

Front-end arendusreeglid

FE arendusreeglid

Version: 1.0

Lühendid

FE – *Front-end*
BE – *Back-end*
FF – Mozilla Firefox
IE – Internet Explorer
UI – *User interface* (kasutajaliides)
SEO – Search engine optimization

Eesmärk ja sihtgrupp

FE arendusreeglid on mõeldud kõigile FE ja BE arendajatele. Reeglite järgimine on kohustuslik.

Veebilehitsejate tugi

- A klass: IE10+, Chrome (viimane versioon), FF (viimane versioon), Microsoft Edge (viimane versioon)

Lõpptulem ei tohi kujundusest erineda, välja arvatud veebilehitsejate implementatsioonist tulenevad erandid (näide: selectboxi native kujundus on veebilehitsejates omavahel erinev ja see on lubatud)

- B klass: Safari

Visuaalselt korrektne, aga võib erineda algsest kujundusest. Optimeerimise ja kodeerimise lihtsustamise eesmärkidel ei pea näiteks toetama ümarad nurkasid, gradiente jne.

Stiililehed

Üks peamine stiilileht nimega main.css, kus sees loogilises järjestuses kommentaaridega eristatult kõik rakenduse stiilid.

Printimise jaoks on print media abil eraldi kuvatav stiilifail, mille eesmärk on kuvada sisuosa ilma illustreerivate elementideta. Ära peidetud on navigatsiooni elemendid, mis printitud versioonis ei ole olulised.

Kataloogide struktuur

gfx – CSSid koos CSSis defineeritud piltidega.

images – HTMLis lingitud pildid. Alankataloogid vajadusel komponendi põhiste elementidele, kui on ühes komponendis kasutusel rohkem kui 15 pilti.

fonts – rakenduses kasutatavad fondi failid.

scripts – rakenduses kasutatavad JS failid.

CSSi haldamine

CSSi stiilidel on üks haldaja ja vastutaja, kes on kursis kõigi muudatustega. Tema kohustus on üle vaadata teiste poolt kirjutatud CSS ja vastavalt vajadusele muuta või saata tagasi arendajale parandamiseks.

CSSi haldamise eesmärk on, et ainult CSSi haldaja ja/või FE meeskond kirjutab CSSi. Arendajad peavad pöörduma CSSi haldaja ja/või FE meeskonna poole, kui on muudatusi vaja. CSSi haldaja koordineerib teiste arendajate tööd.

BE arendajad tohivad stiile kirjutada ainult põhilise stiilifaili lõppu pärast vastavat kommentaari. Enda arendused peavad olema kommenteeritud nii, et CSSi haldaja saaks võimalikult kiirelt koodi üle vaadata ning sobivasse asukohta paigutada ja vajadusel koodi muuta.

Eelnev kirjeldab olemas oleva koodi muutmist või väikseid uusi arendusi olemas olevas raamistikus. Uute arenduste korral tuleb tellida FE antud ala spetsialist, kellele on edastatud antud dokument. Kui uue koodi kirjutada ei ole CSSi haldaja on haldajal kohustus ülevaadata uus kood.

Illustreerivad pildid

- Illustreerivad pildid peavad olema lisatud CSSi abiga.
- Pildid on optimeeritud. Salvestamisel valitud sobiv laiend vastavalt suurusele ja kvaliteedile. Võimalusel kasutatud Fireworks'i, PNG optimizerit või antud hetkel sobivat vahendit vähendamaks piltide mahtu.
- Photoshopis tuleks pildid salvestada *save for web* abil. JPG puhul võib salvestada 75-80% ilma, et oleks visuaalset kadu. Kui pildi suurus on u 150KB, siis kontrollida üle optimeerimise võimalused.
- CSSis ja HTMLis lingitud pildid asuvad eraldi kataloogis.
- CSSis lingitavad illustreerivad pildid asuvad CSSiga samas kataloogis. Eelistada illustreerivate piltide kuvamist läbi CSSi.
- HTMLis lingitud piltide nimetamisel arvestada SEOd ja ühtset stiili.
- CSSi pildid nimetada sarnase loogikaga. Ikonide alguses on icon_, taustapiltidel bg_, logodel logo_, nimekirjadel ul_ ja komponentide spetsiifiliste piltidel puhul selle klassi nimetus.
- Pildi nimed on läbivalt inglise keeles ja semantiliste nimetusega. Ei tohi kirjeldada pildil olevat värvi.

Semantika

Front-end kood peab vastama semantika reeglitele, välja arvatud juhul, kui kokkulepitud kasutatav raamistik ei võimalda seda saavutada. Kodeerimisel tuleb arvestada *responsive* (kohandub iseenesest ümber erinevatele ekraanisuurustele) versioonile üleminekuga.

