

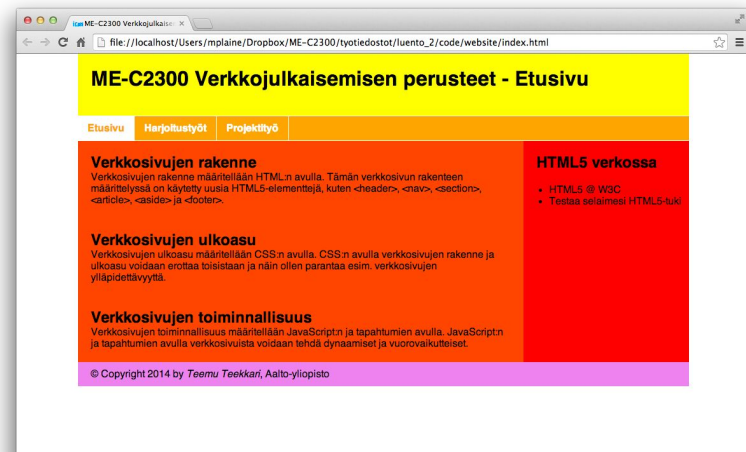
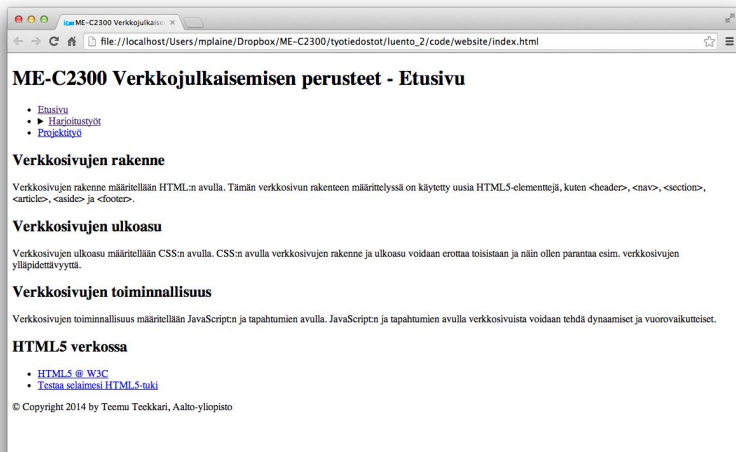
# Verkkosivujen ulkoasu (CSS)

Verkojulkaisemisen perusteet (5 op)

Pauli Laine

Aalto, tuntiopettaja

***”Luennon jälkeen osaan toteuttaa verkkosivuille ulkoasun CSS:llä.”***



*Ennen (vain HTML)*

*Jälkeen (HTML ja CSS)*

# Luennon sisältö

Kertausta

Cascading Style Sheets

Ulkoasun määrittäminen

Elementtien valinta

HTML sivun kokeilu Aallon verkkopalvelimella

Elementtien sijoittelu ja näkyvyys - Grid

Esimerkkikoodit

Responsiivinen suunnittelu

Harjoittelua (aina välillä)

# Luennon käytänteistä

Ensin luento jossa selitetään valitsimet ja parametrit

Sitten kokeilua opettajan tukemana

Harjoittele joko kirjoittamalla koodia koneellasi tai  
Online-editorilla (jossa näet muutokset heti)

Et (tod.näk.) opi kaikkea tänään. Voit lukea lisää ja tehdä  
lisää harjoituksia:

[https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction\\_to\\_CSS](https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS)

<https://www.w3schools.com/css/>

# CSS käytänteistä

CSS on yksinkertaisesti käytettynä helppo järjestelmä.

CSS vaikeutuu kuitenkin nopeasti, jos HTML on monimutkainen ja käytetään useita tyylikerroksia.

Mahdollistaa hyvin monipuoliset ja monimutkaiset tyylimuunnokset.

CSS-on laaja ammattilaistason tyyliohjekieli, jonka hyvä hallinta vaatii vuosien harjoittelun. Tällä kurssilla riittää perustason osaaminen.

# CSS - yleistä

- CSS on monimutkainen ohjelmointikieli, jolla HTML-sivun tyyli voidaan määrittää
- Suunnittelijan on tiedettävä/ymmärrettävä muokkaamansa sivun rakenne
  - Mitkä ovat käytetyt elementit
  - Elementtien hierarkia
  - Elementtien ID:t ja luokat
  - Joko niin, että html-suunnittelija ja graafinen suunnittelija ovat sama henkilö, tai todella hyvässä yhteistyössä!

# HTML+CSS yhdessä

- Sivuston kehittäminen yleensä iteratiivinen prosessi
- Kun kehität rakennetta, voit joutua muuttamaan tyylejä
  - Esim uudet otsikkotasot
  - Uudet muotoilut, palstat
- Joskus uusi sisältö pakottaa muuttamaan CSS määrittämiä
- Vähän kuin ratkaisisi ristisanatehtävää. Voi olla hidasta.



# Pari sanaa graafisesta suunnittelusta

- Selkeys
- Yksinkertaisuus
- Harmoniset värit (ei kaikkia värisävyjä, sävyjen sukulaisuus)
  - Joillain väreillä on merkityksiä: linkki, varoitus, klikattu
- Tyhjä tila
- Mielellään vain yhtä fonttiperhettä

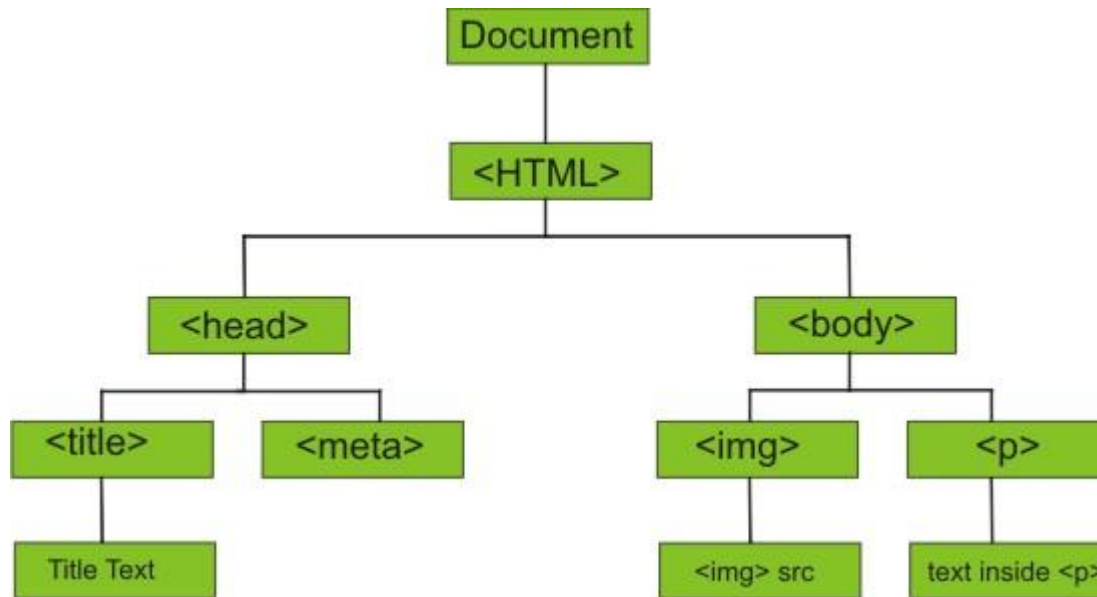
---

# Kertausta

# Index.html

Lisäyksenä viime viikkoon:

**Index.html on kansiossa se kotisivu, jonka verkkoselain tulkitsee kotisivuksi.**





- Opiskelu ja ohjelmat
- Tiede ja tutkimus
- Yhteistyö ja kumppanuudet
- Korkeakoulu ja yksiköt
- Tietoa yliopistosta



