

Klassid ja ID-d

HTML lubab määrata igale elemendile klassi (parameeter „class“) ja / või identifikaatori (parameeter „id“). Klassi ja identifikaatori peamine erinevus seisneb selles, et ID peab olema lehekülje piires unikaalne, ehk siis „id“ väärtust, mis määratud ühele elemendile, ei saa enam anda teisele. Ühte klassi võib samas aga kuuluda piiramatult arv elemente. Lisaks võib üks element kuuluda korraga mitmesse klassi.

ID-ks võib olla mõni numbriline/täheline kominatsioon (nt nimi, kood, vmt). Identifikaatorite selekteerimiseks kirjutatakse elemendi nime järel sümbol „ # “ (trellid), mille järel omakorda identifikaatori nimi.

element#identifikaator { ... }

Nii reglementeeritakse HTML-elemente:

```
div#1tekst{font-size:12px}
```

võib olla ka

```
#1tekst{font-size:12px}
```

HTML-is lisatakse ID järgnevalt:

```
<td id="1tekst">Tekst</td>
```

Klasside selekteerimiseks kirjutatakse elemendi nime järel punkt „ . “, peale mida kirjutatakse klassi nimi. Klasselement võib esineda ühel veebilehel korduvalt. Süntaks järgmine:

element.klassinimi { ... }

Klasselemendi näide:

```
div.salmi_joendus{text-align:right}
```

võib ka olla

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

HTML-is defineeritakse klasselement:

```
<p class="salmi_joendus">Salm 3</p>
```

Klass- ja ID-elementide nimetused ei ole määratletud ja võivad olla suvalise pikkusega. Kuid vältida tuleks klass- ja ID-elementide liigset pikkust.

Näide:

Pikk klasselemendi nimetus:

```
.kassipildi_ees_olev_lahter{background:red}
```

Optimaalsema pikkusega:

```
.kass{background:red}
```

Klasse ja ID-sid saab kenasti kasutada koos ning kombineerida neid teiste selektoritega.

Stiili omistamine mitme klasselemendi abil

Kui mitme elemendi reeglid peaksid sisaldama ühte sama definitsiooni ja teisi erinevaid definitsioone, siis ei saaks neid elemente grupeerida, vaid samast definitsioonist tuleks luua eraldi klasselemendi reegel. Seejärel aga viidata HTML-koodis mitmele klasselemendile.

Näide:

Erinevad elemendid ühe sarnase definitsiooniga:

```
.andmed{width:500px;height:250px; background:green}
```

```
.pealkirjalahter{font-style:italic; background:green}
```

```
.vasakmenyy{text-align:left; background:green}
```

Sarnasest definitsioonist on loodud eraldi reegel:

```
.andmed{width:500px;height:250px}
```

```
.menyy{text-align:left}
```

```
.jalus{font-style:italic}
```

```
.taust{background:green}
```

HTML-koodis töötab eelnev järgnevalt:

```
<div class="andmed taust">Andmed</div>
```

```
<div class="menyy taust">Menüü</div>
```

```
<div class="jalus taust">Uudised</div>
```

Klassid ja ID-d annavad võimaluse lisada HTML-ile semantilist (täenduslikku) infot. Päris kindlasti vajavad aga klassi ja / või ID määramist elemendid „div“ ja „span“, millel muidu puudub igasugune semantiline tähendus.

Pseudoklassid

Pseudoklassid võimaldavad lisada erinevaid efekte mingitele selektoritele või nende osadele. Käesolevas materjalis piirdume pseudoklasside kõige vanema rakendusega - erinevate stiilide määramisega viidetele vastavalt nende staatusele (küllastamata, küllastatud, aktiivne). Pseudoklassid on tõenäoliselt kõige populaarsem CSS-i kasutusvaldkond, mida tunnevad ka need, kes muidu oma lehekülgedel absoluutselt CSS - i abi ei kasuta.

