**Mikä on CSS**

CSS tarkoittaa Cascading Style Sheets -työkalua. Se on tyylitietokieli, jota käytetään kuvaamaan merkintäkielellä kirjoitetun asiakirjan ulkoasua ja muotoilua. Se tarjoaa lisäominaisuuden HTML: lle. Sitä käytetään yleensä HTML: n kanssa verkkosivujen ja käyttöliittymien tyylin muuttamiseen. Sitä voidaan käyttää myös minkä tahansa XML-asiakirjan, kuten tavallisen XML: n, SVG: n ja XUL: n, kanssa.

CSS: ää käytetään yhdessä HTML: n ja JavaScriptin kanssa useimmissa verkkosivustoissa luomaan käyttöliittymiä web-sovelluksiin ja käyttöliittymiin useille mobiilisovelluksille.

**Mitä CSS tekee**

    Voit lisätä vanhoihin HTML-dokumentteihin uusia ulkoasuja.

    Voit täysin muuttaa verkkosivustosi ulkoasua vain muutamalla CSS-koodin muutoksella.

**Nämä ovat CSS: n kolme suurta hyötyä:**

1) Ratkaisee suuren ongelman

Ennen CSS: ää tagit, kuten fontti, väri, tausta tyyli, elementtien linjaukset, raja ja koko oli toistettava jokaisella verkkosivulla. Tämä oli hyvin pitkä prosessi. Esimerkiksi: jos kehität suurta verkkosivustoa, jossa fontit ja väritiedot lisätään jokaiselle sivulle, siitä tulee pitkä ja kallis prosessi. CSS luotiin tämän ongelman ratkaisemiseksi. Se oli W3C: n suositus.

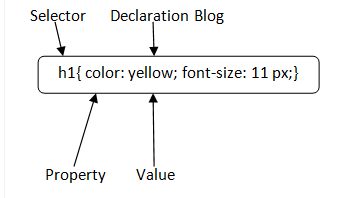
2) Säästää paljon aikaa

CSS-tyyppiset määritelmät tallennetaan ulkoisiin CSS-tiedostoihin, joten koko verkkosivustoa voidaan muuttaa vain yhdellä tiedostolla.

3) Antaa enemmän attribuutteja

CSS tarjoaa yksityiskohtaisempia attribuutteja kuin tavallinen HTML määritellä ulkoasun verkkosivuilla.

CSS syntaksi



**Selector eli Valitsin:** valitsin näyttää tyylin HTML-elementin. Se voi olla mikä tahansa tagi, kuten <h1>, <title> jne.

**Declaration Block eli** Julkilauseke: Ilmoituslohko voi sisältää yhden tai useamman ilmoituksen, joka on erotettu puolipisteellä. Edellä esitetyllä esimerkillä on kaksi ilmoitusta:

 väri: keltainen;

 fontin koko: 11 px;

color: yellow;

font-size: 11 px;

Jokainen ilmoitus sisältää omaisuuden nimen ja arvon, kaksoispiste erotettuna.

**Property eli Ominaisuus**: Kiinteä ominaisuus on HTML-elementin attribuutti. Se voi olla väri, raja jne.

**Values eli Arvo**: Arvot määritetään CSS-ominaisuuksille. Yllä olevassa esimerkissä arvo "keltainen" määritetään väriominaisuudelle.



CSS-valitsimia käytetään valitsemaan haluamasi sisältö. Valitsimet ovat osa CSS-sääntöjoukkoa. CSS-valitsimet valitse HTML-elementit sen id, luokan, tyypin, attribuutin jne mukaan.

CSS: ssä on useita erilaisia ​​valitsimia.

    1 CSS-elementtivalitsin

    2 CSS-tunnuksen valinta

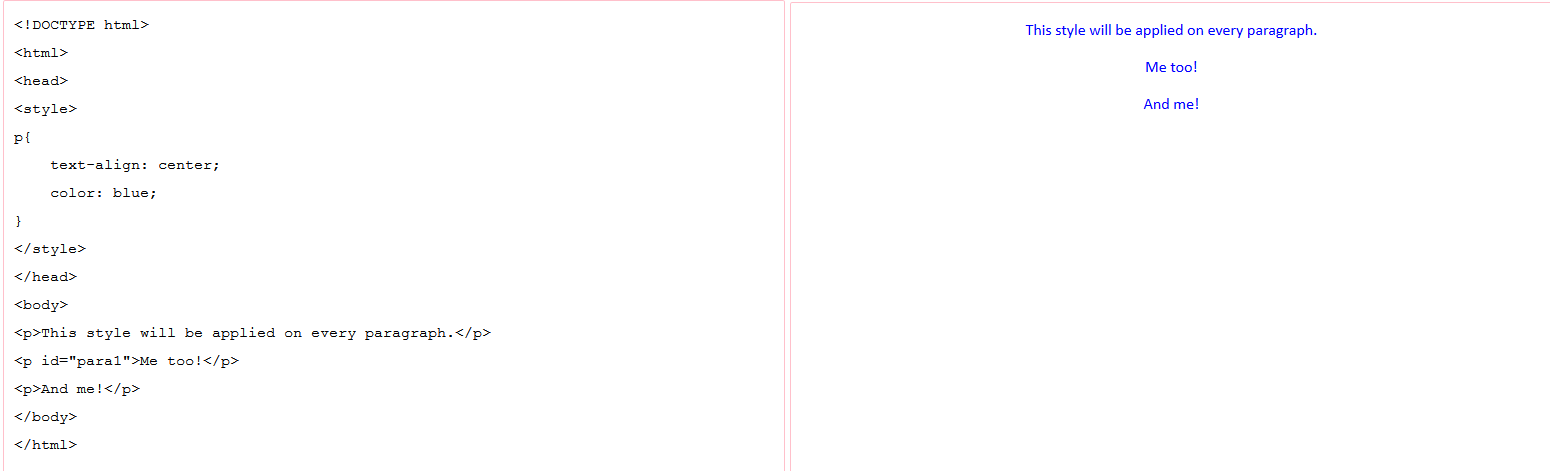
    3 CSS-luokan valitsin

    4 CSS-yleisvalitsin

    5 CSS-ryhmän valitsin

**1 CSS element Selector**

Elementtivalitsin valitsee HTML-elementin nimen mukaan.



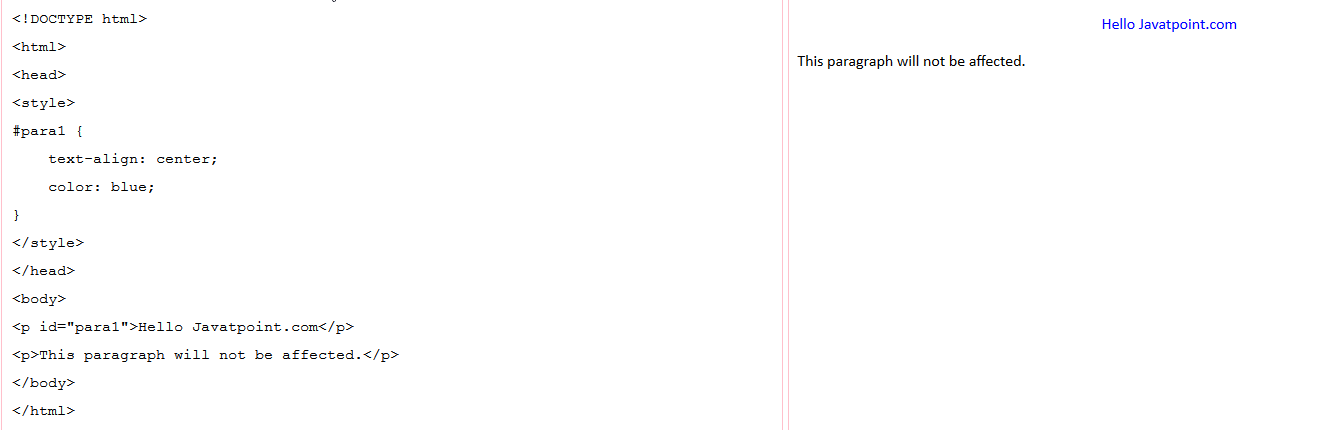
**2 CSS id selector**

id-valitsin valitsee HTML-elementin id attribuutin tietyn **elementin valitsemiseksi**. Id on aina ainutlaatuinen sivulla.

Se on kirjoitettu hash-merkillä (#), jota seuraa elementin id.

