

VEEBIRAKENDUSTE

LOOMINE

MTAT.03.230 (6 EAP)

1. Loeng

Helle Hein

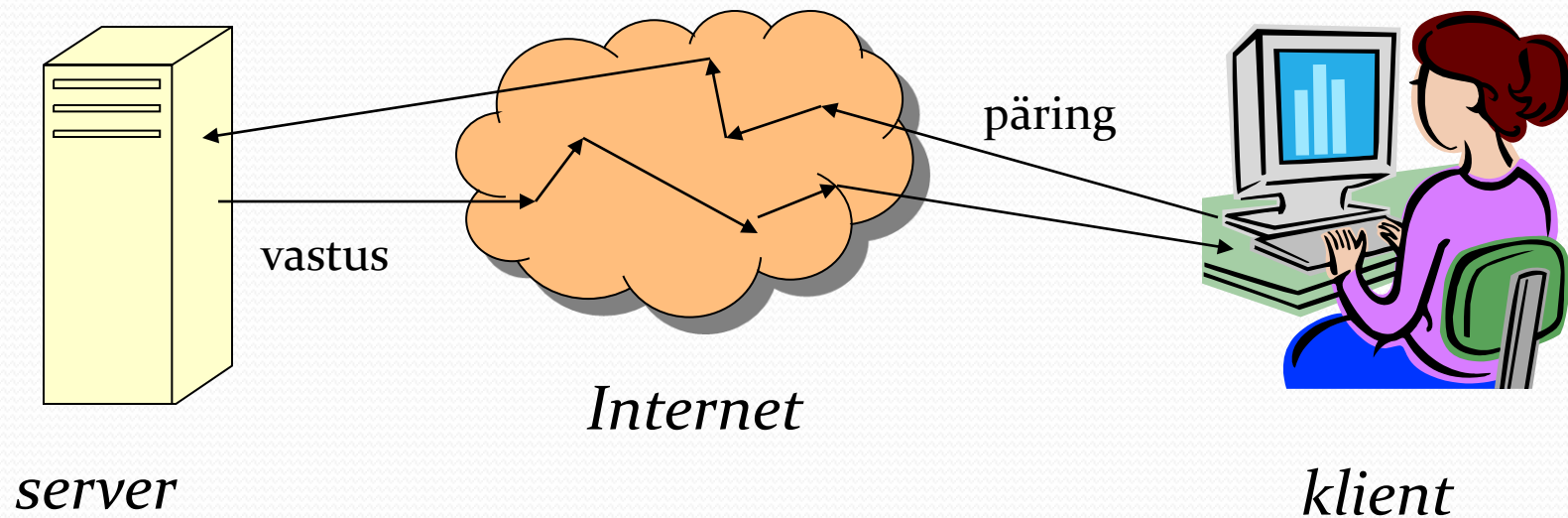
Teemad: HTML, CSS

- HTML ajalugu
- HTML vs XHTML
- HTML
- CSS
- World Wide Web Consortium (W₃C)

WWW - World-Wide Web

- Tim Berners-Lee - 1990, CERN
- Põhineb klient - server mudelil:
 - kliendi arvutis töötab veebilehitseja (brauser, nt IE või Mozilla Firefox), mis pärib infot serverilt
 - serveri arvutis töötab veebiserver (nt. Microsoft IIS või Apache), mis kuulab päringuid ja vastab päringutele

Klient-Server Mudel



Veebiprogrammeerimise elemendid

- HTML: Hyper-Text Markup Language. Teksti ja graafika vormistamine, seosed veebilehtede vahel.
- CSS: Cascading Style Sheets. Võimaldab veebilehe struktuuri ja vormistuse eraldamist.
- JavaScript: kliendipoolne programmeerimiskeel, mis võimaldab veebilehe dünaamilist sisu.

- Java
- Serveripoolne programmeerimine (CGI, PHP, ASP.NET, JSP, JSF,...). Võimaldavad juurdepääsu andmebaasi ja serveri teistele ressurssidele.

Hüpertekst

- Linkidega ühendatud dokumentide kollektsioon;
- Paul Otlet, philosophical treatise (1934);
- Vannevar Bush, hüpoteetiline Memex süsteem (1945)
- Ted Nelson tõi sisse mõiste *hüpertekst* (1968)

Märgistuskeeled (Markup Languages)

- Tekstile struktuuri lisamiseks
- Charles Goldfarb – SGML 1974
- Standard Generalized Markup Language, SGML (1986)
<http://www.sgmlsource.com/8879/n0029.htm>

HTML

- Lihtne, selged printsiibid;
- HTML kirjeldab dokumendi *loogilist struktuuri*;
- Brauseritel on olnud vaba voli tõlgendada märgiseid erinevalt;

- HTML on “kergekaaluline” failiformaat
- Faili ”Tere tudeng!” suurus:

HTML	277
MS Word 2007	10039
PDF	11 767
Postscript	11 274

```
<html>
  <head>
    <title>
      Tere
    </title>
  </head>
  <body>
    <p>
      Tere Tudeng!
    </p>
  </body>
</html>
```

HTML dokumendi üldine struktuur

```
<html>  
  <head>  
    <title>  
      Dokumendi pealkiri  
    </title>  
  </head>  
  <body>  
    ...  
  </body>  
</html>
```

HTML ajalugu

1992: **HTML 1.0, Tim Berners-Lee**

1993: Mosaic (multimeedia toetus),

1994: HTML 2.0, standard (samal aastal loodi W3C)