Pseudoklassi üldkuju on järgmine:

```
selektor:pseudoklass { omadus: väärtus }
```

Pseudoklasse on viis:

- x **link** - külastamata link
- x **visited** - külastatud link
- x **hover** - element kursori ülelibistamise ajal
- x **active** - element hiirevajutuse ajal
- x **focus** - element on valitud (näiteks vormielement, mis ootab teksti sisestamist)

Näiteks võime omistada elemendi „a“ pseudoklassidele järgmised stiilid:

```
a:link { color: blue }
```

```
a:visited { color: gray }
```

```
a:hover { color: red }
```

```
a:active { color: purple }
```

Sellisel juhul on külastamata lingid sinised ja külastatud hallid. Kui hiirekursoriga üle lingi libistada, siis muutub see punaseks, kui hiire esimene klahv lingil olles alla vajutada, siis muutub värv violetseks.

- x Järgnevalt väärrib meie tähelepanu asjaolu, et pseudoklasside deklareerimise järjekord on oluline. Kui eelpool toodud järjekorda muuta, siis hakkab lehekülg töötama teistviisi:

```
a:link { color: blue }
```

```
a:active { color: purple }
```

```
a:hover { color: red }
```

```
a:visited { color: gray }
```

Sellise järjestuse puhul on tulemuseks see, et külastamata lingid on endiselt sinised ja lingist kursoriga üle käies muutub see punaseks, aga hiire klahvi vajutamise peale ei juhtu midagi. Hoopis imelik on aga lugu külastatud linkidega, sest need on ja jäävad halliks ükskõik kui palju sa ka nende peal kursorit ei liigutaks ega klikiks.

Mõnikord on selline linkide käitumine just see, mida sa soovid, aga enamasti siiski

mitte. Seega järjekord on tähtis, aga miks just selline, ja miks käituvad elemendid toodud järjestuste puhul just nõnda? Põhjus on selles, et element võib üheaegselt kuuluda mitmesse erinevasse pseudoklassi. Näiteks võib link olla üheaegselt küllastamata ning kursori all. Kuna pseudoklasside "kaalud" on võrdsed, siis jääb sellises olukorras peale see pseudoklass, mis on deklareeritud viimasena. Vaadeldes viimast näidet näeme nüüd, et link ei muutunud violetseks (purple) kordagi seetõttu, et alati kui lingi peale vajutati asus hiir lingi kohal ning „hover“-iga määratud stiil kirjutas „active“-stiili üle. Samuti, kui link oli juba küllastatud, siis varjutas see nii „hover“ kui „active“ staatuse, sest oli deklareeritud viimasena.

Seega tasub meeles pidada järjekord „link“, „visited“, „hover“ ja „active“. Klasside „link“ ja „visited“ omavaheline järjekord pole tegelikult oluline, sest link ei saa üheaegselt olla küllastatud ja küllastamata, aga on lihtsalt loogilisem kirjeldada küllastamata link enne. Pseudoklass „focus“ võiks asuda aga kõige lõpus - kui muidugi soovitakse, et näiteks tekstikast oleks teksti sisestamise ajal punase taustaga hoolimata sellest, kas kursor on selle kohal või mitte.

x Pseudoklasse võib ka omavahel kombineerida määramaks täpselt mis juhtub, kui element kuulub korraga mitmesse klassi.

```
a {  
  text-decoration: none;  
  font-size: large;  
  font-weight: bold  
}  
a:link { color: green }  
a:link:hover { color: olive }  
a:link:hover:active { color: #0f0 }  
a:visited { color: red }  
a:visited:hover { color: purple }  
a:visited:hover:active { color: #ff0 }
```

Siin määratakse esmalt, et kõik lingid, hoolimata olekust, on allajoonimata, suured ja rasvased. Järgmised deklaratsioonid määravad, et kui link on küllastamata, siis on ta sinine, kursori all oliivikarva ja klikkimise ajal neonroheline. Küllastatud lingid on punased, kursori all violetsed ning klikkimise ajal erk-kollased.