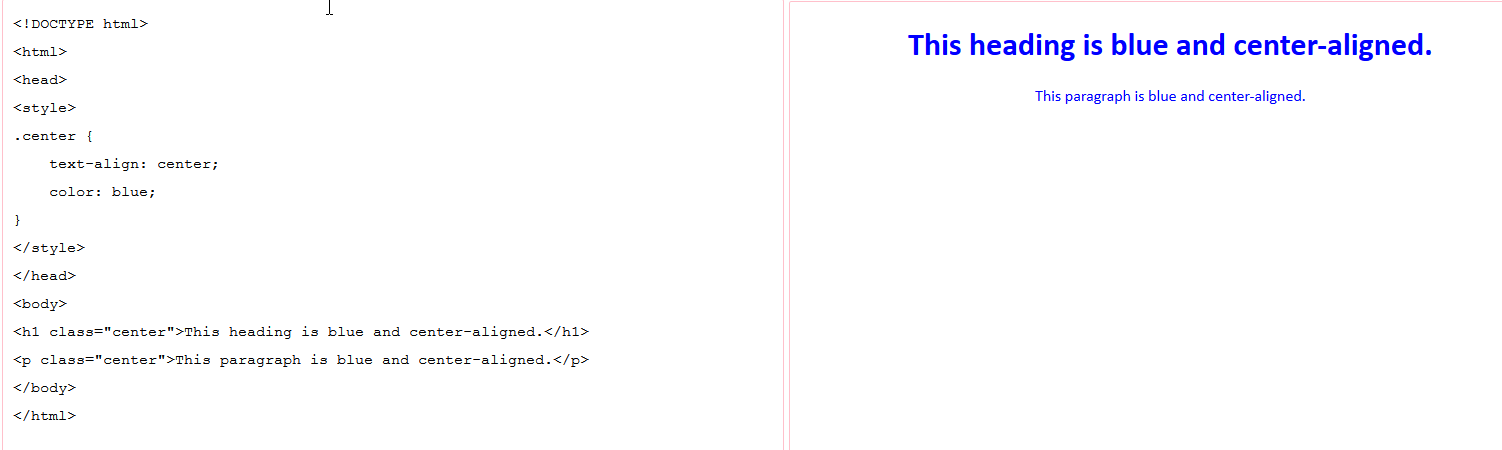
Otetaan esimerkki tunnuksella "para1".

<**p** id=”para1”>Hello…</**p**>



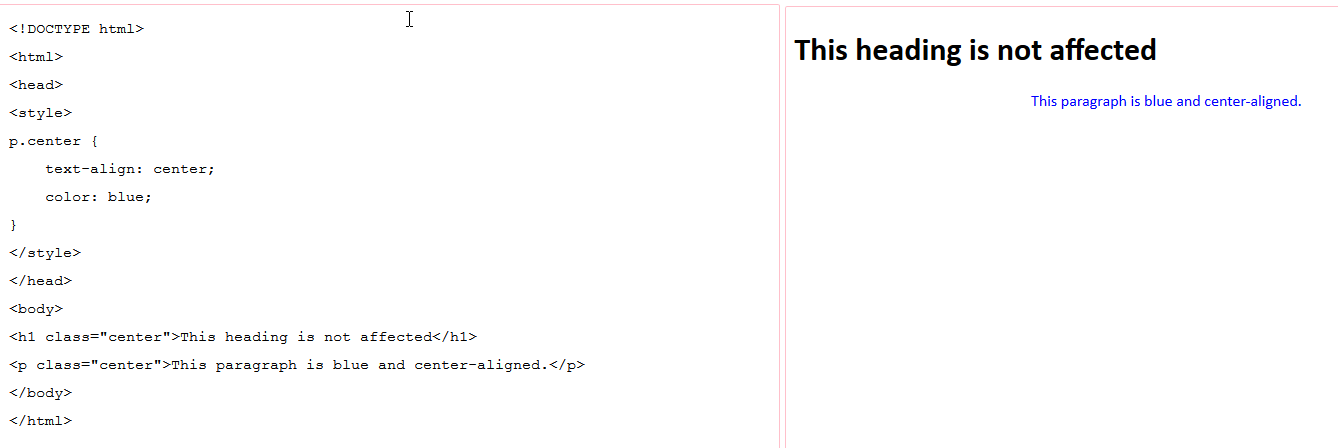
**3 CSS class selector (luokkaselektori)**

Luokan valitsin valitsee HTML-elementit, joilla on tietty luokkaattribuutti. Sitä käytetään ajanjaksolla. Merkitään . (pisteellä) jota seuraa luokan nimi. Esimerkissä .center {}



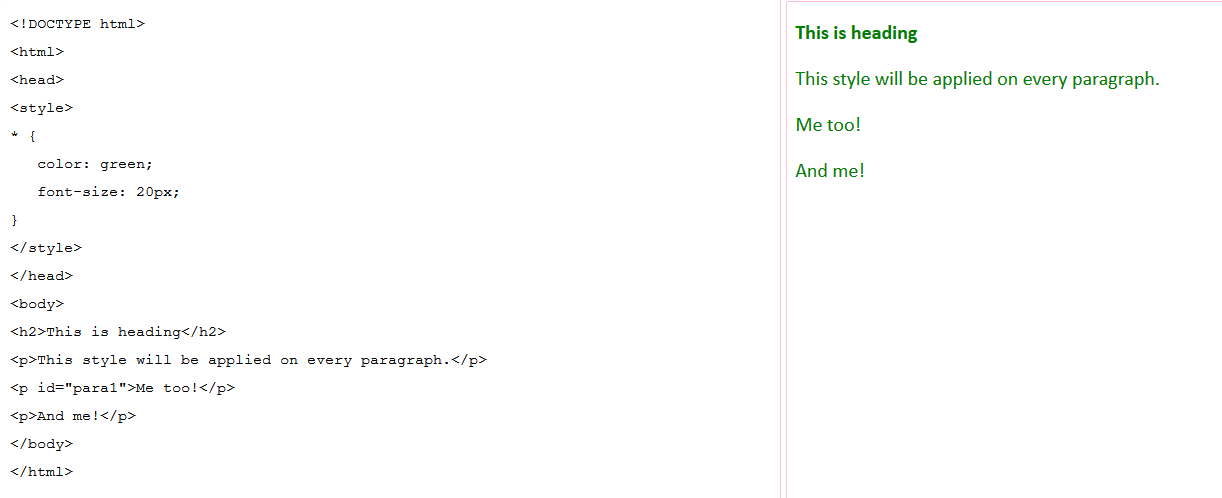
**Css luokka Selector tietylle elementille**

Jos haluat määrittää, että vain yhteen tiettyyn elementtiin vaikuttaa selektori, käytä elementin nimeä luokan valitsimella. eli p.center {}



**4 CSS universal Selector**

Yleisvalitsinta käytetään jokerimerkkiä. Se valitsee kaikki sivut samanlaisiksi.



Ryhmävalitsin **Css group selector.**

Voit valita ryhmässä samanlaiset tyylit. Alla h1, h2 ja p laitettu keskelle ja väriksi valittu sininen.



**Kuinka lisätä CSS**

CSS lisätään HTML-sivuille asiakirjan muotoilemiseksi tyyliarkin tietojen mukaisesti. HTML-asiakirjoissa on kolme tapaa lisätä CSS: n.

    Inline (linjassa) CSS

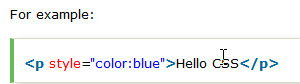
    Sisäinen CSS

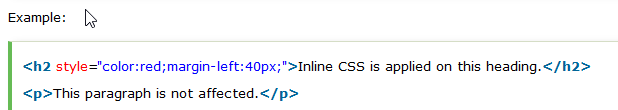
    Ulkoinen CSS

**Inline Css**

Inline CSS: tä käytetään CSS: n käyttämiseen yhdellä rivillä tai elementillä. Css koodi on html koodin sisällä. Älä käytä tätä tapaa!

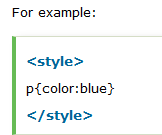
(ei hyvä tapa)