Aalto.fi uudistuu Lue lisää Anna palautetta

```
<script type="text/javascript">try{Typekit.load();}catch(e){}</script>
<script>...</script>
<style type="text/css">...</style>
<div id="bottom_renewal_bar" style="background-image: url(http://www.aalto.fi/style2013/img/renewal2013_fi.gif); width: 417px; display: block;">
  <a style="width: 170px;" class="close" href="#close">&nbsp;</a>
  <a style="width: 110px;" href="http://www.aalto.fi/fi/current/news/view/2013-05-22/">&nbsp;</a>
  <a style="width: 135px;" href="http://bit.ly/12P5Bu8">&nbsp;</a>
</div>
<script type="text/javascript">...</script>
<!-- Google Code for Main list - koko sivusto -->
<!-- Remarketing tags may not be associated with personally identifiable information or placed on pages related to sensitive categories. For instructions on adding this tag and more information on the above requirements, read the setup guide: google.com/ads/remarketingsetup -->
<script type="text/javascript">...</script>
<script type="text/javascript" src="//www.googleadservices.com/pagead/conversion.js">
</script>
<iframe name="google_conversion_frame" title="Google conversion frame" width="300" height="13" src="http://googleads.g.doubleclick.net/pagead/viewthroughconversion/10249_
Su java=true&u npluq=14&u nmime=98&url=http%3A//www.aalto.fi/fi/&frm=0" frameborder="0" marginwidth="0" marginheight="0" vspace="0" hspace="0" allowtransparency="true" scrolling="no">
...</iframe>
</noscript>
<script type="text/javascript">...</script>
</body>
</html>
```

Computed Style Show inherited  
Styles  
element.style { width: 170px; }  
Matched CSS Rules  
#bottom\_renewal\_bar a { www.aalto.fi/fi/:1001 display: block; float: left; height: 36px; margin: 0px; padding: 0px; }  
.close { bootstrap\_min.css:858 float: right; font-size: 20px; font-weight: bold; line-height: 20px; color: #000000; text-shadow: 0 1px 0 #ffffff; opacity: 0.2; }

---

# Cascading Style Sheets



Eri mediatyyppien  
*tyylimäärittelykieli*

Verkkosivun rakenne (HTML)  
ja ulkoasu (CSS) erotetaan

Koostuu *säännöistä*

Verkkosivun *ulkoasu*

Caskading viittaa  
*tyylhierarkioihin*

Dynaaminen tyyli (värit, fontit  
layout l. elementtien  
*sijoittelu*)

# Syntaksi



```
valitsin {  
  ominaisuus: arvo;  
}
```

# CSS

```
7 h1 {  
8   color: red;  
9   font-family: cursive;  
10  font-size: 30px;  
11 }
```

```
1  /** Kommentti **/  
2  
3 ▼ valitsin {  
4     ominaisuus: arvo;  
5 }
```

# Käyttö

*\*.css / <style> / style="..."*

# KÄYTTÖTAVAT \*

- 1) Selaimen oletustyylimäärittelyt
- 2) Ulkoinen tyylimäärittely (\*.css –tiedosto)
- 3) Sisäinen tyylimäärittely (<style> –elementti)
- 4) Elementin sisäinen tyylimäärittely (style –attribuutti)

Lisäksi erityisiä tyylielementtejä, kuten <strong> ja <em>. @import –komennon avulla \*.css –tiedostoja voi liittää toisiinsa.

# SELAIMEN OLETUSTYYLIMÄÄRITTELYT

Selaimet määrittelevät elementeille oletustyyliä. Selainten tyyliominaisuuksien välillä pieniä eroja.

## *Oletustyylien nollaaminen*

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

## *Oletustyylien yhtenäistäminen*

<http://necolas.github.io/normalize.css/>

# ULKOINEN TYYLIMÄÄRITTELY

Tyylit voidaan määritellä ulkoisessa tiedostossa (**\*.css**).

Ulkoinen tyylitiedosto voidaan linkittää verkkosivuun ja näin ollen samoja tyylimäärittelyitä voidaan käyttää usealla eri verkkosivulla (helpottaa ylläpitoa). Selaimet voivat myös tallentaa ulkoisen tyylitiedoston välimuistiin (nopeuttaa verkkosivujen lataamista). **Suosittelavin käytötapa.**

***Kannattaa huomata, että opetteluvaiheessa ulkoisen tyylitiedoston käyttö on hiukan hitaampaa/monimutkaisempaa.***

## ***Käyttäminen verkkosivuilla \****

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

# SISÄINEN TYYLIMÄÄRITTELY

Tyylit voidaan määrittellä verkkosivun sisällä (`<style>`).

Sisäinen tyylimäärittely keskittää verkkosivun tyylimäärittelyt yhteen paikkaan (helpottaa hieman ylläpitoa). Samoja tyylimäärittelyjä ei kuitenkaan voi käyttää muilla verkkosivuilla eivätkä selaimet pysty tallentamaan niitä välimuistiin.

## *Käyttäminen verkkosivuilla \**

```
<style type="text/css">...</style>
```



# ELEMENTIN SISÄINEN TYYLIMÄÄRITTELY

Tyylit voidaan määrittellä elementin sisällä (***style="..."***).

Style –attribuuttia on helppo ja nopea käyttää. Niiden ylläpidettävyys verkkosivulla on kuitenkin hankalaa.

Lisäksi sen käyttö aiheuttaa toisteisuutta (virhealttius kasvaa), sillä mahdollisesti samat tyylimäärittelyt on tehtävä erikseen jokaiselle elementille. Style –attribuutin tyylimäärittelyjä ei voi yliajaa muilla käyttötavoilla.

## ***Käyttäminen verkkosivuilla \****

```
<div style="...">...</div>
```

HTML

CSS

HTML

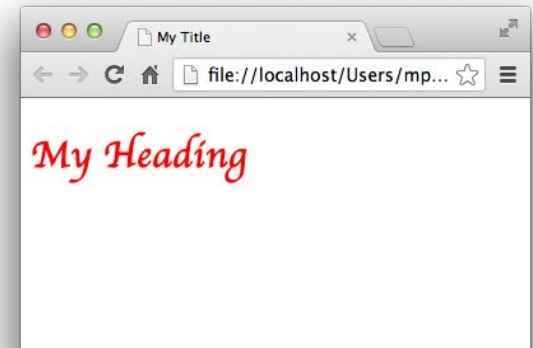
```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8"/>
5     <title>My Title</title>
6     <style type="text/css">
7       h1 {
8         color: red;
9         font-family: cursive;
10        font-size: 30px;
11      }
12    </style>
13  </head>
14  <body>
15    <h1>My Heading</h1>
16  </body>
17 </html>
```

HTML

CSS

HTML

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8"/>
5     <title>My Title</title>
6     <style type="text/css">
7       h1 {
8         color: red;
9         font-family: cursive;
10        font-size: 30px;
11      }
12    </style>
13  </head>
14  <body>
15    <h1>My Heading</h1>
16  </body>
17 </html>
```



---

# Ulkoasun määrittäminen

# Teksti ja fontit

Tekstiominaisuuksien avulla voidaan määrittää **tekstin ulkoasu**, kuten käytettävä tasaus ja koristelu. Alla olevassa esimerkissä <p> –elementtien teksti on tasattu keskelle sekä alleviivattu.

## Ominaisuuksia tekstille \*

text-align

text-decoration

```
1 p {  
2   text-align: center;  
3   text-decoration: underline;  
4 }
```

Fonttiominaisuuksien avulla voidaan määrittää **tekstin ulkoasu**, kuten käytettävä kirjasinlaji ja sen koko. Eri tietokoneille on asennettu eri fontit, joten kaikki fontit eivät näy kaikissa tietokoneissa. **serif**, **sans-serif** ja **cursive** kuitenkin löytyvät kaikilta tietokoneilta. Fonttivaihtoehtoja voi antaa useamman, joista valitaan ensimmäinen käyttäjän tietokoneelta löytyvä fontti.

## Ominaisuuksia fontille \*

font (lyhyt muoto)

font-family

font-size

