

TOILA GÜMNAASIUM

**HTML – Hypertext Markup
Language**

Abimaterjal Veebitehnoloogia kursusel

Avar Ambrosius Pentel

Jõhvis 2001 (täiendatud 2009)

Sisukord

| | |
|---|----|
| Sissejuhatus..... | 2 |
| Mis programmiga HTMLi kirjutada ja kuidas salvestada? | 3 |
| HTML elementide üldine struktuur..... | 4 |
| HTML lehekülje üldine struktuur | 5 |
| HTML elementide atribuudid | 6 |
| Värvid | 8 |
| Üksikutele tekstilõikudele erinevad värvid..... | 9 |
| Elementide paigutust toetavad elemendid ja atribuudid..... | 11 |
| Struktureeritud HTML elemendid..... | 13 |
| Loetelud | 13 |
| Tabelid | 14 |
| Vormielemendid..... | 15 |

Sissejuhatus

HTML – *Hypertext Markup Language* – on keel, milles kirjeldatakse elektroonilise dokumendi struktuuri. Eelkõige on see loodud semantilist struktuuri silmas pidades, mis oleks arusaadav ühtviisi inimesele ja arvutile. See tähendab, et HTML keele abil eristatakse dokumendi tähenduslikult erinevad osad: pealkirjad, alapealkirjad, tekstilõigud, lingid, pildid, tabelid jne. Selles kontekstis peab semantiline eristamine olema eelkõige ühetähenduslikult arusaadav arvutile. Semantilisele vastandub siis visuaalne/kujunduslik eristamine, mis on arusaadav vaid inimesele. Näiteks võib tuua kaks dokumenti, millest esimese pealkirjad on kujundatud vaid visuaalselt paksema ja suurema kirjaga ja teisel on kasutatud pealkirjade vormistamisel semantilist eristamist spetsiaalse pealkirja elemendiga. Esimesel juhul saab lehekülge vaatav inimene aru, et tegu on pealkirjaga, kuid arvuti jaoks on vajalik spetsiaalne märgistamine.

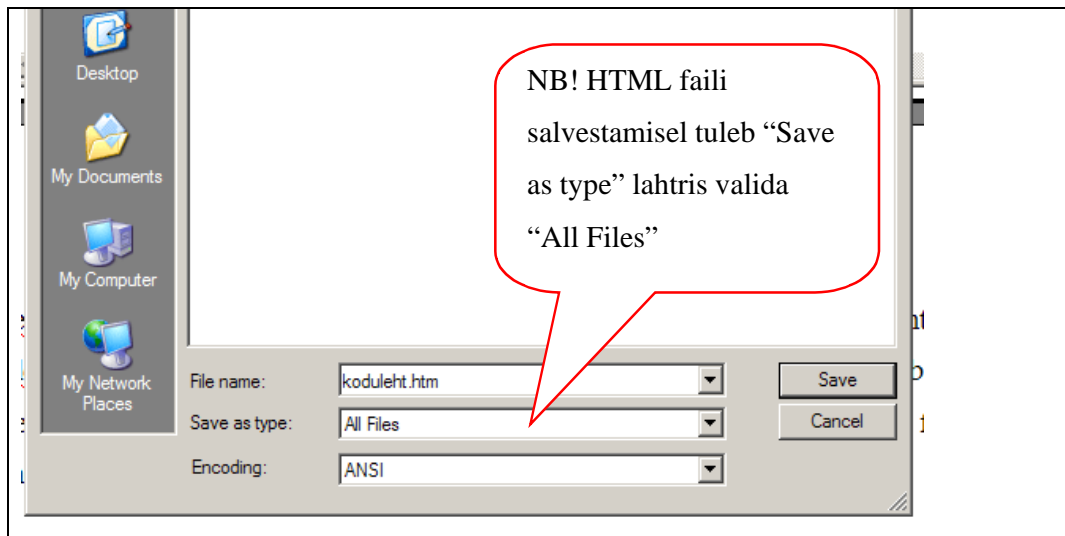
Veebilehekülgede arenedes sooviti lisaks semantikale arendada ka kujunduslikku poolt ja seetõttu lisati HTML keelele elemente ja atribuute, millel arvuti jaoks semantilist tähendust pole. Nüüd on tulnud tagasi algse taotluse juurde ja püütakse hoida veebilehtede kujunduslik ja tähenduslik pool teineteisest lahus. Selleks on

loodud uus keel CSS – *Cascading Style Sheets*, mis teenib puhtalt kujunduslikke eesmäärke ja annab ka märksa suurema vabaduse lehekülje kujundamiseks.

