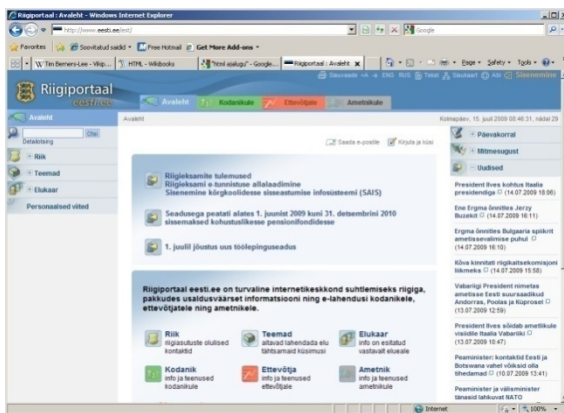


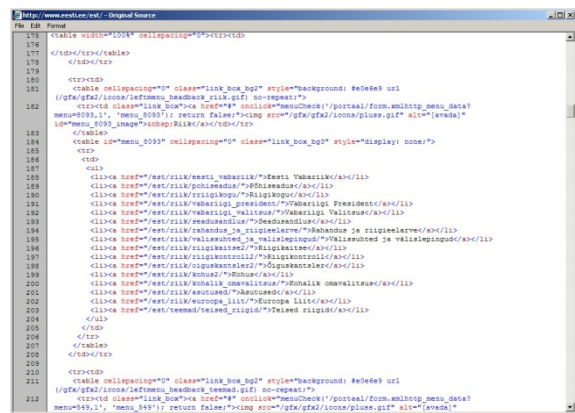
HTML

HTML on lahtiseletatult hüperteksti märkimise keel (*Hypertext Markup Language*) ehk keel, millega märgitakse üles hüperteksti dokumendid. Hüpertekst, kus võtmesõnad klõpsates pöörduakse nende sõnadega seotud dokumentide juurde, on tegelikult veebileht oma esialgses, kitsas tähenduses. Seega on HTML keel veebilehtede ülesmärkimiseks (*markup*), loomiseks.

Kui vaatame mõnda veebilehte, siis paistab see reeglina ilus värviline, sisaldab pilte, videoid jne. Tegelikult aga koosneb veebileht HTML dokumendist, milles on märgitud tekst mida näidata, milliseid pilte, videoid jms näidata, kuidas ja kus. HTML dokument on sisuliselt tekstifail, milles sisu ja kujundus vastavate märgendite (*tag*) abil kirjeldatakse. Veebilehitseja interpreteerib etteantud HTML dokumenti ja kuvab soovitud välimusega veebilehe (Joonis 1.) Kasutaja võib vaadata ka lehe lähtekoodi (*source*) (Joonis 2).



Joonis 1 veebileht eesti.ee



Joonis 2 osa veebilehe eesti.ee lähtekoodist

HTML keelt teab maailm juba 1990. aasta detsembrist, kui sündis WWW.

Vanim HTML dokument, mille praegu veebist leida võib, pärineb 13. novembrist 1990. See leht sisaldab ka täiesti töötavat linki. Lehe leiab aadressil: <http://www.w3.org/History/19921103-hypertext/hypertext/WWW/Link.html>

Aastate jooksul on tehnoloogiate arenedes arenenud ka veebilehtede võimalused, esialgsele tekstile ja piltidele on lisandunud videod, heli, animatsioonid, voogmeedia, erinevad skriptimiskeeled, mitmed interaktiivsed tehnoloogiad nagu Flash ning nüüd ka Microsoft Silverlight. Kuigi tänapäeval võib veebileht koosneda vaid näiteks Silverlight rakendusest, on vahendiks, millega sisu veebilehitsejale edastatakse ning sobivalt ekraanile paigutatakse, ikka HTML või siis selle edasiarendus XHTML. Koos lisandunud võimalustega on arenenud ka HTML keel ise ning sellest on aastate jooksul ilmunud mitu erinevat versiooni.

- Esimene ametlik standard – HTML 2.0 kuulutati välja 1995. aasta 24 novembril.
- 1997. aasta jaanuaris kuulutati välja HTML 3.2, mis lisas hulga kujundamisega seotud elemente ja atribuute.
- 1997. detsembris kuulutati välja HTML 4.0, mis võimaldas kujunduse juba täielikult veebilehe sisust eraldada.

Praegu kasutatakse peamiselt 1999. aastal välja kuulutatud versiooni 4.01 kuid tulemas on juba ka versioon 5.

WWW algaastatel toimus veebilehtede loomine ehk HTML dokumentide kirjutamine käsitsi, tekstiredaktorit kasutades märgendeid ja teksti kirjutades. Sellise ebamugava olukorra lahenduseks on loodud hulk veebitoimetamisprogramme, mis võimaldavad veebilehte tavalise tekstidokumendi kombel kirjutada ning kujundada ja loovad automaatselt vastava HTML dokumendi. Selliste programmide hulka kuuluvad näiteks Microsoft Frontpage, Adobe Dreamweaver jpt. Luues veebirakendusi uusi tehnoloogiaid kasutades, näiteks MS Silverlight või Adobe Flash, luuakse koos rakenduse failidega ka vajalik HTML dokument ilma, et autor sellest suurt midagi teaks.

Kuna autoril võib olla vajadus või soov muuta viisi, kuidas loodud rakendust veebilehitsejas näidatakse, lisada rakendus mõnele olemasolevale veebilehele vms, siis on HTML põhiteadmised veebidisaineritele endiselt vajalikud!

Dokumendi tüübid

Veebilehe loomisel on hea (HTML) kuid sageli lausa kohustuslik (XHTML) lehe algusesse lisada dokumendi tüübi deklaratsioon. Isegi lihtsa HTML koodi puhul võib see aidata soovitud tulemust saavutada.

Suure osa autorite jaoks on olemas justkui lihtsalt HTML kuid tegelikult on olemas mitu erinevat versiooni, praegu on põhiliselt kasutusel versioon 4.01.

Selle versiooni puhul on olemas sellised tüübid nagu: HTML 4.01 Strict, HTML 4.01 Transitional, XHTML 1.0 Strict jne. Kõiki neid tüüpe kirjeldavad W3C spetsifikatsioonid ja standardid ning nad määravad HTML dokumendi legaalsed elemendid, atribuudid ja struktuuri.

Dokumenditüüpide definitsiooni tähiseks on DTD (*Document Type Definition*).

Brauser peab teadma, millise tüübiga on tegemist, selleks lisataksegi veebilehe algusesse vastav deklaratsioon, näiteks:

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

või:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Uusima, HTML5 puhul on vaid üks DTD ja see on väga lihtsa kujuga:

```
<!DOCTYPE HTML>
```

Dokumendi tüüp on oluline selleks, et brauser suudaks veebilehte korralikult ekraanile tuua, elemente tõlgendada, nii toimub lehe avamine ka kiiremini.

Dokumendi tüübi deklareerimine on vajalik ka veebilehe koodi süntaksi kontrollimiseks HTML validaatori abil (<http://validator.w3.org/>).

Loetelu soovitatavate dokumenditüüpidega leiab aadressilt:
<http://www.w3.org/QA/2002/04/valid-dtd-list.html>

HTML õigekiri

Veebilehte kirjeldab HTML dokument, mille faililaiendiks ongi html, näiteks tavaline veebilehestiku pealeht kannab enamasti nimetust: **index.html**

NB! Käesolevas materjalis on püütud järgida XHTML 1.0 transitional standardit!

HTML dokument koosneb elementidest. Elementidel on reeglina algusmärgend (*tag*) ja lõpumärgend, millede vahele jääb elemendi sisu. Lõpumärgendi tunnuseks on kaldkriips elemendi nime ees.

```
<elemendinimi> Sisu </elemendinimi>
```

On olemas ka nn tühjad elemendid (*empty elements*), millel pole eraldi lõpumärgendit!

```
<elemendinimi>
```

NB! XHTML puhul peab selliste elementide märgendi lõppu lisama lõpetava kaldkriipsu!

```
<elemendinimi />
```

Mitmetel elementidel on lisaks nimele olemas veel kohustuslikke ja/või valikulisi atribuute. Atribuudid lisatakse elemendi algusmärgendisse kujul:

```
nimi="väärtus"
```

Nii võib mõni element olla järgmisel kujul:

```
<elemendinimi atribuut1="väärtus" atribuut2="väärtus" ...> Sisu </elemendinimi>
```

Üks tavaline atribuut, mida saab ära kasutada ka CSS stiililehtede rakendamisel, on identifikaator ehk id, näiteks:

```
<elemendinimi id="eriline"> Sisu </elemendinimi>
```

HTML dokumendi kirjutamisel ei pea kasutama reavahetust (*line break*) sest veebilehitseja interpreteerib kõike elementide kaupa! Inimesele on muidugi lihtsam, kui elemendid on paigutatud eraldi ridadele!

