

Tallinna Ülikool  
Informaatika Instituut

## CSS3 3D transformatsioonid

Autor: Kristina Õim  
Juhendaja: Andrus Rinde

Tallinn 2015

## **Autorideklaratsioon**

Deklareerin, et käesolev bakalaureusetöö on minu töö tulemus ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

.....

(kuupäev)

.....

(autor)

# Sisukord

Sissejuhatus .....	4
1 CSS .....	5
1.1 CSS transformatsioonid .....	5
1.2 Transformatsiooni omadused .....	6
1.3 2D transformatsiooni meetodid .....	9
2 CSS3 3D transformatsioonid .....	12
2.1 3D transformatsiooni meetodid .....	13
2.1.1 Translate .....	14
2.1.2 Scale .....	18
2.1.3 Rotate .....	22
2.2 Testimise kokkuvõte .....	25
Kokkuvõte .....	27
Kasutatud kirjandus .....	28
Lisa 1 .....	30

## Sissejuhatus

Aastaid on veebilehtedele erinevate efektide ja animatsioonide lisamiseks kasutatud Adobe Flashi. Nüüd on hakatud Adobe Flash'ist loobuma. Praeguseks kasutab Flashi kogu maailma veebilehekülgedest 11,6%. See protsent väheneb pidevalt (W3Techs, 2014). Flash on ressursinõudlik ja kinnine tehnoloogia, mistõttu vajab Flashi player'i ehk plugina olemasolu. Praegu arendatav CSS3 oma transformatsioonide ning nendel põhinevate animatsioonidega, peaks vähemalt osaliselt kui mitte täielikult selle rolli üle võtma.

CSS3 võimaldab kasutada 2D ja 3D transformatsioone, mida saab animeerida vastavalt vajadusele. Kuna CSS3 on alles arendusjärgus siis transformatsioonid on vähe levinud. Lisaks vähemale tuntusele ei toimi need veel ka kõikide veebilehitsejatega. Kasutajate vajaduste rahuldamiseks on väljatöötamisfaasis uued transformatsioonid ning olemas olevaid on tublisti korrigeeritud ning paremaks muudetud. Transformatsioonide toimimiseks veebis on olemas eesliited, nii peaksid nad toimima ka tuntumate veebilehitsejate uuemate versioonide peal.

Käesoleva seminaritöö eesmärgiks on anda ülevaade CSS3 3D transformatsioonidest, tutvustada nende võimalusi, testida nende toimimist erinevate veebilehitsejatega.

Eesmärkide saavutamiseks annab autor kirjanduse põhjal ülevaate transformatsioonidest, nende omadustest ja toimimisest. Pärast transformatsioonidega tutvumist loob autor veebilehe, millel katsetab erinevaid 3D transformatsioone. Lehekülge testitakse erinevate veebilehitsejatega ning erinevatel seadmetel.

# 1 CSS

CSS (Cascading Style Sheets) defineerib kuidas HTML-i (HyperText Markup Language) ja XHTML-i (Extensible HyperText Markup Language) elemente välja kuvatakse (w3schools, 2015). CSS võimaldab lihtsalt hallata kogu lehekülje kujundust ning on seetõttu ka alustalaks veebikujundusele. CSS-i on aja jooksul arendatud edasi ning praeguseks on jõutud CSS3-ni. Nüüdseks on lisandunud ka võimalused objektide transformeerimiseks ja animeerimiseks.

Erinevalt CSS 2-st on CSS 3 jaotatud mooduliteks, kirjeldus on eristatud dokumentidena. Iga moodul lisab uusi võimalusi ja/või avardab CSS 2 funktsioone säilitades tagasiühilduvuse. Samuti arendatakse iga moodulit iseseisvalt. Kõige varasemad kavandid (drafts) avalikustati 1999. aastal. Modulariseerimise tõttu on erinevatel moodulitel erinev stabiilsus ja staatused. 2012. aastast on avaldatud üle 50 CSS mooduli CSS Working Group-i poolt, 4 neist ametlike soovitusena:

- Meedia päringud (*Media Queries*)
- Nimeruumid (*Namespaces*)
- Selektorid (*Selectors Level 3*)
- Värv (*Color*) (Wikipedia, 2014)

On olemas ka CSS transformatsiooni moodul, mis defineerib 2D ja 3D transformatsioone (pöörlemist jt), mida saab rakendada elementidele pärast tavakuju, ümbritsevaid elemente ei muudeta. Need kõik on hetkel W3C järgi töös olevad kavandid ning ootavad lõplikku soovitust. Transformatsioonid on olemas ka *transform* atribuudi väärtustena (w3, 2014). Nt. *transform: rotateY(120deg);* .

## 1.1 CSS transformatsioonid

Transformatsioone rakendatakse kasutades CSS - i omadusi, mis lasevad panna HTML elementidele transformatsioone. Sinna alla lähevad näiteks elemendi pööramine (*rotate*) ja nihutamine (*translate*). See toimib nii 2D kui ka 3D mõõtmetes. Kaheks peamiseks on *transform* ja *transform-origin*. Kui 2D puhul määratakse omadus lihtsalt koodi kirjutades siis 3D-ga on vaja kõigepealt määrata 3D ruum, mida saavutatakse *perspective* omadust kasutades. See omadus tagab 3D ruumilisuse ning ilme. Mida suuremana element paistab seda lähemal on ta vaatajale. Teisena on vaja määrata

*perspective – origin*, elemendi vaataja vaatepositsioon. Vaikimisi on see alati keskel (MDN, 2014).

