

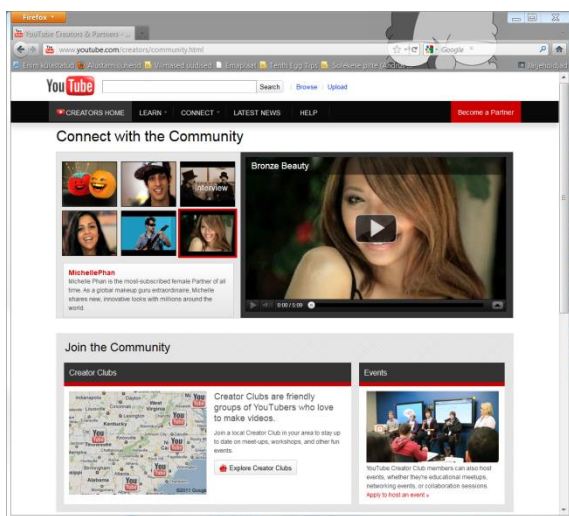
Sisukord

HTML JA SELLE AJALUGU	3
HTML ÕIGEKIRI.....	4
HTML ELEMENDID	6
DOKUMENDI TÜÜP	6
VEEBILEHE PÕHIOSAD.....	6
HTML PÄISE ELEMENDID	7
<i>Veebilehe pealkiri</i>	<i>7</i>
<i>Metaandmed</i>	<i>8</i>
Tekstikodeering	8
Info veebilehe kohta (kirjeldus, võtmesõnad, autor)	9
Suunamine teisele lehele	9
Vaateava.....	9
<i>Teiste ressursside linkimine</i>	<i>11</i>
Favicon	13
<i>Vaikeväärtused hüperlinkidele</i>	<i>14</i>
<i>Kujunduse lisamine</i>	<i>14</i>
HTML DOKUMENDI KEHA ELEMENDID.....	15
<i>Erinevad alajaotused</i>	<i>15</i>
Lehe sisu põhiosa	15
Artikkel	16
Sektsioon	16
Ääremärkused/täiendav info.....	16
Päis ja jalus	17
Navigatsiooniplokk	17
Täiendav info	17
Dialoogiaken.....	18
Alajaotus ehk allüksus	18
Peidetud sisu javascriptiga näitamiseks	19
<i>Tekst</i>	<i>19</i>
Pealkirjad.....	19
Tekstilõigud	19
Teksti poolitamine	20
Loendid.....	20
Nummerdatud loend.....	20
Täpploend	20
Mitmetasemeline loend	21
Kirjelduste loend	21
Teksti vormindamine.....	22
Arvuti stiilid	23
Tsitaadid, definitsioonid.....	23
Esiletõstetud tekst.....	24
<i>Hüperlink</i>	<i>24</i>
Lingid sama lehe piires	26
<i>Pilt veebilehel.....</i>	<i>27</i>
Kohanduv pilt	28
Fotode ja jooniste tähistamine veebilehel	28
Interaktiivne pilt	29
<i>Tabel</i>	<i>30</i>

Veergude ja ridade grupeerimine.....	32
HTML vahendid tabeli ja selle lahtrite vormindamiseks.....	33
Tabeli näidis.....	33
<i>Raam teise dokumendiga</i>	<i>34</i>
<i>Horisontaalne joon</i>	<i>36</i>
<i>Meediaelemendid veebilehel.....</i>	<i>36</i>
Pistikprogramme kasutatavad objektid.....	36
Element <object>	36
Element <embed>	37
Heli ja video veebilehel.....	37
Heliklipid veebilehel	37
Video veebilehel.....	38
Audio- ja videofailide määramine	39
Skriptiga loodud graafika ja animatsioon	40
<i>Arvväärtustega seotud elemendid.....</i>	<i>40</i>
Mõõdik	40
Edenemisriba	41
Aja märkimine	42
HTML ELEMENTIDE ATRIBUUDID.....	42
<i>Üldised atribuudid</i>	<i>42</i>
<i>Keelega seotud atribuudid.....</i>	<i>43</i>
<i>Sündmustega seotud atribuudid</i>	<i>44</i>
Lehe laadimisega seotud atribuudid	44
Piltidega seotud sündmuste atribuudid	45
Klaviatuuriga seotud sündmuste atribuudid	45
Hiirega seotud sündmuste atribuudid	45
ERISÜMBOLID.....	45

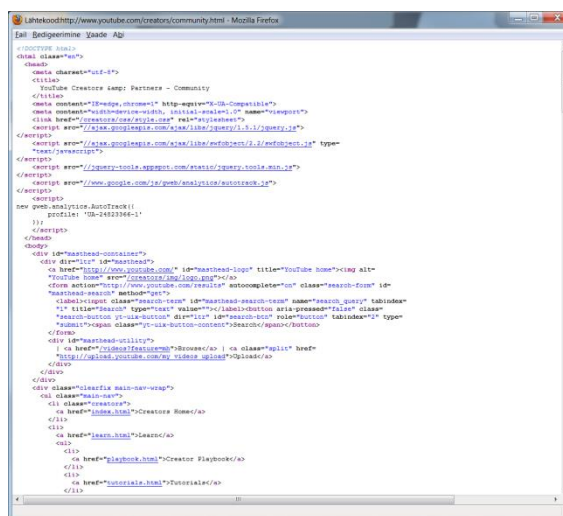
HTML ja selle ajalugu

HTML on lahti seletatult hüperteksti märgendikeel (*Hypertext Markup Language*) ehk keel, millega märgitakse üles hüperteksti dokumendid. Hüpertekst, kus võtmesõnadel klõpsates pöörduetakse nende sõnadega seotud dokumentide juurde, ongi veebileht oma esialgses, kitsas tähenduses. Seega on HTML keel veebilehede ülesmärkimiseks (*markup*), loomiseks. Kui vaatame mõnda veebilehte, siis paistab see reeglina ilus värviline, sisaldab pilte, videoid jne. Tegelikult aga on veebileht HTML dokument, milles on kirjas 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

<http://www.youtube.com/creators/community.html>



Joonis 2 osa kõrval oleva veebilehe 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. Selle kood näeb välja järgmine:

```
<title>Hypertext Links</title>
<h1>Links and Anchors</h1>
A link is the connection between one piece of
<a href=WhatIs.html>hypertext</a> and another.
```

Lehe leiab näiteks aadressil: <http://www.w3.org/History/19921103-hypertext/hypertext/WWW/Link.html>

Esimene ettepanek ametlikuks HTML spetsifikatsiooniks avaldati 1993. aasta keskel ("*Hypertext Markup Language (HTML)*" *Internet-Draft*, Tim Berners-Lee ja Dan Connolly).

HTML



Joonis 3 W3C loodud ajalooline HTML logo

Aastate jooksul on tehnoloogiate arenedes arenenud ka veebilehede võimalused, esialgsele tekstile ja piltidele on lisandunud videod, heli, animatsioonid, voogmeedia, erinevad skripti keeled, 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.

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.
- 1999. aastal kuulutati välja HTML 4.01, mis jäi pikaks ajaks viimaseks.
- 22. jaanuaril 2008 valmis esimene HTML5 spetsifikatsiooni mustand.
- Aastast 2012 oli HTML 5 W3C soovitus kandidaat ja sai soovitusena 2014. aasta 28. oktoobril.
- 1. novembril 2016 sai W3C soovitusena saama juba HTML 5.1.
- 14. detsembril 2017 sai W3C soovitusena HTML 5.3.
- 2017 algas töö HTML 5.3 spetsifikatsiooniga.



Joonis 4 HTML 5 logo (W3C)

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 HTML 5 standardit!

HTML dokument koosneb elementidest. Elementidel on reeglina algumärgend (*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* või *void elements*), millel pole eraldi lõpumärgendit!

```
<elemendinimi>
```

Elementidel on lisaks nimele olemas veel hulk atribuute (mõnedel ka kohustuslikke, enamasti valikulised). Atribuudid lisatakse elemendi algumärgendisse kujul:

```
nimi="väärtus"
```

Nii võib mõni element olla järgmisel kujul:

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

Üks tavaline atribuut, mida saab kasutada näiteks Javascriptiga sidumiseks või ka CSS stiililehede rakendamisel, on identifikaator ehk *id*, näiteks:

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

NB! Ühegi HTML elemendi *id* atribuudi väärtust ei tohi alustada numbriga!

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 **plokielemidid** (*block*) – hõivavad veebilehel kogu rea, ei luba teisi elemente enda kõrvale ja osa **rea-elementid** (*inline*). Plokielementide hulka kuuluvad näiteks pealkirjad (<h1>...<h6>), tekstilõigud (<p>) ja alajaotused (<div>). Plokielemente 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 koodi kirjutamisel tuleb meeles pidada, et:

- Elemendid peavad olema korrektselt pesastatud (*nested*)!
<i>Siin on paks ja kaldu tekst</i>
- Elemendid peavad enamasti olema suletud (vajavad lõpumärgendit või sulgevat kaldkriipsu märgendi lõpus)!

```
<p>See on lõik!</p>
```


Osa elemente võib jätta sulgemata, sest järgmist sarnast elementi ei saa nagunii alustada ilma eelnevat sulgemata. Selliseid elemente nimetatakse isesulguvateks (*self-closing*). Sellisteks on näiteks: html, head, body, p, dt, dd, li, option, thead, th, tbody, tr, td, tfoot, colgroup.