Elemendid võivad üksteisele järgneda kuid võivad olla lisatud ka üksteise sisse ehk pesastatud (*nested*), näiteks:

```
<esimene_element> <teine_element>Sisu</teine_element></esimene_element>
```

Osa elemente on blokielemendid (*block*) – hõivavad veebilehel kogu rea, ei luba teisi elemente enda kõrvale ja osa rea-elementid (*inline*). Blokielementide hulka kuuluvad näiteks pealkirjad (<h1>...<h6>), tekstilõigud (<p>) ja alajaotused (<div>). Blokielemente ei kirjutata reaelementide sisse!

Veebilehitsejad interpreteerivad HTML dokumenti lugemise järjekorras ning kõik elemendid paigutatakse reeglina üksteise järele nii nagu nad on HTML failis järjestatud.

Autoritel soovitatakse oma HTML koodi ka kommenteerida, et seda lihtsam lugeda oleks. Kommentaar pannakse kirja elemendiga <!-- ...-->. Näiteks:

```
<!--Kommentaari tekst, mida veebilehitseja veebilehel ei näita-->
```

HTML ja/või XHTML

Alates aastast 2000 on HTML kõrval olemas XHTML (*Extensible HyperText Markup Language*), mis lihtsalt öeldes on XML (*Extensible Markup Language*) keeles realiseeritud HTML.

XML on keel, kus kõik peab olema korrektselt ülesmääratud, tulemuseks on hästi formuleeritud dokumendid. XML on loodud andmete kirjeldamiseks (*describe*), HTML aga andmete kuvamiseks/näitamiseks (*display*).

Erinevad XHTML versioonid

Ka XHTML-ist on olemas mitu erinevat versiooni

XHTML 1.0

Esimene W3C soovituslik XHTML versioon oli lihtsalt HTML 4.01 lahendatuna XML-is. Olemas kolm DTD-d, mis on vastavuses HTML 4.01 omadega.

- **XHTML 1.0 Strict** on sama mis HTML 4.01 Strict, kuid kasutab XML-i süntaksit.
- **XHTML 1.0 Transitional** on sama mis HTML 4.01 Transitional, kuid kasutab XML-i süntaksit. Selles sisaldub kogu XHTML 1.0 Strict. Lisaks lubab kasutada ka presentatsiooniga seotud elemente ja atribuute. See lihtsustab ületulekut varasematelt HTML-i versioonidelt.
- **XHTML 1.0 Frameset** on sama mis HTML 4.01 Frameset, kuid kasutab XML-i süntaksit. Võimaldab veebilehekülje tegemisel kasutada nn raamisid (*frame*).

XHTML 1.1

Viimane W3C soovituslik XHTML-i versioon on mooduli põhine. Sisuliselt on tegemist parandatud XHTML 1.0-ga. Eemaldatud on kõik presentatsiooniga seotud elemendid, ka need, mis olid lubatud XHTML 1.0-es. Presentatsiooniks kasutatakse vaid CSS-i.

XHTML 2.0 tööversioon

2008 aasta jaanuari seisuga ei ole XHTML 2.0 spetsifikatsioon veel lõplikult valmis. Praegu töös olev variant elimineeriks täielikult tagasiühilduvuse. Sellisel juhul poleks tegemist enam XHTML keele uue versiooniga, vaid juba täiesti uue keelega, mis võimaldaks vältida praegu kasutatavate (X)HTML-i versioonide piiranguid.

XHTML ja HTML erinevused

XHTML pole väga erinev HTML versioonist 4.01!

Kõige olulisem on meeles pidada, et:

- XHTML dokumendil peab olema nn juurelement (*root element*)!
- XHTML elemendid pannakse alati kirja väiketähtedega! Atribuutide nimed peavad olema väiketähtedega!
- XHTML elemendid peavad olema korrektselt pesastatud (*nested*)!
Näiteks HTML lubab isegi sellist ülesehitust:

`<b><i>Siin on paks ja kaldu tekst</b></i>`

Aga XHTML nõuab korrektset pesasamist:

`<b><i>Siin on paks ja kaldu tekst</i></b>`

- XHTML elemendid peavad alati olema suletud (vajavad lõpumärgendit või sulgevat kaldkriipsu märgendi lõpus)!
Näiteks HTML lubab selliseid elemente:

`<p>See on lõik!`

```
<br>
<hr>

```

Aga XHTML nõuab korrektset lõpetamist:

```
<p>See on lõik!</p>
<br />
<hr />

```

- XHTML elemendid peavad olema korrektselt pesastatud (*nested*)! Näiteks HTML lubab isegi sellist ülesehitust:

```
<b><i>Siin on paks ja kaldu tekst</b></i>
```

Aga XHTML nõuab korrektset pesasamist:

```
<b><i>Siin on paks ja kaldu tekst</i></b>
```

- Atribuutide väärtused peavad XHTML puhul olema jutumärkides!
- XHTML-is on keelatud atribuutide „lühendatult“ kirja panemine! Näiteks HTML lubab isegi sellist ülesehitust:

```
<option selected>
```

Aga XHTML nõuab korrektset pesastamist:

```
<option selected="selected">
```

- Keele atribuudi lang kasutamisel XHTML-is tuleb lisada ka xml:lang atribuut! NB! Keele atribuuti saab kasutada pea-aegu kõigil XHTML elementidel! Näiteks:

```
<div lang="et" xml:lang="et">Ja nii ongi see lugu!</div>
```

- XHTML dokumendil peab olema dokumendi tüübi deklaratsioon (DTD), <html>, <head>, <title> ja <body> elemendid!
- Näiteks peab miinimumnõuetele vastav XHTML dokument olema selline:

```
<!DOCTYPE Siia tuleb DTD>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Siia tuleb veebilehe pealkiri, mis läheb brauseri tiitliribale</title>

</head>
<body>

</body>
</html>
```

HTML elemendid

Ülevaate kõigist HTML ja XHTML elementidest ning nende kehtivusest erinevate dokumenditüüpide puhul saab: http://www.w3schools.com/tags/ref_html_dtd.asp

HTML dokumendi esimeseks elemendiks on reeglina dokumendi tüübi deklaratsioon (DTD). Näiteks:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

NB! Peagi kasutuselevõetava HTML 5 puhul pole see deklaratsioon enam vajalik!

Järgmiseks elemendiks on HTML ise, mille sisuks ongi kogu veebileht:

```
<html> Kogu veebileht </html>
```

NB! Sellel elemendil tuleb atribuudina kirja panna ka viide dokumendi tüübi kirjeldusele, näiteks:

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

Veebileht jaguneb kaheks oluliseks üksteisele järgnevas osaks, <head> – päis ja <body> – keha. Päises on elemendid, mida veebilehe vaataja reeglina otseselt ei näe (viide lehe kujundust määravale stiililehele, võtmesõnad otsingumootoritele, kasutatav tekstikodeering jms). Kehaosas on kõik veebilehe sisuna nähtavad elemendid.

Tüüpiline veebilehe ülesehitus on seega järgmine:

```
<DTD ehk dokumendi tüübi deklaratsioon>
<!--siit algab html kood-->
<html dokumenditüübikirjelduseviide>
<!--siit algab html dokumendi päis-->
<head>
  päise sisu
</head>
<!--siit algab html dokumendi sisu-->
<body>
  veebilehe sisu
</body>
</html>
```

Suurem osa veebilehe elemente (tekstid, tabelid jms) võimaldab ka kujundamist.

Põhimõtteliselt saab kujundamiseks vajalikke atribuute kirja panna elemendi algusmärgendisse kuid see võimalus on tänapäeval vähem kasutusel. Praegu kasutatakse kujundamiseks peamiselt eraldi stiililehte (css), mis lingitakse HTML dokumendiga!

HTML dokumendi keha osa elemendid

Veebilehe nähtavaks sisuks on kõik see, mis HTML dokumendis kirjutatud elemendi <body> sisse.

Pealkirjad

Nagu tavalistes tekstides, ajalehtedes ja mujal, on ka veebilehel pealkirjad (*heading*). Pealkirjadel on teksti liigenduse eesmärgil kasutada 6 taset ning nende kirjeldamiseks kasutatakse märgendeid <h1> ... <h6> (tulenevad inglisekeelsest sõnast *heading*). Näiteks:

```
<h1>Veebilehe suur pealkiri</h1>
```

Pealkirjad on blokielemendid (*block*)!

Tekst

Tavalist teksti saab HTML dokumenti ka lihtsalt teiste elementide vahele kirjutada kuid selleks, et seda ka korralikult kujundada saaks (css stiililehtede abil), kasutatakse lõigu elementi <p> (tuleneb inglisekeelsest sõnast *paragraph*):

```
<p>Lõigu jagu teksti</p>
```

Lõigu element on blokielement (*block*)!