## 1.2 Transformatsiooni omadused

Järgnevas loetelus on transformatsiooni omadused, millede suhtes transformatsioon toimub,

- *transform* - rakendab elemendile 2D ja 3D transformatsiooni, selle omaduse atribuutidena kasutataksegi transformatsiooni meetodeid (w3school, 2014).

```
{
  -webkit-transform: rotate(40deg); /* Chrome, Safari, Opera */
  transform: rotate(40deg);
}
```

- *transform – origin* - võimaldab muuta transformeeritava elemendi positsiooni ehk on võimalik muuta x-telje, y-telje ja z-telje koordinaate (w3schools, 2014).

```
{
  -webkit-transform: rotate(45deg); /* Chrome, Safari, Opera */
  -webkit-transform-origin: 80% 50%; /* Chrome, Safari, Opera */
  transform: rotate(45deg);
  transform-origin: 80% 50%;
}
```

- *transform – style* – täpsustab, kuidas pesastatud (*nested*) elemendid on renderdatud 3D ruumis ehk siis kuidas üksteise sees olevad elemendid on visualiseeritud 3D ruumis (w3schools, 2014).

```
#div1 {
  -webkit-transform: rotateY(-60deg); /* Chrome, Safari, Opera */
  transform: rotateY(-60deg);
}
#div2 {
  -webkit-transform: rotateY(60deg); /* Chrome, Safari, Opera */
  -webkit-transform-style: preserve-3d; /* Chrome, Safari, Opera */
  transform: rotateY(60deg);
  transform-style: preserve-3d;
}
```

Pesastatud elemendid:

```

<div id="div1">
    <div id="div2"> </div>
</div>

```

Kui mõlemaid kujundeid muuta siis arvestatakse 3D ruumilisuse hoidmisega (*preserve-3d*).

- *perspective* - täpsustab perspektiivi, kust punktist 3D elemente nähakse. Defineerib mitu pikslit on 3D element paigutatud objekti vaatajast. Toimib ainult transformeeritud 3D elementidel. Töötab child elemendil (w3schools, 2014). Järgnevas näites on child elemendiks div, mille klassiks on „*cube*”. Kõigele, mis jääb *cube* divi alla, rakendub *perspective* omadus ehk kuup, mida veebileheküljel näeme on 3D ruumis.

```

<div class="wrap">
    <div class="cube">
        <div class="front">EESMINE</div>
        <div class="back">TAGUMINE</div>
        <div class="top">PEALMINE</div>
        <div class="bottom">ALUMINE</div>
        <div class="left">VASAK</div>
        <div class="right">PAREM</div>
    </div>
</div>

{
    -webkit-perspective: 500px; /* Chrome, Safari, Opera */
    perspective: 500px;
}

```

Kui *perspective* omadust pole pandud siis kujund näib transformatsiooni tagajärjel rohkem laiiali laotuna ning ei teki vajalikku 3D efekti ning tunnetust (Joonis 1). Kõikidele joonistele on pandud raam ümber, et ära näidata nähtav ala, mille sees kujund liigub.

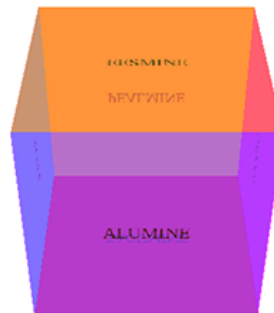
## Näide



Joonis 1 - perspective omadus puudub

Kui transformatsioonis on defineeritud lisaks ka *perspective* omadus siis me näeme kujundit ilusti ning ruumiliselt (Joonis 2). Kogu liikumine toimub samuti ruumiliselt.

## Näide



Joonis 2 - perspective omadus olemas

- *perspective - origin* – täpsustab, kus on kujund paigutatud x-teljel ja y-teljel. Töötab child elemendil samamoodi nagu *perspective*’gi (w3schools, 2014).

```
{  
-webkit-perspective: 150px; /* Chrome, Safari, Opera */  
-webkit-perspective-origin: 30% 30%; /* Chrome, Safari, Opera */  
perspective: 150px;  
perspective-origin: 30% 30%;  
}
```

- *backface - visibility* – defineerib, kas elemendi tagumine külg on nähtav või mitte kui element asetseb ekraani poole. Elemendi tagumine pool on alati läbipaistev taust, peegeldades esimest poolt, kui tal lastakse olla nähtav (MDN,



2014). Kui näiteks element pöörleb siis me saame kasutada seda omadust, et näeme ainult tema esikülge. Kui tagumine külg on peidetud siis kujund justkui kaob korra ekraanilt ning esiküljele pööreldes ilmub taas tagasi.

```
{  
-webkit-backface-visibility: hidden; /* Chrome, Safari, Opera */  
backface-visibility: hidden;  
}
```

### 1.3 2D transformatsiooni meetodid

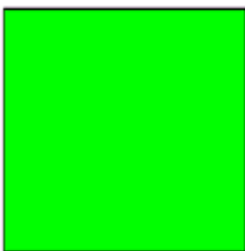
CSS3 võimaldab nii 2D kui ka 3D transformatsioone. Järgnevalt on ära toodud 2D transformatsioonide loend. Omadustel kasutatakse ainult kahte telge – x ja y.

- *translate(x,y)* – element nihutatakse mööda x-telge ja y-telge.



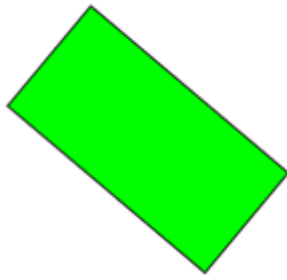
Joonis 3- *translate(x,y)*

- *translateX(n)* – element nihutatakse mööda x-telge, n tähistab piksleid
- *translateY(n)* – element nihutatakse mööda y-telge, n tähistab piksleid
- *scale(x,y)* – muudab elementide laiust ja pikkust



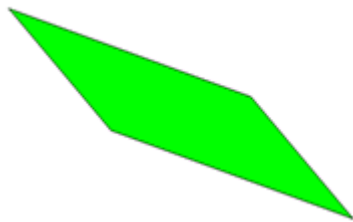
Joonis 4 - *scale(x,y)*

- *scaleX(n)* - muudab elemendi laiust, n on suhtarv
- *scaleY(n)* – muudab elemendi pikkust, n on suhtarv
- *rotate(angle)* – defineerib 2D elemendi pööramist, nurk täpsustatakse parameetris



**Joonis 5 - rotate(angle)**

- $skew(x-angle, y-angle)$  – pöörab elemendi viltu mööda x- ja y-telge



**Joonis 6 - skew(x-angle, y-angle)**

- $skewX(angle)$  – pöörab elemendi viltu mööda x-telge
- $skewY(angle)$  – pöörab elemendi viltu mööda y-telge (w3schools, 2014)
- $matrix(n,n,n,n,n,n)$  – defineerib 2D transformiooni kasutades selleks maatriksit 6 parameetriga, mis sisaldab endas matemaatilisi funktsioone. Seetõttu hõlmab endas kõiki 2D transformatsioone.