Olulisemad punktid

- Classi nimed on üldised või komponendi spetsiifilised
- Classi nimedes ei tohi olla kasutatud visuaalseid kirjeldusi.
 - Näiteks värvid, suurused jne.
- Kasutada semantiliselt korrektseid HTMLi elemente.
 - Näiteks nimekirjade puhul peab olema kasutatud UL või OL tagi vastavalt sisule.
 - BR ei tohi kasutada.
 - Layouti loomisel ei tohi kasutada tabeleid.
 - Inline elemendi sisse ei tohi panna block elemente.
 - Paragraafide sisse ei tohi kirjutada DIVE.
- Vältida tühjade HTMLi elementide kasutamist.

Re-usable front-end

1. Enne kodeerimist tutvuda olemasolevate komponentidega.
2. Võimalusel kasutada olemasolevaid elemente.
3. Kujunduses olevate väikeste erinevuste puhul pakkuda välja ühtlustamise võimalusi ja lahendusi.
4. Kasutada üldiseid classi nimetusi ja arvestada kodeerimisel, et antud stiile oleks võimalik kasutada läbivalt. Erandite puhul kasutada eraldi classi nimetust.
5. Kodeerimisel arvestada, et uue elemendi muutmine ja haldamine oleks võimalikult lihtne. Näiteks arvestada keelsusega. Võimalusel vältida keelte jaoks erandite tegemist.

HTMLi reeglid

- HTML peab valideeruma kasutades <https://validator.w3.org> HTML valideerimisvahendit.
- Inline CSSi ei tohi kirjutada ja kasutada.
- *Cellpadding*'uid ja *cellspacing*'uid ei tohi kasutada HTMLis.
- Võimalusel kirjutada optimeeritud koodi ja vältida üleliigseid elemente ja defineeringuid.
- Kasutada läbivalt ühtset stiili.
 - Jutumärkideks on topelt jutumärgid
 - näide class="style01"
 - Arenduse failid on ühe tabi abil trepitud.
- Arvestada SEO soovitusi
 - Vajadusel illustreeritavatel piltidel korrektsed alt väärtused.
 - Meta tagid vastavalt kaasaegsetele nõudmistele.
- Piltide või ikoonide puhul, mis on kasutusel linkidena, kasutada title tagi. Title sisu peab kasutajale andma informatsiooni, et kuhu see link kasutaja suunab.
 - Näide logo title väärtus on "Avalehele"
- HTMLis vältida ja eemaldada üleliigsed kommentaarid.

CSSi reeglid

Kehtib FE arendaja poolt loodud stiiliiehtede kohta. Ei kehti juhul, kui kokkulepitud kasutatav raamistik ei võimalda seda saavutada.

- CSS peab valideeruma kasutades <https://jigsaw.w3.org/css-validator/> CSS valideerimisvahendit.
- CSSi treppimine
 - Selektori ja loogeliste sulgude vahele üks tühik.
 - Üks tab enne atribuuti (laiuselt võrdne nelja tühikuga).
 - Selektorid ja atribuudid on eraldi ridadel.
 - Näide

```
ul*{ ***margi  
n:*0; }
```

- Selektori blokkide vahel tühjad read eraldamaks erinevaid elemente aga üldiselt ilma tühja reata.

```

h1 {
    font-size:
34px; }
h2 {
    font-size:
24px; }
a {
    color:
#08c; }
a:focus,
a:hover {
    color:
#ff6; }

```

- Esimese taseme ja teise taseme kommentaari ees ja järgi üks tühi rida.
- CSSis blokkide järjekord:
 - normalize/reset, global
 - common
 - vastavalt struktuurile header, content ...
 - Plugina CSSid, kui on tegemist pluginaga, mida kasutatakse samade stiilidega erinevates asukohtades, siis antud CSS lisada eraldi faili.
 - media queries
- CSS Atribuutide järjekord:

```

display: ;
position: ;
top: ;
float: ;
clear: ;
width: ;
height: ;
overflow: ;
margin: ;
padding: ;
outline: ;
border: ;
background: ;
list-style: ;
table-layout: ;
border-collapse: ;
vertical-align: ;
text-align: ;
text-indent: ;
text-transform: ;
line-height: ;
word-spacing: ;
color: ;
font: ;
cursor: ;
visibility: ;
opacity: ;
z-index: ;
content: ;
quotes: ;
CSS3