Vajadusel saab lisada manuaalse reavahetuse (*manual line break*), mille puhul alustatakse uut rida kuid jätkub sama lõik. Selleks kasutatakse märgendit `<br />` (inglisekeelsest sõnast *break*), millel pole eraldi algus- ja lõpumärgendit. Näiteks:

tekst ühel real `<br />` tekst jätkub uuel real

Lõigu sees mingite fraaside erinevaks kujundamiseks tarvitatakse elementi `<span>`, mis ise visuaalselt midagi ei muuda kuid mille jaoks saab luua css stiile. Näiteks:

`<p>Tekstilõik, milles <span>eriline fraas</span> on eraldi välja toodud</p>`

Hüperlink

Hüperlink ehk lihtsamalt link, ongi see, millest sai alguse hüperteksti ehk veebi idee. Lingi lisamiseks HTML dokumenti kasutatakse elementi `<a>` (inglisekeelsest sõnast *anchor*), mille algus- ja lõpumärgendi vahele paigutatakse sisu, millel klõpsamisel link töötab. Vaikimisi on link allajoonitud, külastamata link sinist värvi, külastatud link lillakat ning aktiivne link punast värvi.

Hüperlingi loomiseks tuleb elemendile `<a>` lisada järgmine atribuut:

- *href* – lingitava materjali URL (aadress), teine veebileht, pilt, pdf fail vms;

Näiteks lisab järgnev rida veebilehele teksti "Tallinna Ülikool" millel klõpsates avatakse TLÜ koduleht:

`<a href="http://www.tlu.ee">Tallinna Ülikool</a>`

Lingi saab lisada ka teistele objektidele, näiteks pildile! Näiteks järgmine rida lisab veebilehele pildi, millel klõpsates avatakse TLÜ koduleht:

`<a href="http://www.tlu.ee"> </a>`

Hüperlingile võib lisada atribuudi:

- *target* – võimaldab määrata akent, kus link avatakse. Kasutada saab järgmiseid väärtusi:
 - *_blank* – link avaneb uues aknas (uuel vahekaardil (*tab*));
 - *_self* – link avaneb samas aknas, kus klõpsati;
 - *_parent* – link avaneb „ema“ raamistikus (*parent frameset*);
 - *_top* – link avaneb terves aknas (kui varem oli näiteks raamistiku raamis);
 - raaminimi (*frame name*) – link avatakse nimetatud raamis.

Näiteks lisab järgmine rida veebilehele lingi, mis avab TLÜ kodulehe uues aknas:

`<a href="http://www.tlu.ee" target="_blank">Tallinna Ülikool</a>`

Linke saab lisada ka ühe veebilehe piires selle erinevatele osadele! Selleks tuleb HTML dokumendis kõigepealt määrata nn järjehoidja (*bookmark*) ning seejärel saabki talle lingiga viidata!

Järjehoidja lisamiseks kasutatakse elementi `<a>` koos atribuudiga *name*:

`<a name="osa2"></a>`

Seejärel saab luua lingi, mis sellele järjehoidjale viitab:

`<a href="#osa2">Vaata ka osa 2</a>`

Teise HTML dokumendi kindlale järjekohajale viitamiseks tuleb tema aadressile lisada järjekohaja nimi! Näiteks:

```
<a href="http://www.seejasee.ee/#osa2">Vaata ka selle ja selle asja osa 2</a>
<a href="http://www.seejasee.ee/info.html#osa2">Vaata ka selle ja selle asja info osa 2</a>
```

Järjekohajate asemel võib viidata mistahes id atribuudiga elemendile veebilehel! Näiteks määrame veebilehe ühele pealkirjale nime „osa3“:

```
<h2 id="osa3">Veebilehe kolmas osa</h2>
```

Seejärel saab luua lingi, mis sellele pealkirjale viitab:

```
<a href="#osa2">Vaata ka kolmandat osa!</a>
```

Vaikeväärtused hüperlinkidele

Juhul kui suur osa linke veebilehel viitavad samale domeenile või peavad avanema ühtmoodi aknas (target), siis saab määrata vaikimisi kasutatavaid väärtuseid. Selleks kasutatakse elementi <base>, mis lisatakse kindlasti veebilehe päisesse (<head>).

HTML puhul on tegemist tühja (empty) elemendiga, millel puudub lõpumärgend. XHTML puhul peab <base> element kindlasti korralikult suletud!

Näiteks:

```
<base href="http://www.seejasee.ee/" />
<base target="_blank" />
```

Selle elemendi kasutamisel võib veebilehele lisatud linkides, mis viitavad vaikimisi määratud aadressile, kasutada lihtsalt relatiivseid aadresse selle suhtes. Kõikidel linkide puhul, millele veebilehel eraldi target atribuuti ei määrata, kasutatakse vaikimisi määratud väärtust!

Elektronposti link

Elementi <a> saab kasutada ka veebilehele elektronposti linkide lisamiseks, millel klõpsates saab ettenatud aadressil elektronposti saata.

NB! Elektronpostiaadresside sellisel moel veebilehtede lisamine pole hea tava ning annab võimaluse mitmesugusteks rünnakuteks, näiteks on see kättesaadav rämpsposti levitajatele!

Elektronposti lingi ehk *mailto link*’i lisamiseks tuleb elemendi <a> href atribuudi väärtuseks kirjutada mailto:elektronpostiaadress! Näiteks:

```
<a href="mailto:keegi@seal.ee">Saada e-kiri</a>
```

Viisakas oleks lisada kohe ka kirja teema (*subject*), selleks lisatakse href atribuudi väärtusele küsimärk ja teema nagu järgnevas näites:

```
<a href="mailto:keegi@seal.ee?Subject=Palun%20lugege%20seda!">Saada e-kiri</a>
```

NB! Siinkohal on tühikud teemas asendatud sümbolite kombinatsiooniga „%20“, mis tagab, et veebilehitseja teksti korrektselt näitab!

Sarnaselt saab lisada aadresse, kellele saata kirjast koopia (*cc* ehk *carbon copy*), pimekoopia (*bcc* ehk *blind carbon copy*) ja isegi vaikimisi kasutatav kirja sisu (*body*)!

NB! Siinjuures tuleb järgnevad lisad eraldada juba mitte küsimärgi vaid & märgiga!

```
<a href="mailto:keegi@seal.ee?cc=teine@siin.ee&bcc=kolmas@mujal.ee
&Subject=Palun%20lugege%20seda!&body=Lugemisostkust tuleb edendada!">
Saada e-kiri</a>
```

Nagu eespool mainitud, pole elektronpostiaadressi avalikult veebilehel hoidmine hea praktika. Seetõttu üritatakse seda igati moonutada ning ühe võttena võib @ märgi asendada mingi muu sümboli või sümbolite kombinatsiooniga, näiteks:

```
<a href="mailto:keegi[at]seal.ee">Saada e-kiri</a>
```

Loendid

Loendid (*lists*) võivad veebilehel olla nagu tavapärares tekstideski nummerdatud või täpploendid (*bulleted*).

Mõlemat tüüpi loendi sisuks on loendiüksuse elemendid `<li>` (inglisekeelsest mõistest *list element*).

Nummerdatud loend

Nummerdatud loendi jaoks on kasutusel element `<ol>` (tuleneb inglisekeelsest mõistest *ordered list*).

Näiteks nummerdatud loendi kirjeldus ja selle tulemus:

<code></code>	
<code>Esimene loendi element</code>	1. Esimene loendi element
<code>Teine loendi element</code>	2. Teine loendi element
<code>Kolmas loendi element</code>	3. Kolmas loendi element
<code></code>	

Täpploend

Täpploendi jaoks kasutatakse elementi `<ul>` (tuleneb inglisekeelsest mõistest *unordered list*).

Näiteks täpploendi kirjeldus ja selle tulemus:

<code></code>	
<code>Esimene loendi element</code>	• Esimene loendi element
<code>Teine loendi element</code>	• Teine loendi element
<code>Kolmas loendi element</code>	• Kolmas loendi element
<code></code>	

Mitmetasemeline loend

Loomulikult on võimalik luua ka mitmetasemelisi loendeid ning seejuures täpp- ja nummerdatud loendeid omavahel kombineerida. Näiteks liigendatud loend, milles esimene tase on nummerdatud, teine tase täpploend:

<code></code>	
<code>Esimene loendi element</code>	1. Esimene loendi element
<code></code>	
<code>Esimene alamelement</code>	o Esimene alamelement
<code>Teine alamelement</code>	o Teine alamelement
<code></code>	
<code></code>	
<code>Teine loendi element</code>	2. Teine loendi element
<code></code>	
<code>Esimene alamelement</code>	o Esimene alamelement
<code>Teine alamelement</code>	o Teine alamelement
<code></code>	
<code></code>	
	3. Kolmas loendi element

```
</li>Kolmas loendi element</li>
</ol>
```

NB! Sellise mitmetasemelise (kombineeritud loetelu puhul on alamelemente sisaldav loetelu kõrgema taseme loetelu elemendi osaks!