Nende laiendamisel kolmanda mõõtmega saadaksegi 3D transformatsioon.



## 2 CSS3 3D transformatsioonid

CSS3 lubab muuta elemente veebilehel kasutades selleks 3D transformatsioone (w3schools, 2014). Kuna transformatsioonide moodul on alles arenduses siis kõik veebilehitsejad ei toeta veel täielikult 3D transformatsioone. Transformatsioonide toimimiseks on vaja eesliiteid. Ilma eesliideteta toetavad transformatsioone Mozilla Firefox, Opera, Chrome Androidile, Androidi põhine veebilehitseja ning uuemad versioonid Chrome-i. Internet Explorer-i puhul osaline toetus. Safari ja IOS Safari vajavad eesliiteid. 2D transformatsioone toetavad kõik tuntumad veebilehitsejad.

Erinevate veebilehitsejate eesliited:

- -webkit- - Google Chrome, Safari
- -moz- - Mozilla Firefox
- -ms- - Internet Explorer
- -o- - Opera

Transformatsioone võib kasutada (aga ei pea) ka CSS transition omadustega koos:

- *transition – property* – kasutatakse defineerimaks CSS elemente, millele transformatsiooni rakendatakse. CSS süntaks - *transition-property: none|all|property|initial|inherit;*
- *transition – duration* – täpsustab kui kaua läheb aega, et animatsioon algatada (sekundid, millisekundid). CSS süntaks - *transition-duration: time|initial|inherit;*
- *transition – timing – function* – määrab transformatsiooni kiiruse. CSS süntaks - *transition-timing-function: ease|linear|ease-in|ease-out|ease-in-out|cubic-bezier()|initial|inherit;*
- *transition – delay* – väärtus, kaua oodatakse enne kui transformatsioon toimuma hakkab. CSS süntaks - *transition-delay: time|initial|inherit;*
- *transition* – defineerib oleku kahe elemendi vahel, nõ lühim variant, saab kombineerida nelja eelnevat. Sellisel juhul tuleb nad kirjutada samas järjekorras nagu loendis ülevalpool. CSS süntaks - *transition: property duration timing-function delay|initial|inherit;*  
(w3schools, 2014).



Joonis 7 - enne transition omadust

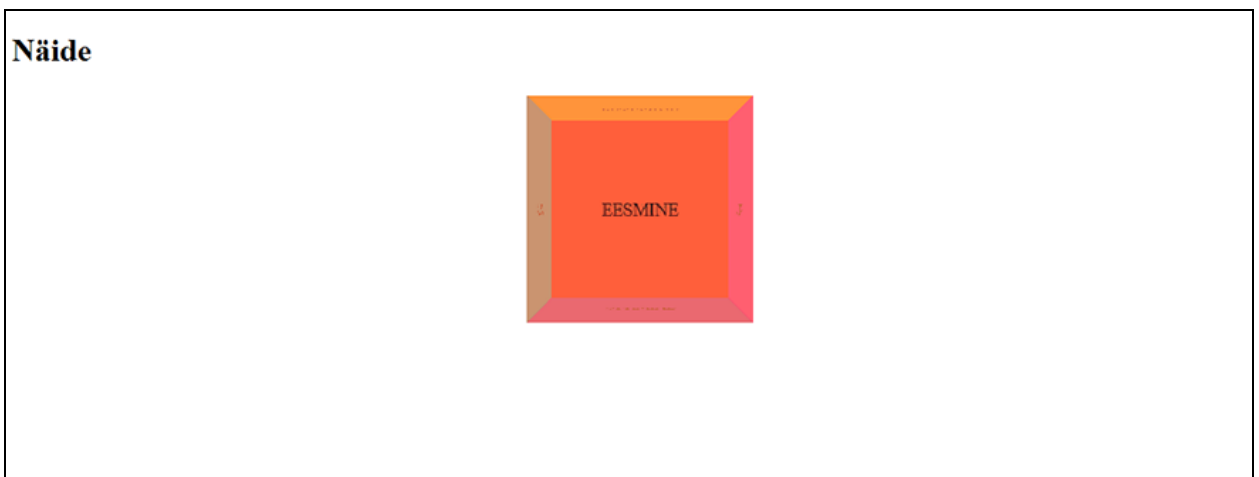


Joonis 8 - pärast transition omadust

## 2.1 3D transformatsiooni meetodid

Järgnevalt on toodud ära 3D transformatsiooni meetodid, mis sarnaneb 2D-le aga sisaldab endas ka kolmandat mõõdet ehk z-telge. Kõikidele on koodi sisse kirjutatud interaktiivsus, kujund liigub oma algse asukoha ning transformatsiooniga tekitatud uue asukoha vahel 1 sekundilise intervalliga. Kasutaja saab ise määrata kui palju mingi telje suunas liikumine toimub vastavalt sellele, millist tulemust soovitakse.

Kujundi esialgne asukoht ning välimus veebileheküljel. Kuubi kokkupanekuks ning ruumiliseks muutmisel kasutati CSS3 3D transformatsioone – rotate ja translate. Nendega antakse tahkudele kohad, et nad üksteisega kokku sobituksid. *Perspective* omadus muudab tahud ruumiliselt vaadeldavaks.



Joonis 9 - esialgne kuju ja asukoht

- *matrix 3d* - defineerib 3D transformatsiooni kasutades 16 väärtusest koosnevat 4x4 maatriksi. Sisaldab endas kõiki teisi transformatsioone nagu ka 2D puhul.