Mis programmiga HTMLi kirjutada ja kuidas salvestada?

HTML lehekülg on tavaline tekstidokument, milles kasutatakse erinevate lehe osade eristamiseks spetsiaalseid HTML elemente ehk märgendeid (ingl. *Tag*). Tekstidokument tähendab siin seda, et tegu peab olema puhta tekstiga (selle kohta öeldakse ka ASCII tekst), milleks ei sobi näiteks *Wordi* või *Wordpadi* tavaline dokument. MS Windowsi programmidest sobib näiteks *Notepad* nimeline programm, mille leiab tavaliselt *Start -> Programs -> Accessories -> Notepad* alt. Samuti võib Notepadi käivitamiseks kirjutada otse Start menüü *Run* aknasse *notepad* ja vajutada *enter* klahvile.

Vaikimisi pakub *Notepad* dokumentide salvestamiseks faililaiendiks *.txt*. Sellised failid on seostatud *Notepadiga* ja kui nende peal klõpsata, siis avanevad nad uuesti *Notepadis*. HTML failid on mõeldud avanemiseks internetibrauseris (*Internet Explorer, Mozilla, Opera, Chrome* või muu) ja et nad seda teeksid, tuleb faili salvestamisel valida faili tüübiks *all files* ja kirjutada faili nimelahtrisse failile laiendiks *.htm* või *.html* (näiteks: *koduleht.htm*).



Kui salvestamisel ununeb *Save as type* lahter muutmata, siis kirjutab süsteem teie failinimele ja laiendile veel lõppu oma laiendi .txt ja fail saab tegelikult nimeks koduleht.htm.txt ning seetõttu ei avane brauseris.

Paljudel arvutitel on operatsioonisüsteemid seadistatud nii, et faililaiendeid ei näidata. Et neid näha selleks peaks avama *My Computer -> Tools -> Folder Options -> View* ja võtma seal linnukese eest ära real: *hide extensions for known file types*. Selleks tegevuseks peavad olema õigused, piiratud õigustega kasutajakontol sisse logides see ei õnnestu.

HTML elementide üldine struktuur

Iga html element eristatakse < ja > märkidega (väiksem, suurem). Me võime Internetibrauserit ette kujutada kui inimest, kes loeb iga HTML dokumendi sümbolit ning kohates märki < tõlgendab ta seda korraldusena ja loeb korraldust kuni märgini > ning seejärel asub antud korraldust täitma. Korralduse täitmise lõpetab ta siis, kui ilmub tühistav korraldus, mis näeb välja samasugune, kuid teiseks sümboliks on alati kaldkriips - </element>.

Näiteks selleks, et anda korraldus kuvada mingi tekstilõik paksuna:

```
tavaline tekst <b>paks tekst</b> tavaline tekst jätkub
```

Eeltoodud näites interpreteerib brauser alates sümbolite < ja > vahele jäävaid *b* tähti korraldustena ja ning elementide vahele jääv tekst tehakse paksuks. Samamoodi saab teha kaldkirja ja allajoonitud kirja:

```
tavaline kiri <i>kaldkiri</i> ja <u>allajoonitud kiri</u>
```

Kui soovime kasutada mitut elementi korraga, näiteks soovime, et tekst oleks paks ja ka allajoonitud, siis tuleb järgida elementide paigutamise hierarhilist struktuuri järgmiselt:

```
<u><b>allajoonitud paks kiri</b></u> või  
<b><u>allajoonitud paks kiri</u></b>
```

kuid kindlasti mitte nii nagu järgmises näites:

```
<b><u>paks allajoonitud kiri</b></u>
```

Reeglina on igal HTML elemendil lõpetav paariline. Kuid mitte kõik korraldused ei ole sellise iseloomuga, et vajaksid eraldi lõpetavat korraldust ja seetõttu on ka ilma paailiseta HTML elemente. Kõige lihtsamaks ja levinumaks näiteks on reavahetuse sümbol `
` ja horisontaaljoone sümbol `<hr>`. Neile ei panda lõpetavat sümbolit `</br>` ja `</hr>`. Samuti pildi element `` ja sisestuselement `<input>` on täiesti eraldiseisev ja ei vaja lõpetavat paarilist.