Definitsioonide loend

Kui on tarvis loetleda terve hulk mõisteid/definitsioone, siis selleks kasutatakse spetsiaalset loendielementi `<dl>` (*Definition List*).

Selle loendi elemendid defineeritakse elementidega `<dt>` (*Term*) ja nende selgitused – kirjeldused elementidega `<dd>` (*Description*).

Näiteks:

```
<dl>
  <dt>Arvuti</dt>
    <dd>Masin, mis pakub meelelahutust, infot ja paljut muud</dd>
  <dt>Monitor</dt>
    <dd>Muudab arvutis toimuva nähtavaks</dd>
  <dt>Printer</dt>
    <dd>Kirjutab arvutis toimuva paberile</dd>
</dl>
```

```
Arvuti
  Masin, mis pakub meelelahutust, infot ja paljut muud
Monitor
  Muudab arvutis toimuva nähtavaks
Printer
  Kirjutab arvutis toimuva paberile
```

Joonis 3 Definitsioonide loend veebilehel

Teksti vormindamine

Teksti vormindamiseks suuremate plokkide kaupa (lõik jne) kasutatakse küll stiililehte (css) aga näiteks üksikute fraaside rõhutamiseks jms võib erijuhtudel siiski HTML standard-elemente kasutada.

NB! XHTML 1.0 Strict puhul on järgnevad vahendid keelatud! Kasutada tuleb css-i!

- Paks tekst (*bold*), element `<b>`, näiteks:
tavatekst `<b>paks tekst</b>` tavatekst jätkub
- Kaldkiri (*italic*), element `<i>`, näiteks:
tavatekst `<i>kaldkiri</i>` tavatekst jätkub
- Rõhutatud kiri, element `<em>`, näiteks:
tavatekst `<em>kaldkiri</em>` tavatekst jätkub
- Eriti tugevalt rõhutatud tekst, element `<strong>`, näiteks:
tavatekst `<strong>rõhutatud tekst</strong>` tavatekst jätkub
- Suurendatud tekst, element `<big>`, näiteks:
tavatekst `<big>suurem kiri</big>` tavatekst jätkub

NB! Elementi `<big>` HTML 5 ei toeta!

- Väiksem tekst, element `<small>`, näiteks:

tavatekst <small>väiksem tekst</small> tavatekst jätkub

- Ülaindeks (*superscript*), element <sup>, näiteks:
tavatekst ^{ülaindeks} tavatekst jätkub
- Alaindeks (*subscript*), element <sub>, näiteks:
tavatekst _{alaindeks} tavatekst jätkub
- "Vahelelisatud" (*inserted*) tekst, element <ins>, tavaliselt allajoonitud, näiteks:
tavatekst <ins>alaindeks</ins> tavatekst jätkub
- "Kustutatud" (*deleted*), element , näiteks:
tavatekst alaindeks tavatekst jätkub
- Teksti suund, element <bdo>, millel on argument dir (väärtused ltr – vasakult paremale või rtl – paremalt vasakule), näiteks:
<bdo dir="rtl">See on tagurpidine tekst!</bdo>

NB! Element <bdo> võib ise olla mõne plokielemendi sees (näiteks lõigu <p>) aga mitte vastupidi!

Neid kujunduselemente võib kasutada ka näiteks eelpool kirjeldatud pealkirjade ning loetelude puhul, tabelites, hüperlinkides ning ka kombineeritult (üksteise sees). Näiteks:

<h1><big><i>Ülimalt</i> rõhutatud eriline</big> pealkiri</h1>

Mis annab tulemuseks:

***Ülimalt* rõhutatud eriline** pealkiri

Joonis 4 Kujundatud tekst

- Eelnevalt vormindatud tekst, kogu tekst esitatakse samasugusel kujul nagu see on algtekstis paika pandud (tühikud, reavahetused jms säilivad), element <pre>, näiteks:
<pre>Siin tekstis on vaja
säilitada reavahetus ja see pikem tühikutega toksitud vahe!</pre>

Lisaks on kasutada eeldefineeritud stiilid arvutivaldkonnaga seotud tekstilõikude kujundamiseks ning üldiste tsitaatide ja definitsioonide jaoks.

Arvuti stiilid

Arvuti stiilid (*computer output*) on loodud silmas pidades vajadust veebilehel näidata programmikoodi jms.

- Programmide koodilõigud (tavalise fondiga), element <code>, näiteks:
<code>x:=y*Math.random();</code>
- Koodinäites muutujate kirjanepuudeks (tavaliselt kaldkirjas), element <var>, näiteks:
<code><var>x</var>:=<var>y</var>*Math.random();</code>
- Koodinäite stiil (tavaline font), element <samp>, näiteks:
<samp>x:=y*Math.random();</samp>
- Klaviatuuri tekst (tavalise fondiga), element <kbd>, näiteks:
<kbd>Sisestame klaviatuurilt vajalikud väärtused</kbd>

- Teletype ehk kirjutusmasina tekst (*typewriter*, *monotype* fondiga), element `<tt>`, näiteks:

`<tt>Vanasti oli kontorites hulk kirjutusmasinaid</tt>`

NB! Elementi `<tt>` HTML 5 ei toeta!

Tsitaadid, definitsioonid

Tsitaatide ja lühendite vormindamiseks on samuti loodud hulk eeldefineeritud stiile.

- Lühend, veebilehel näidatakse lühendit, hiirega sellele liikumisel tuuakse nähtavale pikk variant, element `<abbr>`, näiteks:

`<abbr title="Tallinna Ülikool">TLU</abbr>`



Joonis 5 Lühend (`abbr`) veebilehel

- Akronüüm, sarnane lühendiga, element `<acronym>`, näiteks:

`<acronym title="World Wide Web">WWW</acronym>`

NB! Elementi `<acronym>` HTML 5 ei toeta!

- Aadress (tavaliselt kaldkirjas), dokumendi autori/omaniku kontaktinfo, element `<address>`, näiteks:

`<address>Tallinn, Narva mnt 25</address>`

NB! Element `<address>` on plokielement ja ei tohi asuda teise plokielemendi (näiteks lõigu `<p>`) sees!

- Pikk tsitaat mille ette ja taha jäetakse tühja ruumi, element `<blockquote>`, näiteks:
`<blockquote>Õppida, õppida, õppida!</blockquote>`
- Lühike tsitaat, brauser lisab jutumärgid, element `<q>`, näiteks:
`<q>Õppida!</q>`
- Tsitaat (tekst kursiivis), element `<cite>`, näiteks:
`<cite>Vähem on rohkem!</cite>`
- Definitsioon (tekst esitatakse tavaliselt kaldkirjas), element `<dfn>`, näiteks:
`<dfn>internet on ülemaailmne arvutivõrke ühendav võrk</dfn>`

Pilt veebilehel

Veebilehel saab kasutada peamiselt rastergraafikat, toetatakse jpg, png ja gif failivorminguid sest need on piisavalt universaalsed (toetatud erinevate arvutiplatvormide poolt) ning kasutavad kompressiooni (failimahud on väikesed).

Pildi lisamiseks on `<img>` element, millel pole eraldi lõpumärgendit ja millel on kaks kohustuslikku atribuuti:

- `src` – viide pildifailile (inglisekeelsest sõnast *source*), mille sisuks on soovitud pildifaili aadress ehk URL;

- *alt* – alternatiivtekst (inglisekeelsest mõistest *alternative text*), mille sisuks on pildi lühidalt kirjeldav tekst näiteks nägemispuuetega inimestele ekraani sisu ettelugeva tarkvara jaoks. Seda teksti näidatakse veebilehel ka siis, kui mingil tehnilisel põhjusel ei õnnestu pilti näidata.

Pildi element võib seega välja näha selline:

```

```

Lisaks võib kasutada atribuute pildi suuruse määramiseks kuigi soovitatav on mõnda fototöötlusprogrammi kasutades pildid veebilehe jaoks vajalikku mõõtu seada!

- *width* – laius, väärtuseks täpne arv pikselites või protsentides.
- *height* – kõrgus, väärtuseks täpne arv pikselites või protsentides.

NB! Väärtus protsentides määrab pildi suuruse teda sisaldava elemendi suuruse suhtes!

Näiteks:

```

```

või

```

```

Interaktiivne pilt

Interaktiivne pilt on pilt veebilehel, millel on määratud klõpsatavad, linkidega varustatud piirkonnad.

Interaktiivse pildi loomiseks luuakse element `<map>`, mille sees defineeritakse mingi pildi jaoks piirkonnad, millel saab klõpsata. Nende piirkondade määramisel kasutatakse koordinaatidena kaugust pildi ülemisest vasakust nurgast pikselites. Nende koordinaatide kindlaksmääramisel saab kasutada mistahes graafikaprogrammi abi (näiteks ka MS Paint).