## 2.1.1 Translate

- *translate 3d(x,y,z)* - defineerib 3D elemendi nihutamise kasutades kõiki kolme telge

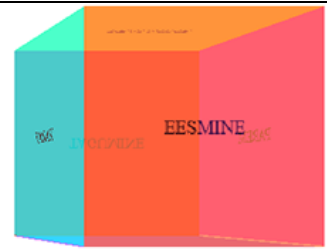


**Joonis 10 - translate3d näide 1**

Kujundi liikumine toimub mööda x-telge, y-telge kui ka z-telge (Joonis 10). Ka koodis kirjutatakse pikslite arvud selles järjekorras. Negatiivne pikslite arv viib kujundi mööda x-telge vasakule. Kujund tundub suurem, sest liigub vaateväljas lähemale.

```
#translate3d .cube{  
    -webkit-transform: translate3d( -350px,60px,40px ); Chrome, Safari, Opera */  
    transform: translate3d( -350px,60px,40px );  
    -moz-transform: translate3d( -350px,60px,40px ); mozilla */  
    transform: translate3d( -350px,60px,40px );  
    -webkit-transition: all 1s ease;  
    transition: all 1s ease;  
}
```

## Näide



### Joonis 11 - translate3d näide 2

Teises näites on negatiivne väärtus tõstetud y – koordinaadi ette ning x – koordinaat on vaikimisi positiivse väärtusega. See nihutab kujundi paremale ning y – koordinaadi tõttu vaateväljas kõrgemale (Joonis 11). Kui negatiivne väärtus oleks kasutuses ka z – koordinaadi ees liiguks kujund vaateväljas tahapoolle ehk paistaks väiksem kuna asuks vaatelejast kaugemal.

```
#translate3d .cube{
```

```
  -webkit-transform: translate3d( 350px,-60px,-40px ); /*Chrome, Safari, Opera */
```

```
  transform: translate3d( 350px,-60px,-40px );
```

```
  -moz-transform: translate3d( 350px,-60px,-40px ); /* mozilla */
```

```
  transform: translate3d( 350px,-60px,-40px );
```

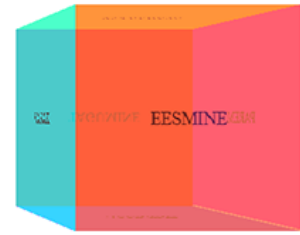
```
  -webkit-transition: all 1s ease;
```

```
  transition: all 1s ease;
```

```
}
```

- *translateX(x)* - defineerib 3D elemendi nihutamise ainult X-telje suunal

## Näide



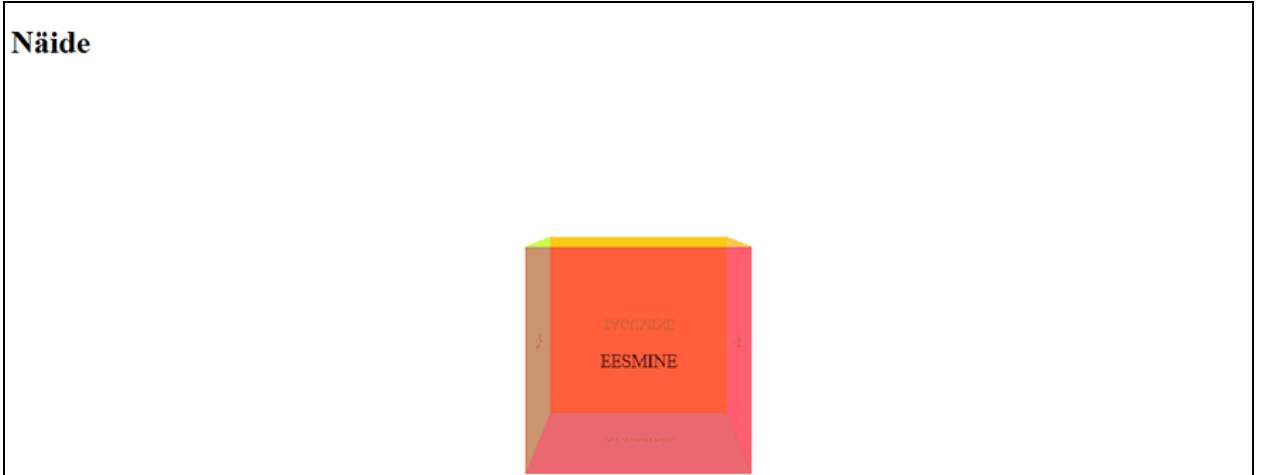
### Joonis 12- translateX

335px viib kujundi ekraani paremale poole, liikumine toimub mööda x-telje (Joonis 12). Veebilehel toimub pidev liikumine algse asukoha ning transformatsiooni tagajärjel saadud uue asukoha vahel, intervalliks on üks sekund.

```
#translateX .cube{  
    -webkit-transform: translateX(335px); /*Chrome, Safari, Opera */  
    transform: translateX(335px);  
    -moz-transform: translateX(335px); /*mozilla */  
    transform: translateX(335px);  
    -webkit-transition: all 1s ease;  
    transition: all 1s ease;  
}
```