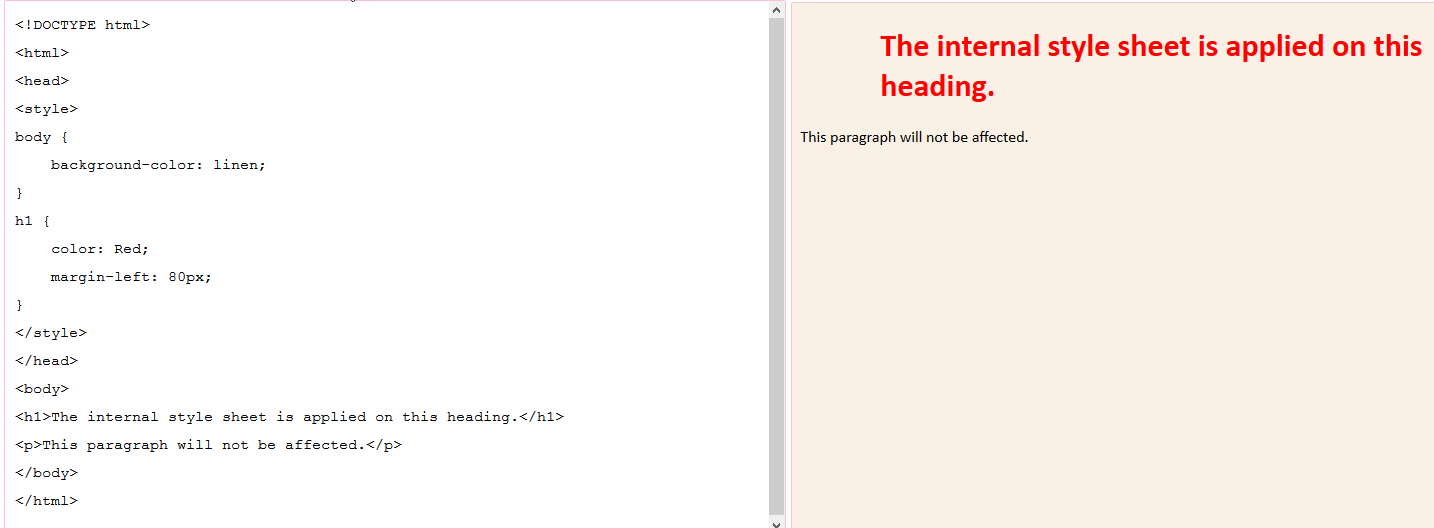




**Sisäinen CSS**

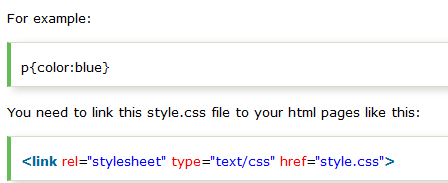
Sisäistä CSS: tä käytetään CSS: n käyttämiseen yhdellä asiakirjalla tai sivulla. Se voi vaikuttaa kaikkiin sivun osiin. Se on kirjoitettu tyylitunnisteen sisällä <style></style>html-sivulla. Tämäkin tapa on huono.





**Ulkoinen CSS**

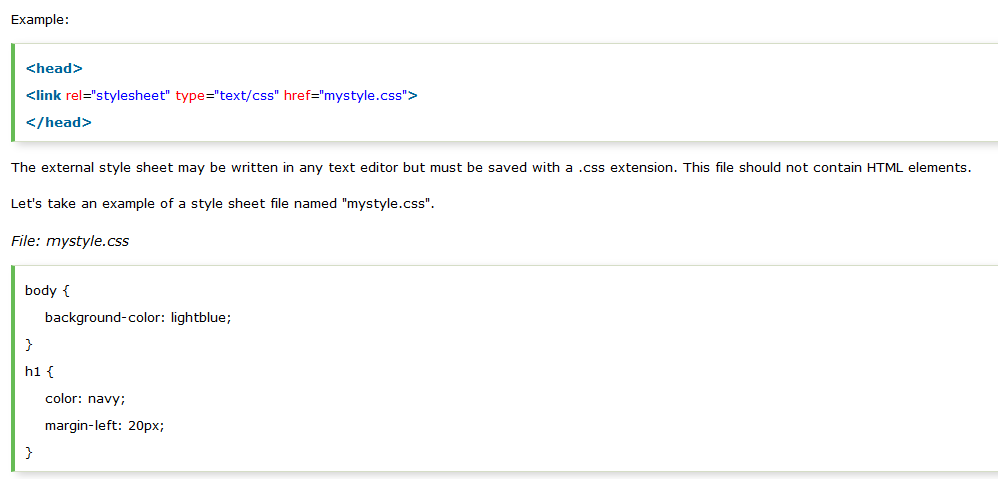
Ulkopuolista CSS: ää käytetään CSS: n soveltamiseen useilla sivuilla tai kaikilla sivuilla. Täällä kirjoitamme kaikki CSS-koodin css-tiedostoon. Sen laajennuksen on oltava .css esimerkiksi style.css. KÄYTÄ TÄTÄ TAPAA!



Link rel stylesheet on oltava html-sivulla.

Käytä stylesheetiä kun haluat tehdä muutoksia useilla sivuilla. Se sopii mainiosti tähän, koska se muuttaa koko sivuston ulkoasua muuttamalla vain yhtä tiedostoa.

Se käyttää <link> -merkintää kaikilla sivuilla ja <link> -tunniste on sijoitettava otsikon osiin.



Älä käytä välilyöntiä. Esimerkiksi: Sen pitäisi olla marginaali-vasen: 20px ei marginaali-vasen: 20 px.

**Kommentointi**

CSS-kommentit kirjoitetaan yleensä koodin selittämiseksi. On erittäin hyödyllistä käyttäjille, jotka lukevat koodisi, jotta he voivat helposti ymmärtää koodin.  
  
Selaimet eivät huomioi kommentteja.  
  
Kommentit ovat yhden tai usean rivin kommentteja ja kirjoitettu /\* \*/

    /\* This is a single-line comment \*/



**CSS Background**

CSS-tausta-ominaisuutta käytetään elementtien taustavaikutusten määrittämiseen. HTML-elementteihin vaikuttaa 5 CSS-taustaominaisuutta:

taustaväri

taustakuva

taustan toisto

taustan paikka, eli sijainti

kiinteä taustakuva

background-color -taustaväri

background-image -taustakuva

background-repeat -taustakuvan toisto

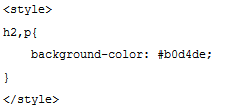
background-position -taustakuvan sijainti

background-attachment -kiinteä taustakuva

**CSS background-color**

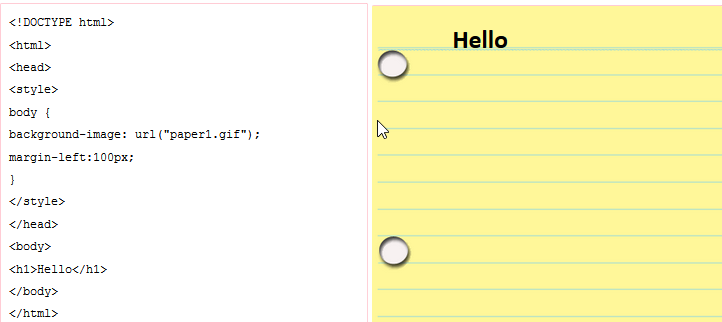
Taustaväri-ominaisuutta käytetään elementin taustan värin määrittämiseen.

Voit asettaa taustavärin näin:



**Css background-image**

Taustakuva-ominaisuutta käytetään asettamaan kuva elementin taustana. Oletusarvoisesti kuva kattaa koko elementin. Voit asettaa taustakuvan näin.

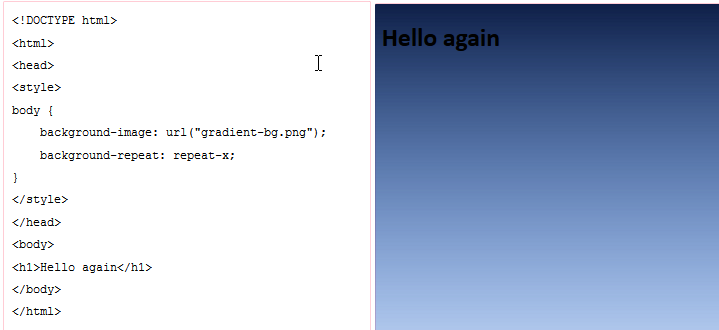


Huomaa: Taustakuva on valittava tekstin värin mukaan. Teksti- ja taustakuvan huono yhdistelmä saattaa olla syynä huonosti suunniteltuun ja luettavaan verkkosivuun.

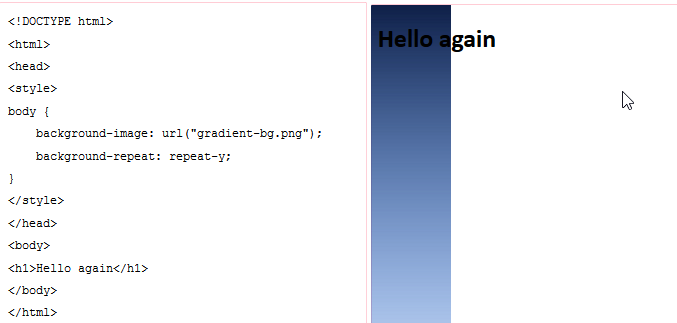
**CSS background-repeat**

Oletuksena on, että taustakuva ominaisuus toistaa tausta kuvan vaaka- ja pystysuunnassa. Jotkin kuvat toistuvat vain vaakasuunnassa tai pystysuorassa. Tausta näyttää paremmalta, jos kuva toistetaan vain vaakasuunnassa.