HTML lehekülje üldine struktuur

Internetibrauser suudab kuvada veebilehte ka siis, kui te korrektset HTML lehe struktuuri ei järgi. Näiteks eeltoodud näite paksu kirja, kaldkirja ja allajoonitud kirjaga võib kopeerida notepadi, salvestada (järgides seda, et *file type* oleks *all files* ja faililaiendiks saaks `.htm` või `.html`) ning juba saabki neid html failidena avada ja näha sisestatud HTML elementide mõju.

Siiski on olemas teatud põhistruktuur, mida on soovitatav iga HTML lehe puhul järgida.

Lehekülj algab ja lõpeb <html> elemendiga ja lehe nähtav osa on elementide <body> ja </body> vahel. Nii <html> kui <body> element esineb kodulehel vaid üks kord (lõpetavat paarilist mitte arvesse võttes):

```
<html>
<body>
Siin on lehekülje sisu ja muud html elemendid
</body>
</html>
```

HTML elementide atribuudid

Paljude HTML elementide juures saab nende mõju muuta lisades elemendile atribuute. Näiteks lingi loomisel peab ju kuidagi osutama kuhu see link viib. Ja seetõttu lingi tegemise element <a> on iseenesest ilma atribuudita täiesti kasutu. Soovides teha linki google.com lehele, kirjutame järgmise koodilõigu:



```
<a href="http://www.google.com">Link google lehele</a>
```

Eeltoodud näites on elemendi <a> atribuudiks *href*, mis osutab sellele kuhu peab antud link sellele klikkides viima. Tekst “Link google lehele” värvub siin lingile

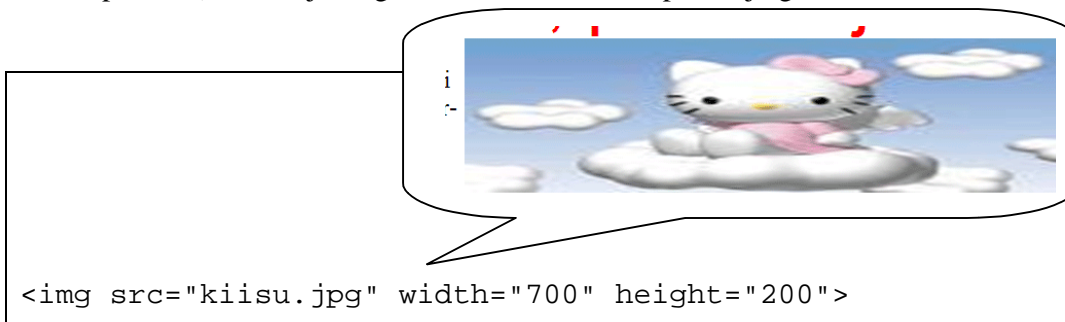
omastes toonides (siniseks kui pole määratud teisiti) ja joonitakse alla ning hiirekursor muutub selle kohal käekujuliseks.

Samuti on pildi tegemisel vajalik määrata pildi asukoht. Pildid pole HTML dokumendi osad, need on eraldiseisvad failid, millele HTML dokumendis viidatakse. Näiteks soovides näidata oma HTML dokumendis pilti *kiisu.jpg*, mis asub selle HTML dokumendiga samas kataloogis, siis kirjutame järgneva koodilõigu:



Eeltoodud näites on elemendi `` atribuudiks *src*, mis viitab pildi allikale (selleks võib olla ka mingi pildi internetiaadress). Atribuudid on elemendispetsiifilised. Atribuut, mis toimib ühe elemendi juures, ei pruugi omada mõju teise juures. Elemendil `` puhul ei saa näiteks kasutada atribuuti *href* ja element `<a>` ei saa aru atribuudist *src*.

Elemendil saab olla rohkem kui üks atribuut. Näiteks saab pildi puhul määrata sellele uue suuruse. Vaikimisi kuvatakse pilt originaalmõõtmetes, kuid seda saab muuta atribuutidega *width* ja *height*. Kasutades vaid ühte atribuuti kahest (*width* ja *height*) muudetakse teist automaatselt ja pilti suurendatakse või vähendatakse proportsionaalselt. Kui määrata mõlemate atribuutide väärtused, siis kuvatakse pilt just nendes proportsioonides (pilti saab nii laiuse või kõrguse suunas venitada või kokku pressida). *Width* ja *height* atribuudid lisatakse pildile järgmiselt:



Värvid

Atribuutidega on seotud ka kõik värvidega seonduv leheküljel. Kogu lehekülje teksti ja taustavärvi ning linkide värvi saab määrata <body> elemendi atribuutidega. Body elementi saab olla HTML dokumendis vaid üks. Lehekülje nähtav osa algab sellega ja lõpeb sellega. Tausta ja tekstivärvid lisatakse body elemendile järgmiselt:

```
<body bgcolor="yellow" text="red">
```

Eeltoodud näide muudab lehekülje taustavärvi kollaseks ja tekstivärvi punaseks. Enamkasutatavate värvide puhul suudavad internetibrauserid tõlgendada ka sõnalisi värvinimetusi nagu red, blue, green ning paljud teised, aga spetsiifilisemate värvide kuvamiseks peaks siiski kasutama värvi koodi.