```
1  p {  
2      font-family: Verdana, sans-serif;  
3      font-size: 30px;  
4  }
```

Viewing 819 of 819 font families

Roboto  
Christian Robertson (12 styles)



All their  
equipment and  
instruments are  
alive.

Pangolin  
Kevin Burke (1 style)



A red flair  
silhouetted the  
jagged edge of a  
wing.

Sansita  
Omnibus-Type (8 styles)



I watched the storm,  
so beautiful yet  
terrific.

Open Sans  
Steve Matteson (10 st

Almost  
knew it,  
left the

Slabo 27px  
John Hudson (1 style)



A shining crescent  
far beneath the  
flying vessel

Lato  
Lukasz Dziedzic (10 styles)



It was going to be  
a lonely trip back.

Fira Sans Condensed  
Carrois Apostrophe (18 styles)



Mist enveloped the  
ship three hours  
out from port

Roboto Condensed  
Christian Robertson (t

My two r  
memory

Try typing directly into the text fields.

GOT IT



# Yksiköt ja arvot

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

Ominaisuuksien arvot ovat *yleensä* joko jotain tietotyyppiä (esim. *pituus*) tai ennalta määriteltäviä arvoja (esim. *center*). Pituus voidaan ilmoittaa käyttäen joko absoluuttista yksikköä (esim. *px* tai *mm*) tai suhteellista yksikköä (esim. fr, *em* tai *vw*).

Elementin fonttikoko voidaan siis ilmoittaa esimerkiksi muodossa *2em*, jolloin sen fonttikooksi määräytyy kaksi kertaa nykyinen fonttikoko.

# Värit ja taustat

Tekstin väri voidaan antaa useassa eri muodossa: **nimenä** (esim. red), **heksadesimaaliarvona** (esim. #FF0000), **rgb-funktiolla** (esim. rgb(255, 0, 0)) tai **rgba-funktiolla** (esim. rgba(255, 0, 0, 1.0)). Värit saattavat näyttää erilaisilta eri tietokoneissa johtuen käytössä olevista väriprofiileista. Värien läpinäkyvyys puolestaan voidaan määrittää numerolla 0.0 (täysin läpinäkyvä) ja 1.0 (täysin näkyvä) väliltä.

## Arvojen muuntaminen

<http://hex.colorrrs.com/>

```
1 p {  
2   color: #FF0000; /* red */  
3   opacity: 0.3;  
4 }
```

# CSS-funktioita

CSS:n on määritetty apufunktioita, joiden avulla voidaan mm. suorittaa laskutoimituksia tai määritellä värejä.

- `rgba()` määrittelee värin
- `calc()` mahdollistaa laskutoimituksen
- `radial-gradient()` liukuväri

Taustaominaisuuksien avulla voidaan määrittää *elementin tausta*, kuten käytettävä väri tai kuva.

## *Ominaisuuksia taustalle* \*

background (lyhyt muoto)

background-color

background-image

```
1 p {  
2   background-color: #FF0000;  
3 }
```

---

# Elementtien valinta

# Selectors

*<http://www.w3.org/TR/selectors/>*



Valitsijoiden (engl. **selectors**) avulla verkkosivulta voidaan valita halutut elementit, joihin tyylimäärittelyt kohdistetaan. Elementtejä voi valita monella eri tapaa, kuten niiden *nimen*, *attribuutin*, *lapselementtien*, *luokan* tai *ID:n* perusteella. Myös näiden yhdistelmät ovat sallittuja ja kaikki elementit voidaan valita \* –valitsimen avulla.

Pseudoluokkavalitsijoiden (engl. **pseudo-class selector**) avulla valituista elementeistä voidaan valita vain osajoukko tai tietty tila. Esimerkiksi viimeinen lapsi (huomaa kaksoispiste): **elementinnimi:last-child { ... }**

**.class ja #id**

Tyylimäärittelyt voidaan kohdistaa elementeille muutenkin kuin niiden nimien perusteella. Esim. ***elementinnimi { ... }***

Luokkien (***class*** –attribuutti elementissä) avulla sama tyylimäärittely voidaan kohdistaa useampaan – vaikka erilaiseen – elementtiin verkkosivulla. Esimerkiksi (huomaa *piste*): ***.attribuutarvo { ... }***

ID:t (***id*** –attribuutti elementissä) puolestaan mahdollistavat tyylimäärittelyn kohdistamisen yhteen *uniikkiin* elementtiin. Esimerkiksi (huomaa *risuaita*): ***#attribuutarvo { ... }***

*body{ ... }*

*.kaupungit { ... }*

*#kissateksti { ... }*

# Pseudoluokka - Pseudo-class

Apukäsite, joka CSS:ssä mahdollistaa elementtien luokittelun samaan tapaan kuin luokkien avulla mutta HTML-dokumentin ulkopuolisiin seikkoihin perustuen, esim. sen mukaan, onko linkkiä seurattu vai ei.

<http://jkorpela.fi/styles/termit.html#class>

```
/* unvisited link */  
a:link {  
    color: green;  
}
```

```
/* visited link */  
a:visited {  
    color: green;  
}
```

```
/* mouse over link */  
a:hover {  
    color: red;  
}
```

```
/* selected link */  
a:active {  
    color: yellow;  
}
```

# Yhdistelmävalitsimet - Combinators

- Viittaavat HTML-rakenteen alirakenteisiin (esim. divin sisällä oleva toinen div tms.). Nämä on hyvä tunnistaa.
- `()`
- `>`
- `+`
- `~`

# HUOMAA!

**Jos käytät monimutkaisia valintoja**  
**- esim elementtien nimiä, ID:tä yms yhdessä**  
**-aliluokilla lisäominaisuuksia**

## **TYYLIN KÄYTTÖ JA RAKENNE MONIMUTKAISTUU!**

---

# Elementtien sijoittelu ja näkyvyys



# block vs. inline

HTML –elementit voidaan jakaa karkeasti **block** ja **inline** –tason elementteihin.

**block** –tason elementit (esim. **div**) käyttävät koko käytössä olevan leveyden tai vaihtoehtoisesti toteuttavat niille määritellyn leveysarvon mahdollisen korkeusarvon lisäksi. Ne myös pakottavat seuraavan elementin uudelle riville.

### **block –tason elementit**

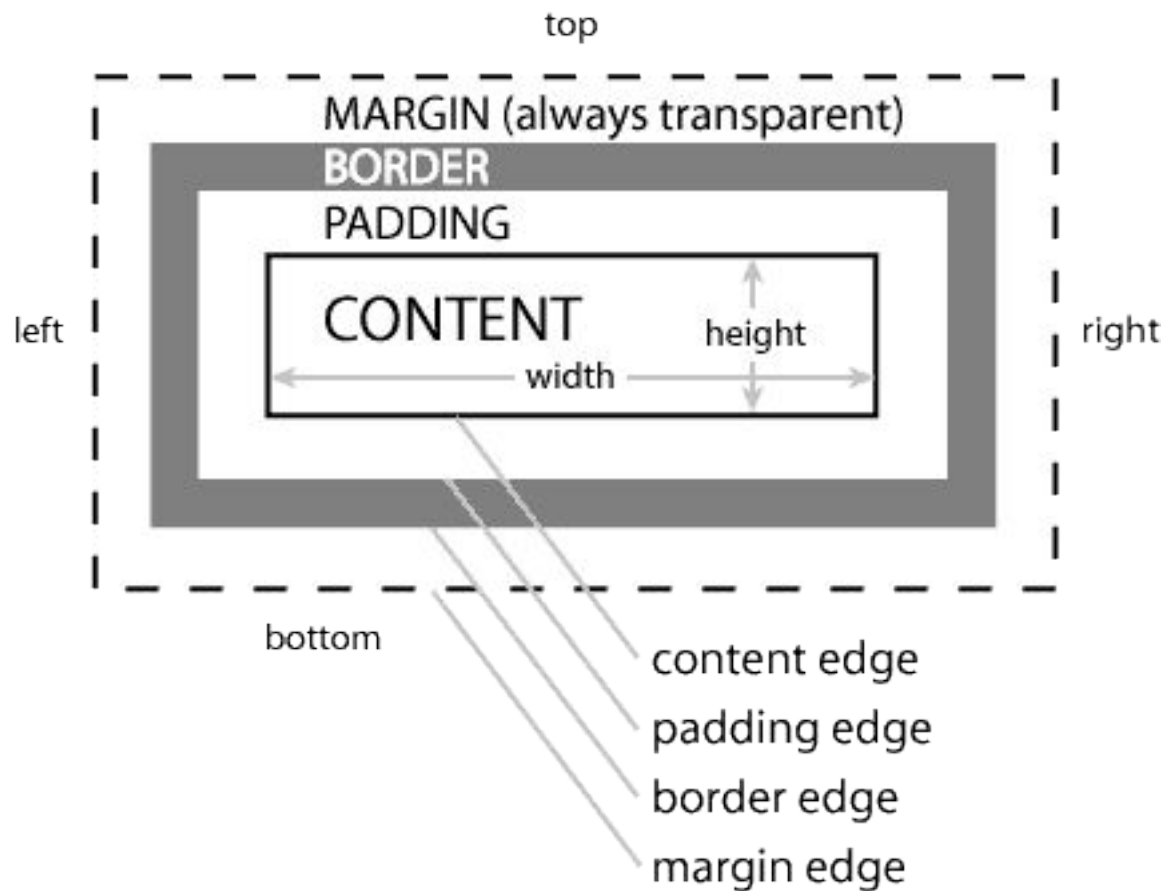
[https://developer.mozilla.org/en-US/docs/HTML/Block-level\\_elements](https://developer.mozilla.org/en-US/docs/HTML/Block-level_elements)

*inline* –tason elementit (esim. *span*) käyttävät vain sisällön tarvitseman tilan eli leveys- ja korkeusarvot eivät koske niitä. Ne eivät myöskään pakota seuraavia elementtejä uudelle riville.

### *inline-tason elementit*

[https://developer.mozilla.org/en-US/docs/HTML/Inline\\_elements](https://developer.mozilla.org/en-US/docs/HTML/Inline_elements)

# CSS-laatikkomalli



Sisällön leveys ja korkeus määritellään *width* ja *height* –ominaisuuksien avulla (*block* –tason elementit). Näille voidaan asettaa myös minimi- ja maksimiarvot (esim. *min-width* ja *max-width*). Taustamäärittelyt vaikuttavat sisältöalueeseen.

## *Ominaisuuksia sisällölle \**

height  
width