Elemendil `<map>` on kohustuslikud atribuudid:

- *name* – loodava „kaardi“ nimi, mille abil see vastava pildi elemendiga (`<img>`) seotakse!
- *id* – loodava „kaardi“ identifikaator. Selle väärtuseks võib kirjutada sama, mis nimeatribuudile. Ta on vajalik veebilehe korrektseks valideerimiseks.

Näiteks:

```
<map name="eestilinnad" id="eestilinnad">
...
</map>
```

Loodud „kaardi“ sidumiseks vastava pildiga tuleb elemendile `<img>` lisada atribuut `usemap`!

Näiteks:

```

```

Klõpsatavad piirkonnad luuakse elemendi `<area>` abil. Sellel elemendil on järgmised atribuudid:

- *shape* – piirkonna kuju, mis võib olla:
 - *rect* – ristkülik, selle koordinaatidena tuleb kirja panna diagonaali otspunktid (ülemise vasaku ja seejärel alumise parema nurga koordinaadid);
 - *circle* – ring, selle koordinaatidena tuleb kirja panna kolm arvu: keskpunkti koordinaadid ja raadius;
 - *poly* – hulknurk, kirja tuleb panna kõikide soovitud tippude koordinaadid;

- *alt* – alternatiivtekst (kohustuslik);
- *href* – viide veebilehele (või muule ressursile), kuhu satutakse vastaval piirkonnal klõpsates;
- *target* – millises aknas href atribuudiga määratud ressurss avatakse;
- *coords* – koordinaadid pikselite arvudena, komadega eraldatuna järjest vajalike punktide x ja y koordinaadid (pikselites).

Näiteks:

```
<map name="eestilinnad" id="eestilinnad">
  <area shape="rect" coords="197,57,244,79" alt="Tallinn" href="http://www.tallinn.ee"
  target="_blank" />
  <area shape="circle" coords="220,204,5" alt="Pärnu" href="http://www.parnu.ee"
  target="_blank" />
  <area shape="circle" coords="291,206,5" alt="Viljandi" href="http://www.viljandi.ee"
  target="_blank" />
  <area shape="circle" coords="363,204,8" alt="Tartu" href="http://www.tartu.ee"
  target="_blank" />
</map>
```

NB! Kui pilti, millele „kaart“ lisatakse, ei näidata originaalsuuruses vaid väiksemana, siis peab ka koordinaadid samas proportsioonis ümber arvutama!

Tabel

Tabelit kasutatakse veebilehtedel ka tavapäraselt andmete korrastatult esitamiseks aga enamasti hoopis kujundamise eesmärgil. Tabeli abil on väga hõlbus objekte veebilehel korrapäraselt ning soovitud kohtadele paigutada.

Tabel luuakse elemendi `<table>` abil ning tema sees kirjeldatakse ära read, veerud ning nende sisu:

```
<table>
tabeli sisu
</table>
```

Rea loob tabeli sisse element `<tr>`. Näiteks kahe reaga tabel:

```
<table>
<tr>
esimese rea sisu
</tr>
<tr>
teise rea sisu
</tr>
</table>
```

Lahtrid luuakse ridade sisse elemendiga `<td>`. Näiteks tabel kahe reaga, milles kummaski kaks lahtrit (ehk veergu):

```
<table>
  <tr>
    <td>esimese rea, vasaku lahtri sisu</td>
    <td>esimese rea, parema lahtri sisu</td>
  </tr>
  <tr>
    <td>teise rea, vasaku lahtri sisu</td>
    <td>teise rea, parema lahtri sisu</td>
  </tr>
</table>
```

Kasutada saab ka eraldi elementi `<th>` tabeli veergude pealkirjade jaoks (*table heading*). Elementi `<th>` kasutatakse tabeli esimesel real (element `<tr>`) täpselt nagu elementi `<td>` ning neid saab ka samamoodi kujundada (vt. allpool).

```
<table>
  <tr>
    <th>esimese veeru päis</th>
    <th>teise veeru päis</th>
  </tr>
  <tr>
    <td>esimese rea, vasaku lahtri sisu</td>
    <td>esimese rea, parema lahtri sisu</td>
  </tr>
  <tr>
    <td>teise rea, vasaku lahtri sisu</td>
    <td>teise rea, parema lahtri sisu</td>
  </tr>
</table>
```

NB! Mitmed veebilehitsejad ei suuda korrektselt näidata tabelis tühjaks jäetud lahtreid! Seetõttu soovitatakse lahtri täiesti tühjaks jätmise asemel sinna lisada "kõva tühik" ` ` (*non-breaking space*)!

Huvitav võimalus on kogu tabel jagada kolmeks põhiosaks: päis (*head*), jalus (*footer*) ja keha (*body*), selleks kasutatakse elemente `<thead>`, `<tfoot>` ja `<tbody>`, millede sisse lisatakse vastava osa read jms. Tabeli päis paigutatakse automaatselt tabeli ülemisse serva ja jalus automaatselt tabeli allaserva isegi, kui nad on vales järjekorras kirja pandud.

Tabelile saab lisada ka pealkirja (*caption*), milleks lisatakse tabeli algusmärgendi järele, esimese rea ette element `<caption>`.

```
<table>
<caption>Väga oluline tabel</caption>
<tr>
  <th>esimese veeru päis</th>
  ...
```

Tabeli veergusid võib ka grupeerida, et neid siis koos kujundada või näiteks nende gruppide vahele ääriseid paigutada! Veergude grupi loomiseks kasutatakse elementi `<colgroup>`. Element `<colgroup>` lisatakse tabeli sisse ja tema laiuse määramiseks (mitu veergu gruppi kuulub) kasutatakse atribuuti `span`.

Näiteks:

```
<table>
<colgroup span="3"></colgroup>
<colgroup span="2"></colgroup>
<tr>
  ...
```

Tabeli, ridade ja lahtrite vormindamine

Tabeli vormindamiseks saab kasutada järgmiseid atribuute, mis kirjutatakse tabeli algusmärgendisse:

- *width* – laius, määrab, kui suure osa veebilehitseja aknas tabel enda alla võtab, väärtuseks täpne arv pikselites või protsentides;

- *border* – ääris ehk raamjoon, määrab tabelit ja tema lahtreid ümbritseva joone paksuse, väärtuseks täpne arv pikselites, 0 tähendab äärise puudumist ning see on ka vaikeväärtuseks;
- *cellspacing* – lahtrite vahekaugus, väärtuseks täpne arv pikselites;
- *cellpadding* – lahtrite vooderdus, määrab lahtri serva ja sisu omavahelise kauguse igast küljest, väärtuseks täpne arv pikselites;
- *frame* – määrab, millised tabeli välimised äärised (raamjooned) on nähtavad. Võimalikud väärtused on: *void* – ääriseid ei näidata; *above* – näidatakse ülemist äärist; *below* – näidatakse alumist äärist; *hsides* – näidatakse ülemist ja alumist (horisontaalsed) äärist; *vsides* – näidatakse vasakut ja paremat (vertikaalsed) äärist; *lhs* – näidatakse vasakut äärist; *rhs* – näidatakse paremat äärist; *box* või *order* – näidatakse kõiki välimisi ääriseid;
- *rules* – määrab, milliseid sisemisi ääriseid näidatakse. Võimalikud väärtused on: *none* – jooni ei näidata; *groups* – jooned on veergude ja ridade gruppide vahel; *rows* – jooned on ridade vahel; *cols* – jooned on veergude vahel; *all* – näidatakse kõiki jooni.

Näiteks tabel, mille äärise paksus on 4 pikselit, lahtrite vahekaugus 10 ja vooderdus samuti 10 pikselit ning igas lahtris on üks pilt:

```
<table border="4" cellspacing="10" cellpadding="10">
<tr>
<td>

</td>
<td>

</td>
</tr>
<tr>
<td>

</td>
<td>

</td>
</tr>
</table>
```

Välja näeb see selline:



Joonis 6 Tabel piltidega, ääris 4, lahtrite vahekaugus 10 ja lahtrite polsterdus 10 pikselit

Rea vormindamiseks saab kasutada järgmiseid atribuute, mis kirjutatakse rea algusmärgendisse:

- *align* – joondus, määrab kuidas sisu rea lahtritesse paigutatakse, võimalikud väärtused: *right* (parem), *left* (vasak, mis on ka vaikeväärtuseks), *center* (keskel) ja *justify* (rööpselt);
- *valign* – vertikaalne joondus, määrab kuidas sisu rea lahtritesse vertikaalsuunal paigutatakse, võimalikud väärtused: *top* (ülaserva, mis on ka vaikeväärtuseks), *middle* (keskele) ja *bottom* (allaserva).