Värvi koodi saab sisestada näiteks kuuekohalise 16-nd süsteemi numbriga. 16-nd süsteemi numbrid on lisaks kümnendsüsteemis kasutatavatele 0-9 veel tähed a, b, c, d, e, f (s.t puuduvad 6 numbrit on võetud tähestiku algusest). Värvi kood näeb siis välja näiteks selline: fa07c1. Koodi tähendust on hea meelde jätta lühendiga RGB (Red, Green, Blue) -esimesed kaks numbrit tähistavad punast, keskmised kaks rohelist ja viimased kaks sinist värvi. Elektroonilised põhivärvid on erinevad sellest, mis toimivad kunstniku värvimolbertil – elektroonilistest põhivärvidest puudub kollane, aga seal on hoopis roheline. Kõik värvid ekraanil saadakse nende – punase, rohelse ja sinise valguse segunedes. Kui valgust ei ole siis on ekraan must ja selle kood on 000000. Maksimaalne valgus on valge ja selle kood on ffffff. Puhas punane on ff0000, puhas roheline on 00ff00 ja puhas sinine on 0000ff.

Järgnevas näites on osutatud, kuidas need värvikoodid lisatakse.

```
<body bgcolor="00ff00" text="ffffff">
```


Peale tausta ja tekstivärvi saab *body* elemendi atribuutidega muuta veel lingivärvi. Selleks on atribuut `link`, millele antakse väärtuseks soovitud värv:

```
<body bgcolor="00ff00" text="ffffff" link="ccffcc">
```

Tuleb arvestada sellega, et lingil on kolm olekut ja need kõik võivad olla tähistatud eri värvidega:

1. tavaline link (vaikimisi sinine),
2. aktiivne link (see on link, mille peal on just kõpsatud, aga uus lehekülg pole jõudnud veel laadida)
3. ja külastatud link (vaikimisi lilla) – see on link mingile lehele, kus te olete juba käinud.

Kõigi nende olekute jaoks saab seada endavalitud värvid. Selleks tuleb lisada *body* elemendi veel atribuutideks `vlink` (külastatud link – visited link) ja `alink` (aktiivne link):

```
<body bgcolor="00ff00" text="ffffff" link="ccffcc"
alink="red" vlink="aabbcc">
```

Üksikutele tekstilõikudele erinevad värvid

Eeltoodud näidetes defineeritakse värvid kogu lehele. See tähendab, et kogu tekst lehel on ühte värvi ja taustavärv on terve lehe ulatuses samasugune. Samuti on kõik lingid samasugused. Taustavärvi saab HTMLis määrata peale *body* elemendi veel tabelile ja selle igale lahtrile eraldi ja nii võib siiski lehekülje eri osi eri värvi taustal kuvada.

Kuid tabel ei võimalda muuta teksti või lingivärvi.

Eraldiseisva tekstilõigu värvi muutmine toimub HTML keeles elemendi abil. Soovitatav tekst eristatakse elementidega ja määratakse font elemendile atribuut color:

see tekst on tavaline, aga see tekst on punane

```
see tekst on tavaline, aga <font color="red">see tekst on punane</font>
```

Font elemendiga saab veel määrata fonditüüpi ning kirjasuurust. Need atribuudid on vastavalt *face* ja *size*. Kirjasuurus on number vahemikus 1-6 ja fonditüüp on suvaline font nagu *Arial*, *Verdana*, *Courier* vm. Siiski on soovitatav kasutada levinud fonditüüpe, sest fontide näitamise võimalikkus sõltub sellest, kas veebilehe vaataja arvutis on see font olemas. Font element koos kõikide atribuutidega näeb välja järgmine:

See tekst on tavaline, aga see tekst suur, punane ja Arial

```
See tekst on tavaline, aga  
<font color="red" size="7" face="arial">see tekst suur,  
punane ja Arial</font>
```


Atribuutide järjekord HTML elementides ei oma mingit tähtsust. Seetõttu pole vahet, kas kirjutada:

```