```
1  p {  
2    height: 100px;  
3    width: 200px;  
4  }
```

Täyte määrittelee tyhjän tilan (*padding*) sisältöalueen ympärille (leveyden ja korkeuden lisäksi). Täyte voidaan määritellä laitakohtaisesti. Taustamäärittelyt vaikuttavat täytealueeseen.

## *Ominaisuuksia täytteelle*

padding (lyhyt muoto)

padding-top

padding-right

padding-bottom

padding-left

```
1 p {  
2   padding: 5px 10px 5px 10px;  
3 }
```

Reunan paksuus, tyyli ja väri määritellään ***border-width***, ***border-style*** ja ***border-color*** –ominaisuuksien avulla. Reuna voidaan määritellä laitakohtaisesti ja sen paksuus lasketaan sisältöalueen ja täytealueen leveyden ja korkeuden lisäksi. Taustamäärittelyt eivät vaikuta reunaan.

## ***Ominaisuuksia reunalle \****

border-top	border (lyhyt muoto)
border-right	border-width
border-bottom	border-style
border-left	border-color

```
1  p {  
2    border: 5px solid #000000;  
3  }
```



Marginaali määrittelee tyhjän (läpinäkyvän) tilan (*margin*) reunan ympärille. Reuna voidaan määritellä laitakohtaisesti ja sen paksuus lasketaan sisältöalueen, täytealueen ja reunan paksuuden lisäksi. Taustamäärittelyt eivät vaikuta marginaaliin.

## *Ominaisuuksia marginaalille*

margin (lyhyt muoto)

margin-top

margin-right

margin-bottom

margin-left

```
1  p {  
2    margin: 10px 20px 10px 20px;  
3  }
```

# Media query ja Breakpoint (murtumispiste)

Kaikista CSS3:n tuomista ominaisuuksista media queryt ovat ehdottomasti responsiivisen suunnittelun kannalta tärkein uudistus. Käytännössä media queryt antavat luoda sääntöjä, jotka koskevat esimerkiksi vain tiettyä ruudunleveyttä tai näkymää.

Murtumispisteellä tarkoitetaan kokoa, jossa verkkosivun ulkoasu, muuttuu media queryjen ansiosta merkittävästi. Yleensä nämä ovat kokoja, joissa kehittäjä olettaa kävijän käyttävän tablettia tai älypuhelimta ja sivun layout muuttuu vastaavasti. (Niko Kauppinen, Responsiivinen web-design)

# Media query

```
@media only screen and (min-device-width :  
768px) {  
    CSS käsky (tee tekstistä punaista)  
}
```

Niin kauan kuin näkymän leveys on vähintään 768 pikseliä, teksti on punaista. Jos leveys on vähemmän kuin 768 pikseliä, sääntö ei enää ole voimassa.

# Sijoittelu

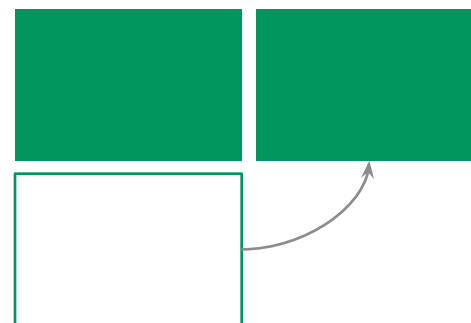
Tämä osio korvautuu pääosin  
grid-osiolla

*float, clear, position, ...*

Elementtien vakiosijoittelua (vasemmalta oikealle ja ylhäältä alas) verkkosivuilla voidaan muuttaa **float**-ominaisuuden avulla. Ominaisuus mahdollistaa elementtien kelluttamisen siten, että ne asettuvat vierekkäin sisällön mukaan joko vasemmalle tai oikealle. Alla olevassa esimerkissä **float: left;**

## **Ominaisuuden arvovaihtoehdot**

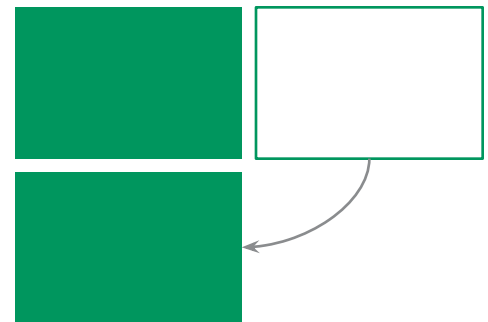
left  
right  
none



**clear** –ominaisuuden avulla elementtien sijoittelu verkkosivuilla voidaan palauttaa takaisin vakiosijoitteluun (vasemmalta oikealle ja ylhäältä alas). Ominaisuus mahdollistaa elementtien palauttamisen joko vasemmalta tai oikealta puolelta. Alla olevassa esimerkissä **clear: both;**

## **Ominaisuuden arvovaihtoehdot**

left  
right  
both  
none



Elementtejä voidaan sijoitella verkkosivuilla mielivaltaisesti **position** –ominaisuuden avulla. Ominaisuutta käytetään yhdessä **top**, **left**, **bottom** ja **right** –ominaisuuksien (arvot *pituutena*) kanssa. Ominaisuuden avulla elementtejä voidaan sijoitella *absoluuttisesti* (**absolute**), *suhteellisesti* (**relative**) tai liimata *tietyyn kohtaan* (**fixed**).

## Ominaisuuden arvovaihtoehdot

absolute

relative

fixed

static (vakio)