1995: Netscape omadused

1996: Netscape ja IE omadused

1996: HTML 3.2, brauserite sõja lõpp

1997: HTML 4.0, stiilide (CSS) sissetoomine

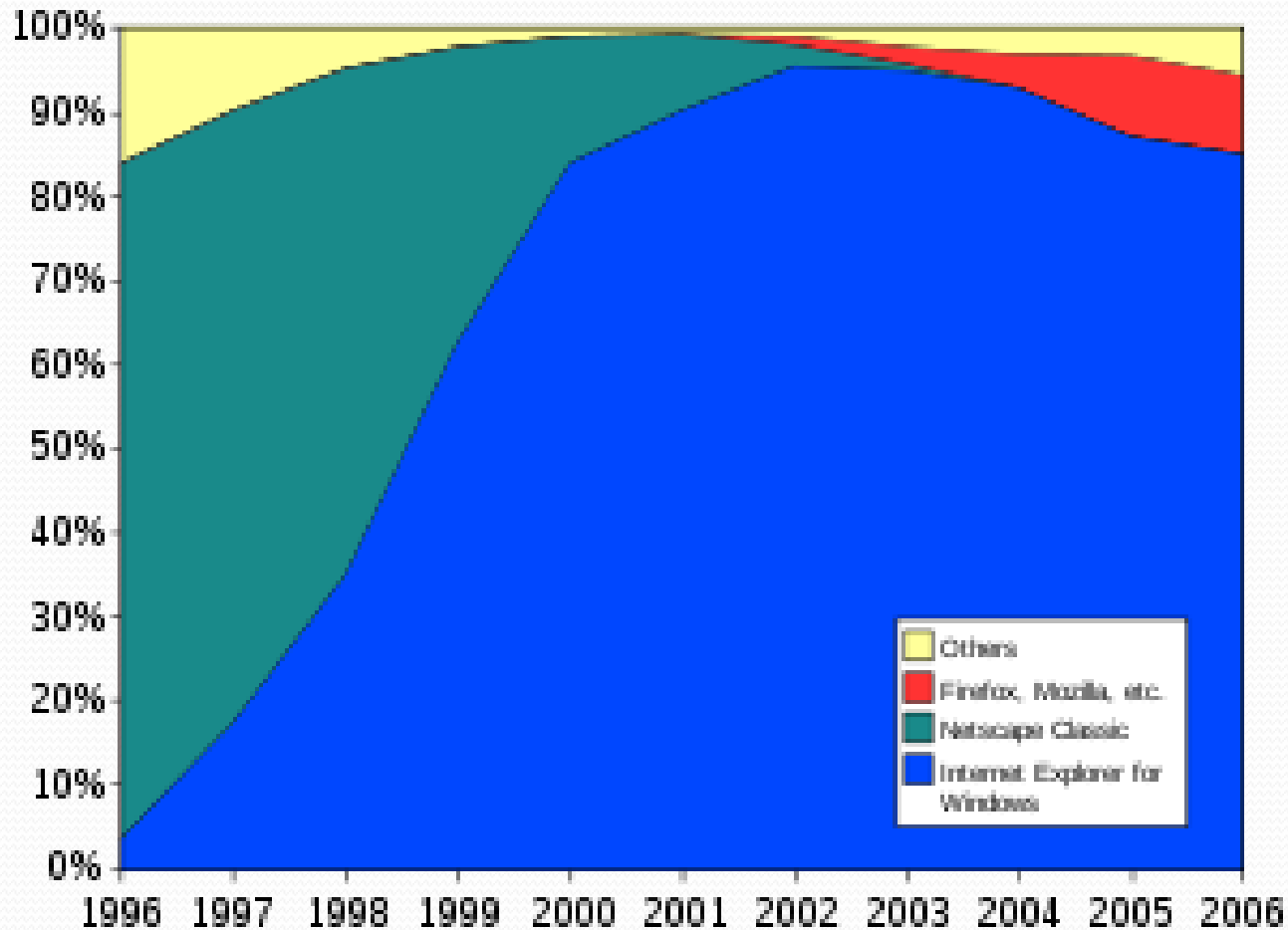
1999: HTML 4.01

2000: XHTML 1.0, HTML 4.01 versioon XML s

2001: XHTML 1.1

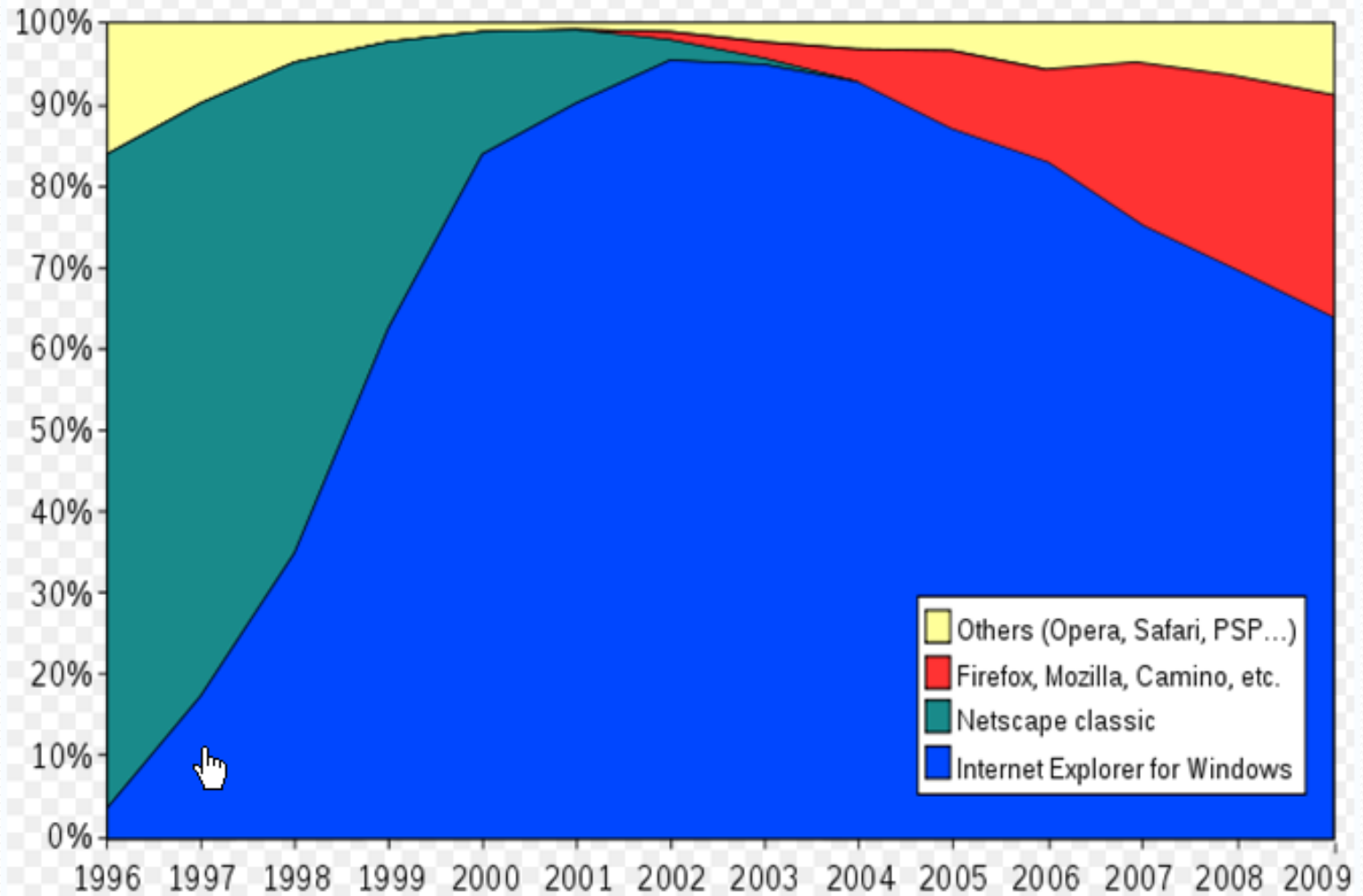
2002: XHTML 2.0, lihtsustatud ja üldistatud