**background-repeat: repeat-x;**

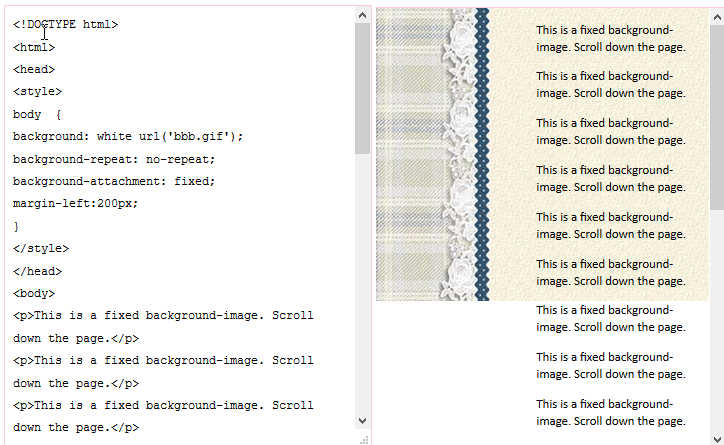


**background-repeat: repeat-y;**



**Background-attachment**

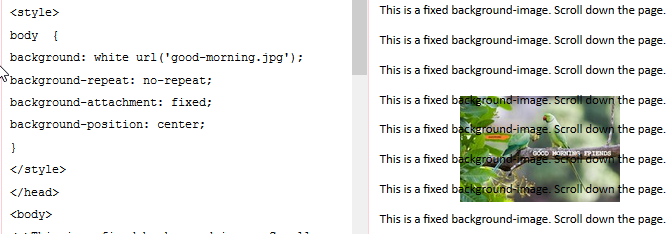
Jos asetat kiinteän taustakuvan, kuva ei liiku selaimen vierityksen aikana. Otetaan esimerkki kiinteällä taustakuvalla.



jatkuu…

**CSS Background-position**

Taustan sijainti -ominaisuutta käytetään taustakuvan alkuperäisen sijainnin määrittämiseen. Oletusarvoisesti taustakuva asetetaan verkkosivun vasempaan yläkulmaan.  
  
Voit asettaa paikat seuraavasti:  
  
     keskelle  
     ylös  
     alas  
     vasemmalle  
     oikealle



**CSS Border**

CSS reunuksilla määritetään reunuksen tyyli, väri ja koko. Ominaisuudet:

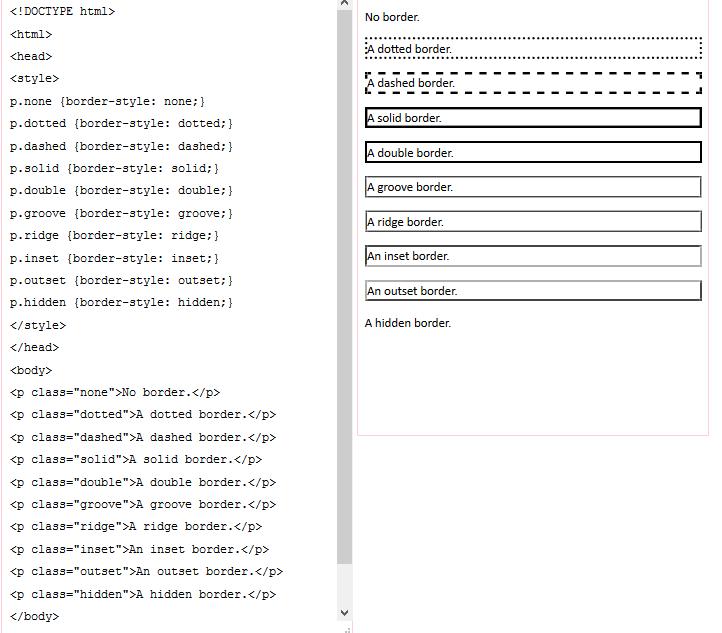
border-style, eli reunuksen tyyli

border-color, eli reunuksen väri

border-width eli reunuksen leveys

border-radius eli reunuksen säde

|  |  |
| --- | --- |
| **arvo** | **kuvaus** |
| none | ei reunuksia |
| dotted | pilkkureunukset |
| dashed | katkoviivat |
| solid | yhtäjaksoinen viiva |
| double | kaksi reunaviivaa |
| groove | kolmiuloitteinen uritettu reunus |
| ridge | kolmiulotteinen reunus |
| inset | kolmiuloiteinen reunus, värit |
| outset | 3d aloitusrajan määritys |
|  |  |

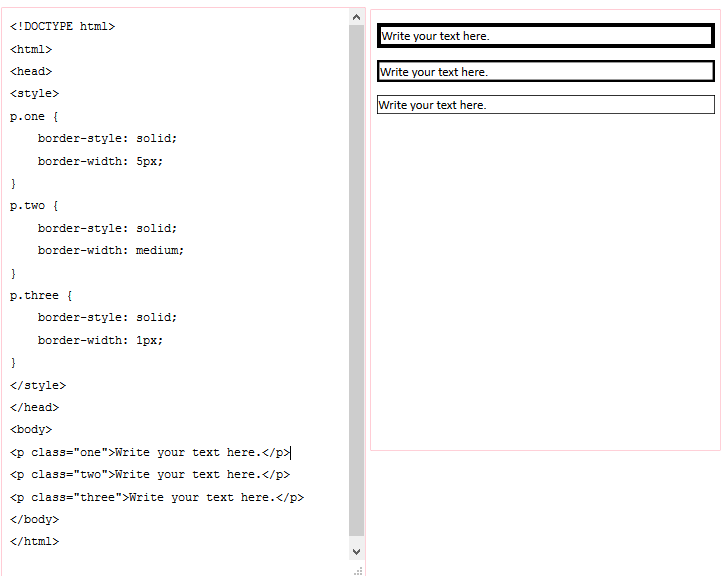


</html>

**CSS border-width**

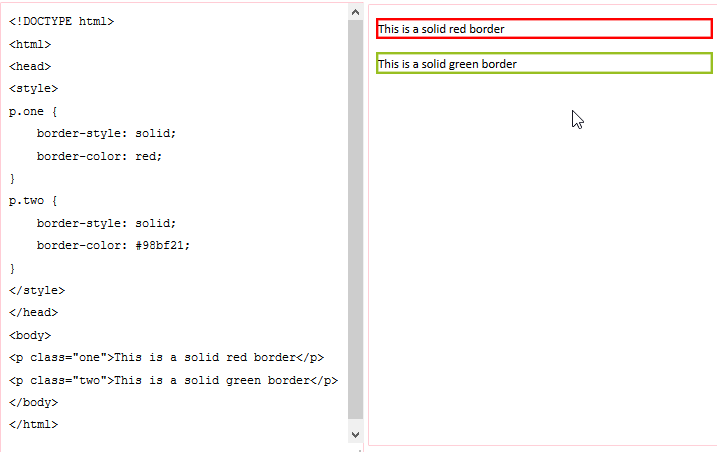
**Rajan leveysominaisuutta** käytetään rajan leveyden määrittämiseen. Se asetetaan pikseleinä. Voit myös käyttää yhtä kolmesta ennalta määritetystä, ohut, keskikokoisesta tai paksuisesta arvosta rajan leveyden määrittämiseen.

Huomaa: Rajan leveysominaisuutta ei käytetä yksinään. Sitä käytetään aina muiden raja ominaisuuksien kuten "border-style" -ominaisuuden kanssa rajan asettamiseksi ensin, muussa tapauksessa se ei toimi.



**CSS border-color**

**Rajan värin** asettamiseen on kolme tapaa.  
  
     Nimi: Määrittää värin nimen. Esimerkiksi: "red".  
     RGB: Määrittää värin RGB-arvon. Esimerkiksi: "rgb (255,0,0)".  
     Hex: Määrittää värin heksadesimaalin. Esimerkiksi: "# ff0000".  
  
Myös reunaviiva on "läpinäkyvä". Jos reunusväriä ei ole asetettu, se periytyy elementin väritiedosta.  
Huomaa: Raja-väri-ominaisuutta ei käytetä yksinään. Sitä käytetään aina muiden raja ominaisuuksien kuten "border-style" -ominaisuuden kanssa rajan asettamiseksi ensin, muussa tapauksessa se ei toimi.



**CSS display**

**CSS-näyttö** on CSS: n tärkein ominaisuus, jota käytetään elementin ulkoasun ohjaamiseen. Se määrittää, kuinka elementti näytetään.  
  
Jokaisella elementillä on oletusarvoinen näyttöarvo sen luonteen mukaan. Jokainen verkkosivujen elementti on suorakaiteen muotoinen laatikko, ja CSS-ominaisuus määrittää kyseisen suorakaiteen laatikon käyttäytymisen.

**CSS displayn oletusominaisuudet**

|  |  |
| --- | --- |
| default value | inline |
| inherited | no |
| animation supporting | no |
| version | css1 |
| javascript syntax | object.style.display="none" |
|  |  |

Syntax

display:value;

**CSS display arvot**

Seuraavat display arvot:

1. display: inline;
2. display: inline-block;
3. display: block;
4. display: run-in;
5. display: none;

**CSS display inline**

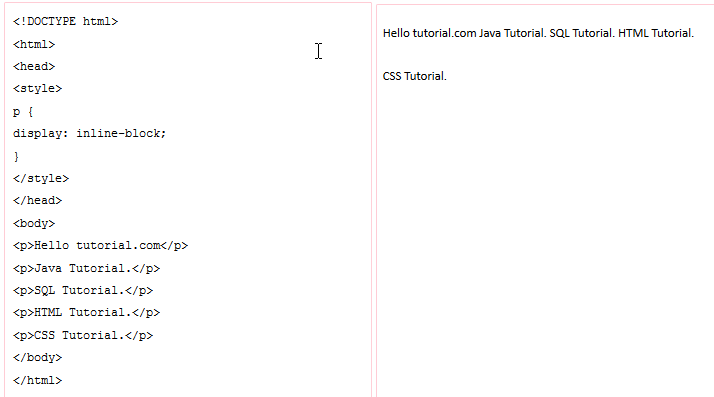
Vie vain vaaditun leveyden. Ei pakotettua rivinvaihtoa.

inline-elementit ova <span>, <a>, <em>, <b> jne…



**CSS display inline-block**

CSS-näyttö inline-block -elementti on hyvin samanlainen kuin inline-elementti, mutta ero on se, että pystyt asettamaan leveyden ja korkeuden.