```
1 p {  
2   position: absolute;  
3   top: 100px;  
4   left: 200px;  
5 }
```

***absolute*** –*arvo* määrittää elementin *absoluuttisen* sijainnin suhteessa edelliseen isäntäelementtiin (jolla jokin muu kuin vakioarvo) tai sellaisen puuttuessa selainikkunaan.

***relative*** –*arvo* määrittää elementin *suhteellisen* sijainnin suhteessa siihen, mihin se normaalisti olisi sijoittunut.

***fixed*** –*arvo* liimaa elementin *tiettyyn kohtaan* selainikkunaa. Sen sijainti ei muutu edes verkkosivua vieritettäessä.

***static*** –*arvo* määrittää vakiokäyttäytymisen.



# Näkyvyys

*display, visibility, opacity, z-index, ...*

Elementtien näkyvyyttä voidaan säädellä usean eri ominaisuuden avulla, kuten *display*, *visibility*, *opacity*, ja *z-index*.

*display* –ominaisuus määrittää elementin ”CSS-laatikon” (arvoina esim. *block*, *inline-block* tai *inline*), ja näin ollen miten se käyttäytyy viereisiin elementteihin nähden. Sen avulla elementti voidaan myös poistaa kokonaan näkyvistä (arvona *none*), mukaan lukien sen viemä tila.

**visibility** –ominaisuus määrittää elementin näkyvyyden. Sen avulla elementti voidaan esimerkiksi näyttää (arvona **visible**) tai piilottaa näkyvistä (arvona **hidden**). Piilotettuna elementti ei näy, mutta siitä huolimatta se vie sen tarvitseman tilan.

**opacity** –ominaisuuden avulla voidaan säädellä elementin läpinäkyvyyttä (arvot väliltä **0...1**), josta on hyötyä esimerkiksi elementtien animoinnissa.

**z-index** –ominaisuus määrittää elementin *korkeustason*. Mitä suurempi numeroarvo sille annetaan (arvot voivat olla myös negatiivisia, **0** puolestaan on *vakioarvo*), sitä ylemmäksi se asetetaan piirtopinossa. Tämän avulla voidaan hallita päällekkäisten elementtien näkyvyyttä.

<https://developer.mozilla.org/en-US/docs/Web/CSS/z-index>

Elementtien näkyvyyttä voidaan myös säädellä hieman harvinaisempien **overflow** ja **clip** –ominaisuuksien avulla.

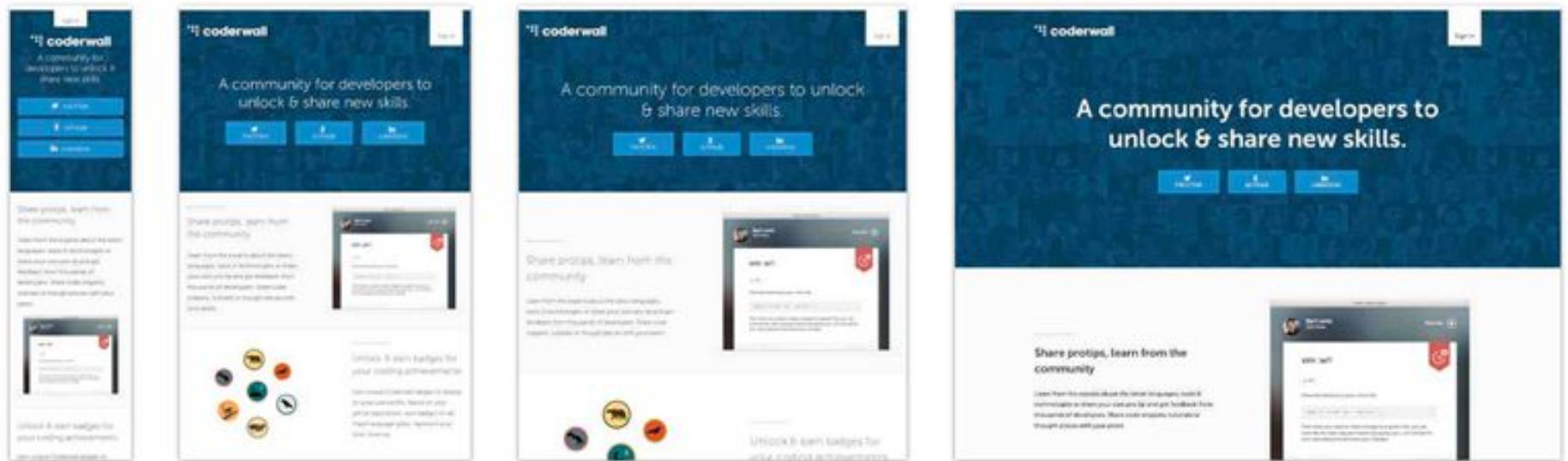
---

# Responsiivinen suunnittelu

# Yleiskuvaus

- Responsiivinen suunnittelu (engl. Responsive Web Design, RWD) on suunnitteluperiaate, jonka tavoitteena on luoda dynaamisesti taittavia (ja optimaalisia) verkkosivuja eri päätelaitteille
- Toteutuksen pääperiaatteet
  - 1) Joustava ruudukkomainen sivupohja
  - 2) Joustava (+optimoitu) mediasisältö
  - 3) Mediakyselyt
    - Sekä joustava mobiiliversio
- Läheisesti liittyvät konseptit
  - Mobile first
  - Progressive enhancement

## Coderwall



This responsive web design was created by [@coderwall](#) and suggested for inclusion in this collection by [@uggedal](#). It has had 4214 unique views since 2013-08-14.

Tagged with [fluid grids](#), [media queries](#), [flexible images](#) and [responsive web design](#).

# Media Queries

**Subscription Handbook**  
The complete handbook For  
choosing the right  
subscription billing service.



via Ad Packs

## Muumilaakso



This responsive web design was created by [@joonasy](#) and suggested for inclusion in this collection by [@joonasy](#). It has had 1292 unique views since 2013-12-27.

Tagged with [fluid grids](#), [media queries](#), [flexible images](#) and [responsive web design](#).



# Joustava ruudukkomainen sivupohja

- Kiinteän kokoinen sivupohja ei joustaa
  - Selainikkunan käydessä liian pieneksi, joutuu käyttäjä vierittämään sivua nähdäkseen sen kokonaan
- Joustava sivupohja mukautuu selainikkunan koon mukaan
  - Toteutus perustuu suhteellisten yksiköiden käyttöön (esim. palstojen leveydet) sekä maksimi- ja minimileveysrajoitukseen

# Joustava (+optimoitu) mediasisältö

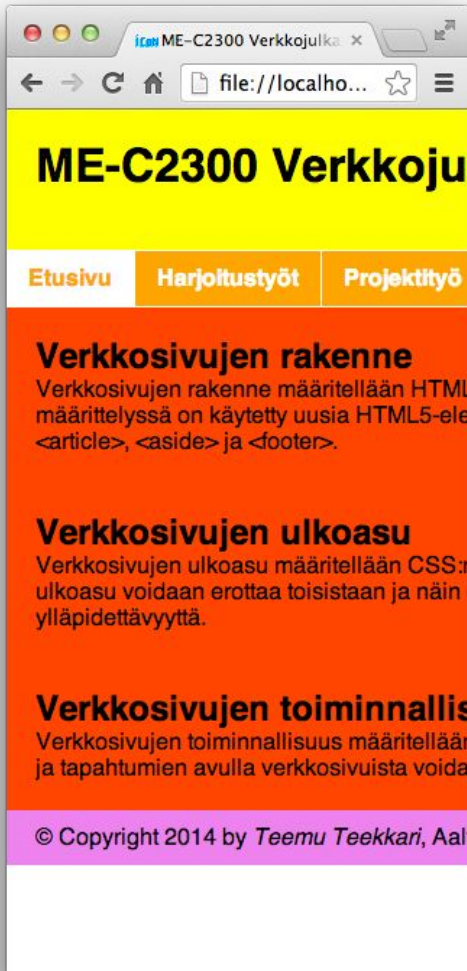
- Kiinteän kokoinen mediasisältö (kuvat, videot, pitkät tekstit) ei jousta
  - Selainikkunan käydessä liian pieneksi, mediasisältö ei enää mahdu isäntäelementin sisään ja näin ollen rikkoo sivupohjan