```

- Ühikud
 - Eelistatult läbivalt kasutada pixleid.
 - 0 väärtuse korral ühik pole vajalik. 0em == 0px == 0
- Class' ja ID nimetused läbivalt väikeste tähtedega, semantilised ja eralduseks kasutatud sidekriipsu.
 - Näide: data-table-outer
- Classide nimetamisel kasutada läbivalt inglise keelt.
- ID kasutada ainult juhul, kui ollakse täiesti kindel, et seda elementi tulevikus ei kasutata lehel korduvalt. Näiteks lehe layout päis, jalus, menüü.
- CSSis on kaks taset kommentaare.
 1. esimene tase


```

/**
=Kommentaar
*/

```
 2. teine tase


```

/** =Kommentaar */

```

- Kommentaarid läbivalt inglise keeles. Vajadusel rakenduse spetiifilisi mõisteid kasutada eesti keeles.
- Välja kommenteeritud koodi ei tohi jätta alles.
- Margin ja paddingute korrektne kasutamine. Paddingut kasutada antud elemendi sisse nn õhu jätmiseks ja marginit väljapoole.
- Tekstide joondamisel vältida line-heighti abil joondamist ja eelistada paddingut.
- Floatide clearimiseks on class clear, mis läheb floatitud elementide ümber olevale tagile.

```
.clear:before,
.clear:after
{
    content: " ";
    display: table;
}
.clear:after
{ clear:
  both;
}
```

- CSS kirjutada ainult selleks mõeldud faili. Inline CSSi ja HTMLi failis eraldi <style> kasutada ei tohiks. Võimalusel vältida ka visuaalsete stiilide defineerimist JS failides.
- Pildi URLi lisamisel jutumärke ei ole vaja kasutada.
- Vältida veebilehitsejaspetiifilisi hääke.
- Kodeerimisel arvestada mõistlikkuse piirides "Web Content Accessibility Guidelines (WCAG) 2.0" soovitusi.
 - <http://www.w3.org/WAI/WCAG20/quickref/>
 - <http://kristjankure.com/>

Optimeerimine FE tehnikate abil

- https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Writing_efficient_CSS
 - tbody#select-01 => #select-01
 - #navigation li a => #nav a
 - tr.tr-name td td-name => .td-name
- HTMLis olevatele piltide määrata kõrgus ja laius, kui on võimalik. Eesmärk vähendada re-flow'ed.
- Gradientide, ümarate nurkade jne puhul kasutada CSS3 ning vanematel veebilehitsejatel peab kujundus välja nägema visuaalselt korrektne aga ei pea olema visuaalselt identne. Kontrollida veebilehitseja võimalusi siit <http://caniuse.com/>.
- Kui on võimalik illustreerivad pildid nt nooled, bulletid asendada CSS3 lahendusega, siis seda võimalust kasutada. Kui vajadus tühja elementi kasutada, siis selle asemel kasutada CSSis before'i võimalust. Selle juures tuleks arvestada veebilehitsejate tuge ja kasutajamugavust mõistlikkuse piirides. Informatsioon peab olema arusaadav ja kättesaadav ka B klassi veebilehitseja toele.
- Reflow ja repaint'iga arvestamine.
- CSSi reseti asemel kasutada normalize.
- Värvid kirjutada koodi abil (soovitavalt hex aga vajadusel rgba) ja läbivalt väikeste tähtedega. Võimalusel lühendada kolme tähelisteks.
 - #ffffff => #fff
 - #ff88aa => #f8a, #ff0000 => #f00
- Võimalusel CSSi väärtused tõsta kokku ühele reale.
 - Border-top, border-left, border-bottom, border-right => border
- CSSis ei tohi kasutada * lahendust määramaks kõigile elementide CSSi väärtuseid.
- Important kasutamist CSSi failis vältida.
- Normalize/reseti puhul ei tohiks olla defineeritud elemente, mida praktikas ei kasutata. Uue projekti alguses tuleks reset/normalize optimeerida ja vajadusel jooksvalt juurde lisada elemente.
- Võimalusel ja vajadusel kasutada sprite tehnikat.
- Pluginatega kaasa tulevat CSSi peab ülevaatama, optimeerima ja järjekorda uuendama.

Kujundus

- Linkidele, nuppudele alati korrektsed :hover, :focus ja :active staatused määrata.
 - hoveri ja fookuse puhul on eesmärk tuua visuaalselt element rohkem esile ja paremini loetavaks.
 - aktiivse staatuse puhul tuleks jätta mulje, et nupule või lingile on vajutatud. Visuaalselt poolelt näiteks gradient on vastupidine või lingi värv on tagasihoidlikum.
- Arvestada olemasoleva kujundusega ja jälgida sellele sarnast stiili.
- Ühtlustada ja jälgida lõpptulemuse joonduseid.