Browser Wars



http://en.wikipedia.org/wiki/Browser_wars

Browser Wars



Keele defineerimine sisaldab kahte aspekti: süntaks ja semantika.

Arvutikeele *süntaks* defineerib, milliseid märgijadasid võib dokument sisaldada.

Keele *semantika* on kirjeldus, mida süntaktiliselt korrektse dokumendi elemendid tähendavad.

Keele semantika defineeritakse tavaliselt inimkeeles.
HTML 4.01 semantika on defineeritud samuti inimkeeles

<http://www.w3.org/TR/html401/>

HTML 4.01 defineerimiseks kasutatakse SGML (Standard Generalized Markup Language), mis muudeti standardiks 1986. See keel on ka liiga üldine HTML defineerimiseks. HTML dokumendi parsimine sisaldab elementide puu loomist.

Keele *abstraktne süntaks* defineerib keele abstraktse süntaksi puu tasemel. See määrab, millised:

- elemendid võivad puus olla;
- atribuudid võivad elemendil olla;
- võivad olla atribuutide väärtused;
- alamelemendid (child element) võivad olla ja millises järjekorras.

Keele *konkreetne süntaks* defineerib, kuidas see puu struktuur on esindatud keeles. HTML ja XHTML korral hõlmab see alama taseme detaile, nt milliseid märke võib kasutada märgistes, kas nimed on tõstutundlikud, kuidas atribuutide väärtusi kirjutada jne.

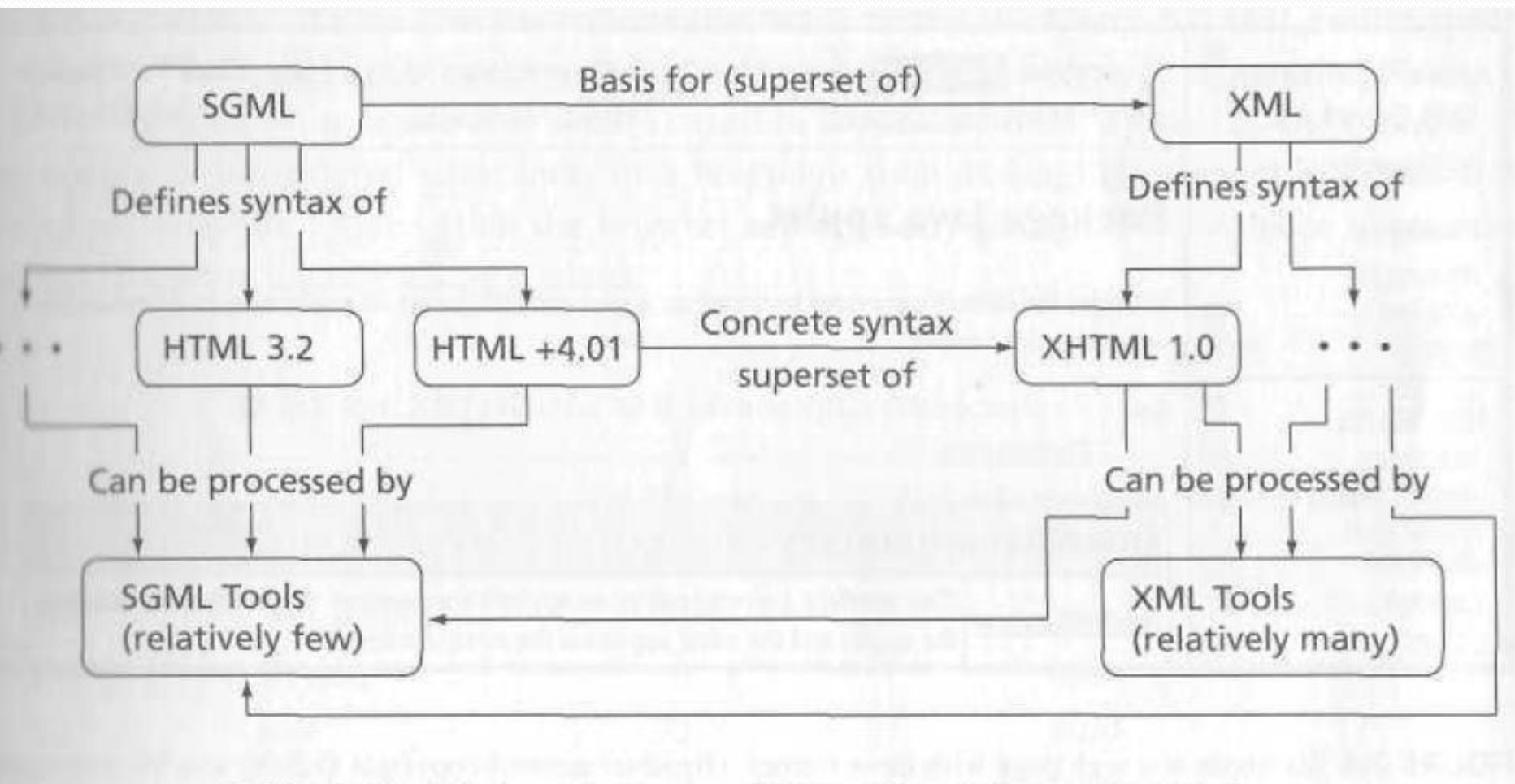
Keeled HTML 4.01 ja XHTML 1.0 on ekvivalentsed semantika ja abstraktse süntaksi tasemel, kuid **erinevad vaid konkreetse süntaksi tasemel.**