```


Või

pildi paigutamise omab atribuut *align* veel ühte olulist mõju pildile. Kui sisestada mingi leheküljest kitsam pilt pikema tekstilõigu algusesse ja määrata pildile *align* atribuudiga paiknemine vasakule või paremale, siis ümbritsetakse pilt tekstiga, vastasel korral algab tekst pildi kõrvalt ja teksti esimene rida on samal tasemel pildi alumise servaga:



 ilma *align* atribuudita:


Lattelecom on Läti internetisõjas astunud järgn keskpaaigas hakatakse ettevõtte fiiber-koju võrgus pakkuma ühendus Lattelecom ei ole ise veel teadet kommenteerinud, kuid uudist vaheer uudistekanalid ja blogid. Väidetavalt on 500 Mbit/s üks maailma kiire kommertsinterneti teenustest, vahendab Arvutimaailm. Fiiber-koju te kodu Riias, lisandumas on veel 700 klienti. Lattelecomiga konkureer Starman Eestis Elioniga, pakkudes üle telekaabli 100 Mbit/s ühendus



 align atribuudiga:

Lattelecom on Läti internetisõjas astunud järgmise sammu ja novembri keskpaaigas hakatakse ettevõtte fiiber-koju võrgus pakkuma ühendust kiirusega e veel teadet kommenteerinud, kuid uudist vaheer uudistekanalid ja blogid. Väidetavalt on 500 Mbit/s üks maailma kiireimast koju jõudvatest kommertsint vahendab Arvutimaailm. Fiiber-koju teenust kasutavad praegu Riias, lisandumas on veel 700 klienti. Lattelecomiga konkureerib kaabellvifirma Eestis Elioniga, pakkudes üle telekaabli 100 Mbit/s ühendust DOCSIS tehnoloogi

s astunud järgmise sammu ja novembri fiiber-koju võrgus pakkuma ühendust kiirusega e veel teadet kommenteerinud, kuid uudist ite uudistekanalid ja blogid. Väidetavalt on 500 koju jõudvatest kommertsinterneti teenustest koju teenust kasutavad praegu 2600 klienti. Lattelecomiga konkureerib kaabellvifirma Eestis Elioniga, pakkudes üle telekaabli 100 Mbit/s ühendust DOCSIS tehnoloogi



 align atribuudiga:

Starman Eestis Elioniga, pakkudes üle telekaabli 100 Mbit/s ühendust DOCSIS tehnoloogi

Struktureeritud HTML elemendid

Mõned HTML elemendi kujutavad endast hierarhilist struktuuri, kus selle üksikuid osi eraldiseisvalt kasutada ei saa. Tuntumad sellistest struktuursetest elementidest on loetelud, tabelid ja rippmenüüd.

Loetelud

HTML keeles on loetelusid kahte liiki: nummerdatud ja nummerdamata loetelud. Loetelud koosnevad kahest erinevast elemendist. Välimine element määrab, kas tegu on nummerdatud (element `` - *ordered list*) või nummerdamata loeteluga (element `` - *unordered list*). Mõlema loetelu sisemised elemendid on samasugused (`` - *list item*) ning need tähistavad ühte alapunkti loetelus.

Nii on numbrilise loetelu kood järgmine:

| | |
|--|---|
| <pre> Loetelu esimene rida Loetelu teine rida Loetelu kolmas rida </pre> | <pre>1. Loetelu esimene rida 2. Loetelu teine rida 3. Loetelu kolmas rida</pre> |
|--|---|

Ja nummerdamata loetelu on järgmine:

| | |
|--|--|
| <pre> Loetelu esimene rida Loetelu teine rida Loetelu kolmas rida </pre> | <pre>• Loetelu esimene rida • Loetelu teine rida • Loetelu kolmas rida</pre> |
|--|--|

Tabelid

Loeteludest keerulisema struktuuriga on tabelid. Tabel koosneb vähemalt kolmest erinevast elemendist `<table>`, `<tr>` ja `<td>`.