Lahtri vormindamiseks saab kasutada järgmiseid atribuute, mis kirjutatakse lahtri algusmärgendisse:

- *align* – joondus, määrab kuidas sisu lahtrisse paigutatakse, võimalikud väärtused: *right* (parem), *left* (vasak), *center* (keskel) ja *justify* (rööpselt);
- *valign* – vertikaalne joondus, määrab kuidas sisu lahtrisse vertikaalsuunal paigutatakse, võimalikud väärtused: *top* (ülaserva), *middle* (keskele) ja *bottom* (allaserva);
- *colspan* – määrab üle mitme veeru antud lahter ulatub, väärtuseks täisarv;
- *rowspan* – määrab üle mitme rea antud lahter ulatub, väärtuseks täisarv.

Ühele või mitmele veerule korraga atribuutide määramiseks saab kasutada elementi `<col>`. See võimaldab vältida igal real vastavatele lahtritele üha uuesti samade atribuutide määramist. Kui igale veerule määratakse eraldi omadused, siis tuleb tabelisse lisada niipalju `<col>` elemente, kui on veergusid. Korraga mitmele järjestikusele veerule omaduste määramiseks kasutatakse atribuuti `span`, mille väärtuseks on mõjutatavate veergude arv. Näiteks kolmeveerulises tabelis esimese veeru taust on valge, kahel järgmisel hall (määratud `style` atribuudi ja `css`-iga):

```
<table>
  <col style="background-color:white" />
  <col span="2" style="background-color:gray" />
  <tr>
    <td>esimese rea, vasaku lahtri sisu</td>
    <td>esimese rea, keskmise lahtri sisu</td>
    <td>esimese rea, parema lahtri sisu</td>
  </tr>
  <tr>
    <td>teise rea, vasaku lahtri sisu</td>
    <td>teise rea, keskmise lahtri sisu</td>
    <td>teise rea, parema lahtri sisu</td>
  </tr>
</table>
```

Ka sellel elemendil saab kasutada atribuute `width`, `align` ja `valign`

NB! Elementi `<col>` saab kasutada vaid tabeli või `<colgroup>` elemendi sees!

Järgmises näites on püütud suurem osa tabeli elemente ja vormistusvõtteid ära kasutada:

```
<!-- Tabeli algus, raamjoon, lahtrite vahe, polsterdus, laius, välisraam, sisejooned -->
<table border="1" cellspacing="15" cellpadding="10" width="800" frame="vsides" rules="groups">
  <!-- lisan pealdise - caption -->
  <caption>Tabeli loomise võimalusi tutvustav tabel</caption>
  <!-- Veergude grupid -->
  <colgroup span="2"></colgroup>
  <colgroup span="1"></colgroup>
  <!-- algab tabeli päis -->
```

```

<thead>
<tr>
  <!--päise ainsa rea ainus lahter on tabeli 3 lahtri laiune-->
  <td colspan="3">See on tabeli päis</td>
</tr>
<tr> <!--Tabeli veergude päised, kasutab th elemente -->
  <th>Vasaku veeru päis</th>
  <th>Keskmise veeru päis</th>
  <th>Parema veeru päis</th>
</tr>
</thead>
<!--algab tabeli jalus-->
<tfoot>
<tr>
  <!--jaluse ainsa rea ainus lahter on tabeli 3 lahtri laiune-->
  <td colspan="3">See on tabeli jalus</td>
</tr>
</tfoot>
<!--Algab tabeli kehaosa-->
<tbody>
<!--Esimene rea lahtrite sisu on keskele joondatud ning vertikaalselt allaserva-->
<tr align="center" valign="bottom"> <!--siit algab tabeli esimene rida-->
  <td>
    <p>Esimene rida, vasak</p><p>Siin on ainsana kaks rida</p>
  </td>
  <td>
    <p>Esimene rida, keskmine</p>
  </td>
  <td>
    <p>Esimene rida, parem</p>
  </td>
</tr> <!--Teine rida-->
<tr>
  <td>
    <p>Teine rida, vasak</p>
  </td>
  <td>
    <p>Teine rida, keskmine</p>
  </td>
  <td>
    <p>Teine rida, parem</p>
  </td>
</tr>
</tbody>
</table>

```

Välja näeb see järgmine:

Tabeli loomise võimalusi tutvustav tabel		
See on tabeli päis		
Vasaku veeru päis	Keskmise veeru päis	Parema veeru päis
Esimene rida, vasak		
Siin on ainsana kaks rida	Esimene rida, keskmine	Esimene rida, parem
Teine rida, vasak	Teine rida, keskmine	Teine rida, parem
See on tabeli jalus		

Joonis 7 Tabel, kus pea-aegu kõik elemendid kasutatud

Alajaotus ehk sektsioon

HTML keeles on üks huvitav element `<div>` (inglisekeelsest sõnast *division*), mille abil saab veebilehele luua alajaotust, mille sisse tavaliselt grupeeritakse objekte, et neid siis stiile kasutades koos kujundada.

```
<div>
alajaotuse sisu (tekst, pildid, objektid vms)
</div>
```

Alajaotus on blokielement (*block*)!

Stiilide abil saab alajaotuse välimust määrata, sealhulgas ka alajaotuse asukohta veebisirvija aknas ning näiteks seda isegi teiste objektide peale asetada!

Raam teise dokumendiga

Sageli on tarvis ühe veebilehe sees näidata mõne teise veebilehe sisu. Varem kasutati sellisel eesmärgil raamistikku (*frameset*), mis jagas terve veebilehe raamideks kus iga raami sisse paigutati eraldi HTML-dokument. Praegu ei soovitata sellist lähenemist ning HTML versiooni 5 puhul on raamistik lausa keelatud!

Teise dokumendi lisamiseks veebilehele kasutatakse elementi `<iframe>`!

Sellel elemendil on kohustuslikuks atribuudiks `SRC` (*source*), mille väärtuseks on viide teisele dokumendile. Näiteks:

```
<iframe src ="http://www.seejasee.ee/oluline"></iframe>
```

Selle elemendi sisuks võib kirjutada näiteks teate, et veebilehitseja ei toeta seda elementi või siis näiteks lisada sinna pildi, mida näidatakse kui määratud sisu ei õnnestu näidata. Sisu pole siinjuures kohustuslik.

Seda `<iframe>` elementi saab CSS stiilide abil kujundada nagu teisigi HTML elemente. Lisaks kohustuslikule `SRC` atribuudile saab tal kasutada veel atribuute:

- *width* – laius, väärtuseks täpne arv pikselites või protsentides.
- *height* – kõrgus, väärtuseks täpne arv pikselites või protsentides.
- *frameborder* – äärisjoon, väärtusteks 0 – äärist ei näidata või 1 – äärist näidatakse.
- *longdesc* – viide (URL) veebilehele, kus on antud raami sisu pikk kirjeldus.
- *marginwidth* – raami ülemine ja alumine veeris pikslites (px).
- *marginheight* – raami vasak ja parem veeris pikslites (px).
- *name* – raami nimi.
- *scrolling* – määrab kerimisribade kasutamise, võimalikud väärtused *yes*, *no* ja *auto*.

NB! Elementi `<iframe>` ei saa kasutada, kui kasutatakse dokumendi tüübi deklaratsiooni (DTD) HTML 4.1 Strict!

Erinevate objektide lisamine veebilehele

Siinkohal peame silmas selliseid objekte nagu heliklipid, videod, Java apletid, ActiveX objektid, pdf dokumendid, Flash animatsioonid ja muidugi ka Microsoft Silverlight rakendused.

Selliste objektide lisamiseks kasutatakse <object> elementi:

```
<object>  
Koodiread objekti kuvamisega seotud parameetritega  
</object>
```

Kahjuks on selle elemendi interpreteerimisel erinevate veebilehitsejate puhul erinevusi. Erinevad veebilehitsejad kasutavad sama objekti laadimiseks erinevaid koodiridu. Probleemi lahenduseks kirjutatakse <object> elemendi sisse (algus- ja lõpumärgendi vahele) hulk erinevaid koodiridu, erinevatele veebilehitsejatele erinevad read, millega tagatakse objekti kuvamine.

Näiteks DeepZoom veebialbumi lisamise veebilehele:

```
<object data="data:application/x-silverlight-2," type="application/x-silverlight-2" width="800px"  
height="600px">  
  <param name="source" value="DeepZoomProject.xap"/>  
  <param name="background" value="white" />  
  <param name="minRuntimeVersion" value="2.0.31005.0" />  
  <param name="autoUpgrade" value="true" />  
<a href="http://go.microsoft.com/fwlink/?LinkId=124807" style="text-decoration: none;">  
    
</a>  
</object>
```

Horisontaalne joon

Erinevate objektide, veebilehe sektsioonide eraldajaks kasutatakse sageli horisontaalset joont, mille lisamiseks on lihtne element: <hr />.

HTML päise elemendid

HTML dokumendi päises (element <head>) on elemendid, mida veebilehe vaataja reeglina otseselt ei näe. Nende abil määratakse siiski mitmeid veebilehe väljanägemist määravaid parameetreid (META andmed nagu tekstikodeering, stiilileht, märksõnad otsingumootorite jaoks jpm).