XHTML 1.0 vs HTML 4.01

Peamised erinevused konkreetse süntaksi tasemel on:

- Elemendi (märgis - *tag*) ja atribuudi nimed peavad olema väikeste tähtedega;
- Atribuutide väärtused peavad olema jutumärkides;
- Kõik elemendid tuleb lõpetada;
- Igal atribuudil on väärtus;
- Dokument on alati korrektne;

- Atribuutide väärtustes kaotatakse üleliigsed tühikud;
- Atribuuti *name* asendab atribuut *id*;
- XHTML 1.0 dokument on erijuht XML dokumendist



Raamatust J. Jackson Web Technologies

On kolm eri varianti XHTML 1.0 ja HTML 4.01 spetsifikatsioonidest:

Strict: See on W3C ideaal HTML jaoks alates 1997;

Transitional: Sisaldab Strict HTML ja mõningaid vananenud (deprecated) elemente ja atribuute. Need on niisugused elemendid, mida ei peaks kasutama kui võimalik.

Frameset: Sisaldab Transitional HTML, mis lubab brauseri kliendipiirkonnas alamaknaid (raame), nt Java API:

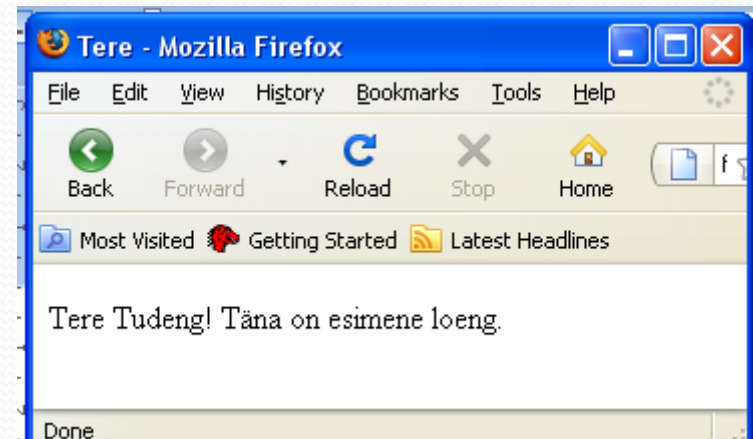
Soovituslikud deklaratsioonid on järgmised:

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

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

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

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang=
    "en-US" lang="en-US">
<head>
    <title>
        Tere
    </title>
</head>
<body>
    <p>
        Tere Tudeng!
        Täna on esimene loeng.
    </p>
</body>
</html>
```

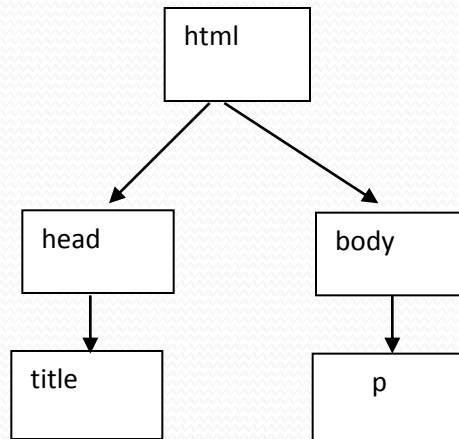


<!DOCTYPE – dokumendi tüübi deklaratsioon – HTML version, mida kasutatakse.

XHTML 1.0 standardi järgi peab `html` algusmärgis sisaldama atribuuti `xmlns` koos väärtusega:

```
<html xmlns=
```

HTML dokument esitab puud, mis antud näite korral on järgmine:



XHTML tekstilise info sees olevaid erimärke (white space) käsitletakse kui sõnade eraldajaid, nende jada asendatakse ühe tühikuga:

Märk	ASCII kood	Unicode standardväärtus(Hex)
Carriage return	13	000D
Line feed	10	000A
Space	32	0020
Tab	9	0009

```
<body>
```

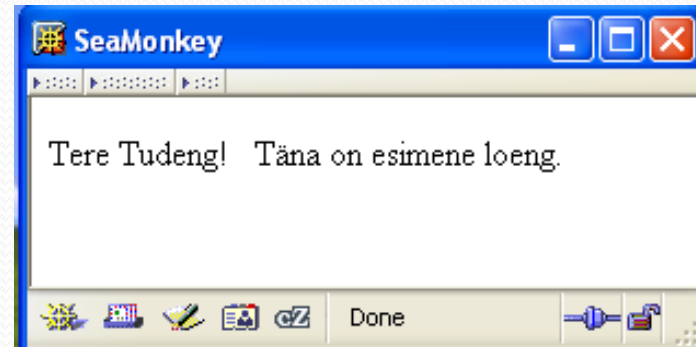
```
<p>
```

```
Tere Tudeng!
```

```
Täna on esimene loeng.
```

```
</p>
```

```
</body>
```



```
<p>
```

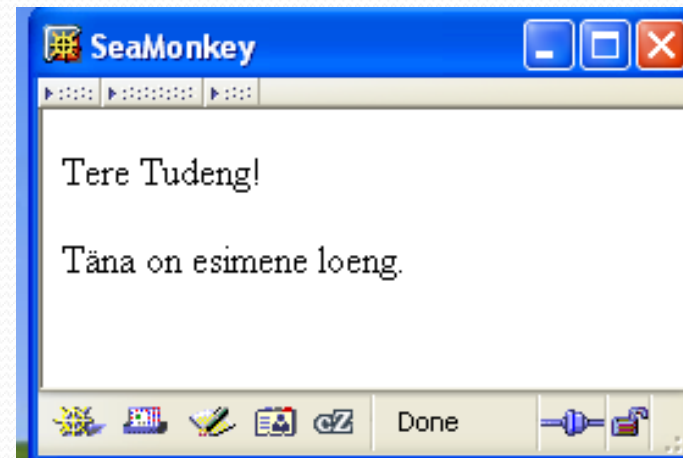
```
Tere Tudeng!
```

```
</p>
```

```
<p>
```

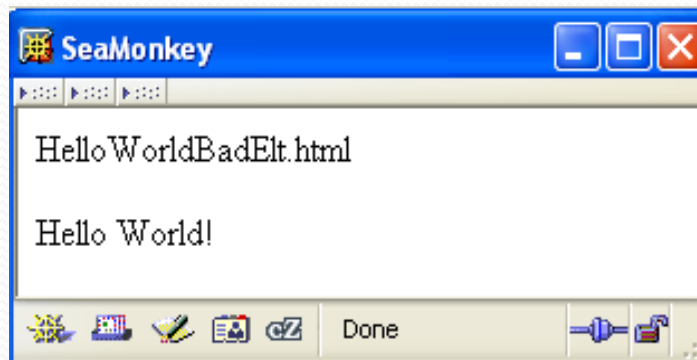
```
Täna on esimene loeng.
```