Spetsialistid siiski soovivad neid kõiki lõpetada, et kirjutatav kood oleks selgem ja, et vältida lohakust.

- Elementid pannakse alati kirja väiketähtedega! Atribuutide nimed peavad olema väiketähtedega!
- Atribuutide väärtused peavad olema jutumärkides!

HTML elemendid

Ülevaate kõigist HTML5 elementidest saab näiteks: <http://w3schools.com/html5/>

Dokumendi tüüp

Veebilehe loomisel tuleb lehe algusesse lisada dokumendi tüübi deklaratsioon DTD (*Document Type Declaration*).

Dokumendi tüüp on oluline selleks, et veebilehitseja suudaks veebilehte korralikult ekraanile tuua, elemente tõlgendada, nii toimub lehe avamine ka kiiremini. Isegi lihtsa HTML koodi puhul võib see aidata soovitud tulemust saavutada.

Veebilehitseja peab teadma, millise tüübiga on tegemist, selleks lisataksegi veebilehe algusesse vastav deklaratsioon. Varasemate HTML versioonide puhul oli olemas mitmeid erinevaid versioone, millel kõigil olid erinevad DTD-d. HTML5 puhul on vaid üks DTD ja see on väga lihtsa kujuga:

```
<!DOCTYPE html>
```

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

NB! HTML5 puhul pole see deklaratsioon enam tegelikult vajalik!

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

Veebilehe põhiosad

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

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

NB! HTML5 puhul pole sellel elemendil enam vaja dokumendi tüübile viitavat atribuuti!

Veebileht jaguneb kaheks oluliseks üksteisele järgnevaks 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 HTML 5 veebilehe ülesehitus on seega järgmine:

```
<!DOCTYPE html>
<html lang="et">
<head>
  <title>Veebileht</title>
  <meta charset="UTF-8">
  päise sisu
</head>
<body>
  veebilehe sisu
</body>
</html>
```

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) aga ka veebilehe pealkiri.

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 5 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 näiteks aadressil:

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

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 HTML5 dokumendi päisesse <meta>-element uue atribuudiga `charset`:

```
<meta charset="kodeering">
```

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

```
<meta 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 soovitada kasutada kodeeringut utf-8, ehk siis vastav metaandmete element oleks järgmine:

```
<meta charset="utf-8">
```

Tekstikodeeringute ja nende erinevuste kohta saab lugeda näiteks aadressil:

https://www.w3schools.com/html/html_charset.asp

Nimekirja HTML standardsetest tekstikodeeringutest leiab näiteks 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 meta-elementi `content` atribuudi väärtus ei tohiks olla pikem kui 250 sümbolit! Seda meta-elementi kasutatakse otsimootorite poolt küll järjest vähem kuid selles sisalduvad sõnad on ka veebilehe enda sisus, siis teda arvestatakse!

Interneti otsingumootorid kasutavad enamasti ja peamiselt veebilehete päises kirjas olevaid 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 tavapärasest 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!

Vaateava

Vaateava ehk *viewport* on HTML5 uuendus, mis võimaldab disaineritel luua kohanduvat (*responsive*) kujundust vastavalt veebilehe vaatamiseks kasutatava seadme ekraanile.

Vaateava annab brauserile juhised, kuidas veebilehe mõõtusi hallata, elemente suurendada/vähendada.

Vaateava defineeritakse järgmise meta-elementiga:

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

Atribuudi *content* väärtus määrab siin vaateava atribuudid. Antud näites on tavapärased atribuudid tavapäraste väärtustega:

- *width=device-width* – käsib lehe suuruse seada vastavusse kasutatava seadme ekraaniga;
- *initial-scale=1.0* – määrab lehe laadimisel kasutatava esialgse suurendusastme (*zoom level*).

Atribuutidel võib olla ka teistsuguseid väärtuseid.

- *width* väärtuseks võib olla ka konkreetne arv piksleid, näiteks *width=640*. Seda kasutatakse, kui on teada, et leht vajab minimaalselt teatavat laiust aga mõnel seadmel võib olla sellest väiksem ekraan.

Kindla suurusega vaateava saab määrata näiteks nii:

```
<meta name="viewport" content="width=640">
```

Olemas on veel atribuudid:

- *height* – mis on kasulik, kui elementide asukoht muutub vastavalt vaateava kõrgusele. Üks võimalikke väärtuseid on *device-height*.
- *maximum-scale* – määrab maksimaalse võimaliku suurendusastme, väärtuste vahemik on > 0 kuni 10 ja vaikeväärtuseks 5;
- *minimum-scale* – määrab minimaalse võimaliku suurendusastme, väärtuste vahemik on > 0 kuni 10 ja vaikeväärtuseks 0,25;
- *user-scalable* – määrab, kas kasutaja saab suurendusastet muuta, väärtusteks „yes“ (vaikeväärtus) ja „no“.

Näiteks võib olla soov luua kindla suurusega veebirakendust, mille suurust muuta ei saa:

```
<meta name = "viewport" content = "user-scalable=no, width=device-width">
```

Vaateava mõõte kasutatakse CSS kujunduskeelega abil elementide paigutust ja suurust seades.

Pärast vaateava kasutuselevõttu saab CSS-is kasutada järgmiseid mõõtühikuid:

- *vw* – 1% vaateava laiuusest (*viewport width*);
- *vh* – 1% vaateava kõrgusest (*viewport height*);
- *vmin* – 1% vaateava lühemast servast (*viewport minimum*), kas laiuusest või kõrgusest olenevalt kumb on lühem;
- *vmax* – 1% vaateava pikemast servast (*viewport maximum*), kas laiuusest või kõrgusest olenevalt kumb on pikem.

Nendega saab määrata veebilehe elementide mõõte.

NB! Ka tavapärane protsentides viitamine veebilehe suurusele asendatakse nüüd protsentidega vaateala suurusest.

Teiste ressursside linkimine

Väliste ressursside (CSS stiililehed, favicon jms) sidumiseks veebilehega kasutatakse elementi `<link>`. Tegemist on tühja (*empty*) elemendiga.

`<link>` elementi võib veebilehe päises olla mitu.

Sellel elemendil on kasutada järgmised atribuudid:

- *href* – lingitava materjali URL (aadress).
 - *hreflang* – võimaldab määrata lingitud ressursi keelt (kasutatav vaid koos atribuudiga *href*).
 - *crossorigin* – määrab, kuidas käsitletakse allikatevahelisi (*cross-origin*) päringuid;
 - *rel* – kohustuslik atribuut, mis määrab antud veebilehe ja välise ressursi vahelise seose. Võimalikud väärtused on:
 - *alternate* – sama dokumendi alternatiivne versioon;
 - *author* – link käesoleva dokumendi autorile;
 - *license* – link dokumendi litsentsitingimustele (*copyright*);
 - *help* – link abiinfot sisaldavale dokumendile;
 - *icon* – impordib ikooni, mis esindab antud dokumenti (favicon);
 - *prev* – loogiliselt käesolevale eelnev dokument;
 - *next* – loogiliselt käesolevale järgnev dokument;
 - *pingback* – määratakse *pingback* serveri aadress, mis haldab antud lehe lingipinge (*pingback*);
 - *preconnect* – määrab, et brauser peaks juba looma ühenduse määratud ressursi asukohaga;
 - *prefetch* – lingitud dokumenti tuleks puhverdada, sest ilmselt on oodata sellele navigeerimist;
 - *preload* – määrab, et brauser peaks lingitud lehe eelnevalt puhverdama, seotud atribuudi as antud sihtkohaga;
 - *prerender* – määrab, et lingitud leht tuleks brauseris taustal ennetavalt valmis visualiseerida (*render*). Kui kasutaja sellele lehele edasi navigeerib, siis see kiirendab tööd;
- NB!** Prerender atribuut võib osutada lihtsalt ribalaiuse raiskamiseks!
- *dns-prefetch* – eelnevalt tuleks teha lingitud dokumendi asukoha DNS päring;
 - *search* – käesoleva dokumendi otsinguvahend;
 - *stylesheet* – lingitud dokument on CSS stiilileht.
- *sizes* – määrab lingitud ressursi suuruse.

NB! See atribuut on kasutatav vaid atribuudi rel väärtuse *icon* puhul!

Atribuudi *sizes* väärtuseks võib olla:

- *any* – määrab, et ikoon on skaleeritav (näiteks *svg* pilt);
- *kõrgusxlaius* – konkreetne mõõt pikslites.
Tühikutega eraldades või määrata mitu suurust.

Näiteks:

```
<link rel="icon" href="myicon.png" sizes="16x16 32x32" type="image/png">
```

- *type* – lingitud ressursi MIME (*Multipurpose Internet Mail Extensions*) tüüp.

Näiteks:

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

- *media* – meediapäring, millise seadmega kuvamiseks on antud ressurss mõeldud.

Võimalikud väärtused on:

- *all* – kõik seadmed (vaikeväärtus);
- *aural* – kõnesüntesaatorid;
- *braille* – Braille seadmed (pimedatele);
- *handheld* – pihuseadmed;
- *projection* – projektorid;
- *print* – printerid;
- *screen* – arvutimonitorid;
- *tty* – teletaipseadmed jms, püsisammkirjaga (*monospace*) kirjaga seadmed;
- *tv* – televiisorid jms.

Nende väljundseadmete puhul saab kasutada loogikaoperaatoreid *and*, *not* ja *“*,*“* (or) ning järgmiseid väärtuseid:

- *width* – nähtava ala laius (kasutada saab eesliiteid „*min-“* ja „*max-“*);
- *height* – nähtava ala kõrgus (kasutada saab eesliiteid „*min-“* ja „*max-“*);
- *device-width* – väljundseadme/paberi laius (kasutada saab eesliiteid „*min-“* ja „*max-“*);
- *device-height* – väljundseadme/paberi kõrgus (kasutada saab eesliiteid „*min-“* ja „*max-“*);
- *orientation* – väljundi orientatsioon („*portrait“* või „*landscape“*);
- *aspect-ratio* – väljundi laiuse/kõrguse suhe (näiteks 16/9, kasutada saab eesliiteid „*min-“* ja „*max-“*);
- *device-aspect-ratio* – väljundseadme/paberi laiuse/kõrguse suhe (näiteks 16/9, kasutada saab eesliiteid „*min-“* ja „*max-“*);

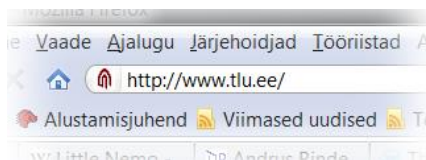
- *color* – väljundseadme ekraani värvussügavus (kasutada saab eesliiteid „*min-*“ ja „*max-*“);
- *color-index* – värvide arv, mida väljundseadme ekraan suudab näidata (kasutada saab eesliiteid „*min-*“ ja „*max-*“);
- *monochrome* – bittide arv piksli kohta monokroomse väljundi puhul (*bits per pixel in a monochrome frame buffer*) (kasutada saab eesliiteid „*min-*“ ja „*max-*“);
- *resolution* – väljundi punktihedus dpi või dpcm (kasutada saab eesliiteid „*min-*“ ja „*max-*“);
- *scan* – kasutatava TV-ekraani reaotusmeetod („*progressive*“ või „*interlace*“);
- *grid* – kas väljundseade on võrgustikuga (*grid*, kindel arv ridu ja sümboleid) või rastergraafika (*bitmap*) seade („1“ – *grid*, „0“ – muu variant).

Näiteks:

```
<link rel="stylesheet" type="text/css" href="printimiseks.css" media="print">
```

Favicon

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



Joonis 6 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

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 enamusi 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="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 ">
```

Vaikeväärtused hüperlinkidele

Juhul kui suur osa linke (hüperlinke) 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.

Näiteks:

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

või

```
<base target="_blank">
```

või

```
<base href="http://www.seejasee.ee/" 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! Seega, kui näiteks `<base>` elemendiga määratakse veebilehtede avanemine uues aknas (vahekaardil) ja erandina vajatakse avamist samas aknas (vahekaardil), siis tuleb konkreetset hüperlingil kasutada ka atribuuti **target** väärtust „*self*“.

Kujunduse lisamine

Veebilehtede kujundus lisatakse tänapäeval küll eelkõige väliseid CSS stiililehti kasutades, kuid vajadusel saab CSS-stiili kirjutada ka otse HTML-koodi sisse, selleks kasutatakse elementi `<style>`. Näiteks:

```
<style>
  veebilehe erinevate elementide kujundusreeglid
</style>
```

NB! Veebilehel võib olla mitu `<style>` elementi.

Sellel elemendil saab kasutada ka mõningaid atribuute:

- *media* – määratleb meedia/väljundseadme, mille jaoks on kujundus loodud. Võimalikud väärtused on samad, mis veebilehe päise elemendi `<link>` samal atribuudil.
- *scoped* – määrab kujunduse vaid antud `<style>` elemendi vanem-elementile (mis teda sisaldab, *parent*) ja selle tütarelementidele.

NB! Atribuut *scoped* laseb `<style>` elementi kasutada lehe mistahes elemendi sees.

Näiteks:

```
<div>
  <style scoped>
    <div>-elemendi ja selle tütarelementide kujundusreeglid
  </style>
</div>
```

HTML dokumendi keha elemendid

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

Erinevad alajaotused

Järgnevalt kirjeldame veebilehe osi, mis jaotavad veebilehe semantilisteks osadeks (seksioonideks, navigatsiooni osaks jne), millede sisse paigutatakse sisu.

Lehe sisu põhiosa

Veebilehe sisu põhiosa defineerimiseks võib kasutada elementi `<main>`, mille sisuks võivad olla artiklid, ääremärkused jne.

Põhiosa peab sisaldama lehe unikaalset sisu, mis ei kordu lehestiku servadel ja mujal.

Element `<main>` ei tohi olla `<article>`, `<aside>`, `<footer>`, `<header>` ja `<nav>` elementide tütarelementideks.

Näiteks:

```
<main>
  <header> ... </header>
  <article>
    <section> ... </section>
    <section> ... </section>
    ...
  </article>
  <article> ... </article>
  <footer> ... </footer>
</main>
```

NB! Varasemalt võis veebilehel võib olla vaid üks <main> element!

Alates versioonist HTML 5.2 tohib veebilehel olla mitu <main> elementi, kuid korraga tohib nähtaval olla vaid üks. Ülejäänud peavad olema hidden atribuudiga peidetud. Näiteks:

```
<main hidden>
  <article>
  ...
  </article>
</main>
```

Artikkel

Artikkel ehk element <article> sisaldab iseseisvat sisu (mida saab ka ülejäänud lehest eraldi levitada). Tüüpiliselt on <article> elemendi sisuks foorumi või blogi postitused, uudislood või kommentaarid.

```
<article>
  <h1> Kaspersky Lab piilus 2030. aastasse: inimene ei kao</h1>
  <p> Tehnoloogia arengu praeguse tempo püsimine viib inimkonna väga kiiresti uude,
  täiendatud tegelikkuse, kvantarvutite, pilvehoidlate ning mitte inimeste, vaid masinate vahel
  peetavate infosõdade ajastusse.</p>
</article>
```

Sektsioon

Sektsioon jagab dokumendi sisu osadeks, näiteks peatükkideks. Sektsioon võib olla näiteks osa artiklist.

```
<section>
  <h2>Pealkiri</h2>
  <p>See siin ongi sisu, mis antud sektsiooni lisatud</p>
</section>
```

Ääremärkused/täiendav info

Ääremärkuste laadse täiendava info lisamiseks mingi sisuga elemendile kasutatakse elementi <aside>. Selle elemendi sisu peaks loomulikult olema seotud ümbritseva sisuga.

Näiteks:

```
<p>Tallinn oli keskajal tuntud kaubanduskeskus ning kuulus Hansa Liitu</p>
<aside>
  <h5>Hansa Liit</h5>
  <p>Hansa Liit oli 13.–17. sajandil tegutsenud Põhja-Saksamaa, Skandinaaviamaade,
  Madalmaade ja Liivimaa linnade kaubanduslik ja poliitiline liit.</p>
</aside>
```

<aside> elemendi sisu võib paigutada näiteks artikli (element <article>) kõrvale.

Päis ja jalus

Terve dokumendi või selle sektsiooni sissejuhatava informatsiooni või navigatsiooni jaoks kasutatakse päist ehk elementi `<header>`. Näiteks:

```
<section>
  <header>
    <h1>Esimene peatükk</h1>
    <p>See siin on pikem jutt päises</p>
  </header>
</section>
```

Terve dokumendi või sektsiooni jalus defineeritakse elemendiga `<footer>`. Jalusesse lisatakse info teda sisaldava elemendi kohta (info autori kohta, dokumendi loomise kuupäev jms peenes kirjas info). Näiteks:

```
<section>
  <footer>
    <p>Selles peatükis oli hulk tarka juttu!</p>
  </header>
</section>
```

Navigatsiooniplokk

Hüperlinkide kogumi ehk navigatsiooniploki defineerimiseks kasutatakse elementi `<nav>`. Element on kasulik näiteks vaegnägijatele mõeldud ekraanilugejate puhul, mis võivad selliselt grupeeritud linkide ettelugemise vahele jätta.

Näiteks:

```
<nav>
  <a href="teine.html">Teine</a>
  <a href="kolmas.html">Kolmas</a>
  <a href="neljas.html">Neljas</a>
</nav>
```

Täiendav info

Täiendava informatsiooni, mida kasutaja saab soovi korral avada ja vaadata, lisamiseks veebilehele on element `<details>`.

Sellel elemendil saab kasutada ka üht atribuuti:

- *open* – elemendi sisu on veebilehel kohe avatud (*open* ehk *expanded*).

Täiendavale infole nähtava pealkirja määramiseks kasutatakse elementi `<summary>`, mis peab olema elemendi `<details>` tütarelemendiks.

Näiteks:

```
<details open>
  <summary>HK</summary>
  <p>HK tähistab Haapsalu kolledžit, mis on Tallinna Ülikooli üks üksus. Haapsalu kolledžis
  õpetatakse 4 õppekava.</p>
</details>
```



Joonis 7 Täiendav info suletud



HK tähistab Haapsalu kolledžit, mis on Tallinna Ülikooli üks üksus. Haapsalu kolledžis õpetatakse 4 õppek

Joonis 8 Täiendav info avatud

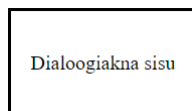
NB! Täiendava info sisuks võib kasutada mistahes infot!

Dialoogiaken

Erinevatel veebilehtedel on üsna tavaline vajadus avada dialoogiaknaid, modaalaknaid kasutajale erineva info kuvamiseks. HTML 5.2 pakub selleks uut elementi `<dialog>`. Sellel elemendil on tõeväärtusega (*boolean*) atribuut `open`, mis määrab dialoogiakna avatuks. Näiteks:

```
<dialog open>
  <p>Dialoogiakna sisu</p>
</dialog>
```

Vaikimisi näidatakse `<dialog>` elementi musta raamjoonega ristkülikuna.



Joonis 9 `<dialog>` elemendi vaikimisi välimus

Dialoogiakna avamiseks, sulgemiseks ja kõige pealmise modaalaknana avamiseks on kasutada meetodid `show()`, `close()` ja `showModal()`.

NB! 2022 aasta jaanuari seisuga toetavad `<dialog>` elementi veebilehitsejad Google Chrome MS Edge ja Opera. Firefox'i puhul on see vaja konfiguratsioonis lubada ning Safari seda elementi ei toeta.

Alajaotus ehk allüksus

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

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

Alajaotus on plokielement (*block*)!

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

Peidetud sisu javascriptiga näitamiseks

Et lisada veebilehele sisu, mis alguses lehe laadimisel peab olema peidetud ning mida võib olla vaja lausa korduvalt kasutada, näidata, kasutatakse spetsiaalset malli elementi `<template>`.

Näiteks:

```
<template>
  <h2>Üllatav sisu</h2>
  <p>Kes oleks seda noodanud!</p>
  
</template>
```

Tekst

Järgnevalt ülevaade erinevatest tekstielementidest.

Pealkirjad

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

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

Pealkirjad on plokielemendid (*block*)!

Tekstilõigud

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

```
<p>Lõigu jagu teksti</p>
```

Lõigu element on plokielement (*block*)!

NB! Lõigu elemendi `<p>` tütarelementidena saab kasutada vaid rea-elemente (*inline*).

Vajadusel saab lisada manuaalse reavahetuse (*manual line break*), mille puhul alustatakse uut rida kuid jätkub sama lõik. Selleks kasutatakse tühja elementi `
` (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 ``, mis ise visuaalselt midagi ei muuda kuid mille jaoks saab luua css stiile. Näiteks:

```
<p>Tekstilõik, milles <span style="color:red">eriline fraas</span> on eraldi välja toodud</p>
```

Teksti poolitamine

Poolitamine on seni veebis olnud problemaatiline kuid HTML5 lisab uue elemendi `<wbr>` (*Word Break Opportunity*), mis tuleb lisada pikkades sõnades neile kohtadele, kus poolitamine oleks sobilik.

Näiteks:

```
<p>Siin tekstis on pikk sõna desoksü<wbr>ribo<wbr>nukleiin<wbr>hape, mida võib poolitada päris mitmest kohast!</p>
```

NB! Poolitusmärki siiski ei lisata!

NB! Seda elementi ei toeta Internet Explorer!

Loendid

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

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

Nummerdatud loend

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

Näiteks nummerdatud loendi kirjeldus ja selle tulemus:

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

1. Esimene loendi element
2. Teine loendi element
3. Kolmas loendi element

Nummerdatud loendile saab määrata, millega numeratsioon algab. Selleks kasutatakse atribuuti `start`, mis oli HTML 4.01 standardis iganenuks tunnustatud kuid nüüd taas kasutusel! Selle atribuudi väärtuseks on soovitud täisarv. Näiteks:

```
<ol start="17">
  <li>Esimene loendi element</li>
  <li>Teine loendi element</li>
  <li>Kolmas loendi element</li>
</ol>
```

17. Esimene loendi element
18. Teine loendi element
19. Kolmas loendi element

Vajaduse korral saab loendi ka tagurpidisesse järjestusse seada (numeratsioon kahanevas järjekorras). Selleks kasutatakse atribuuti `reversed`, mille ainsaks võimalikuks väärtuseks on „*reversed*“.

Täpploend

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

Näiteks täpploendi kirjeldus ja selle tulemus:

```
<ul>
  <li>Esimene loendi element</li>
  <li>Teine loendi element</li>
  <li>Kolmas loendi element</li>
</ul>
```

- Esimene loendi element
- Teine loendi element
- Kolmas loendi element

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:

```
<ol>
  <li>Esimene loendi element
    <ul>
      <li>Esimene alamelement</li>
      <li>Teine alamelement</li>
    </ul>
  </li>
  <li>Teine loendi element
    <ul>
      <li>Esimene alamelement</li>
      <li>Teine alamelement</li>
    </ul>
  </li>
  <li>Kolmas loendi element</li>
</ol>
```

1. Esimene loendi element
 - Esimene alamelement
 - Teine alamelement
2. Teine loendi element
 - Esimene alamelement
 - Teine alamelement
3. Kolmas loendi element

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

Kirjelduste loend

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

NB! Varasemate HTML versioonide puhul nimetati seda definitsioonide (*definition*) loendiks!

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 palju 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 10 Definiitsioonide 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. Lisaks on kasutada eeldefineeritud stiilid arvutivaldkonnaga seotud tekstilõikude kujundamiseks ning üldiste tsitaatide ja definiitsioonide jaoks.

- Paks tekst (*bold*), element ``. HTML 5 puhul tuleb seda kasutada n.ö. viimase võimalusena kui muud võimalused (näiteks `` või ``) ei sobi. Näiteks:
tavatekst `paks tekst` tavatekst jätkub
- Kaldkiri (*italic*), element `<i>`. HTML 5 korral kasutatakse seda, kui on tarvis teksti osa ülejäänust erinevaks muuta ning teised võimalused (näiteks `<cite>`, `<dfn>`, `<q>` või `<small>`) ei sobi. Näiteks:
tavatekst `<i>eriline</i>` tavatekst jätkub
- Rõhutatud kiri (tavaliselt kaldkirjas), element ``, näiteks:
tavatekst `rõhutatud` tavatekst jätkub
- Oluline tekst, element ``. Varasemas HTML 4.01 standardis tähistas lihtsalt rõhutatud teksti. Näiteks:
tavatekst `oluline tekst` tavatekst jätkub
- Väiksem tekst, element `<small>`, näiteks:
tavatekst `<small>väiksem tekst</small>` tavatekst jätkub
- Ülainedeks (*superscript*), element `<sup>`, näiteks:
tavatekst `^{ülainedeks}` tavatekst jätkub
- Alainedeks (*subscript*), element `<sub>`, näiteks:
tavatekst `_{alainedeks}` tavatekst jätkub
- "Vahelelisatud" (*inserted*) tekst, element `<ins>` (tavaliselt allajoonitud).
Kasutada saab kahte atribuuti:
 - *cite* – väärtuseks URL dokumendini, milles selgitatakse, miks teksti lisati;
 - *datetime* – aeg, millal teksti lisati (YYYY-MM-DDThh:mm:ssTZD).Näiteks:
tavatekst `<ins cite="http://www.seejasee.ee/muudatused.txt">uus tekst</ins>` tavatekst jätkub
- "Kustutatud" (*deleted*), element `` (tavaliselt läbikriipsutatud). Kasutatakse samu atribuute nagu elemendil `<ins>`. Näiteks:

tavatekst `<del cite="http://www.seejasee.ee/kustutatud.txt">alaindeks` tavatekst jätkub

- Väär, ebakorrektnene või ebaoluline tekst, element `<s>` (tavaliselt läbikriipsutatud).

Näiteks:

```
<p>Osa tekstist on <s>vale</s>tõde!</p>
```

- Teksti suund, element `<bdo>` (*Bi-Directional Override*), millel on argument `dir` (väärtused `ltr` – vasakult paremale või `rtl` – paremalt vasakule), näiteks:

```
<bdo dir="rtl">See on tagurpidine tekst!</bdo>
```

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

```
<h1><strong><i>Ülimalt</i> rõhutatud </strong> eriline <small>pealkiri</small> </h1>
```

Mis annab tulemuseks:

***Ülimalt* rõhutatud eriline pealkiri**

- 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>
```

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>
```
- Muutujate kirjapanekuks (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:

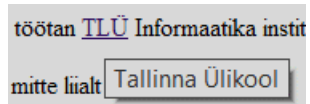
```
<kbd>Sisestame klaviatuurilt vajalikud väärtused</kbd>
```

Tsitaadid, definitsioonid

Tsitaatide ja lühendite vormindamiseks on samuti loodud hulk eeldefineeritud stile.

- Lühend, veebilehel näidatakse lühendit, hiirega sellele liikumisel tuuakse nähtavale pikk variant, element `<abbr>`, millel on vajalik atribuut `title`. Näiteks:

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



Joonis 11 Lühendi kasutamine veebilehel (täendus ilmub, kui hiirekursor lühendi peale liigutada)

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

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

Kui `<address>` element on lisatud `<article>` elemendi sisse, siis defineerib ta artikli autori/omaniku kontaktinfo!

- 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>
```

NB! Jutumärgide välimus sõltub keelest, näiteks inglisekeelsel tekstil on jutumärgid tsitaade ees ja järel mõlemad üleval, eestikeelsel aga tsitaadi ees all!