- Joustava mediasisältö mukautuu isäntäelementin koon mukaan
  - Toteutus perustuu maksimi- ja minimileveysrajoituksen asettamiseen mediasisällölle
- Kuvien resoluutio voidaan optimoida esim. seuraavasti
  - CSS, SVG, HTML5 <picture> ja Retina.js
- Videoiden kuvasuhde voidaan optimoida esim. seuraavasti
  - FitVid.js ja CSS

# Mediakyselyt (media-query)

- Joustavan sivupohjan ulkoasu ei enää toimi
  - Selainikkunan käydessä liian pieneksi tai isoksi, joustavan sivupohjan ulkoasu muuttuu epäkäytännölliseksi
- Mediakyselyt mahdollistavat murtumispisteiden lisäämisen sivupohjaan
  - Toteutus perustuu CSS-tyylimäärittelyihin (esim. width, display ja font-size), joita sovelletaan murtumispisteissä (esim. laitteen leveys tai resoluutio)
  - Tyylimäärittelyiden avulla on myös mahdollista näyttää/piilottaa selainikkunan elementtejä/elementtiryhmiä

# Joustava mobiiliversio

- Mobiiliselainten aloitusnäky on lukukelvoton
  - Sisältöä ei pysty lukemaan, koska mobiiliselaimet zoomaavat kauas sisällöstä
- Viewport:n käyttö korjaa tilanteen
  - HTML: `<meta name="viewport" content="initial-scale=1" />`
  - CSS: `@viewport { zoom: 1.0; width: extend-to-zoom;}`
  - Lisätietoa:  
<https://webdesign.tutsplus.com/articles/quick-tip-dont-forget-the-viewport-meta-tag--webdesign-5972>



*Ennen (original)*



*Jälkeen (responsive)*

---

# Esimerkki-koodit ja demo

# Verkkosivu ennen (vain HTML)



The screenshot shows a web browser window with the following content:

ME-C2300 Verkojulkaisemisen perusteet - Etusivu

- [Etusivu](#)
- ► [Harjoitustyöt](#)
- [Projektityö](#)

## Verkkosivujen rakenne

Verkkosivujen rakenne määritellään HTML:n avulla. Tämän verkkosivun rakenteen määrittelyssä on käytetty uusia HTML5-elementtejä, kuten <header>, <nav>, <section>, <article>, <aside> ja <footer>.

## Verkkosivujen ulkoasu

Verkkosivujen ulkoasu määritellään CSS:n avulla. CSS:n avulla verkkosivujen rakenne ja ulkoasu voidaan erottaa toisistaan ja näin ollen parantaa esim. verkkosivujen ylläpidettävyyttä.

## Verkkosivujen toiminnallisuus

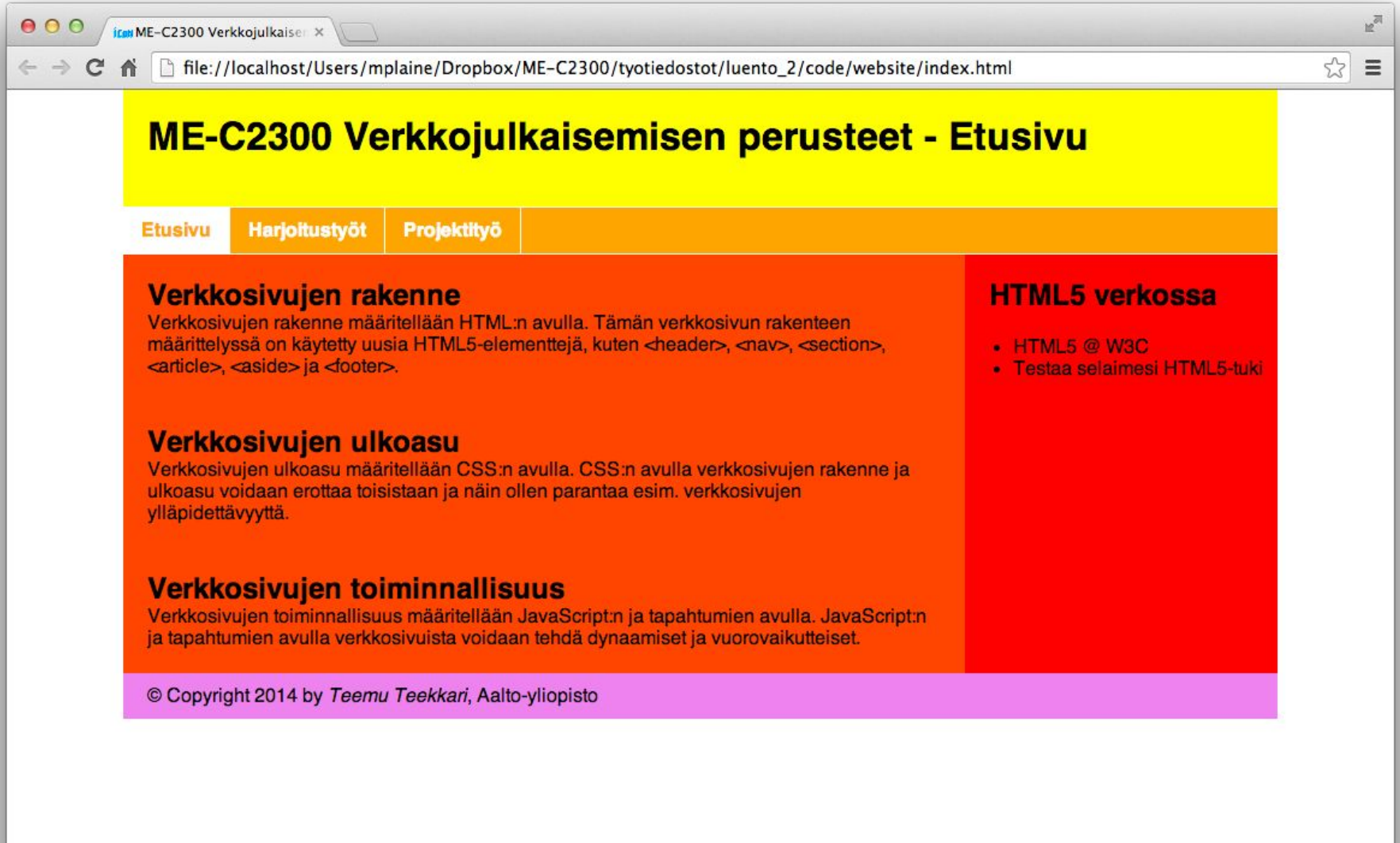
Verkkosivujen toiminnallisuus määritellään JavaScript:n ja tapahtumien avulla. JavaScript:n ja tapahtumien avulla verkkosivuista voidaan tehdä dynaamiset ja vuorovaikutteiset.

## HTML5 verkossa

- [HTML5 @ W3C](#)
- [Testaa selaimesi HTML5-tuki](#)

© Copyright 2014 by Teemu Teekkari, Aalto-yliopisto

# Verkkosivu jälkeen (HTML ja CSS)



The screenshot shows a web browser window with the following content:

- Address bar:** file:///localhost/Users/mplaine/Dropbox/ME-C2300/tyotiedostot/luento\_2/code/website/index.html
- Page Title:** ME-C2300 Verkojulkaisemisen perusteet - Etusivu
- Navigation:** Etusivu | Harjoitustyöt | Projektityö
- Main Content:**
  - Verkkosivujen rakenne**

Verkkosivujen rakenne määritellään HTML:n avulla. Tämän verkkosivun rakenteen määrittelyssä on käytetty uusia HTML5-elementtejä, kuten <header>, <nav>, <section>, <article>, <aside> ja <footer>.
  - Verkkosivujen ulkoasu**

Verkkosivujen ulkoasu määritellään CSS:n avulla. CSS:n avulla verkkosivujen rakenne ja ulkoasu voidaan erottaa toisistaan ja näin ollen parantaa esim. verkkosivujen ylläpidettävyyttä.
  - Verkkosivujen toiminnallisuus**

Verkkosivujen toiminnallisuus määritellään JavaScript:n ja tapahtumien avulla. JavaScript:n ja tapahtumien avulla verkkosivuista voidaan tehdä dynaamiset ja vuorovaikuttiset.
- Right Sidebar:**
  - HTML5 verkossa**
  - HTML5 @ W3C
  - Testaa selaimesi HTML5-tuki
- Footer:** © Copyright 2014 by Teemu Teekkari, Aalto-yliopisto



# Mobile-first design

# Mobile-first suunnittelu

- Tarkoittaa suunnittelua ensisijaisesti mobiililaitteille
- Selkeämpi käyttöliittymä mobiilikäyttäjille
- Lyhyemmät latausajat