```
</p>
```



Brauserid ei kaeba selle üle, kui dokument sisaldab vigu vaid püüavad siiski näidata kogu lehte. Näiteks

```
<head>  
  <titl>  
    HelloWorldBadElt.html  
  </title>  
  Hello World!  
</head>
```



Kuidas lisada teksti nt märke <, > jne?

Selleks kasutatakse olemiviida (*entity reference*) tüüpi märkimist, mis algab märgiga & ja lõpeb semikooloniga ;

Märgiviida (*character reference*) korral järgneb märgile # Unicode standard väärtus.

Olemiviit

&#38;

>

<

"

'

©

α

∀

õ

ä

ö

ü

...

Märk

&

>

<

"

'

©

α

∀

õ

ä

ö

ü

Märgiviit (Decimal)

&

>

<

"

'

©

α

∀

õ

ä

ö

ü

Atribuudid

Kõik XHTML atribuudid peavad olema sama kujuga:

```
value = "See on atribuudi väärtus"
```

Võib kasutada nii ühe- kui ka kahekordseid jutumärke. Atribuudi väärtus peab algama tähega, võib sisaldada tähti ja numbreid ja nelja märki _:.- (alakriips, koolon, punkt, sidekriips).

HTML algusmärgis, mis on kooskõlas nii HTML 4.01 kui ka XHTML 1.0 ga, on järgmine:

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

Mõned põhilised HTML elemendid

- Päiseelemendid *h1*, *h2*, *h3*, *h4*, *h5*, *h6* – taseme järgi kahanev;
- Reavahtetus *br* – tühi element, ei tohi sisaldada midagi: `
`;
- Eelsisestatud tekst *pre* – enamuse brausereid arvestab kasutatud reavahtetusega ja kasutab *monospace* fonti – kasulik nt Java teksti näitamiseks;
- Teksti formaatimine: *span* – võimaldab eraldatud tekstile määrata stiili ja teisi atribuute;

```
<span style = "font-style:italic"> eraldi tekst</span>
```

Tekst blokina: *div*

Teksti rõhutamine:

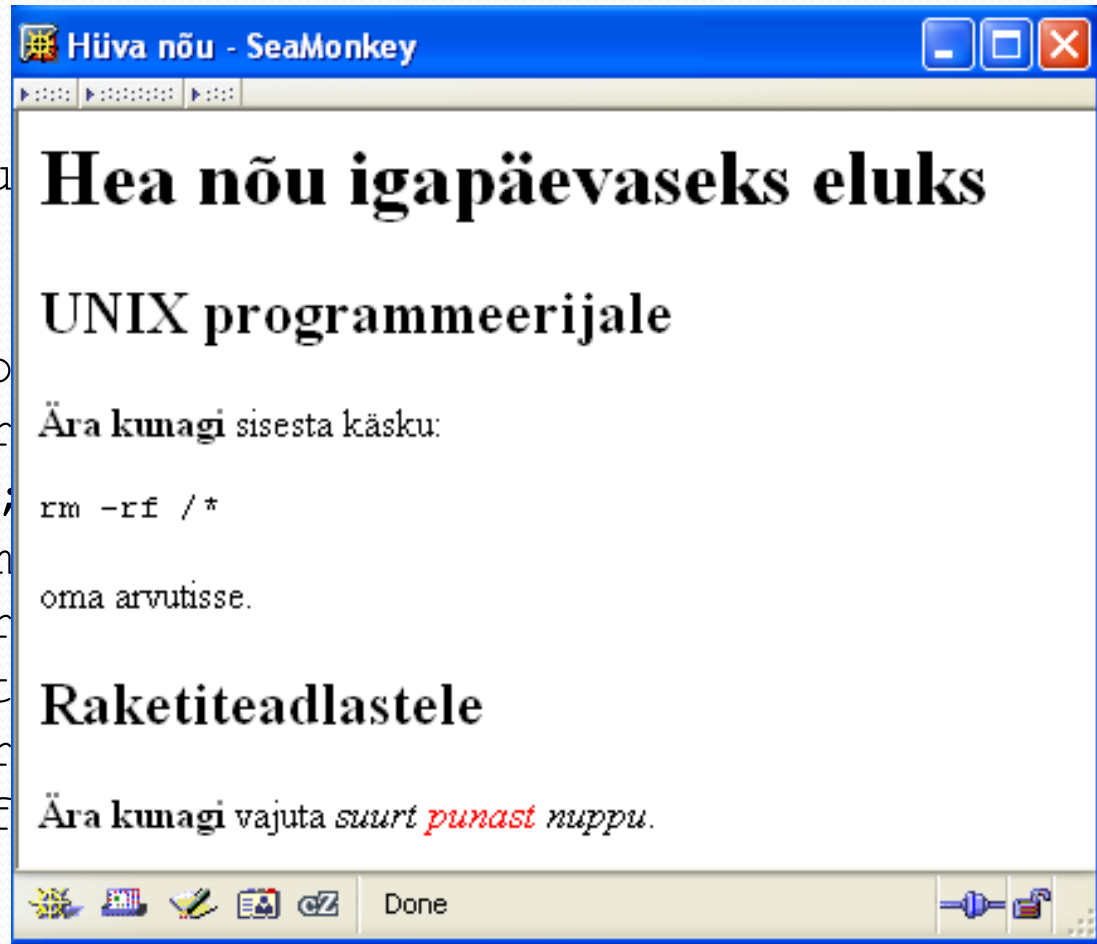
- *strong* – muutmine boldface-iks;
- *em* – rõhuasetus (enamus brausereid näitab kaldkirjas);
- *b* – Boldface;
- *i* – Italics;
- *tt* – Monospace;
- *big* – Suurendatud fondi suurus;
- *small* – Vähendatud fondi suurus;

Lihtne vormistamine:

```
<html>
  <head>
    <title>H&uuml;va n&otilde;u</title>
  </head>
  <body>
    <h1>Hea n&otilde;u igap&auml;evaseks eluks</h1>
    <h2>UNIX programmeerijale</h2>
    <b> &Auml;ra kunagi</b> sisesta k&auml;sku:
    <p><tt>rm -rf /*</tt></p>
    <p>oma arvutisse. </p>
    <h2>Raketiteadlasele</h2>
    <b> &Auml;ra kunagi</b> vajuta
    <i>suurt <font color="red">punast</font> nuppu</i>.
  </body>
</html>
```

Lihtne vormistamine:

```
<html>
  <head>
    <title>H&uu
  </head>
  <body>
    <h1>Hea n&o
    <h2>UNIX pr
      <b> &Auml;
    <p><tt>rm
    <p>oma arvutisse.
    <h2>Raketiteadlastele
    <b> &Auml;
    <i>suurt <f
  </body>
</html>
```

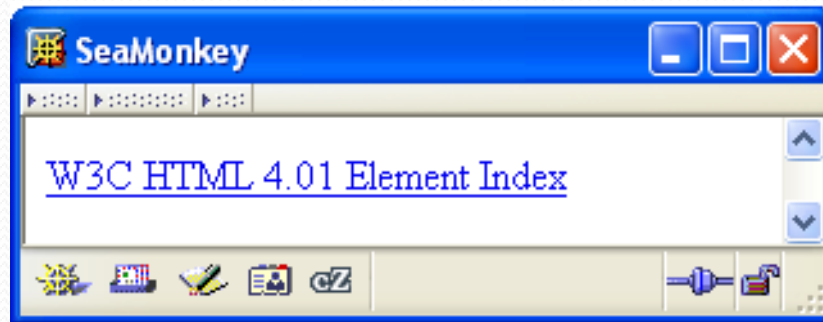


1>

u</i>.

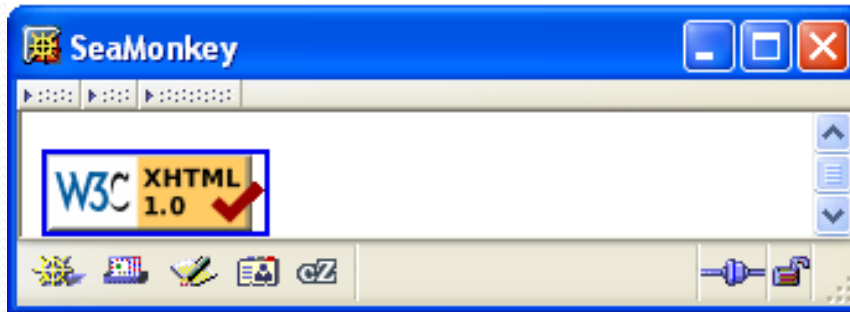
Lingid: *a* – ankur (anchor)

```
<a href="http://www.w3.org/TR/html4/index/elements.html">  
  W3C HTML 4.01 Element Index  
</a>
```



Teksti asemel kasutatakse sageli ka pilti:

```
<a href="http://validator.w3.org/check/referer">  
    
</a>
```



Lingil on kaks otsa (lõppu) ja suund. Link algab lähteankruga ja lõpeb sihtankruga, mis võib olla suvaline veebiressurss.

```
<a id = "section1" name = "section1" </a>
```

Sihtankru märkimiseks kasutatakse märki #:

```
<a href =
```

```
http://www.example.org/PageWithAnchor.html#section1/>
```

Kommentaarid

<!--

-->

Loendid

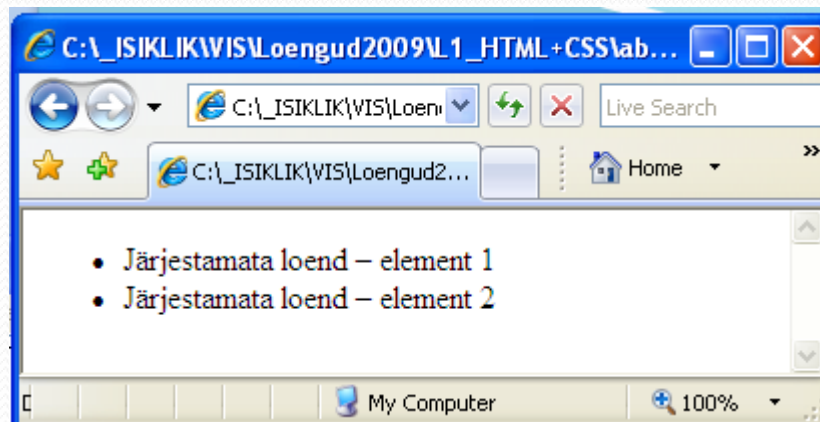
Järjestamata loendid: `<ul type = "disk" | "circle" | "square">`

```
<ul type = "square">
```

```
  <li> Järjestamata loend - element 1 </li>
```

```
  <li> Järjestamata loend - element 2 </li>
```

```
</ul>
```



Järjestatud loendid:

```
<ol type = "1" | "a" | "A" | "i" | "I">
```

```
<ol type = "A">
```

```
  <li> Järjestatud loend - element 1 </li>
```

```
  <li> Järjestatud loend - element 2 </li>
```

```
</ol>
```

1 – araabia numbrid;

a - väikesed tähed (a. b. ...);

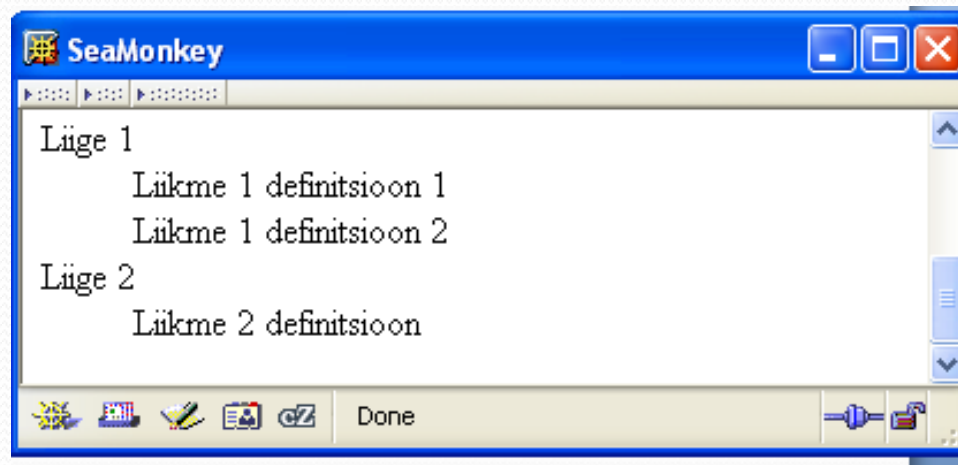
A – suured tähed (A. B. ..);

i – väikesed tähed Rooma numbritena (i ii ...);

I – suured tähed Rooma numbritena (I II ...).

Defineeritud loendid:

```
<dl>  
  <dt>Liige 1</dt>  
    <dd>Liikme 1 definitsioon 1</dd>  
    <dd> Liikme 1 definitsioon 2</dd>  
  <dt>Liige 2</dt>  
    <dd> Liikme 2 definitsioon</dd>  
</dl>
```



Tabelid

```
<table border="2">
  <tr>
    <th> Kuu </th>
    <th>Päevi</th>
  </tr>
  <tr>
    <td> Jaanuar</td>
    <td> 31</td>
  </tr>
  <tr>
    <td> Veebruar</td>
    <td> 28</td>
  </tr>
</table>
```



Ei ole vaja ette anda ridade või veergude arvu, need leitakse automaatselt.

Saab ette anda tühja lahtri: `<td> </td>`.

Parameetrid *cellspacing*, *cellpadding* määravad lahtrite vahelise ja lahtri sees oleva vaba ruumi;

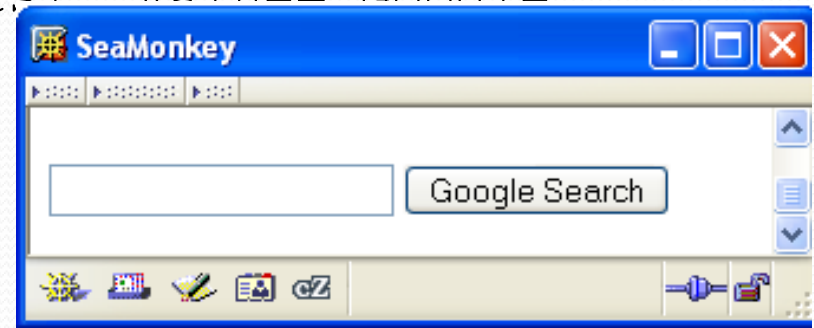
Parameetrid *rowspan* ja *colspan* määravad elemendi ulatuse ridades või veergudes.

Vormid

HTML vorm võimaldab kasutajal andmeid sisestada.

```
<form method="get"
      action="http://www.google.com/search">
  <input type="text" name="query">
  <input type="submit" name="otsi" value="Google
    Search">
</form>
```

`action` - URI, kuhu andmed saata
`method` - andmete saatmise meetod



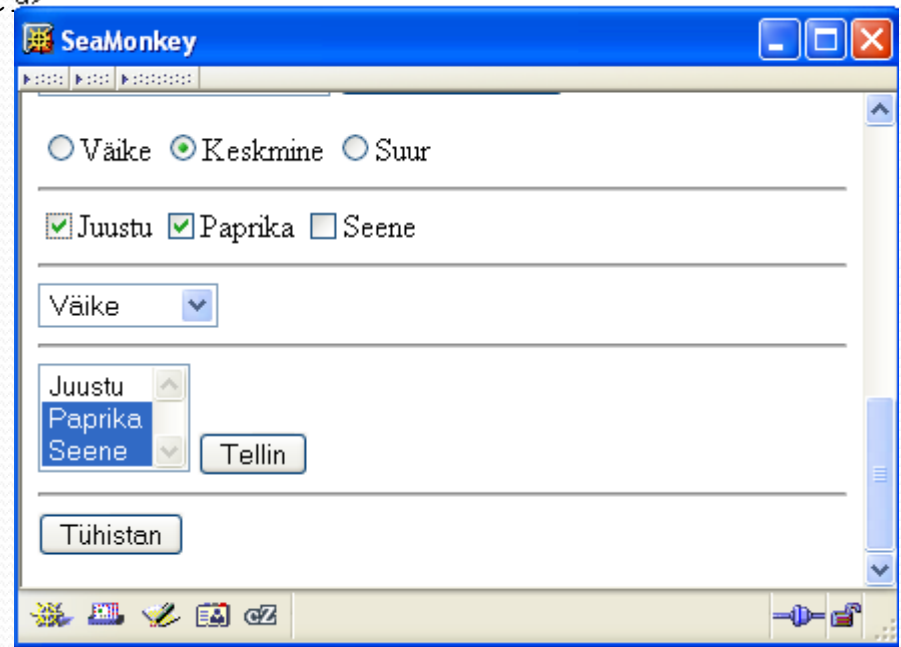

```
<input name="pub" type="radio" value="v">Väike
<input name="pub" type="radio" value="k">Keskmine
<input name="pub" type="radio" value="s">Suur <hr/>
<input name="pizza" type="checkbox" value="j">Juustu
<input name="pizza" type="checkbox" value="p">Paprika
<input name="pizza" type="checkbox" value="s">Seene <hr/>
<select name="pub">
  <option value="v">Väike
  <option value="k">Keskmine
  <option value="s">Suur
</select><hr/>
<select name="pizza" multiple>
  <option value="j">Juustu
  <option value="p">Paprika
  <option value="s">Seene
</select>

<input type="submit" value="Tellin"><hr/>
<input type="reset" value="Tühistan"
```