Veebilehe pealkiri

Veebilehe pealkiri defineeritakse HTML dokumendi päises elemendiga <title>. See pealkiri paigutatakse veebilehitseja tiitliribale, seda kasutatakse veebilehe salvestamisel lemmikute või järjehoidjate (*favorites* või *bookmarks*) hulka. Selle elemendi sisu kasutatakse otsimootorite poolt veebilehe kataloogimiseks ning ta muutub järjest olulisemaks! Seetõttu ei tohiks ta sisaldada mõttetut teksti! Eespool asetsevad sõnad on otsimootori jaoks suurema kaaluga! Näiteks:

```
<title>Tallinna Ülikool</title>
```

Selle elemendi toime avaldub veebilehitseja tiitliribal järgmiselt:



Joonis 8 Veebilehe pealkiri veebilehitseja tiitliribal

NB! Tegemist on veebilehe päise kohustusliku elemendiga!

NB! Selle elemendi sisu ei tohiks olla rohkem kui 64 sümbolit! Mitmed veebilehitsejad ei näita rohkem kui 70 sümbolit!

Metaandmed

Metaandmed (*Metadata*) on info andmete kohta ehk siis veebilehe kontekstis info veebilehe sisu kohta.

Metaandmete lisamiseks kasutatakse HTML dokumendi päises <meta> elementi. Sellel elemendil on kohustuslik atribuut content, milles on kirjas konkreetne metaandmete sisu ning valikulised atribuudid, mis on content atribuudiga seotud:

- *http-equiv* – määrab content atribuudi sisule päise (*header*), kasutatakse veebilehe sisu tekstikodeeringu, veebilehe sisu tüübi jms määramisel;
- *name* – määrab content atribuudi sisule nime, kasutatakse veebilehe kohta käivate võtmesõnade, autori jms määramisel.

NB! Need valikulised atribuudid välistavad üksteist!

NB! name atribuudi väärtuste kohta leiab rohkem infot:

http://www.w3schools.com/tags/att_meta_name.asp

NB! Element <meta> on tühi element ja vajab kaldkriipsuga „lõpetamist“!

Tekstikodeering

Erinevates keeltes esineb väga erinevaid tähemärke, mille korrektseks kuvamiseks on vaja määrata, millist kodeeringut kasutades antud veebilehte avada tuleb. Veebilehitsejad püüavad reeglina küll automaatselt tuvastada, millist kodeeringut peaks kasutama kuid autori poolt tehtud määrangud tagavad kindla tulemuse.

Tekstikodeeringu määramiseks tuleb HTML dokumendi päisesse lisada järgmine rida metaandmetega:

```
<meta http-equiv="Content-Type" content="text/html; charset=kodeering" />
```

Üks tavalisemaid kodeeringuid on ISO-8859-1, mis on ka enamusel veebilehitsejatel vaikimisi kasutatav ning sobib ka eestikeelsete veebilehtede jaoks. Seega näeks vastav metaandmete element välja järgmine

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
```

Kuna ISO-8859 märgikomplektid on piiratud suurusega ning mitmekeelsete keskkondade jaoks sobimatud, siis on väljatöötatud *unicode* standard. Viimasel ajal on hakatudki soovitama kasutada kodeeringut utf-8, ehk siis vastav metaandmete element oleks järgmine:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

Nimekirja HTML standardsetest tekstikodeeringutest leiab aadressil:

http://www.w3schools.com/tags/ref_charactersets.asp

Info veebilehe kohta (kirjeldus, võtmesõnad, autor)

Veebilehe kohta saab ja on lausa kasulik lisada ka pisut kirjeldavat infot, mida kasutavad näiteks otsingumootorid ja mis muudavad veebilehe kergemini leitavaks.

Veebilehe kirjelduse jaoks lisatakse HTML dokumendi päisesse järgmine rida:

```
<meta name="description" content="Veebilehe lühikirjeldus" />
```

NB! Selle metamärgendi content atribuudi väärtus ei tohiks olla pikem kui 250 sümbolit! Seda metamärgendit kasutatakse otsimootorite poolt küll järjest vähem kuid selles sisalduvad sõnad on ka veebilehe enda sisus, siis teda arvestatakse!

Interneti otsingumootorid kasutavad enamasti ja peamiselt veebilehtede päises kirjasolevaid võtmesõnu. Nende lisamiseks tuleb kirjutada järgmine rida:

```
<meta name="keywords" content="võtmesõna1, võtmesõna2, võtmesõna3, ..." />
```

NB! Üldiselt soovitatakse kasutada maksimaalselt 20 märksõna! Optimaalne oleks piirduda 64 sümboliga ehk 6 – 8 sõnaga!

Sarnaselt neile saab kirja panna ka veebilehe autori. Sellisel juhul on atribuudi name väärtuseks „author“.

Suunamine teisele lehele

Veebilehe elutsükli jooksul võib lehe aadress (URL) muutuda. Kuna leht võib olla lingitud mitmete teistel lehtedel ja kantud ka otsingumootorite andmebaasidesse, siis saab vana lehe asendada väikesemahulise lehega, mis suunab kasutajad otse uuele aadressile.

Selleks tuleb suunamiseks loodud veebilehe päisesse lisada järgmine element:

```
<meta HTTP-EQUIV="REFRESH" content="sekundid; url=uue_lehe_url">
```

Näiteks:

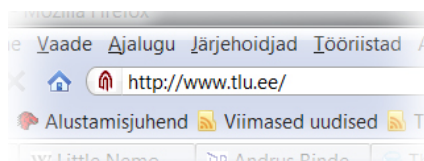
```
<meta HTTP-EQUIV="REFRESH" content="5; url=http://www.seejasee.ee">
```

NB! Igaks juhuks on soovitatav ümbersuunava lehe sisus (<body>) kasutada tavapärast linki nende väheste kasutajate jaoks, kelle veebilehitseja automaatset ümbersuunamist ei toeta!

NB! Jättes kirja panemata soovitud veebilehe URL-i, toimub lihtsalt käesoleva veebilehe värskendamine etteantud ajaintervalli järel!

Favicon

Favicon on pisike ikoon, mida kasutatakse veebilehe tähistamiseks järjehoidjate (*bookmarks*) ja lemmikute (*favorites*) nimekirjas. Nimetus tulenebki inglisekeelsetest sõnadest *favorites icon*.



Joonis 9 Favicon TLÜ veebilehel

Faviconi mõõtudeks on reeglina 16X16 pikselit või 32X32 pikselit (harva ka 48X48). Tavaliselt kasutatakse .ico faile kuid põhimõtteliselt saab kasutada ka jpg, png, gif ja svg faile.

Tabel 1 Veebilehitsejate tugi erinevatele failivormingutele Faviconi kasutamisel (Wikipedia)

Veebilehitseja	ICO	PNG	GIF	animeeritud GIF	JPEG	APNG (animeeritud PNG)	SVG
Google Chrome	X	X	X	-	X	-	X
Internet Explorer	X	-	-	-	-	-	-

Firefox	X	X	X	X	X	X	-
Opera	X	X	X	X	X	X	X
Safari	X	X	X	-	X	-	-

Faviconide loomiseks on olemas hulk vahendeid, sealhulgas ka veebipõhiseid, näiteks: <http://www.favicon.cc/>

NB! Praegusel ajal ei vaja enamus veebilehitsejaid üldse mingit HTML elementi faviconi linkimiseks kui kasutatakse vaikenimetust **favicon.ico** ja see fail asub veebilehestiku juurkataloogis.

Faviconi lisamiseks veebilehele HTML-koodi kasutades tuleb dokumendi päisesse lisada element `<link>` (sarnaselt näiteks css faili linkimisega). Näiteks kõige tavapärasema .ico faili linkimiseks:

```
<link rel="shortcut icon" href="ico_faili_URL">
```

Kasutades jpg, png või gif faili, tuleks määrata ka failitüüp! Näiteks:

```
<link rel="icon" type="image/jpg" href="jpg_faili_URL ">  
<link rel="icon" type="image/png" href="png_faili_URL ">  
<link rel="icon" type="image/gif" href="gif_faili_URL ">
```

Erisümbolid

Mitmesuguste erisümbolite (näiteks matemaatilised võrdlusmärgid, mis muidu on kasutusel HTML märgendite juures jne) jaoks on HTML keeles kasutusel nn olemid (*entity*), mida pannakse kirja kujul:

&oleminimi;

või

&oleminumber;

	"kõva tühik" (<i>non-breaking space</i>)	 	
<	väiksem kui	<	<
>	suurem kui	>	>
&	ampersand	&	&
"	jutumärk (<i>quotation mark</i>)	"	"
€	euro	€	€
§	paragrahv (<i>section</i>)	§	§
©	copyright	©	©
®	registered trademark	®	®

Täieliku nimekirja erinevate sümbolite jaoks vajalike olemite kohta leiab: http://www.w3schools.com/tags/ref_entities.asp

HTML elementide atribuudid

Suur osa HTML elementide atribuutidest on käsitletud eespool koos elementidega, mille juures neid kasutatakse. Siinkohal pöörame tähelepanu mõningatele, mida eespool on osaliselt käsitletud või pole üldse vaadeldud.

Tuumikatribuudid

Tuumikatribuudid (*core*) määravad erinevate elementide üldised omadused.

- `id` atribuudi väärtuseks on elemendile määratud unikaalne identifikaator (nimi).
- `title` atribuudi väärtuseks on tekst, lisainfo elemendi kohta.
- `style` atribuudi väärtuseks on css reastil.
- `class` atribuudi väärtuseks on elemendi kujundamiseks kasutatav css klass.

NB! Neid atribuute ei saa kasutada elementidel `<base>`, `<html>`, `<meta>`, `<param>`, `<script>` ja `<title>`!

Keele atribuudid

Atribuut `lang` on kasutusel tervel veebilehel või selle osades kasutatava keele deklareerimiseks! XHTML puhul tuleb lisada ka `xml:lang` atribuut!

Vastavalt W3C soovitusele tuleks igal veebilehel kasutatav keel deklareerida elemendi `<html>` algusmärgendis!

Näiteks HTML puhul:

```
<html lang="et">
```

XHTML puhul:

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

Kasutatava keele deklareerimiseks määrab koodid standard ISO 639-1 ja vastavalt sellele on kasutusel näiteks järgmised koodid:

Araabia	ar	Hispaania	es	Itaalia	it	Portugali	pt
Saksa	de	Eesti	et	Jaapani	ja	Vene	ru
Kreeka	el	Soome	fi	Leedu	lt	Rootsi	sv
Inglise	en	Prantsuse	fr	Läti	lv	Hiina	zh
Esperanto	eo	Ungari	hu	Poola	pl	Ukraina	uk

Täieliku nimekirja kasutatavatest koodidest leiab:

http://www.w3schools.com/tags/ref_language_codes.asp

Keelega seoses on kasutatav ka veel atribuut `dir`, mis määrab teksti suuna elemendis.

Võimalikud väärtused on:

- *ltr* – määrab teksti suuna vasakult paremale (*left to right*);
- *rtl* – määrab teksti suuna paremalt vasakule (*right to left*);

NB! Keele atribuute ei saa kasutada elementidel `<base>`, `<br>`, `<frame>`, `<frameset>`, `<hr>`, `<iframe>` ja `<script>`!

Sündmustega seotud atribuudid

Alates HTML versioonist 4 on elementidele lisatud atribuute, mille abil saab käivitada teatud toiminguid, näiteks JavaScript'i.

Nendele atribuutidele reageerimiseks peab veebilehega siduma mingis keeles kirjutatud skripti ja selleks kasutatakse element `<script>`, mille sisuks ongi programmikood.

Sellel elemendil on rida atribuute:

- *type* – kohustuslik atribuut, määrab skripti tüübi (võimalikud väärtused on näiteks: *text/javascript*, *text/ecmascript*, *application/ecmascript*, *application/javascript* jms);
- *charset* – määrab skriptis kasutatud tähemärkide kooditabeli;

- *defer* – näitab, et skript ei genereeri dokumendile mingisugust sisu, brauser saab jätkata veebilehe parsimist (väärtus: *defer*);
- *src* – URL, mis viitab välisele skriptifailile, näiteks JavaScript'i *.js failile.

Lehe laadimisega seotud sündmused

Veebilehe laadimisel (elementide `<body>` ja `<frameset>` juures) saab kasutada järgmiseid atribuute:

- *onload* – väärtuseks skript, mis käivitatakse lehe laadimisel;
- *onunload* – väärtuseks skript, mis käivitatakse lehelt lahkumisel.

Näiteks:

```
<body onload="yksFunktsioonJavaScriptiga()">
```

Piltidega seotud sündmused

Piltidele ehk elemendile `<img>` saab lisada vaid ühe sündmusega atribuudi:

- *onabort* – väärtuseks skript, mis käivitatakse, kui pildi laadimine katkestatakse.

Klaviatuuriga seotud sündmused

Klaviatuuriga seotud sündmused on klahvivajutused. Kasutada saab kolme erinevat sündmuse atribuuti:

- *onkeydown* – väärtuseks skript, mis käivitatakse klahvi allavajutamisel;
- *onkeyup* – väärtuseks skript, mis käivitatakse klahv vabastatakse;
- *onkeypress* – väärtuseks skript, mis käivitatakse, kui klahv vajutatakse alla ja vabastatakse.

NB! Neid atribuute saab kasutada kõigi elementide juures välja arvatud `<base>`, `<bdo>`, `<br>`, `<frame>`, `<frameset>`, `<head>`, `<iframe>`, `<meta>`, `<param>`, `<script>`, `<style>` ja `<title>`!

Hiirega seotud sündmused

Hiirega seotud sündmused on seotud klõpsamise (*click*) ja hiire liigutamisega.

- *onclick* – väärtuseks skript, mis käivitatakse hiire nupuga klõpsamisel;
- *ondblclick* – väärtuseks skript, mis käivitatakse hiire topeltklõpsul;
- *onmousedown* – väärtuseks skript, mis käivitatakse hiire nupu allavajutamisel;
- *onmouseup* – väärtuseks skript, mis käivitatakse hiire nupu vabastamisel;
- *onmousemove* – väärtuseks skript, mis käivitatakse hiire liigutamisel;
- *onmouseover* – väärtuseks skript, mis käivitatakse hiirega elemendile liikumisel;
- *onmouseout* – väärtuseks skript, mis käivitatakse hiirega elemendilt lahkumisel.

NB! Neid atribuute ei saa kasutada elementidel `<base>`, `<bdo>`, `<br>`, `<frame>`, `<frameset>`, `<head>`, `<html>`, `<iframe>`, `<meta>`, `<param>`, `<script>`, `<style>` ja `<title>`!

Kokkuvõte

Käesolev HTML juhend on kokkuvõtlik ja katab vaid olulisemad, enimkasutatavad elemendid ja nende enimkasutatavad atribuudid.

Täieliku materjali HTML keele kohta leiab näiteks aadressil: <http://www.w3schools.com>

Veebilehte luues tuleks kindlasti püüda kinni pidada standarditest ja headest tavadest. Kindlasti peaks veebilehe alguses olema dokumendi tüübi deklaratsioon (DTD), päises meta-andmed, sealhulgas tekstikodeering jms. Oma veebilehe vastavust standarditele saab kontrollida HTML validaatori abil (<http://validator.w3.org/>).

Näitena esitame siinkohal veebilehe, millel on pealkiri, kaks sissejuhatavat tekstilõiku, üks link, liigendatud loetelu ja tabel kahe rea ja kahe veeruga, milledes on pildid lilledega:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Lilledega lehekülg</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<meta name="description" content="Nelja ilusa lillega lehekülg" />
<meta name="keywords" content="foto, lill, karikakar, võilill, kaunis kuldking, kibuvits" />
</head>
<body>
<h1><big><strong>Neli kaunist õit</strong></big></h1>
<p>Sellel veebilehel on nelja kauni õie pildid</p>
<p>Neid õisi võib igaüks Eestimaal sageli kohata, erandiks on ehk vaid Kaunis Kuldking, mis on
haruldane ja looduskaitse all!
<br />
Rohkem infot Eesti taimede kohta leiab aadressil: <a href="http://bio.edu.ee/taimed/"
target="_blank">bio.edu.ee/taimed</a>!
</p>
<hr />
<ol>
<li>Esimeses reas</li>
<ul>
<li>Karikakar</li>
<li>Kaunis Kuldking</li>
</ul>
<li>Teises reas</li>
<ul>
<li>Kibuvits</li>
<li>Võilill</li>
</ul>
</ol>
<hr />
<table border="4" cellspacing="10" cellpadding="10">
<tr>
<td>

</td>
<td>

</td>
</tr>
<tr>
<td>

</td>
<td>
</td>
</tr>
```

```
<td>

</td>
</tr>
</table>
<hr>
</body>
</html>
```

Välja näeb see tulemus järgmine:

Neli kaunist õit

Sellel veebilehel on nelja kauni õie pildid

Neid õisi võib igauks Eestimaal sageli kohata, erandiks on ehk vaid Kaunis Kuldking, mis on haruldane ja looduskaitse all!
Rohkem infot Eesti taimede kohta leiab aadressil: bio.edu.ee/taimed/

-
1. Esimeses reas
 - o Karikakar
 - o Kaunis Kuldking
 2. Teises reas
 - o Kibuvits
 - o Võilill
-



Joonis 10 Näidisena valminud veebileht

<http://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/Tags.html>