# Validointi

# CSS LINT

Will hurt your feelings\*  
(And help you code better)

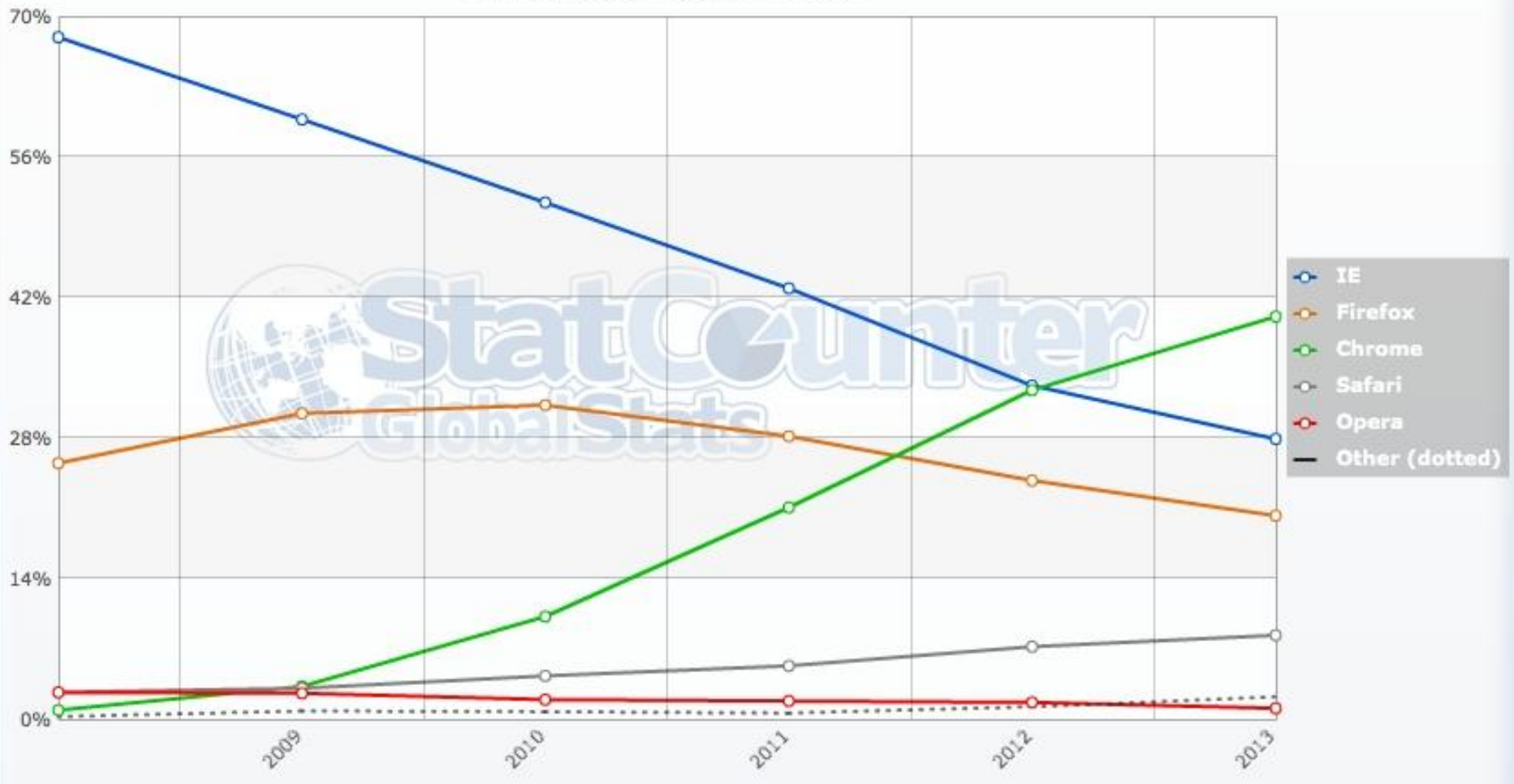
Your CSS goes here. The more, the better. Linting works best when we see the big picture, so give us everything you've got.

LINT!



# Selaintuki

### StatCounter Global Stats Top 5 Browsers from 2008 to 2013



Selaimet tulkitsevat CSS:ää selainmoottoreiden (engl. *layout engine*) avulla. Eri selaimissa on yleensä eri selainmoottorit, mistä johtuen CSS-ominaisuuksien selaintuki vaihtelee selaimesta toiseen.

## *Selaintuen testaus*

<http://css3test.com/>

<http://caniuse.com/>

<http://fmbip.com/litmus>

# CSS Kirjastot

- Mainitaan tässä esimerkin vuoksi, kirjastoja ei kuitenkaan käytetä tällä kurssilla!
- Esimäärittelevät luokkia ja tyylejä, jotta saadaan yhtenäinen ja suhteellisen helposti käytettävä tyyli
- Jopa värit on saatettu kirjaston puolesta kiinnittää (eri väreillä on eri merkityksiä), jolloin suunnittelija ei ole vapaa valitsemaan väriä
- 
- Esim:
  - Normalize.css
  - Bootstrap



# CSS esiprocessorit

<https://www.keycdn.com/blog/sass-vs-less>

CSS on its own can be fun, but stylesheets are getting larger, more complex, and harder to maintain. This is where a preprocessor can help. Sass lets you use features that don't exist in CSS yet like variables, nesting, mixins, inheritance and other nifty goodies that make writing CSS fun again.

---

# Yhteenveto ja tehtävänänto

# Tehtävänanto

- Toteuttakaa verkkosivusto HTML:n ja CSS:n avulla.
- Verkkosivuston pitää liittyä ryhmän **teemaan**.
- Harjoitustyö tehdään **itsenäisesti**.
- Harjoitustyön vaatimukset sekä tarkempi tehtävänanto löytyvät kurssin MyCourses-sivuilta Harjoitustyöt-osiosta.

# Aallon kotisivut ja winscp

Kotisivusi on muotoa:

`https://users.aalto.fi/~lainep8/`

Lisätietoa:

<https://inside.aalto.fi/display/ITPK/Kotisivut+-+users.aalto.fi>

Lataa tiedostosi omaan public html kansioosi winscp:llä

# Harjoitus – Sivuston sijoittaminen palvelimelle

- Avaa selaimessa aallon sivusi uudelleen
- Verkkosivusi pitäisi aueta
- Kokeile myös naapurin sivua!
  
- Muista, että lokaalisti (pöydälläsi olevalla tietokoneella) editoimasi tiedosto on ERI kuin palvelimella oleva. Jotta se olisi palvelimella, se pitää SIIRTÄÄ sinne!
- Varaudu siihen, että tuntien päätteeksi asiat testataan palvelimelle siirretyillä sivuilla. Varaudu myös siihen, että voit näyttää sivuja muille kurssilaisille ja opettajille!

# Users.aalto.fi

- <https://inside.aalto.fi/display/ITServices/www+home+pages+-+users.aalto.fi>
- Lataa tiedostosi omaan public html kansioosi winscp:llä

<https://users.aalto.fi/~lainep8/sivustotiimi.html>

- Testaa myös mobiililaitteella

Selaimet tulkitsevat CSS:ää selainmoottoreiden (engl. *layout engine*) avulla. Eri selaimissa on yleensä eri selainmoottorit, mistä johtuen CSS-ominaisuuksien selaintuki vaihtelee selaimesta toiseen.

## *Selaintuen testaus*

<http://css3test.com/>

<http://caniuse.com/>

<http://fmbip.com/litmus>

---

# CSS grid



# Grid hyödyt

- **Gridin** voit jakaa sivun pääalueisiin, joka vastaa HTML-rakennetta. Tämä tapahtuu helpommin kuin aikaisemmilla CSS-layout komennoilla
  - **CSS Grid Layout** excels at dividing a page into major regions or defining the relationship in terms of size, position, and layer, between parts of a control built from HTML primitives.

# Grid peruskäsitteet

- Responsiivisuus (esim. fr parametri) “css grid track size”
  - <https://www.joomlashack.com/blog/tutorials/determining-the-size-of-the-tracks-in-css-grid/>
- Elementtien sijoittelu rivien, nimen tms perusteella
- Tasaus (alignment)

[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Grid\\_Layout/Basic\\_Concepts\\_of\\_Grid\\_Layout](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout)