- *translateY(y)* - defineerib 3D elemendi nihutamise ainult Y-telje suunal

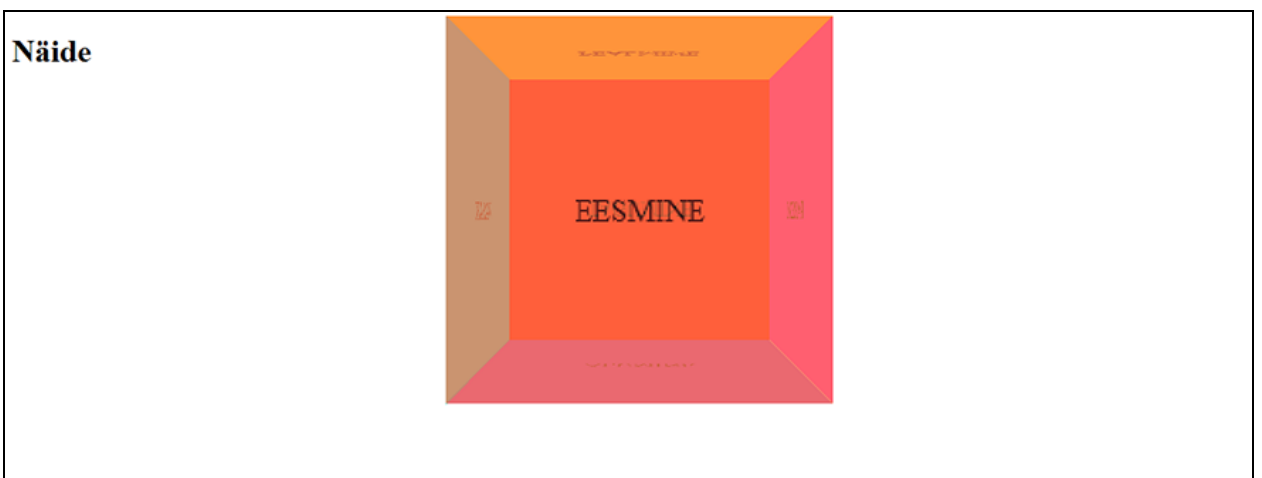


**Joonis 13 - translateY**

Kujundi liikumine toimub mööda y-telge, kujund on „hüpanud” algse asukohaga võrreldes allapoole (Joonis 13).

```
#translatey .cube{
    -webkit-transform: translateY(140px); /* Chrome, Safari, Opera */
    transform: translateY(140px);
    -moz-transform: translateY(140px); /*mozilla */
    transform: translateY(140px);
    -webkit-transition: all 1s ease;
    transition: all 1s ease;
}
```

- *translateZ(z)* - defineerib 3D elemendi nihutamise ainult Y-telje suunal



**Joonis 14 - translateZ**

Transformatsiooni mõjul liigub kujund mööda z-telge. Kuna koodis on pikslite arv kirjutatud positiivse väärtusena, liigub kujund vaateväljale lähemale ning kujund tundub seetõttu suurem (Joonis 14 **Error! Reference source not found.**).

```
#translatez .cube{
  -webkit-transform: translateZ(290px); /* Chrome, Safari, Opera */
  transform: translateZ(290px);
  -moz-transform: translateZ(290px); /*mozilla */
  transform: translateZ(290px);
  -webkit-transition: all 1s ease;
  transition: all 1s ease;
}
```

### 2.1.2 Scale

- $scale3d(x,y,z)$  - defineerib kujundi suuruse X-, Y-, ja Z-telje suhtes

Näide



Joonis 15 - scale3d näide 1

Kujund venitatakse vaataja suunas tahupoole pikemaks. Kasutatakse kõiki kolme telge (Joonis 15 **Error! Reference source not found.**).

```

#scale3d .cube{
    /*-webkit-transform: scale3d(0.5,0.6,2.7); Chrome, Safari, Opera */
    transform: scale3d(0.5,0.6,2.7);
    /*-moz-transform: scale3d(0.5,0.6,2.7); mozilla */
    transform: scale3d(0.5,0.6,2.7);
    -webkit-transition: all 1s ease;
    transition: all 1s ease;
}

```



**Joonis 16 - scale3d näide 2**

Scale3d teine näide näitab elemendi suuruse muutust x – telje suunal. Kujund on venitatud horisontaalselt pikemaks (Joonis 16). Kuna x on negatiivne siis kujund pöörab ennast transformatsiooni käigus ümber. Näeme kujundit tagantpoolt.

```

#scale3d .cube{
    /*-webkit-transform: scale3d(-1.5,0.6,2.7); Chrome, Safari, Opera */
    transform: scale3d(-1.5,0.6,2.7);
    /*-moz-transform: scale3d(-1.5,0.6,2.7); mozilla */
    transform: scale3d(-1.5,0.6,2.7);
    -webkit-transition: all 1s ease;
    transition: all 1s ease;
}

```

- *scaleX(x)* - defineerib elemendi suuruse X-telje suhtes

### Näide



Joonis 17 - *scaleX*

Kujund on kokkusurutud ning omandanud püstristtahuka välimuse (Joonis 17 **Error!** **Reference source not found.**). Transformatsioon toimub x-telje suunas.

```
#scalex .cube{
    /*-webkit-transform: scaleX(0.5); Chrome, Safari, Opera */
    transform: scaleX(0.5);
    /*-moz-transform: scaleX(0.5); mozilla */
    transform: scaleX(0.5);
    -webkit-transition: all 1s ease;
    transition: all 1s ease;
}
```

- *scaleY(y)* - defineerib elemendi suuruse Y-telje suhtes

### Näide

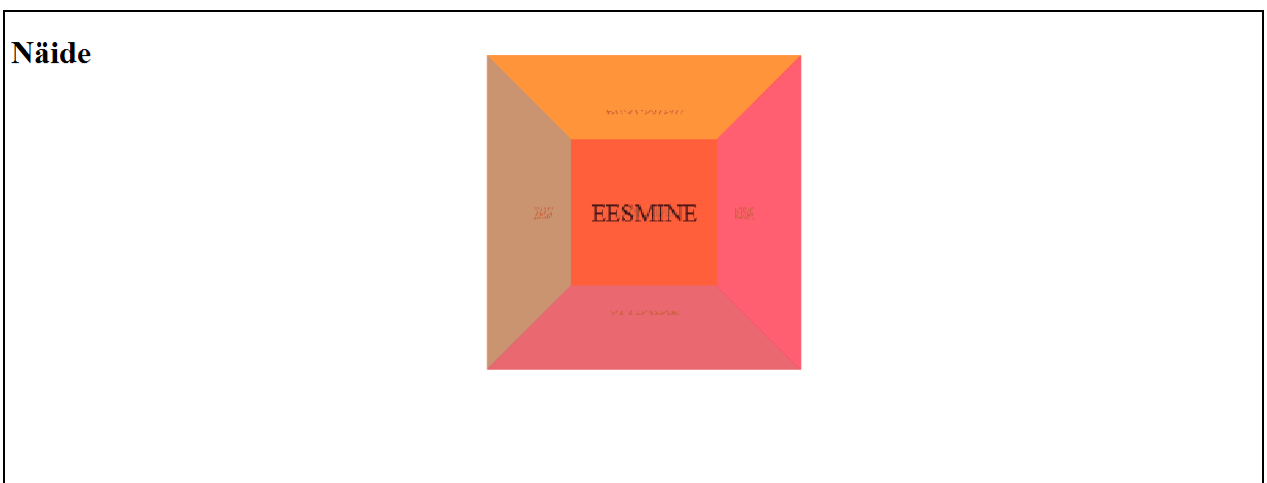


Joonis 18 - *scaleY*

Kujund venitatakse allapoole pikemaks (Joonis 18). Venitus toimub mööda y-telge. Kui y väärtus oleks negatiivne, veniks kujund pikemaks ülespoole.

```
#scaley .cube{  
    /*-webkit-transform: scaleY(1.7); Chrome, Safari, Opera */  
    transform: scaleY(1.7);  
    /*-moz-transform: scaleY(1.7); mozilla */  
    transform: scaleY(1.7);  
    -webkit-transition: all 1s ease;  
    transition: all 1s ease;  
}
```