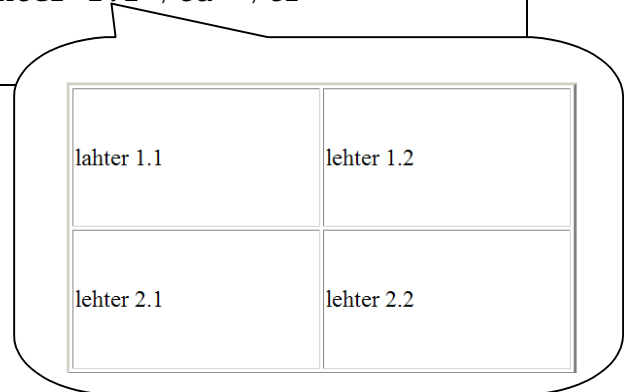
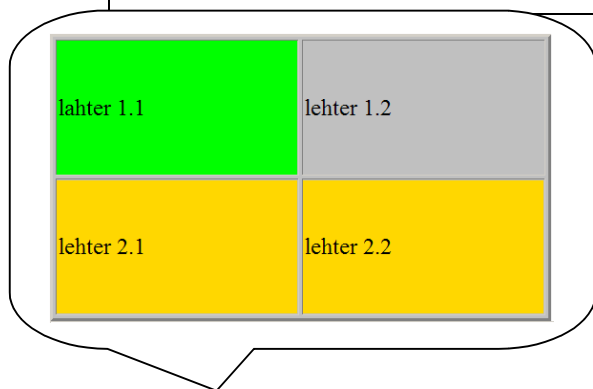
`<table>` element määrab terve tabeli suuruse ja muud omadused nagu raam, paiknemine, taustavärv jne.

`<tr>` määrab tabeli rea

`<td>` määrab tabeli lahtri.

Lihtsa 2 x 2 tabeli loob näiteks järgmine koodilõik:

```
<table border="2" width="350" height="200">
<tr><td>lahter 1.1</td><td>lahter 1.2</td></tr>
<tr><td>lahter 2.1</td><td>lahter 2.2</td></tr>
</table>
```



Tabeli igale elemendile on võimalik määrata erinev taustavärv *bgcolor* atribuudiga:

```
<table border="2" width="350" height="200" bgcolor="silver">
<tr><td bgcolor="lime">lahter 1.1</td><td>lahter 1.2</td></tr>
<tr bgcolor="gold"><td>lahter 2.1</td><td>lahter 2.2</td></tr>
</table>
```

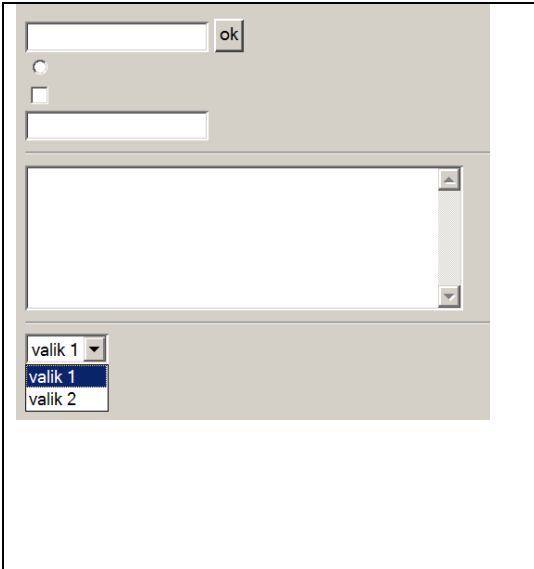
Sarnaselt pildile, saab ka tabelile lisada *align* atribuudi. Tabel paigutatakse siis teda ümbritseva teksti sisse vastavalt atribuudi *align* väärtusele - paremale või vasakule.

Kui tabeli lahtri elemendile `<td>` lisada atribuut *align*, siis määrab see teksti ja muude elementide paiknemist selle tabelilahtri sees. `<td>` elemendil on ka üks unikaalne

paigutusatribuut – *valign* (*vertical alignment*), mis määrab teksti paiknemise lahris vertikaalsuunal (ülal, all või keskel).

Vormielemendid

Üsna kiiresti arenesid veebilehed staatilistest dokumentidest interaktiivseteks rakendusteks, mis nõudsid uusi elemente andmevahetuseks kasutaja ja veebirakenduse vahel. Neid elemente nimetatakse vormielementideks. Vormielementide abil saab kasutaja sisestada ja valida mingeid andmeid ja nendega mingeid protsesse käivitada – teha pangaülekandeid, otsida infot, kirjutada foorumisse, kasutada mingit online kalkulaatorit, veebipõhist mailiklienti vm sellist.

| | |
|--|---|
|  | <pre><input> <input type="button" value="ok"> <input type="radio"> <input type="checkbox"> <input type="password"> <textarea cols="40" rows="7"> </textarea> <select> <option>valik 1</option> <option>valik 2</option> </select></pre> |
|--|---|