```
<div class="wrapper">
```

```
  <div class="one">One</div>
```

```
</div>
```

```
.wrapper {
```

```
  display: grid;
```

```
  grid-template-columns: repeat(3, 1fr);
```

```
  grid-auto-rows: minmax(100px, auto);
```

```
}
```

```
.one {
```

```
  grid-column: 1 / 3;
```

```
  grid-row: 1;
```

# Firefox grid inspector

- [https://developer.mozilla.org/en-US/docs/Tools/Page\\_Inspector/How\\_to/Examine\\_grid\\_layouts](https://developer.mozilla.org/en-US/docs/Tools/Page_Inspector/How_to/Examine_grid_layouts)

# Grid-template-areas

- nimetyt alueet, jotka voivat heijastella HTML-rakennetta, oikein hyvä esimerkki:

<https://developer.mozilla.org/en-US/docs/Web/CSS/grid-template-areas>

# Grid tracks

Grid track (“raita”?) on alue joka on määritelty grid rivien ja sarakkeiden avulla.

We define rows and columns on our grid with the [grid-template-columns](#) and [grid-template-rows](#) properties. These define grid tracks.

[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Grid\\_Layout/Basic\\_Concepts\\_of\\_Grid\\_Layout#Creation\\_of\\_additional\\_tracks\\_to\\_hold\\_content](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout#Creation_of_additional_tracks_to_hold_content)

# Grid fr-yksikkö

Fr-mittayksiköllä (fraction) tarkoitetaan jäljellä olevaa tilaa yläelementissä. Radan kaikille fr-mittayksikköä käyttäville alueille jaetaan tila sen mukaan, kuinka monta fr-yksikköä ne käyttävät (Atkins, Etemad & Atanassov 2015).

Tuomas Velling - Grid Layout ja Flexible Box -moduulit  
CSS-ohjelmistokehyksissä

<http://urn.fi/URN:NBN:fi:amk-2016052810550>

A flexible length or <flex> is a dimension with the **fr** unit, which represents a fraction of the leftover space in the grid container. Tracks sized with **fr** units are called flexible tracks as they flex in response to leftover spaces similar to how flex items fill space in a flex container.





# Holy grail - layout

[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Grid\\_Layout/Grid\\_Template\\_Areas](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Grid_Template_Areas)

<https://bitsofco.de/holy-grail-layout-css-grid/>

---

# Yhteenveto

# Yhteenveto

- Verkkosivujen *ulkoasu* voidaan määritellä CSS:n avulla. Myös elementtien *sijoittelua* ja *näkyvyyttä* voidaan hallita CSS:n kautta.
- CSS:stä on kehittynyt vuosien varrella todella *ilmaisuvoimainen* kieli.
- CSS3 koostuu erillisistä *moduuleista*. Eri moduuleita on lukuisia (50+) ja ne keskittyvät joko yhdessä tai erikseen eri asioiden ratkaisemiseen, kuten digilehden taitto tai animaatio.
- Responsiivinen suunnittelu parantaa käyttökokemusta eri päätelaitteilla.

# Yhteenveto - keskeiset asiat

- **Parametrit:** (väri, koko, fontti, animaatio yms.)
- **Valitsimet:** (tunnus/tagi, nimi/id, luokka)
- **Sijoittelu:** grid

---

# Tuntiharjoitukset

# Harjoitus – Verkkosivuston tyyli

- Tee sivun sisäisellä CSS-tyylitiedolla verkkosivustolle tyyli, jossa on ainakin:
  - Taustaväri, fonttikoko ja fontti vaihdettu oletusarvoista
  - Otsikoiden ympärillä kehykset

# Harjoitus – Verkkosivun tausta ja fontti

- Aloita tyhjästä. Suunnittele ja kirjoita ensin HTML
- Tee sivun sisään sijoitetulla CSS-tyylimäärityksellä seuraavat muutokset:
  - Vaihda taustaväri
  - Vaihda tekstien fontti
  - Laita sivulle n. 0.5 läpinäkyvä kuva

# Harjoitus – Sivuston sijoittaminen palvelimelle

- Avaa selaimessa aallon sivusi uudelleen
- Verkkosivusi pitäisi aueta
- Kokeile myös naapurin sivua!
  
- Muista, että lokaalisti (pöydälläsi olevalla tietokoneella) editoimasi tiedosto on ERI kuin palvelimella oleva. Jotta se olisi palvelimella, se pitää SIIRTÄÄ sinne!
- Varaudu siihen, että tuntien päätteeksi asiat testataan palvelimelle siirretyillä sivuilla. Varaudu myös siihen, että voit näyttää sivuja muille kurssilaisille ja opettajille!



# Harjoitus – Verkkosivuston tyyli

## 2

- Tee toisella CSS-tyylimäärittelyllä verkkosivustolle **täysin erilainen** tyyli, jossa on ainakin:
  - Sivustolle sopiva palstoitus
  - Taustaväri, fonttikoko ja fontti vaihdettu oletusarvoista
  - Otsikoiden ympärillä kehykset

# Harjoitus – kokeile

<http://www.csszengarden.com>

- Lataa jokin <http://www.csszengarden.com> tarjoamista tyyleistä
- Lataa Zen garden pohja, ja pohdi millaisia muutoksia voisit tehdä.
- Kokeile sitä sivustoosi
  - Mitä huomaat
  - Miten sivustosi elementtejä pitäisi muuttaa, jotta zengardenin tyyli sopisi?

# Tuntitehtävä - HG

Laadi 'Holy Grail' tyyppinen sivusto. Voit käyttää omaa teemaasi tms. pohjana

HTML rakenne

CSS määrittelyt

# Harjoitus – Animaatio

- Tee sivun sisään sijoitetulla CSS-tyylimäärittelyksellä seuraavat muutokset:
  - Kun kuvan päällä pitää hiirtää (:hover), niin se kääntyy 90 asteta

# Linkkejä

- <http://www.codecademy.com/courses/web-beginner-en-TlhFi/>
- <http://learn.shayhowe.com/>
- <http://www.w3.org/Style/CSS/current-work>
- <http://www.htmldog.com/reference/cssproperties/>
- <http://caniuse.com/>
- <http://css-tricks.com/>

# Linkkejä

## Media Queries

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

## CSS Device Adaptation

<http://www.w3.org/TR/css-device-adapt/>

## Selainikkunakoot

[https://en.wikipedia.org/wiki/List\\_of\\_displays\\_by\\_pixel\\_density](https://en.wikipedia.org/wiki/List_of_displays_by_pixel_density)

## Responsiivinen suunnittelu

<http://www.mikkokuhna.com/researcher/slides/20130224-responsive-web-design.pdf>

<https://aaltodoc.aalto.fi/handle/123456789/8986>

## Testaus

<http://ipadpeek.com/>

[namco.com/ghostlab/](http://namco.com/ghostlab/)

# Grid-linkkejä

<https://medium.freecodecamp.org/learn-css-grid-in-5-minutes-f582e87b1228>

<https://designshack.net/articles/css/best-ways-to-learn-css-grid/>

<https://alligator.io/css/css-grid-holy-grail-layout/>

[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Grid\\_Layout/Realizing\\_common\\_layouts\\_using\\_CSS\\_Grid\\_Layout](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Realizing_common_layouts_using_CSS_Grid_Layout)

<http://ireade.github.io/holy-grail-css-grid/>

KIITOS!