**CSS display block**

CSS-näyttölohkoelementti vie niin paljon kuin horisontaalista tilaa, kun se voi. Tarkoittaa, että lohkoelementti ottaa täyden käytettävissä olevan leveyden. Riviväli lisätään ennen ja jälkeen.

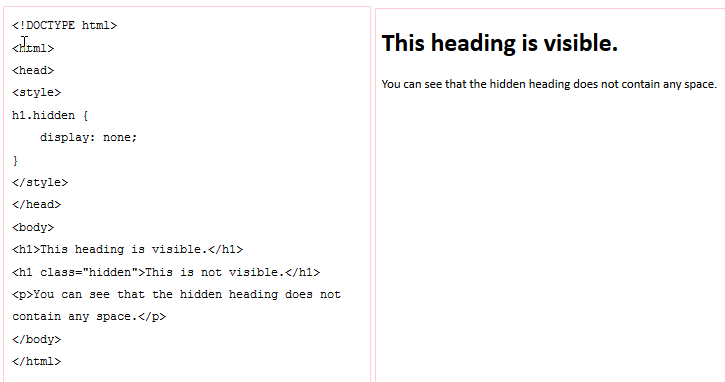


**CSS display run-in**

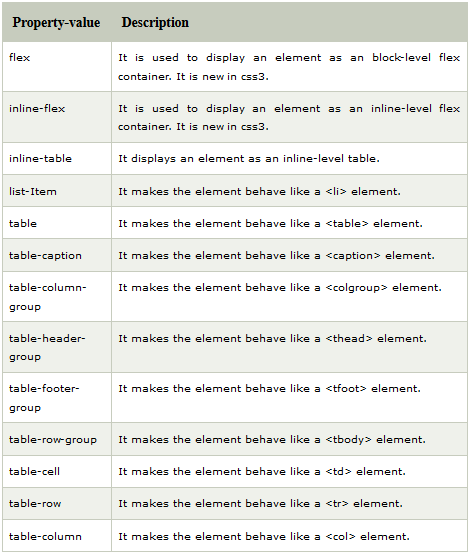


**CSS display none**

"Ei mitään" -arvo poistaa kokonaan osan sivulta. Se ei vie tilaa.

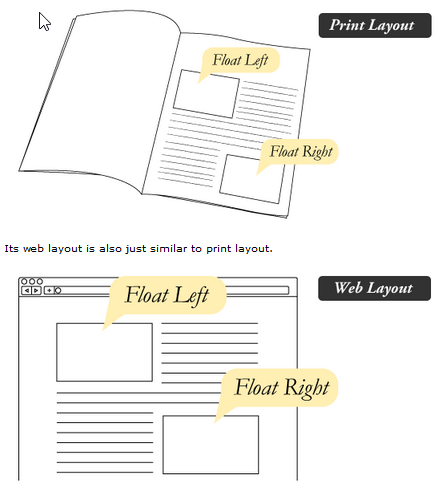


**Muita CSS display arvoja**



**CSS Float**

CSS float -ominaisuus on sijaintiominaisuus. Sitä käytetään työntämään elementti vasemmalle tai oikealle, jolloin toinen elementti voidaan kääriä sen ympärille. Sitä käytetään yleensä kuvien ja ulkoasujen kanssa.  
  
Ymmärrämme sen tarkoituksen ja alkuperän, katsokaamme sen tulostusnäyttöön. Tulostusnäytössä kuva asetetaan sivulle niin, että teksti ympäröi sitä tarpeen mukaan.



Miten se toimii?

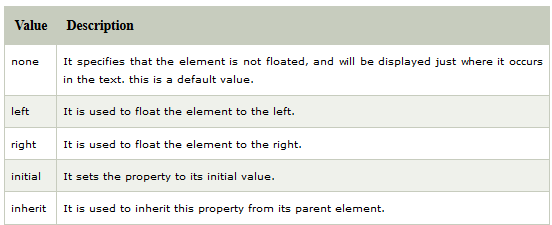
Elementit kelluvat vain vaakasuunnassa. Joten on mahdollista vain kellua elementtejä vasemmalle tai oikealle, ei ylös tai alas.

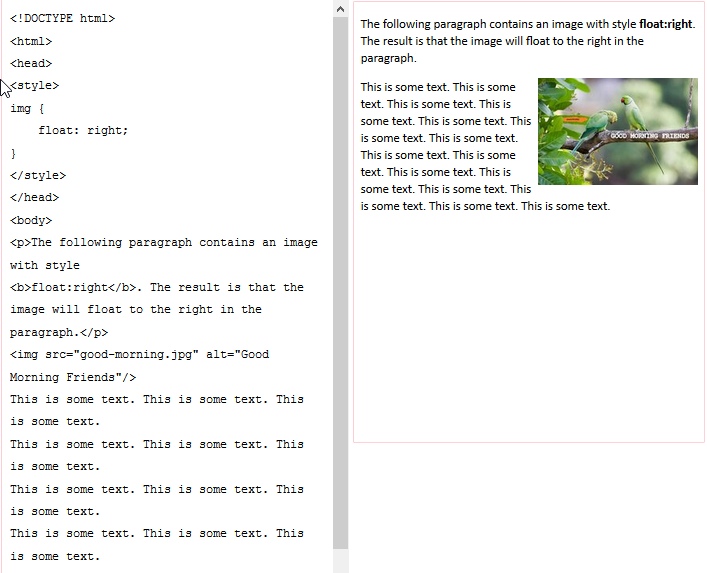
Floated elementtiä voidaan siirtää mahdollisimman kauas vasemmalle tai oikealle. Yksinkertaisesti se tarkoittaa, että kelluva elementti voi näkyä äärimmäisen vasemmalla tai äärioikeistolla.  
     Elementit virtauksen jälkeen virtaavat sen ympärille.  
     Elementit ennen kelluvan elementin ei vaikuta.  
     Jos kuva kelluu oikealle, tekstit kulkevat sen ympärillä vasemmalle ja jos kuva kelluu vasemmalle, teksti kulkee sen ympärillä oikealle.

**CSS Float ominaisuudet**

|  |  |  |
| --- | --- | --- |
| **Property** | **Description** | **Values** |
| clear | tyhjennys | left, right, both, none, inherit |
| float | määrittää kelluuko laatikko vai eikö | left, right, none, inherit (perintä) |

CSS Float





**CSS Font (fontit)**

<https://www.javatpoint.com/css-font>

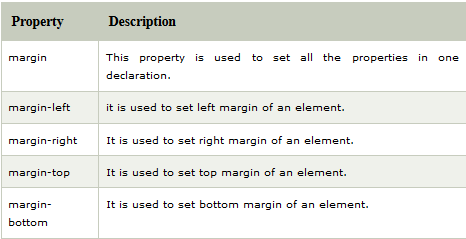
Css Line height (rivin korkeus)

<https://www.javatpoint.com/css-line-height>

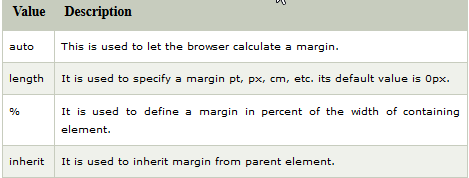
**CSS Margin**

CSS Marginaali -ominaisuutta käytetään määrittämään **elementtien ympärillä oleva tila**. Se on täysin läpinäkyvä eikä sillä ole taustaväriä. Se poistaa elementin ympärillä olevan alueen.  
  
Ylä-, ala-, vasen- ja oikea marginaali voidaan muuttaa itsenäisesti käyttämällä erillisiä ominaisuuksia. Voit myös muuttaa kaikkia ominaisuuksia kerralla käyttämällä lyhyen marginaalin ominaisuutta.  
  