- *scaleZ(z)* - defineerib elemendi suuruse Z-telje suhtes



**Joonis 19 – scaleZ**

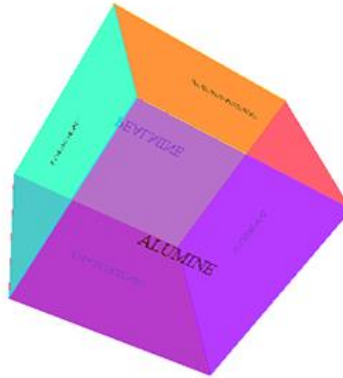
Kujund venitatakse pikemaks vaateväljas lähemale liikudes, kujund suundub nõ vaatajale näkku (Joonis 19).

```
#scalez .cube{  
    -webkit-transform: scaleZ(2.9); /* Chrome, Safari, Opera */  
    transform: scaleZ(2.9);  
    -moz-transform: scaleZ(2.9); /*mozilla */  
    transform: scaleZ(2.9);  
    -webkit-transition: all 1s ease;  
    transition: all 1s ease;  
}
```

### 2.1.3 Rotate

- `rotate3d(x,y,z,angle)` - defineerib 3D objekti pööramist ruumis

Näide



Joonis 20 – rotate3d näide 1

Kujund on pööratud transformiooni tagajärjel põhi ülespoole (**Error! Reference source not found.**). Liikumisel kasutatakse kõiki kolme telge (x,y,z) ja lisaks nurka kraadides.

```
#rotate3d .cube{  
    /*-webkit-transform: rotate3d(120,70,20,65deg); Chrome, Safari, Opera*/  
    transform: rotate3d(120,70,20,65deg);  
    /*-moz-transform: rotate3d(120,70,20,65deg); mozilla */  
    transform: rotate3d(120,70,20,65deg);  
    -webkit-transition: all 1s ease;  
    transition: all 1s ease;  
}
```

Näide

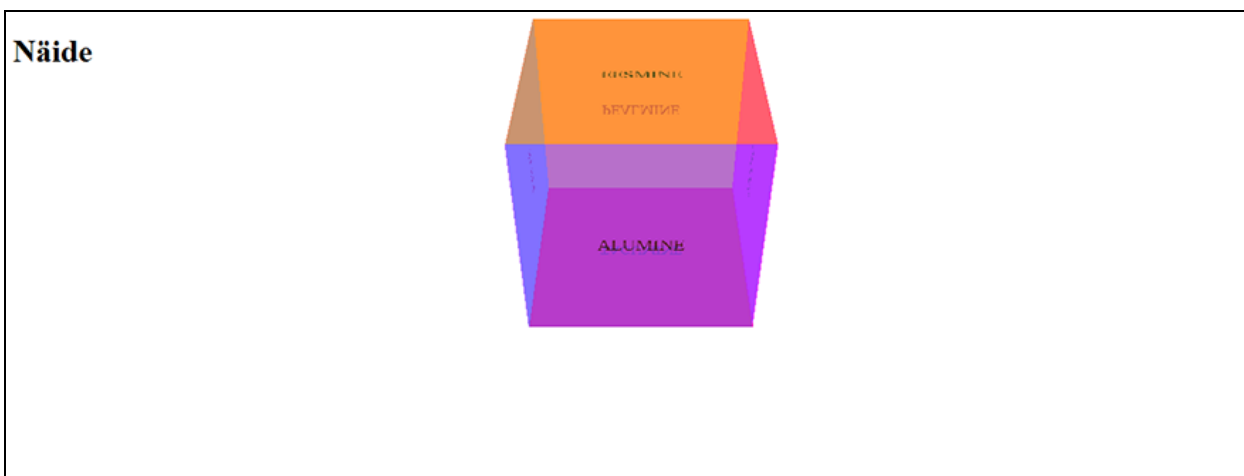


Joonis 21 - rotate3d näide 2

Teisel näitel on kujund pööratud küllili (**Error! Reference source not found.**). Koodis on muudetud transformatsiooni kraade ning x – koordinaat on väljendatud negatiivse väärtusega. Z - telge muutes muutub kujund kas väiksemaks või suuremaks.

```
#rotate3d .cube{
    /*-webkit-transform: rotate3d(-120,70,20,100deg); Chrome, Safari, Opera */
    transform: rotate3d(-120,70,20,100deg);
    /*-moz-transform: rotate3d(-120,70,20,100deg); mozilla */
    transform: rotate3d(-120,70,20,100deg);
    -webkit-transition: all 1s ease;
    transition: all 1s ease;
}
```

- *rotateX(angle)* - defineerib 3D pööramist ümber X-telje



**Joonis 22 - rotateX**

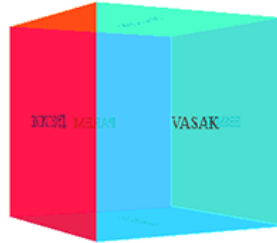
Kujund on pööratud 50 kraadi x-telje suunas nõ külje joone peale (**Error! Reference source not found.**). Alumine külg paistab, mis muidu on suunatud alla ning seetõttu asub vaateväljast väljas.

```
#rotatex .cube {
    /*-webkit-transform: rotateX(50deg); Chrome, Safari, Opera */
    transform: rotateX(50deg);
    /*-moz-transform: rotateX(50deg); mozilla */
    transform: rotateX(50deg);
    -webkit-transition: all 1s ease;
    transition: all 1s ease;
}
```