- x CSS lubab määrata pseudoklasse „hover“, „active“ ja „focus“ kõigile elementidele (mitte ainult linkidele). Kahjuks ei toeta mitmed laialt levinud brauserid (eeskätt Microsoft Internet Explorer) seda võimalust. „focus“-t muidugi lingile määrata ei saagi, aga see selleks.

Näiteks võib määrata, et tekstilõik muutub hiirega üle libistades kontrastsemaks, klikkides aga negatiivseks:

```
p {  
  color: #222;  
  background-color: #ddd;  
}  
p:hover {  
  color: #000;  
  background-color: #fff;  
}  
p:active {  
  color: #fff;  
  background-color: #000;  
}
```

NB! Efekti nägemiseks tuleb kasutada mõnda uuemat brauserit näiteks perekondadest Mozilla, Opera, Konqueror või Safari.

- x CSS annab võimaluse määrata eraldi stiil tekstilõigu esimesele reale ja / või tähele. Selleks saab kasutada pseudoelemente first-line ja first-letter:

```
p:first-line { font-variant: small-caps }  
p:first-letter { font-size: xx-large }
```

Näites määratakse, et esimene on kirjutatud trükitähtedega ning esimene täht on ülejäänutest märgatavalt suurem (oletades, et elemendi „p“ kirjususuurus on „medium“). Tihtipeale soovitakse aga, et esimene täht oleks nn. Drop Cap. Selleks puhuks sobib suurepäraselt parameeter float:

```
p:first-letter {  
  font-size: xx-large;  
  float: left; }
```

Ka pseudoelementide puhul tegelikult järjekord oluline pole, sest pseudoelementid on erineva kaaluga ning kui tekst kuulub korraga mitmesse pseudoelementi, siis saab määravaks spetsiifilisem element. Kahjuks ei kehti see aga Microsoft Internet Exploreri puhul, mis tõlgendab neid võrdse kaaluga olevateks ning arvestab seda, mis määratud viimasena. Näiteks järgmine stiilileht ei anna IE-ga vaadates soovivat tulemust:

```
div:first-letter {  
  color: blue;  
  font-size: xx-large;  
}  
div:first-line { color: red }
```

Siin on elemendi esimene rida punane ning esimene täht suur ja samuti punane. Õnneks on sellele probleemile väga lihtne lahendus - kirjeldada kõigepealt „first-line“ ja alles siis „first-letter“:

```
div:first-line { color: red }  
div:first-letter {  
  color: blue;  
  font-size: xx-large;  
}
```

Stiilinõuded lehekülje disainimisel

Veebikujundusest on tänaseks saanud omaette tõsine töövaldkond, kus töötavad kõrvuti arvutispetsialistid, kunstnikud ja disainerid. Algajale veebisepale võiks aga soovitada järgmist:

- Sisu, sisu ja veel kord sisu... Kui lehekülg on sisutühi, ei aita ka parim disain.
- Lihtne on ilus. Väga meeldiva lehekülje võib koostada ka minimaalsete vahenditega. On muidugi hea, kui autor tunneb kõiki HTML-i võimalusi - see annab tema käsutusse palju mitmekesisemad töövahendid. Samas aga on piisavalt lehekülgi, mille autor justkui karjuks: "Vaadake, mida mina oskan!".
- Hoidu kasutamast viitena sõnapaari "klõpsa siia" (või selle analooge) - viidetena tuleks kasutada lehekülje teemast tulenevaid tähtsamaid märksõnu.
- Ära pane lehekülge üles ainult selleks, et öelda "siin on veel töö pooleli".