```

<input name="pub" type="radio" value="v">Väike
<input name="pub" type="radio" value="k">Keskmine
<input name="pub" type="radio" value="s">Suur <hr/>
<input name="pizza" type="checkbox" value="j">Juustu
<input name="pizza" type="checkbox" value="p">Paprika
<input name="pizza" type="checkbox" value="s">Seene <hr/>
<select name="pub">
  <option value="v">Väike
  <option value="k">Keskmine
  <option value="s">Suur
</select><hr/>
<select name="pizza" multiple>
  <option value="j">Juustu
  <option value="p">Paprika
  <option value="s">Seene
</select>

```



```

<input type="submit" value="Tellin"><hr/>
<input type="reset" value="Tühistan">

```

Element

Atribuudi tüüp

Objekt

input

text

Teksti sisestamine

input

password

Parooli sisestamine

input

checkbox

Valikukast

input

radio

Raadionupp

input

submit

Sisestusnupp

input

image

Pildiga sisestusnupp

input

reset

Tühistamisnupp

input

button

Nupp (skriptid)

input

hidden

Nähtamatu

input

file

Faili valimine

button

submit

Sisestusnupp

button

reset

Tühistusnupp

Stiil – *Cascading Style Sheets* (CSS)

- CSS Level 1 (1996)

<http://www.w3.org/TR/REC-CSS1>

- CSS Level 2 (1998)

<http://www.w3.org/TR/REC-CSS2>

toetab erinevaid meediaspetsiifilisi stiililehti

- CSS Level 3 (draft 2001)

<http://www.w3.org/TR/css3-roadmap/>

Mis on CSS (*Cascading Style Sheets*)?

- Stiilid defineerivad, **kuidas** näidata HTML elemente;
- “**Cascading**” viitab faktile, et iga stiilideklaratsiooni võib “ladustada” üksteise peale, moodustades ülem-alam (*parent-child*) relatsiooni stiilide vahel.

CSS kasutatakse, et öelda brauserile, kuidas tuleb sisu näidata:

- Uus keel, täiesti erinev HTMList.
- Defineeritakse **reeglid** ja see, kus neid reegleid rakendada dokumendis.

NÄIDE

http://www.w3schools.com/css/demo_default.htm

<http://www.csszengarden.com/>

- HTML leht jääb alati samaks;
- Valides erineva CSS muutub välimus

CSS mõned eelised

- HTML täiendamine;
- Veebilehe kujundus eraldatakse sisust;
- Lihtne muuta välimust; arendamise aeg lühem;
- Üks CSS – palju veebilehti;
- Kiirem allalaadimise aeg.

CSS reeglid ja stiililehed

- **Reegel:** Lause ühe või mitme elemendi stilistilise aspekti kohta.
- **Stiilileht:** Kogumik ühest või mitmest reeglist, mis rakenduvad HTML dokumendile.

CSS Süntaks

selector {property: value}

Kolm osa (valija, omadus, väärtus):

selector - HTML märgis

property – valija atribuut, mida tuleb muuta

value – atribuudi väärtus

Näiteks:

```
body {  
    background-color: yellow;  
}  
  
p {  
    font-size: x-large;  
    font-style: italic;  
}
```

Kui väärtus koosneb mitmest sõnast, siis ülakomade vahele:

```
p {font-family: 'sans serif'; }
```

- Ühe valijaga võib defineerida mitu omadust, nt

```
p {text-align: left; color: red; }
```

- Valijaid võib rühmitada, nt

```
h1, h2, h3, hr {color: black;}
```

CSS süntaks – valija Class

- Valijat **class** kasutatakse, et luua erinevaid stiile sama HTML märgise jaoks:

```
p.right {text-align: right;}  
p.center {text-align: center;}
```

- HTML dokumendis kasutatakse atribuuti `class`, nt

```
<p class="right">  
  See lõik rajastatakse paremalt.  
</p>
```

CSS süntaks – valija Class

- Valija **class** võib defineerida ilma märgise nimeta, st kõik, märgised, millel on sama klass, näevad välja ühtemoodi:

```
.right {text-align: right;}
```

```
<h1 class="right">
```

```
    See pealkiri rajastatakse paremalt.
```

```
</h1>
```

```
<p class="right">
```

```
    Nagu ka see lõik.
```

```
</p>
```

CSS süntaks – valija id

- Valijat **id** võib rakendada ühele, unikaalsele elemendile, nt

```
p#para1 { text-align: right;
          color: red; }
```

```
<p id="para1">
  Sisu on punane ja paremale rajastatud.
</p>
```

Stiililehe kasutamine

- Väline
- Sisemine
- Reasisene
- Mitme stiililehe kasutamine

Väline (external) CSS

- Tekstifail, mis ei tohi sisaldada HTML märgiseid – laiend **.css**;
- Sidumiseks kasutatakse märgist `link` ; meedia määramiseks kasutatakse atribuuti `type`

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

Sisemine (internal) CSS

- Stiilikäsud, mis on märgise **head** sees;
- Kasutatakse, kui leht vajab unikaalset stiili

```
<head>
  <style type="text/css">
    p {color: red}
    body {margin-left: 20px}
  </style>
</head>
```


Reasisene (inline) CSS

- Kasutab HTML atribuuti `style`

```
<p style="color: red;  
font-family: arial;  
margin-left: 0.5in;  
margin-right: 0.5in;  
font-style: italic">
```

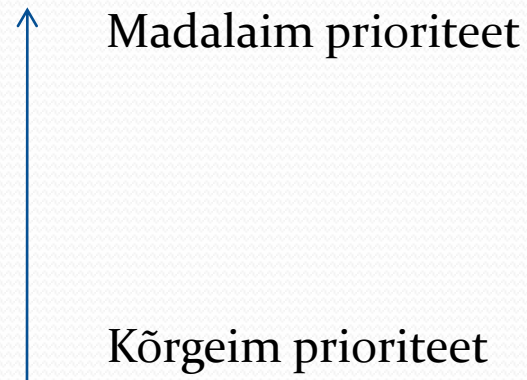
See on lõik.

```
</p>
```

Mitme stiililehe kasutamine

- Ühe dokumendi jaoks võib kasutada mitut stiililehte
 - Neist moodustatakse üks
 - Sisemised stiilid defineerivad üle välised, kui on topelt
- Järjekord:

1. Brauseri vaikestiil
2. Kasutaja stiilileht
3. Väline stiilileht
4. Sisemine stiil
5. Reasisene stiil



HTML dokumentide jaoks võib kasutada
valiidsuse kontrolli programme (HTML validator):

<http://validator.w3.org/>