}

- `rotateY(angle)` - defineerib 3D pööramist ümber Y-telje

#### Näide



Joonis 23 - rotateY

Kujund on pööranud ennast põhja peal vasakule 120 kraadi (Joonis 23).

```
#rotatey .cube{
    /*-webkit-transform: rotateY(120deg); Chrome, Safari, Opera */
    transform: rotateY(120deg);
    /*-moz-transform: rotateY(120deg); mozilla */
    transform: rotateY(120deg);
    -webkit-transition: all 1s ease;
    transition: all 1s ease;
}
```

- `rotateZ(angle)` - defineerib 3D pööramist ümber Z-telje

#### Näide



Joonis 24 - rotateZ



Kujund on ennast pööranud vastupäeva nurga peale. Liikumine toimub z-telje järgi (Joonis 24 **Error! Reference source not found.**).

```
#rotatez.cube {
    /*-webkit-transform: rotateZ(220deg); Chrome, Safari, Opera */
    transform: rotateZ(220deg);
    /*-moz-transform: rotateZ(220deg); mozilla */
    transform: rotateZ(220deg);
    -webkit-transition: all 1s ease;
    transition: all 1s ease;
}
```

## 2.2 Testimise kokkuvõte

Järgnev tabel võtab kokku transformatsioonide testimise põhilistel veebilehtitsetel. Testimisel kasutati arvutit operatsioonisüsteemiga Windows 7 ning nutiseadmeid operatsioonisüsteemidega Android 5.0.2 ja IOS 7.1. Nendele vastavalt kasutati veebilehitsejaid Mozilla Firefox, Google Chrome ja Safari.

	Mozilla Firefox	Google Chrome	Safari
translate3d(x,y,z)	✓	✓	*
translateX(x)	✓	✓	*
translateY(y)	✓	✓	*
translateZ(z)	✓	✓	✘
scale3d(x,y,z)	✓	✓	*
scaleX(x)	✓	✓	*
scaleY(y)	✓	✓	*
scaleZ(z)	✓	✓	✘
rotate3d(x,y,z)	✓	✓	*
rotateX(angle)	✓	✓	*
rotateY(angle)	✓	✓	*
rotateZ(angle)	✓	✓	*

- ✓ Töötab ilma eesliiteta
- \* Näitab transformatsiooni aga 2D ruumis
- ✘ Ei tööta nii eesliitega kui ka ilma

Nagu tabelist näha, toimivad 3D transformatsioonid kõige vähem Apple'i toodete peal. Arvestades, et Apple Flashi ei toeta võiks ta toetada CSS transformatsioone. Osaliselt transformatsioonid toimivad kuid mitte täielikult.

Üldiselt võib väita, et CSS3 3D transformatsioonid toimivad üldjuhul nii nagu spetsifikatsioonis lubatud.

## Kokkuvõte

Käesoleva seminaritöö eesmärgiks oli anda ülevaade CSS3 3D transformatsioonidest üldiselt, testida nende toimimist erinevate veebilehitsejatega ning välja selgitada nende praktilisus. Selleks lõi autor transformatsioone kasutades kuubi, millega proovis erinevad transformatsioonid ja nende omadused läbi.

3D transformatsioonide osa on suuremas osas valmis ning seda peaks saama pea kõikidel platvormidel kõikide veebilehitsejatega kasutada. Spetsifikatsiooni kohaselt on kirjas ka eesliited erinevatele veebilehitsejatele, et transformatsioon töötaks kuid nt Mozilla Firefox ja Google Chrome-iga sai juba ilma. Safari näitab kuubi esitahku ilma eesliideteta 2D vaates kuid koos eesliitega toimub ka liikumine veebileheküljel algasukoha ning transformatsiooni vahel. Siiski näitab Safari ka siis kogu liikumist ainult 2D vaates. Transformatsioone on hea kasutada muutes veebilehte atraktiivsemaks ning huvitavamaks. Samuti toimivad need ka teksti illustreerimiseks.

Autor loeb eesmärgi täidetuks. Transformatsioone testiti erinevates olukordades erinevate veebilehitsejatega. Üllatuseks tuli tõsiasi, et transformatsioonid toimivad ilma eesliideteta kuigi kirjas on, et neid kasutatakse.

Hetkel olemas olevate transformatsioonide põhjal võib oletada, et need koguvad populaarsust ning koos CSS animation omadustega koos kasutades hakatakse veebi üha rohkem lisama animatsioone. Selle arvelt väheneb Flash'i osatähtsus veelgi. Heaks näiteks CSS3 3D transformatsioonidest on teinud Google, luues interaktiivse 3D rubiku kuubiku (Joonis 25). Sellega on võimalik arvutis mängida või luua ise midagi uut juhitudes olemasolevast koodist.

Tööst võiks kasu olla kõigile, kes esmakordselt CSS3 3D transformatsioonidega kokku puutuvad või lihtsalt nende vastu huvi tunnevad.