- Lisa kõigi lehekülgede lõppu tagasiviide - brauseritel on küll "Back"-nupp, kuid kui lugeja on sattunud aadressi kaudu otse mõnele alamlehele (mitte pealeheküljele), siis ei saa ta pealeheküljele, ilma et sisestaks uue aadressi.
- Ära pane ühele lehele üle ühe animatsiooni (OK, pikemal lehel võib kaks olla...)
- Kui kahtled oma keelelistes võimetes, lase lehekülg mõnel asjatundjal üle vaadata. Tuleta meelde, miks Salomon Vesipruulist luuletajat ei saanud....
- Ära pane lehe algusesse suuri pilte ja fotosid, eriti reklaame.
- Aita lugejal oma veebis orienteeruda - pane algusse sisukord, kasuta mõõduka pikkusega ja ühtse kujundusega lehekülgi.
- Arvesta maksimaalselt erinevate lugejatega - välja arvatud juhul, kui leht on tõepoolest mõeldud kitsale lugejateringile. See tähendab üldmõistetavaid sümboleid, lihtsat ja selget esitust jne.
- Igasugused tagasisidevariandid on üldiselt igati teretulnud - E-aadressid, küsitlusvormid, külalisteraamatud. Samas aga peab nende olemasolu korral tagasiside tulemused ka reaalselt näha olema - kui külalisteraamatusse kirjutavad mitu lugejat järjest "Igav!" või saadetakse E-kirjaga lehel tekkinud vea kirjeldus, siis tuleks ka leida aega midagi ette võtta.
- Arvesta tekstibrauseri kasutajatega. Lisa kõigile graafikaobjektidele ALT-osa, suurtele graafilistele lehekülgedele tuleks teha eraldi alternatiivleheküljed, mis sisaldavad ainult teksti. Üks võimalik idee on kasutada ASCII-graafikat - seda tasub teha lihtsate, kuid suure tähtsusega jooniste korral (ainult alternatiivlehel).
- Kontrolli valmis lehtede vastavust standarditega. Validaatoreid on veebis mitmeid, siinkohal mainitagu vaid üht kesksamat - <http://validator.w3.org>.

Mõned näited

Kui stiilid on määratud ära samas failis, kus on sisu.

```
<html>
  <head>
    <title>Näide 1</title>
  </head>
  <style>
    body{background-color:#ffff33;
  }
  h3{text-align: center;
    color: red;
    font-family: courier;
    font-weight: bold;
  }
  p{text-align: center;
    font-style: italic;
  }
</style>
<body>
  <h3>Me oleme Päikese lapsed</h3>
  <p>Me oleme Päikese lapsed.<br />
  Me isaks sai tuline Tuul<br />
  ja ise me ammeks end pakkus<br />
  Suur, tunnetest pakatav Kuu.</p>
  <p>Me oleme Päikese lapsed -<br />
  see kaotatud paradiis!<br />
  Oh, ava vaid kivine kapsel<br />
  las Su hing Sind taas sinna viib!</p>
  <p>Las kannab Sind leekide juurde,<br />
  las sulab Su südamest jää.<br />
  Kui sisemist kutset veel kuuled,<br />
  siis Päikese lapseks sa jääd.</p>
</body>
</html>
```



Kui stiilid on määratud ära faililaiendiga *.css failis.

Alguses tehtud tavaline lehekülg, mis sisaldab viidet *.css failile....

```
<html>
  <head>
    <link rel="stylesheet" href="style.css" type="text/css" />
    <title>Näide 2</title>
  </head>
  <body>
    <h3>Me oleme Päikese lapsed</h3>
    <p>Me oleme Päikese lapsed.<br />
    Me isaks sai tuline Tuul<br />
    ja ise me ammeks end pakkus<br />
    Suur, tunnetest pakatav Kuu.</p>
    <p>Me oleme Päikese lapsed -<br />
    see kaotatud paradiis!<br />
    Oh, ava vaid kivine kapsel<br />
    las Su hing Sind taas sinna viib!</p>
    <p>Las kannab Sind leekide juurde,<br />
    las sulab Su südamest jää.<br />
    Kui sisemist kutset veel kuuled,<br />
    siis Päikese lapseks sa jääd.</p>
  </body>
</html>
```

*.css fail....

```
body{background-color:#ffff33;
}
h3{text-align:center;
color:red;
font-family:courier;
font-weight:bold;
}
p{text-align:center;
font-style:italic;
}
```

Tulemuseks sama lehekülg, mis esimeses näites.