- Tsitaat (tekst kursiivis), element `<cite>`, näiteks:

```
<cite>Vähem on rohkem!</cite>
```
- Definiitsioon (tekst esitatakse tavaliselt kaldkirjas), element `<dfn>`, näiteks:

```
<dfn>internet on ülemaailmne arvutivõrke ühendav võrk</dfn>
```

Esiletõstetud tekst

HTML5 pakub uut elementi esiletõstetud (*marked, highlighted*) teksti jaoks. Element `<mark>` tõstab teksti esile (*highlight*) justkui see oleks paberil markeriga märgitud.

Näiteks:

```
<p>Ärgem unustagem, et veebileht <mark>peab valideeruma</mark>!</p>
```

Ärgem unustagem, et veebileht **peab valideeruma!**

Joonis 12 Märgitud tekst

NB! Suurem osa veebilehitsejaid kasutab selle elemendi puhul vaikimisi musta teksti kollasel taustal!

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, küllastamata link sinist värvi, küllastatud link lillakat ning aktiivne link punast värvi.

Erinevalt varasematest HTML versioonidest, on HTML 5 puhul element `<a>` kasutusel ainult hüperlingina ja ankruna (*anchor*) ehk järjehoidjana enam ei toimi.

Elemendil `<a>` saab kasutada järgmiseid atribuute (kohustuslikke pole):

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

Näiteks lisab järgnev rida veebilehele teksti "W3C koduleht" millel klõpsates avataksegi vastav koduleht:

```
<a href="http://www.w3.org">W3C koduleht</a>
```

Linki saab lisada ka teistele objektidele, näiteks pildile! Näiteks järgmine rida lisab veebilehele pildi, millel klõpsates avatakse W3C koduleht:

```
<a href="http://www.w3.org"> </a>
```

- *hreflang* – võimaldab määrata lingitud ressursi keelt (kasutatav vaid koos atribuudiga *href*). Näiteks:

```
<a href="http://www.tlu.ee" hreflang="et">TLÜ koduleht</a>
```

Kõigi kasutatavate keelte koodid leiab näiteks aadressilt:

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

- *target* – võimaldab määrata akent, kus link avatakse (kasutatav vaid koos atribuudiga *href*). Selle atribuudi võimalikud väärtused on:
 - *_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);

Näiteks lisab järgmine rida veebilehele lingi, mis avab W3C kodulehe uues aknas:

```
<a href="http://www.w3.org" target="_blank">W3C koduleht</a>
```

NB! Lingi atribuutide *href* ja *target* jaoks saab ka vaikeväärtuseid seada, milleks kasutatakse veebilehe päise `<head>` tütarelementi `<base>` (selle kirjeldus on päise elementide peatükis)!

- *rel* – määrab käesoleva ja lingitud ressursi seose. Kasutatavad väärtused on:
 - *alternate* – sama dokumendi alternatiivne versioon;
 - *author* – link käesoleva dokumendi autorile;
 - *bookmark* – järjehoidja seadmiseks kasutatav link;
 - *help* – link abiinfole;
 - *license* – link dokumendi litsentsitingimustele;
 - *prev* – loogiliselt käesolevale eelnev dokument;
 - *next* – loogiliselt käesolevale järgnev dokument;
 - *nofollow* – link, mida näiteks otsingumootor ei peaks järgima (reklaam vms);

- *noreferrer* – veebilehitseja ei peaks lingi järgimisel http päist saatma (*the browser should not send a HTTP referer header if the user follows the hyperlink*);
- *prefetch* – lingitud dokumenti tuleks puhverdada;
- *search* – käesoleva dokumendi otsinguvahend;
- *tag* – käesoleva dokumendi võtmesõna (*tag*).
- *type* – lingitud ressursi MIME (*Multipurpose Internet Mail Extensions*) tüüp.

Näiteks:

```
<a href="www.seejasee.ee/film.mpg" type="video/mpeg" />
```

MIME tüüpide nimekirja leiab näiteks: http://en.wikipedia.org/wiki/Internet_media_type või: <http://www.webmaster-toolkit.com/mime-types.shtml> või <http://reference.sitepoint.com/html/mime-types-full>

HTML5 on hüperlinkidele lisatud veel kaks uut võimalikku atribuuti:

- *media* – määratleb meedia/väljundseadme, mille jaoks on lingitud ressurss optimeeritud. Võimalikud väärtused on samad, mis veebilehe päise elemendi <link> samal atribuudil.

Näiteks:

```
<a href="www.seejasee.ee/lahja.html" media="screen and (min-color-index:256)" />
```

- *download* – lingitud ressurss laetakse alla kasutaja seadmesse, väärtus pole kohustuslik aga võib määrata faailinime.

Näiteks:

```
<a href="/pildid/minupilt.jpg" download>
```

Lingid sama lehe piires

Linke saab lisada ka ühe veebilehe piires selle erinevatele osadele! Selleks tuleb HTML elementidel kasutada atribuuti *id*, neile selle abil nimed määrata ning seejärel saabki selle nime järgi lingiga viidata!

Näiteks määrame veebilehe ühele pealkirjale nime „osa2“:

```
<h2 id="osa2">Veebilehe teine osa</h2>
```

Seejärel saab luua lingi, mis sellele pealkirjale viitab:

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

Teise HTML dokumendi kindla nimega osale viitamiseks tuleb elemendi nimi lisada dokumendi aadressi lõppu! 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>
```

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).

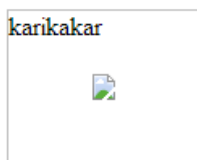
Pildi lisamiseks on `` element, mis on nn tühi element (*empty*) (pole eraldi lõpumärgeid) ja millel on kaks kohustuslikku atribuuti:

- *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 [Joonis 13].

Pildi element võib seega välja näha selline:

```

```



Joonis 13 Pildi koht veebilehel, pilt puudu, näha on alternatiivtekst (Google Chrome)

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

- *width* – laius pikselites.
- *height* – kõrgus pikselites.

Näiteks:

```

```

NB! Erinevalt varasematest HTML-versioonidest ei saa laiuse ja kõrguse määramisel kasutada %-märki!

HTML 5 puhul on kasutada ka uus atribuut:

- *crossorigin* – lubab elemendi `<canvas>` sees kasutada pilte kolmandate osapoolte veebilehtedelt.

Pildiga võib muuta interaktiivseks sidudes sellega `<map>` elemendi, selleks kasutatakse atribuuti:

- *usemap* – määrab, mis nimega `<map>` element selle pildiga seotakse.

Pilt võib ka ise kaardina (*map*) käituda, sellisel juhul saadetakse pildil klõpsamisel koordinaadid serveripoolsele rakendusele (php, asp vms). Selleks on atribuut:

- *ismap* – muudab pildi *server-side image-map*-iks.

Näiteks:

```
<a href="kohtkaardil.php">
  
</a>
```

NB! Atribuuti *ismap* saab kasutada vaid elemendi `<a>` sees oleval pildi puhul!

Kohanduv pilt

Kohanduva veebi (*responsive web*) jaoks pakub HTML5 uut elementi `<picture>`, mis laseb ühe lähtefaili asemel määrata mitu.

Elemendil `<picture>` on kahte tüüpi tütarelemente: `<source>` ja ``.

Tütarelement `<source>` määrab mingisugusele seadmele/ekraanisuurusele vastava pildifaili. Neid elemente võib olla mitu. `<source>` elemendil on järgmised atribuudid:

- *srcset* – määrab pildifaili URL-i (kohustuslik atribuut);
- *media* - määratleb meedia/väljundseadme, mille jaoks vastav pilt mõeldud on. Kõik selle atribuudi väärtused on samad, mis veebilehe päise elemendil `<link>` vastaval atribuudil.
- *sizes* – üks laius, üks meediapäring laiuse kirjeldusega või komadega eraldatud meediapäringute loend;
- *type* – MIME tüüp.

Brauser valib esimese sobiva pildifaili ning ignoreerib ülejäänuid.

Teise tütarelemendi tüübina on ``, mis peab olema viimane (ainult üks), et tagada ühilduvus brauseritega, mis ei toeta `<picture>` elementi. `` elementi kasutatakse ka, kui ükski `<source>` element pole sobiv.

NB! Tütarelement `` peab olema viimane!

Näiteks:

```
<picture>
  <source media="(min-width: 900px)" srcset="suur_pilt.jpg">
  <source media="(min-width: 4005px)" srcset="tilluke_pilt.jpg">
  
</picture>
```

NB! Meediapäring peab olema sulgudes!

Fotode ja jooniste tähistamine veebilehel

Sellise sisu nagu fotod, joonised, diagrammid jms tähistamiseks on HTML 5 puhul kasutusele võetud element `<figure>`.

Elemendi `<figure>` esimeseks tütarelemendiks on pilt `` või `<picture>`, või ka tabel `<table>`.

Teise tütarelemendina kasutatakse piltide jms pealdiste (*caption*) lisamiseks elementi `<figcaption>`.

Näiteks:

```
<figure>
  
  <figcaption>Veebilehtede kursuse tublimad osalejad</figcaption>
</figure>
```

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 kohustuslik atribuut:

- *name* – loodava „kaardi“ nimi, mille abil see vastava pildi elemendiga (``) seotakse!

Näiteks:

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

NB! Kui `<map>` elemendil on ka atribuut *id*, siis HTML5 puhul peab sellel olema sama väärtus, mis atribuudil *name*!

Loodud „kaardi“ sidumiseks vastava pildiga tuleb elemendile `` 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).
- **NB!** Kui pilti, millele „kaart“ lisatakse, ei näidata originaalsuuruses vaid väiksemana, siis peab ka koordinaadid samas proportsioonis ümber arvutama!

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>
```

HTML5 lisab uute atribuutidena:

- *hreflang* – võimaldab määrata lingitud ressursi keelt.
- *media* – määratleb media/väljundseadme, mille jaoks on lingitud ressurss optimeeritud. Vaikeväärtusena on „all“. Kõik väärtused on samad, mis veebilehe päise elemendil <link> vastaval atribuudil. Näiteks:

```
<area shape="rect" coords="25,250,50,50" alt="kandiline koht" href="www.seejasee.ee"
media="screen and (min-color-index:256)" />
```

- *rel* – määrab käesoleva ja lingitud ressursi seose. Kasutatavad väärtused on samad, mis hüperlingi element vastaval atribuudil.
- *type* – lingitud ressursi MIME (*Multipurpose Internet Mail Extensions*) tüüp.

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>
```

esimese rea, vasaku lahtri sisu	esimese rea, parema lahtri sisu
teise rea, vasaku lahtri sisu	teise rea, parema lahtri sisu

Joonis 14 Lihtne tabel (lisatud on raamjoon)

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. Vaikimisi on tegemist paksus kirjas keskele joondatud tekstiga.

```
<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>
```

esimese veeru päis	teise veeru päis
esimese rea, vasaku lahtri sisu	esimese rea, parema lahtri sisu
teise rea, vasaku lahtri sisu	teise rea, parema lahtri sisu

Joonis 15 Tabel veergude pealkirjadega (lisatud on raamjoon)

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*)!

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>
  ...
```

Veergude ja ridade grupeerimine

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 tabelis. kus on 5 veergu, saab veergusid grupeerida nii:

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

Ühele või mitmele veerule veergude grupi <colgroup> sees korraga atribuutide määramiseks kasutatakse 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>
  <colgroup>
    <col style="background-color:white" />
    <col span="2" style="background-color:gray" />
  </colgroup>
  <tr>
  ...
```

Kogu tabeli saab jagada kolmeks põhiosaks: päis (*head*), jalus (*footer*) ja keha (*body*), selleks kasutatakse <table> elemendi tütarelementidena elemente:

- <thead> – tabeli päis;
- <tbody> – tabeli kehaosa.

- `<tfoot>` – tabeli jalus;

Nende kasutamine võimaldab teatud juhtudel tabeli kehaosa päisest ja jalusest eraldi kerida ning pika tabeli printimisel korrata päist/jalust igal leheküljel.

NB! Kõik need kolm elementi peavad omakorda sisaldama vähemalt üht `<tr>` elementi. Päise element peab järgnema elementidele `<caption>` ja `<colgroup>` ning eelnema elementidele `<tbody>`, `<tfoot>` ja `<tr>`. Kehaosa element `<tbody>` peab järgnema elementidele `<caption>`, `<colgroup>` ja `<thead>`.

NB! Jaluse element `<tfoot>` tuleks mõnede allikate järgi paigutada elemendi `<thead>` järele enne elementi `<tbody>` kuid see annab valideerimisel vea. Seega on soovitatav kasutada järjekorda `<thead>`, `<tbody>` ja siis `<tfoot>`.

NB! HTML5 elementidel `<thead>`, `<tbody>` ja `<tfoot>` ei saa kasutada ühtki varasematest HTML versioonidest pärit atribuute.

HTML vahendid tabeli ja selle lahtrite vormindamiseks

Erinevalt varasematest HTML versioonidest on HTML5 puhul alles jäänud vaid üksikud tabeli vormingut puudutavad atribuudid.

Lahtrite elemendi `<td>` ja Veergude pealkirja elemendi `<th>` vormindamiseks saab kasutada järgmiseid atribuute:

- *colspan* – määrab üle mitme veeru antud lahter ulatub, väärtuseks täisarv.
- *rowspan* – määrab üle mitme rea antud lahter ulatub, väärtuseks täisarv.
- *headers* – määrab ühe või mitu pealkirja lahtrit, millega antud pealkirja lahter seotud on. Väärtuseks elemendi `<td>` identifikaator (atribuudi `id` väärtus). Ei oma visuaalset mõju, kasutatakse ekraanilugerite (*screen reader*) puhul.

Elemendil `<th>` on lisaks veel atribuut:

- *scope* – määrab, kas pealkirja lahter on pealkirjaks veerule (*column*), reale (*row*) või ridade/veergude grupile, väärtused vastavalt *col*, *colgroup*, *row* või *rowgroup*. Ei oma visuaalset mõju, kasutatakse ekraanilugerite (*screen reader*) puhul.

Tabeli näidis

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

```
<!-- Tabeli algus, raamjoon -->
<table>
<!--lisan pealdise - caption-->
  <caption>Tabeli loomise võimalusi tutvustav tabel</caption>
<!--algab tabeli päis-->
  <thead>
    <tr>
```

```
<th>Vasak veerg</th>
<th>Keskmine veerg</th>
<th>Parem veerg</th>
</tr>
</thead>
<!--Algab tabeli kehaosa-->
<tbody>
<tr> <!--Esimene rida-->
  <td>Esimene rida, vasak<br>Siin on ainsana kaks rida</td>
  <td>Esimene rida, keskmine</td>
  <td>Esimene rida, parem</td>
</tr>
<tr> <!--Teine rida-->
  <td>Teine rida, vasak</td>
  <td>Teine rida, keskmine</td>
  <td>Teine rida, parem</td>
</tr>
</tbody>
<!--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>
</table>
```

Välja näeb see järgmine:

Tabeli loomise võimalusi tutvustav tabel

Vasak veerg	Keskmine veerg	Parem veerg
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 16 Tabel, kus pea-aegu kõik elemendid kasutatud

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 paigutatigi eraldi HTML-dokument. HTML5 puhul on raamistik keelatud!

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

Sellel elemendil on kohustuslikuks atribuudiks:

- `src` – viide teisele dokumendile (URL). 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.

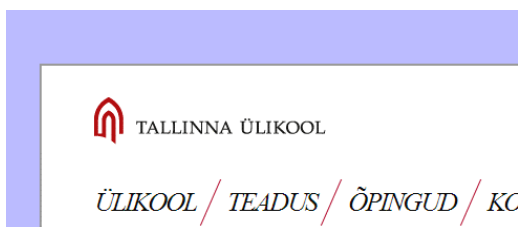
NB! Element `<iframe>` on reaelement (*inline*)!

`<iframe>` elementi saab CSS stiilide abil kujundada nagu teisi HTML elemente.

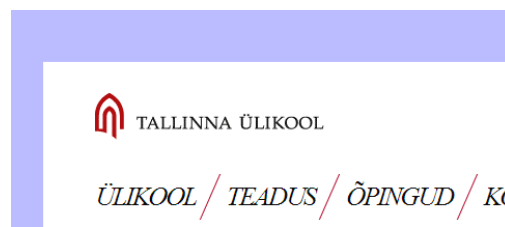
Lisaks kohustuslikule `src` atribuudile saab tal kasutada veel atribuute:

- `srcdoc` – elemendi sees näidatava sisu HTML-kood.
- `width` – laius, väärtuseks täpne arv pikselites.
- `height` – kõrgus, väärtuseks täpne arv pikselites.
- `seamless` – muudab elemendi sisuks oleva dokumendi sarnaseks ülejäänud veebilehega ehk eemaldab muidu kasutatava raamjoone. Näiteks:

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



Joonis 17 TLÜ veebileht elemendis `<iframe>` ilma atribuudita `seamless`



Joonis 18 TLÜ veebileht elemendis `<iframe>` atribuudiga `seamless`

- `sandbox` – võimaldab elemendi sisule piiranguid seada. Võimalikud väärtused:
 - `allow-same-origin` – lubab kasutada vaid dokumente samast domeenist;
 - `allow-top-navigation` – lubab sisuks oleval dokumendil navigeerida põhidokumendis;
 - `allow-forms` – lubab kasutada vorme, andmete edastamist (*form submission*);
 - `allow-scripts` – lubab skripte käivitada.

Näiteks:

```
<iframe src =“http://www.seejasee.ee/oluline“ sandbox=“allow-same origin“></iframe>
```

NB! Kasutada saab mitut väärtust! Need tuleb vaid üksteisest tühikuga eraldada!

- `name` – elemendi nimi.

Kasutades raami nime hüperlingis `target` atribuudi väärtusena, saab luua linke, mis avavad elemendis `<iframe>` erinevaid lehti!

Näiteks:

```
<iframe src="teineleht.html" name="lehekoht"></iframe>  
...  
<a href="http://www.tlu.ee" target="lehekoht">Tallinna Ülikool</a>
```

Horisontaalne joon

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

Joonis 19 Horisontaalne joon

See element toetab globaalseid ja sündmuste atribuute.

Meediaelemendid veebilehel

HTML 5 üks oluline uuendus on just parem multimeediumi tugi.

Pistikprogramme kasutavad objektid

Siinkohal peame silmas mitte-HTML (non-HTML) objekte nagu heliklipid, videod, Java apletid, ActiveX objektid, pdf dokumendid, Flash animatsioonid ja muidugi ka Microsoft Silverlight rakendused, mis vajavad erinevate pistikprogrammide (plug-in) kasutamist.

Element `<object>`

Selliste objektide lisamiseks võib kasutada `<object>` elementi, mis võimaldas varasemate HTML-versioonide puhul kasutada meediatüüpe, mida vastava HTML versiooni loomise ajal veel ei tuntud. Võimalik on ka teise veebilehe lisamine oma lehele.

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

Elemendil `<object>` saab kasutada järgnevat atribuute:

- *data* – kasutatava ressursi aadress (URL).
- *form* – määrab vormi(d), millesse antud `<object>` kuulub. Näiteks:
`<object data="reklaam.mpg" form="kysimus"></object>`
- *height* – objekti kõrgus pikslites.
- *width* – objekti laius pikslites. Näiteks:
`<object data="reklaam.mpg" width="480" height="360"></object>`
- *name* – objekti nimi.
- *type* – määrab kasutatava ressursi MIME tüübi.
- *usemap* – kasutatava `<map>` elemendi nimi.

Meediaobjektide mängimiseks kasutatavale pistikprogrammile saab erinevatid parameetreid edastada elemendiga `<param>`. Sellel elemendil on järgmised atribuudid:

- *name* – parameetri nimi.

- *value* – parameetri väärtus.

Näiteks:

```
<object data="reklaam.mpg" width="480" height="360">
  <param name="autoplay" value="true" />
</object>
```

Kasutusel võib olla hulk erinevaid parameetreid. Näiteks:

- *autoplay* – kas objekti mängimist alustatakse automaatselt.
- *controller* – kas kasutatakse juhtnuppe.

Kahjuks ei toimi see kõigi veebilehitsejatega päris ühtmoodi. Microsofti Internet Exploreri puhul on sageli tarvis kasutatavat ressursi teisiti kirjeldada. Sellisel puhul võib kirjutada ühe `<object>` elemendi IE jaoks ning selle sisse kirjutada teise `<object>` elemendi ülejäänud veebilehitsejate jaoks. Näiteks:

```
<object width="480" height="360">
  <param name="FileName" value="reklaam.mpg" />
  <param name="autoplay" value="false" />
  <param name="controller" value="true" />
  <object type="video/mpeg" data="reklaam.mpg" width="480" height="360">
    <param name="autoplay" value="false">
    <param name="controller" value="true" />
  </object>
</object>
```

Element `<embed>`

Pistikprogramme (*plug-in*) kasutavate meediaelementide lisamiseks (muuhulgas näiteks ka Adobe Flash) on olemas element `<embed>`, millel on järgmised atribuudid:

- *src* – meediafaili aadress (url);
- *type* – sisu tüüp (MIME tüüp);
- *height* – määrab sisu kõrguse, väärtus positiivne arv pikselites;
- *width* – määrab sisu laiuse, väärtus positiivne arv pikselites.

```
<embed src="animatsioon.swf" type="application/x-shockwave-flash" width="200"
height="150"/>
```

Heli ja video veebilehel

Järgnevalt vaatleme heli ja video kasutamist HTML 5 veebilehtedel.

Heliklipid veebilehel

Hetkel toetatakse HTML 5 poolt **mp3**, **wav** ja **ogg** vormingus helifaile. Helikliptide mängimiseks on element `<audio>`.

Elemendi sisuks võib panna teate selliste veebilehitsejate, mis `<audio>` elementi ei toeta, kasutajatele.

Näiteks:

```
<audio>
  Teie veebilehitseja ei toeta audio elementi!
</audio>
```

Sellel elemendil võib kasutada järgmiseid atribuute:

- *autoplay* – määrab, kas heli hakkab automaatselt mängima.
- *controls* – lisab juhtvahendid.
- *loop* – heliklipi korduvesitus.
- *muted* – heli vaigistatakse.
- *preload* – laseb heli mängimiseks koos veebilehe laadimisega valmis laadida (ignoreeritakse, kui on kasutatud ka atribuuti *autoplay*), väärtuseks *auto* – heliklipp laetakse automaatselt koos veebilehega, *metadata* – automaatselt laetakse vaid heliklipi meta-andmed, *none* – automaatselt ei laeta midagi.

Näiteks:

```
<audio controls>
  Teie veebilehitseja ei toeta audio elementi!
</audio>
```



Joonis 20 Audio element juhtnuppudega

NB! Kasutatava helifaili määramiseks on element `<source>`, mida kirjeldatakse peatükis „Audio- ja videofailide määramine“.

Video veebilehel

Veebilehtedel toetatakse **MP4** (H.264 video ACC heliga), **WebM** (VP8 video Vorbis heliga) ja **Ogg** (Theora video Vorbis heliga) videofaile. Video lisamiseks on element `<video>`. Elemendi sisuks võib panna teate selliste veebilehitsejate, mis `<audio>` elementi ei toeta, kasutajatele. Näiteks:

```
<video>
  Teie veebilehitseja ei toeta video elementi!
</video>
```

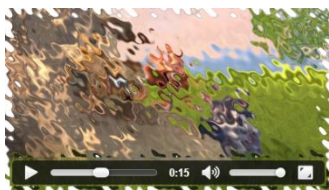
Sellel elemendil võib kasutada järgmiseid atribuute:

- *autoplay* – määrab, kas video hakkab automaatselt mängima kui on valmis., kasutatav väärtus on *auto*, *metadata*, *none*;
- *preload* – laseb video mängimiseks koos veebilehe laadimisega valmis laadida (ignoreeritakse, kui on kasutatud ka atribuuti *autoplay*), võimalikud väärtused on *auto* – video laetakse koos veebilehega; *metadata* – laetakse vaid meta-andmed; *none* – ei laeta midagi;
- *controls* – lisab juhtvahendid, kasutatav väärtus on *controls*;

- *loop* – laseb videot üha uuesti mängida, kasutatav väärtus *loop*;
- *muted* – video heli on vaigistatud;
- *poster* – määrab pildi, mida näidatakse kuni video laetakse, väärtuseks pildi URL;
- *height* – määrab videomängija kõrguse, väärtus pikselites;
- *width* – määrab videomängija laiuse, väärtus pikselites.

Näiteks:

```
<video width="320" height="240" controls preload="auto" muted poster="pildid/poster.jpg">
  Teie veebilehitseja ei toeta video elementi!
</video>
```



Joonis 21 Video element juhtnuppudega

NB! Kasutatava videofaili määramiseks on element `<source>`, mida kirjeldatakse peatükis „Audio- ja videofailide määramine“.

Audio- ja videofailide määramine

Kasutatava heli- või videofaili määramiseks kasutatakse reeglina elementi `<source>`, millel on järgmised atribuudid:

- *src* – heliklipi aadress (URL).

NB! Kuna kõik veebilehitsejad ei toeta kõiki failivorminguid samaväärselt, võib kirja panna mitu erineva *src* atribuudi väärtusega `<source>` elementi.

- *type* – heliklipi MIME-tüüp.
 - Audio puhul on kasutatavad tüübid: *audio/mpeg*, *audio/wav* ja *audio/ogg*.
 - Video puhul on kasutatavad tüübid: *video/mp4*, *video/webm* ja *video/ogg*.
- *media* – määratleb media/väljundseadme, mille jaoks on lingitud ressurss optimeeritud. Kõik väärtused on samad, mis veebilehe päise elemendil `<link>` vastaval atribuudil.

Näiteks heliklipi mängimiseks:

```
<audio controls preload="auto" loop>
  <source src="muusika.mp3" type="audio/mpeg">
  <source src="muusika.ogg" type="audio/ogg">
  Teie veebilehitseja ei toeta audio elementi!
</audio>
```

Näiteks videoklipi mängimiseks:

```
<video width="320" height="240" controls autoplay poster="pildid/poster.jpg">
  <source src="film.mp4" type="video/mp4">
  <source src="film.ogg" type="video/ogg">
  Teie veebilehitseja ei toeta video elementi!
</video>
```

Skriptiga loodud graafika ja animatsioon

Uue elemendina on HTML 5-s olemas `<canvas>`, mille sisuks luuakse skriptiga (reeglina JavaScript) „lennult“ (*on the fly*) joonistusi ja animeeritud graafikat. Selle elemendi sisuks võib lisada ka tavalist teksti, mida näitavad need veebilehitsejad, mis ei toeta elementi `<canvas>`.

Kasutada on kaks atribuuti:

- *height* – määrab elemendi kõrguse pikslites;
- *width* – määrab elemendi laiuse pikslites.

Näiteks:

```
<canvas id="canvas" width="800" height="500">
  Teie veebilehitseja ei toeta HTML5 canvas elementi
</canvas>
```

Joonistus luuakse selle elemendi sisse HTML elemendi `<script>` sisse kirjutatud skriptiga.

Arvväärtustega seotud elemendid

HTML 5 pakub veebiarendajale ka mõningaid arvväärtustega seotud elemente, mis võimaldavad väärtuseid graafiliselt väljendada või näiteks kuupäeva ja kellaajaga seotud väärtuseid masinloetaval kujul kodeerida.

Mõõdik

Arvväärtuseid või esitada etteantud skaalaga mõõdiku kujul. Selleks on element `<meter>`. Elemendi sisuks võib kirjutada andmed arvkujul nende veebilehitsejate jaoks, mis seda elementi ei toeta.

Elemendil `<meter>` on kohustuslikuks atribuudiks:

- *value* – mõõdikule määratud väärtus.

Näiteks:

```
<meter value="0.75">75%</meter>
```



Joonis 22 element `<meter>` väärtusega 0,75

NB! Väärtustena saab kirjutada kümnendmurde. Komakoha eraldajana tuleb kasutada punkti!

Lisaks saab kasutada atribuute:

- *min* – määrab skaala miinimumväärtuse, vaikeväärtus on 0;
- *max* – määrab skaala maksimumväärtuse, vaikeväärtus on 1;
- *low* – määrab piiri, millest allapoole jäävad loetakse madalateks väärtusteks;
- *optimum* – määrab optimaalse väärtuse;
- *high* – määrab piiri, millest ülespoole jäävad loetakse kõrgeteks väärtusteks;

Näiteks:

```
<meter min="0" low="25" optimum="70" high="75" max="100" value="15"></meter>
<meter min="0" low="25" optimum="70" high="75" max="100" value="35"></meter>
<meter min="0" low="25" optimum="70" high="75" max="100" value="55"></meter>
<meter min="0" low="25" optimum="70" high="75" max="100" value="75"></meter>
<meter min="0" low="25" optimum="70" high="75" max="100" value="95"></meter>
```



Joonis 23 Eelneva näidiskoodi mõõdikud veebilehitsejas Google Chrome



Joonis 24 Eelneva näidiskoodi mõõdikud veebilehitsejas Opera

- *form* – väärtuseks element `<form>` atribuudi `id` väärtus, määrab vormi, mille koosseisu see `<meter>` kuulub.

Edenemisriba

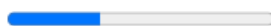
Mingisuguse protsessi edenemise näitamiseks saab kasutada edenemisriba (*progress bar*), mille jaoks on element `<progress>`. Juhuks, kui veebilehitseja seda elementi ei toeta, võib tema sisuks kirjutada edenemise väärtuse teksti kujul.

Kasutada on kaks atribuuti:

- *max* – määrab, mis on maksimaalne väärtus ehk millal protsess lõppenuks loetakse. Vaikeväärtuseks on 1;
- *value* – määrab edenemisriba hetkeväärtuse, väärtus peab jääma vahemikku 0 kuni atribuudi `max` väärtus.

Näiteks:

```
<progress value="35" max="100">Tehtud 35%</progress>
```



Joonis 25 Eelnevale koodinäitele vastav edenemisriba (Google Chrome ja MS Edge)

Vastavalt sellele, kas atribuut `value` on määratud või mitte, on edenemisribal kaks võimalikku olekut (*state*): määratamata (*indeterminate*) või kindlaksmääratud (*determinate*).

Erinevad veebilehitsejad näitavad edenemisriba erinevalt. Näiteks:



Joonis 26 Määramata olekus edenemisriba (Google Chrome ja MS Edge)



Joonis 27 Firefox'i edenemisriba



Joonis 28 Firefox'i määramata edenemisriba



Joonis 29 Internet Explorer'i edenemisriba



Joonis 30 Internet Explorer'i määramata edenemisriba

Aja märkimine

Aeg on selline asi, mida saab väljendada väga erinevalt. Võib kasutada rahvakalendrist pärinevaid päevade nimetusi või määrata arvude abil kindla kuupäeva ja kellaaja.

Aja märkimiseks arvutile arusaadaval viisil on loodud element `<time>`. See ei muuda visuaalselt midagi, kuid võimaldab muidu vaid inimesele arusaadavaid viiteid (näiteks „lastekaitsepäev“) kirja panna konkreetse kuupäeva ja/või kellaaja abil.

Kasutada on atribuut:

- *datetime* – mis määrabki täpselt elemendiga määratud aja.
Selle atribuudi väärtus pannakse kirja kujul: YYYY-MM-DDThh:mm:ssTZD
 - YYYY – aasta;
 - MM – kuu;
 - DD – kuupäev;
 - T – **vajalik** eraldaja, mida kasutatakse, kui määratakse ka kellaag;
 - hh – tunnid (näiteks 17 või 5pm);
 - mm – minutid;
 - ss – sekundid;
 - TZD – ajavööndi tähis.

Näiteks:

```
<p><time datetime="2018-06-01">Lastekaitsepäeval</time> sai tasuta loomaaeda.</p>
```

HTML elementide atribuudid

Eespool on käsitletud peamiselt neid atribuute, mis igal elemendil oma ja erilised on. Lisaks on HTML keeles hulk atribuute, mis on ühised praktiliselt kõigi elementide jaoks.

Varasemate HTML versioonide puhul esines osade atribuutide kasutamisel piiranguid, HTML 5 lubab reeglina kõiki atribuute kasutada kõigi elementide juures kuigi see pole alati vajalik.

Üldised atribuudid

Järgnevalt kirjeldatavaid üldiseid atribuute (*global attributes*) saab kasutada kõigi HTML elementide juures.

- *id* – väärtuseks on elemendile määratud unikaalne identifikaator (nimi). Näiteks:

```
<section id="Uudised">
```

NB! Atribuudi *id* väärtus peab sisaldama vähemalt üht tähemärki ja ei tohi alata numbriga!

- *title* – väärtuseks on tekst, lisainfo elemendi kohta. Näiteks:

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

- *style* – väärtuseks on css reastiil. Näiteks:

```
<p style="color:red">
```

- *class* – väärtuseks on elemendi kujundamiseks kasutatav css klassi nimi. Näiteks:

```
<p class="oluline">
```

- *tabindex* – määrab TAB klahviga sellele elemendile liikumise järjekorra. Väärtuseks on täisarv (1 on esimene). Näiteks:

```
<a href="saladus.html" tabindex="3">Saladuse avaldamine</a>
```

- *accesskey* – määrab kiirklahvi, mida vajutades saab antud elementi aktiveerida, sellele fookust seada.

```
<a href="saladus.html" accesskey="s">Saladuse avaldamine</a>
```

- *contenteditable* – määrab, kas elemendi sisu on muudetav, väärtuseks *true* või *false*. Näiteks:

```
<p contenteditable="true">Lisa siia oma kommentaar! </p>
```

NB! Kui elemendil pole atribuuti *contenteditable* määratud, siis päritakse (*inherit*) see ema-
elemendilt (*parent element*)!

Keelega seotud atribuudid

Atribuut *lang* on kasutusel tervel veebilehel või selle osades kasutatava keele deklareerimiseks!

Vastavalt W3C soovitusel tuleks igal veebilehel kasutatav keel deklareerida elemendi `<html>` algusmärgendis! Näiteks:

```
<html 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 näiteks:

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>`, `
`, `<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.

Näiteks:

```
<script>
  document.getElementById("silt").innerHTML = "See on tähtis teade!";
</script>
```

Sellel elemendil on rida atribuute:

- `async` – määrab, kas skripti täidetakse asünkroonselt ehk kohe, kui see on laetud või mitte. Tegemist on loogika-väärtusega (*boolean*) atribuudiga, ta on olemas (tõene ehk *true*) või puudub (väär ehk *false*);
- `type` – määrab skripti tüübi ja pole HTML 5 puhul kohustuslik (võimalikud väärtused on näiteks: `text/javascript`, `text/ecmascript`, `application/ecmascript`, `application/javascript` jms);
- `charset` – määrab skriptis kasutatud tähemärkide kooditabeli;
- `defer` – määrab, et skript ei käivitu enne, kui terve leht on laetud ja parsitud. Tegemist on loogika-väärtusega (*boolean*) atribuudiga, ta on olemas (tõene ehk *true*) või puudub (väär ehk *false*);
- `src` – URL, mis viitab välisele skriptifailile, näiteks JavaScript'i *.js failile.

Näiteks:

```
<script type="text/javascript" src="kood.js" charset="UTF-8" defer></script>
```

Lehe laadimisega seotud atribuudid

Veebilehe laadimisel (element `<body>`) saab kasutada järgmiseid atribuute:

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

Näiteks:

<body onload="yksFunktsioonJavaScriptiga()">

NB! Soovituslik on Javascript'i funktsioonid veebilehega siduda siiski otse Javascript'i failis!

Piltidega seotud sündmuste atribuudid

Piltidele ehk elemendile saab lisada vaid ühe sündmusega atribuudi:

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

Klaviatuuriga seotud sündmuste atribuudid

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>,
, <frame>, <frameset>, <head>, <iframe>, <meta>, <param>, <script>, <style> ja <title>!

Hiirega seotud sündmuste atribuudid

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>,
, <frame>, <frameset>, <head>, <html>, <iframe>, <meta>, <param>, <script>, <style> ja <title>!

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 näiteks:

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