Seuraavia CSS-marginaalivaatimuksia on olemassa:

CSS Margin Properties

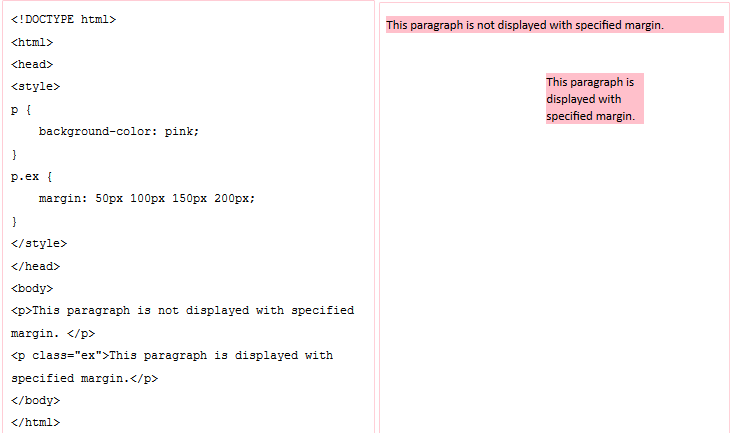


CSS Margin Values



myös negatiiviset arvot on käytössä.

**Marginaali: Kiinteä ominaisuus**  
  
CSS-pikatoimintoa käytetään lyhentämään koodia. Se määrittää kaikki marginaaliominaisuudet yhdessä ominaisuudessa.  
  
Marginaali-ominaisuuden määrittämiseen on neljä tyyppiä. Voit käyttää yhtä niistä.  
  
     marginaali: 50px 100px 150px 200px;  
     marginaali: 50px 100px 150px;  
     marginaali: 50px 100px;  
     marginaali 50px;  
  
1) marginaali: 50px 100px 150px 200px;  
  
Se osoittaa, että:  
  
ylimmän marginaalin arvo on 50px  
  
oikean marginaalin arvo on 100px  
  
alimman marginaalin arvo on 150px  
  
vasemman marginaalin arvo on 200px



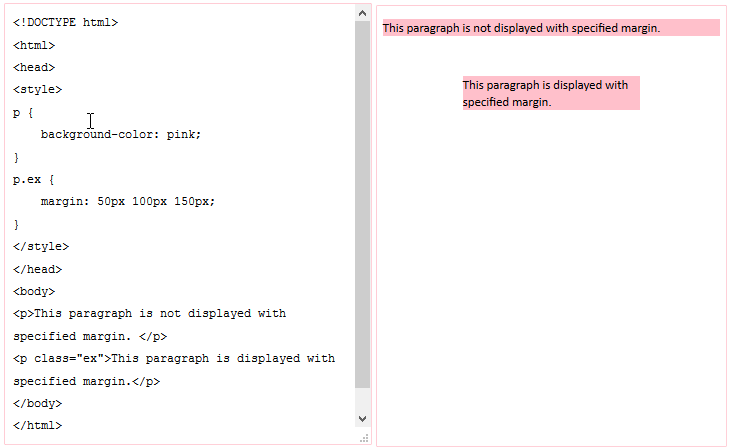
margin: 50px 100px 150px;

**Tarkoittaa että:**

**top** margin value is 50px

**left and right** margin values are 100px

**bottom** margin value is 150px

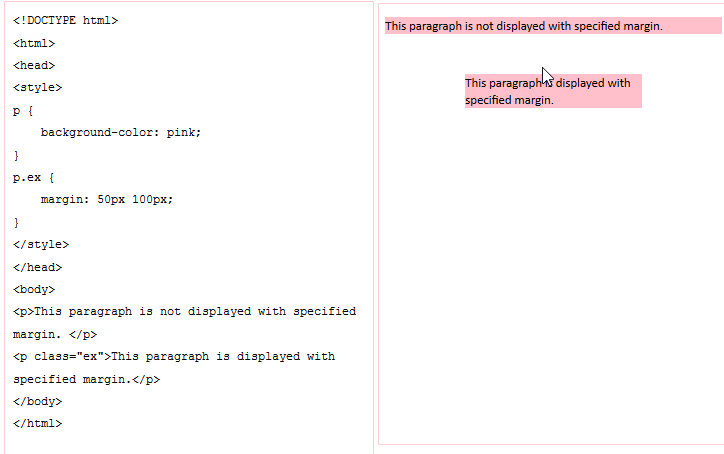


margin: 50px 100px;

Tarkoittaa että:

**top and bottom** margin values are 50px

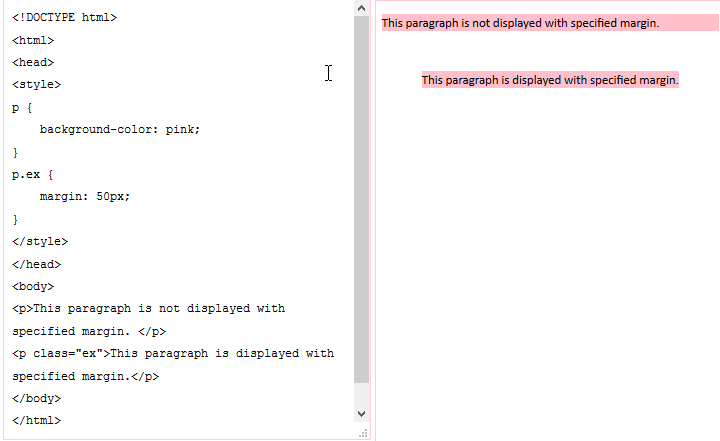
**left and right** margin values are 100px



margin: 50px;

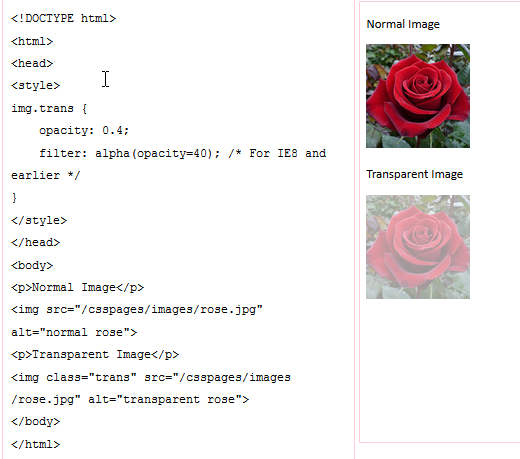
Tarkoittaa että

top right bottom and left margin values are 50px



**Opacity eli läpinäkyvyys**

CSS-peittävyysominaisuutta käytetään elementin läpinäkyvyyden määrittämiseen. Yksinkertaisella sanalla voit sanoa, että se määrittää kuvan selkeyden.  
  
Teknisesti, Opacity määritellään asteeksi, jolla valo saa kulkea kohteen läpi.



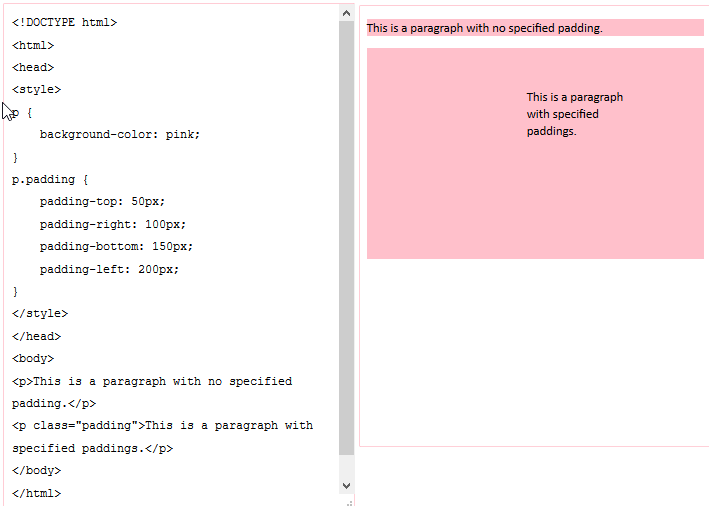
**CSS ylivuoto lohkoihin**

<https://www.javatpoint.com/css-overflow>

**Css padding**

CSS Padding -ominaisuutta käytetään elementti elementin ja elementtirivin välisen tilan määrittämiseen.  
  
Se eroaa CSS-marginaalista tavalla, jolla CSS-marginaali määrittää elementtien ympärillä olevan tilan. Taustaväreihin vaikuttaa CSS-tyyny. Se tyhjentää sisällön ympärillä olevan alueen.  
  
Ylä-, ala-, vasen- ja oikea pehmuste voidaan vaihtaa itsenäisesti käyttämällä erillisiä ominaisuuksia. Voit myös muuttaa kaikkia ominaisuuksia kerralla käyttämällä stenografista täyttöominaisuutta.





