.org/TR/html4/frameset.dtd">
```

Tāad HTML dokumenta sākumam būtu jāizskatās apmēram šādi:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">  
<html>  
<head>
```

...