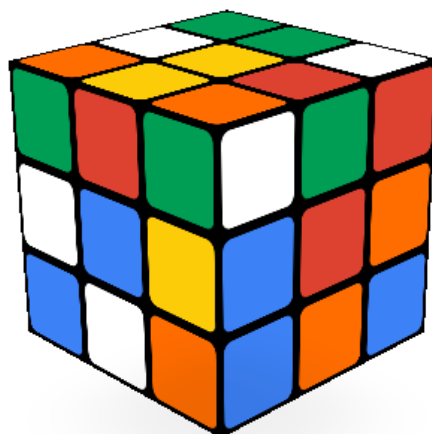
## Kasutatud kirjandus

1. <http://w3techs.com/technologies/details/cp-flash/all/all> - külastatud 02.11.2014
2. <https://developer.mozilla.org/en-US/docs/Web/CSS/backface-visibility> - külastatud 02.11.2014
3. [http://www.w3schools.com/cssref/css3\\_pr\\_perspective.asp](http://www.w3schools.com/cssref/css3_pr_perspective.asp) - külastatud 02.11.2014
4. [http://www.w3schools.com/cssref/css3\\_pr\\_transform-origin.asp](http://www.w3schools.com/cssref/css3_pr_transform-origin.asp) - külastatud 02.11.2014
5. [http://www.w3schools.com/cssref/css3\\_pr\\_perspective-origin.asp](http://www.w3schools.com/cssref/css3_pr_perspective-origin.asp) - külastatud 02.11.2014
6. [http://www.w3schools.com/css/css3\\_3dtransforms.asp](http://www.w3schools.com/css/css3_3dtransforms.asp) - külastatud 02.11.2014
7. [http://www.w3schools.com/css/css3\\_2dtransforms.asp](http://www.w3schools.com/css/css3_2dtransforms.asp) - külastatud 30.10.2014
8. <http://www.w3.org/Style/CSS/specs#transforms> – külastatud 02.11.2014
9. <http://www.w3.org/TR/css3-transitions/#transition-shorthand-property> – külastatud 1.11.2014
10. [http://www.w3schools.com/cssref/trycss3\\_transform-origin\\_3d\\_inuse.htm](http://www.w3schools.com/cssref/trycss3_transform-origin_3d_inuse.htm) - külastatud 02.11.2014
11. <http://css3.bradshawenterprises.com/transitions/> - külastatud 02.11.2014
12. [http://www.w3schools.com/css/css3\\_transitions.asp](http://www.w3schools.com/css/css3_transitions.asp) - külastatud 02.11.2014
13. [https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Using\\_CSS\\_transitions](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Using_CSS_transitions) - külastatud 01.11.2014
14. <http://www.css3.info/preview/css3-transitions/> - külastatud 23.06.2014
15. [http://www.w3schools.com/css/css3\\_intro.asp](http://www.w3schools.com/css/css3_intro.asp) - külastatud 30.11.2014
16. <http://desandro.github.io/3dtransforms/examples/transforms-01-functions.html> - külastatud 23.06.2014
17. [http://www.tutorialspark.com/css3/CSS3\\_3D\\_Transforms.php](http://www.tutorialspark.com/css3/CSS3_3D_Transforms.php) - külastatud 24.06.2014
18. [http://www.w3schools.com/html/html\\_colors.asp](http://www.w3schools.com/html/html_colors.asp) - külastatud 01.03.2015
19. <http://www.tlu.ee/~oimk/seminaritoo/>
20. <http://www.wired.com/2014/05/google-doodle-rubiks-cube/#slide-id-904031> – külastatud 02.11.2014
21. [https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Using\\_CSS\\_transforms](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Using_CSS_transforms) - külastatud 01.03.2015
22. [http://www.w3schools.com/cssref/css3\\_pr\\_transform-style.asp](http://www.w3schools.com/cssref/css3_pr_transform-style.asp) - külastatud 01.03.2015

23. [http://www.w3schools.com/cssref/css3\\_pr\\_perspective-origin.asp](http://www.w3schools.com/cssref/css3_pr_perspective-origin.asp) - külastatud 01.03.2015
24. [http://www.w3schools.com/cssref/css3\\_pr\\_perspective.asp](http://www.w3schools.com/cssref/css3_pr_perspective.asp) - külastatud 01.03.2015
25. [http://www.w3schools.com/cssref/css3\\_pr\\_backface-visibility.asp](http://www.w3schools.com/cssref/css3_pr_backface-visibility.asp) - külastatud 01.03.2015
26. [http://www.w3schools.com/css/css3\\_2dtransforms.asp](http://www.w3schools.com/css/css3_2dtransforms.asp) - külastatud 01.03.2015
27. [http://www.w3schools.com/cssref/css3\\_pr\\_transform-style.asp](http://www.w3schools.com/cssref/css3_pr_transform-style.asp) - külastatud 01.03.2015
28. <https://www.chrome.com/cubelab#tech> – külastatud 01.03.2015
29. <http://www.w3.org/Style/CSS/current-work> - külastatud 01.03.2015
30. [http://www.w3schools.com/css/css\\_intro.asp](http://www.w3schools.com/css/css_intro.asp) - külastatud 01.03.2015
31. [http://www.w3schools.com/css/css3\\_transitions.asp](http://www.w3schools.com/css/css3_transitions.asp) - külastatud 01.03.2015
32. [http://www.w3schools.com/cssref/css3\\_pr\\_transition.asp](http://www.w3schools.com/cssref/css3_pr_transition.asp) - külastatud 01.03.2015
33. [http://www.w3schools.com/cssref/css3\\_pr\\_transition-delay.asp](http://www.w3schools.com/cssref/css3_pr_transition-delay.asp) - külastatud 01.03.2015
34. [http://www.w3schools.com/cssref/css3\\_pr\\_transition-duration.asp](http://www.w3schools.com/cssref/css3_pr_transition-duration.asp) - külastatud 01.03.2015
35. [http://www.w3schools.com/cssref/css3\\_pr\\_transition-property.asp](http://www.w3schools.com/cssref/css3_pr_transition-property.asp) - külastatud 01.03.2015
36. [http://www.w3schools.com/cssref/css3\\_pr\\_transition-timing-function.asp](http://www.w3schools.com/cssref/css3_pr_transition-timing-function.asp) - külastatud 01.03.2015
37. [http://www.w3schools.com/cssref/css3\\_pr\\_perspective-origin.asp](http://www.w3schools.com/cssref/css3_pr_perspective-origin.asp) - külastatud 01.03.2015

## Lisa 1

Interaktiivne Google Rubiku kuubik



4



### Joonis 25 - Rubiku kuubik

Google lõi kasutades kõiki CSS3 3D transformatsioone, Javascripti, Three.js jm kaasaegse Rubiku kuubiku. Selleni jõuti alles hiljaaegu (mai 2014) kuna arvati, et tehnoloogia polnud veel valmis selleks. Veebilehitsejate suure toetuse tõttu 3D transformatsioonidele on see kättesaadav enamus kasutajatele, kellel on ligipääs internetile. (Edidin, „How Google Built It’s 3D Rubiks’s Cube Doodle”, 2014). 3D muudab kujutise justkui elavaks, kui tavalise 2D kujundi puhul jääks väga pealiskaudne mulje.