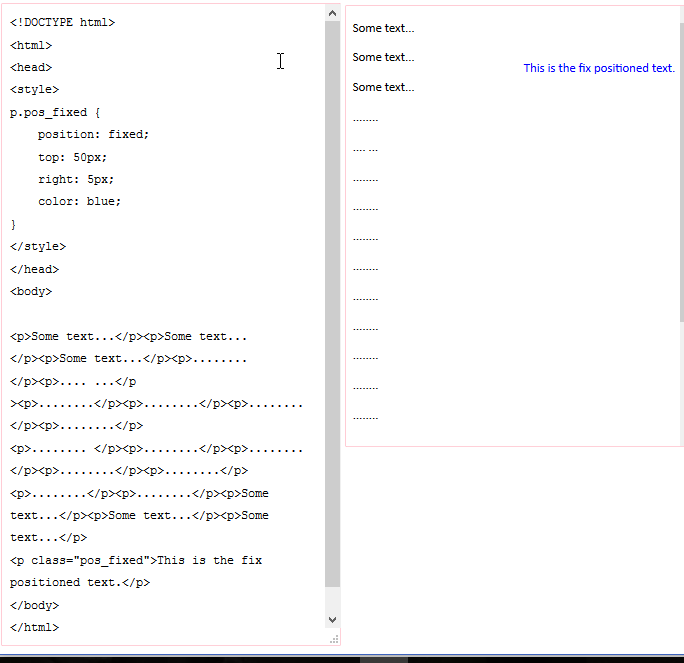
**CSS position eli sijainti**

CSS-sijainti-ominaisuutta käytetään elementin sijainnin määrittämiseen. sitä käytetään myös sijoittamaan elementti toisen taakse ja hyödyllinen myös käsikirjoitetun animaation vaikutukselle.  
  
Voit sijoittaa elementin käyttämällä ylhäältä, alhaalta, vasemmalta ja oikealta. Näitä ominaisuuksia voidaan käyttää vain sen jälkeen, kun asetusominaisuus on ensin asetettu. Asema-elementin laskettu sijainti ominaisuus on suhteellinen, absoluuttinen, kiinteä tai tahmea.  
  
Katsotaanpa seuraavaa CSS-paikannusta:  
  
     CSS staattinen sijainti  
     CSS Kiinteä sijainti  
     CSS-suhteellinen sijainti  
     CSS Absolute Positioning absoluuttinen sijainti

Css staattinen sijainti

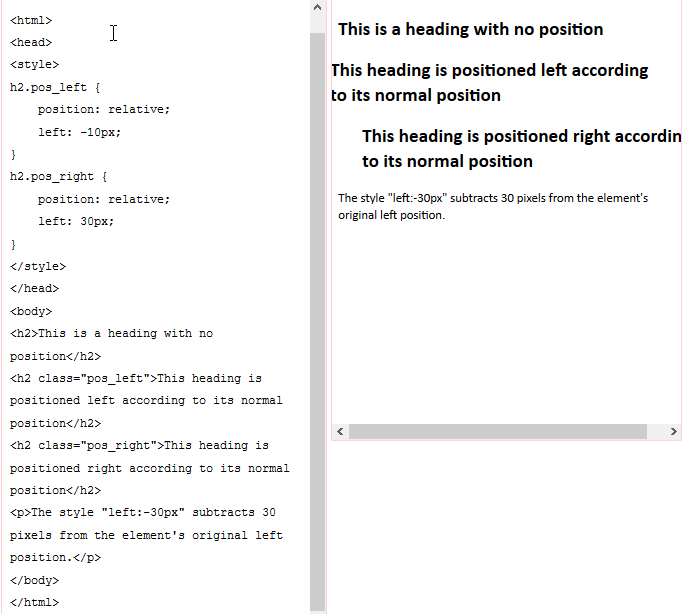
Tämä on oletusasetus HTML-elementeille. Se sijoittaa aina elementin normaalin sivuvirran mukaan. Top, bottom, left and right ominaisuudet eivät vaikuta siihen.

2) CSS Kiinteä sijainti  
  
Kiinteä sijainti auttaa selaimen kiinnittämiseen. Tämä kiinteä testi on sijoitettu suhteessa selainikkunaan eikä siirry edes selaamalla ikkunaa.



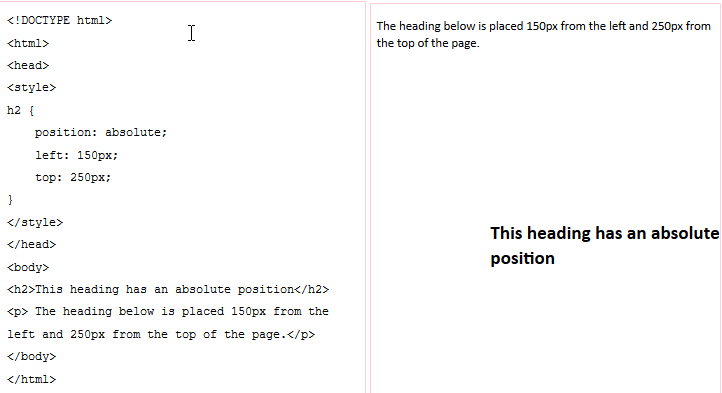
**CSS relative positioning**

Suhteellisen sijaintiominaisuuden avulla elementti asetetaan suhteessa normaaliin asentoonsa.



**CSS Absolute Positioning**

Absoluuttista sijaintia käytetään elementin sijoittamiseen suhteessa ensimmäiseen peruselementtiin, jolla on muu asema kuin staattinen. Jos tällaista elementtiä ei löydy, sisällyttävä lohko on HTML.  
  
Absoluuttisen sijainnin avulla voit sijoittaa elementin mihin tahansa sivulle.

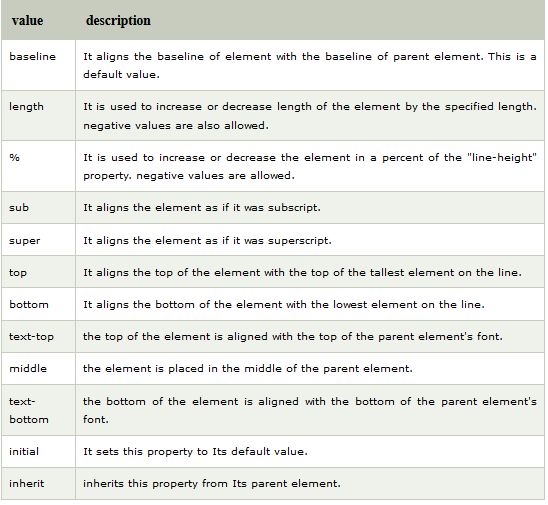


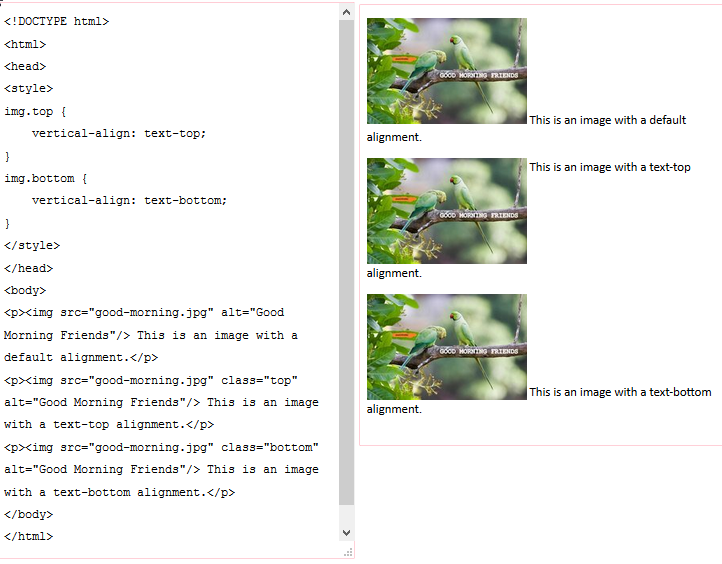


**CSS Vertical Align**

CSS-pystysuuntaisen linja ominaisuuden avulla määritetään sisäisen tai taulukko-solulaatikon pystysuuntainen linjaus. Se on CSS: n itsestään selvä ominaisuus. Se ei ole kovin helppo ominaisuus aloittelijoille.

Mitä se tekee  
  
     Sitä sovelletaan sisäänrakennettuihin tai inline-block-elementteihin.  
     Se vaikuttaa elementin kohdistamiseen eikä sen sisältöön. (paitsi taulukon solut)  
     Kun se on sovellettu pöydän soluihin, se vaikuttaa solun sisältöön, ei itse soluun.



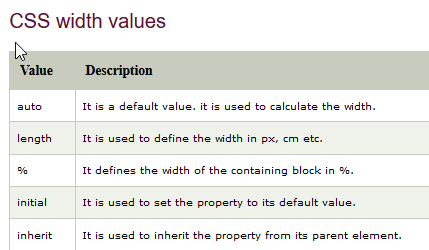


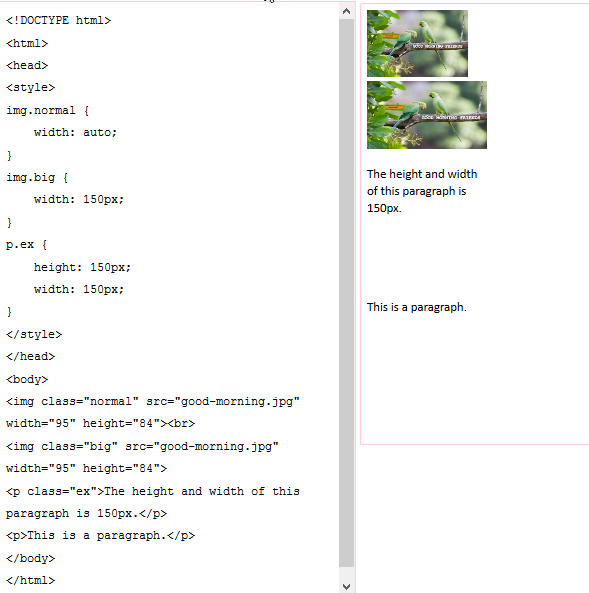
**CSS White Space**

<https://www.javatpoint.com/css-white-space>

**CSS Width**

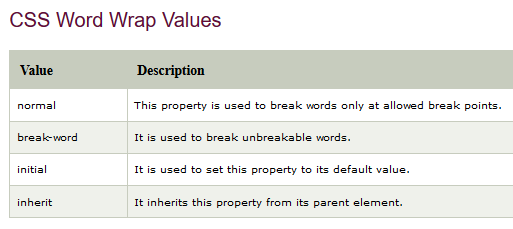
CSS-leveysominaisuutta käytetään määrittämään elementin sisältöalueen leveys.  
  
Se ei sisällä padding-reunoja tai marginaaleja. Se asettaa alueen leveyden elementin pehmusteiden, reunojen ja marginaalin sisällä.

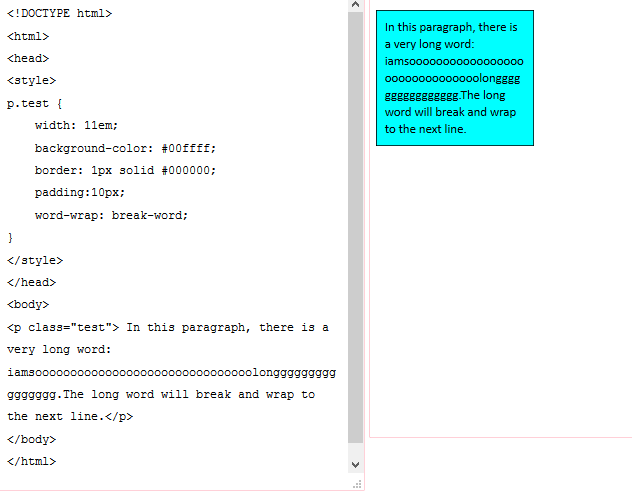




**CSS Word Wrap**

Tätä ominaisuutta käytetään estämään ylivuoto, kun rikkoutumaton merkkijono on liian pitkä sopivaksi sisällyttävään laatikkoon.

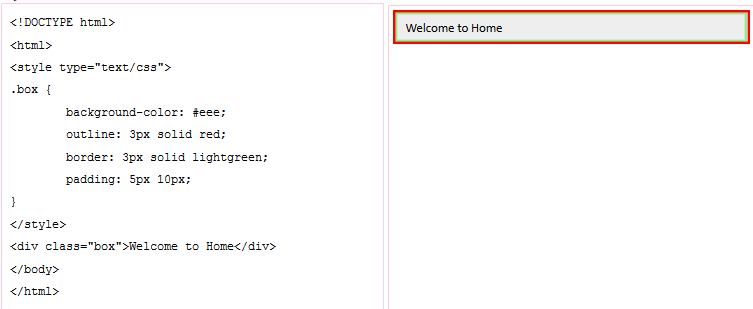


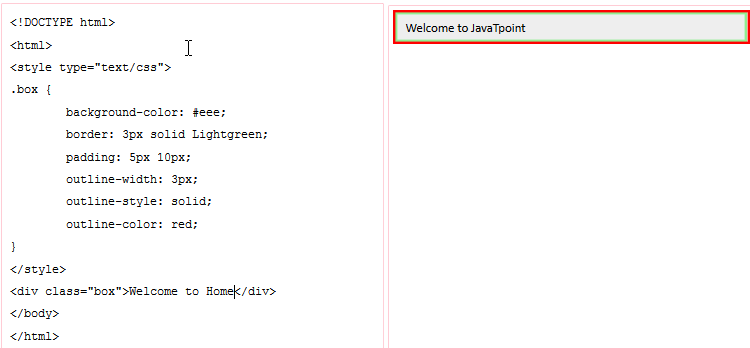


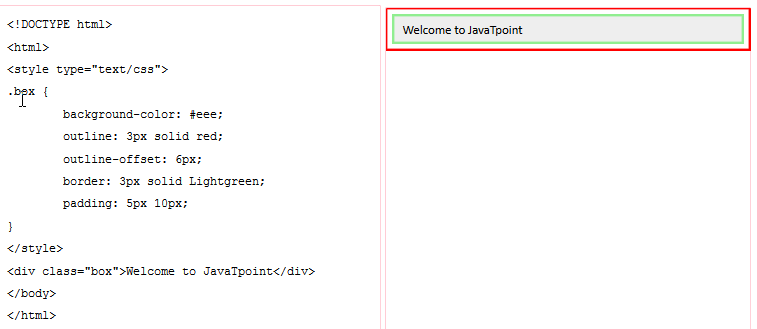
**Css Outline**

<https://www.javatpoint.com/css-outline>

CSS-ääriviiva on kuin CSS-border rajapinta. Sen avulla voit piirtää ylimääräisen rajan elementin ympärille, jotta saat visuaalisen huomion.  
  
Sen käyttö yhtä helppoa kuin reunuksen.







**CSS Visibility**

<https://www.javatpoint.com/css-visibility>

CSS Counter (eräänlainen laskuri, voi käyttää sisällysluettelossa)

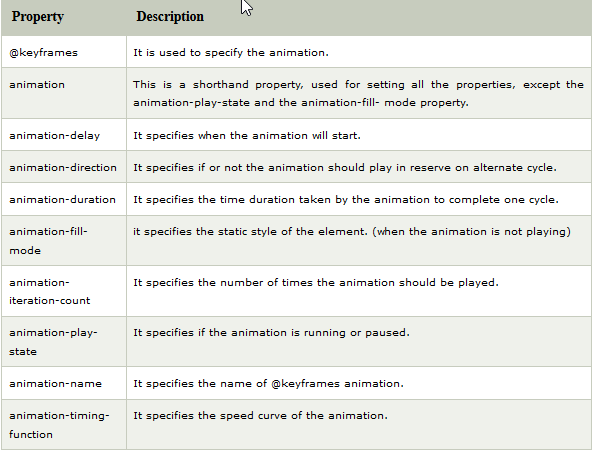
<https://www.javatpoint.com/css-counter>

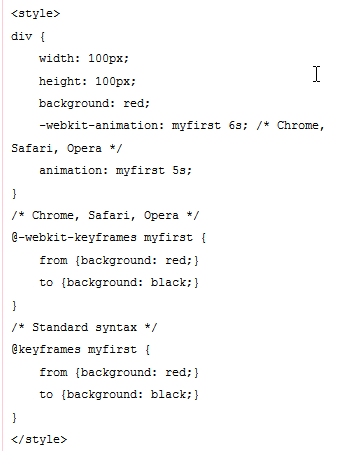
**CSS-animaatio**  
CSS Animation -ominaisuutta käytetään animaation luomiseen verkkosivulla. Sitä voidaan käyttää Flashin ja JavaScriptin luomaa animaatiota korvaavana. Flashia ei kannata käyttää enää!  
  
CSS3 @ keyframes sääntö  
Animaatio luodaan @keyframe -säännössä. Sitä käytetään ohjaamaan välivaiheita CSS-animaatiosekvenssissä.

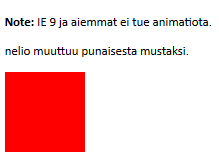
Mikä animaatio tekee  
Animaatio tekee elementistä asteittain muuttuvan tyylistä toiseen. Voit lisätä niin monta kuin lisäät kohteita. Voit myös määrittää muutokset 0% animaation alun ja 100% määritellä sen suorittamisen.

**Miten CSS-animaatio toimii**  
  
Kun animaatio luodaan @ keyframe -säännössä, sen on oltava sidottu valitsimella; muuten animaatiolla ei ole vaikutusta.  
  
Animaatio voidaan sitoa valitsimeen määrittämällä vähintään nämä kaksi ominaisuutta:  
  
     Animaation nimi  
     Animaation kesto

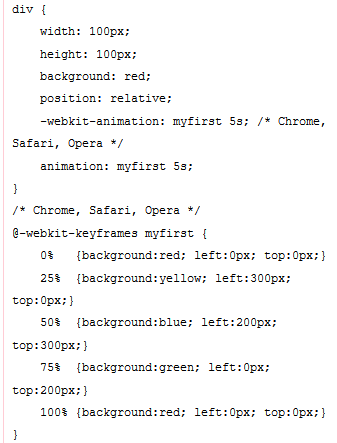
CSS animation properties

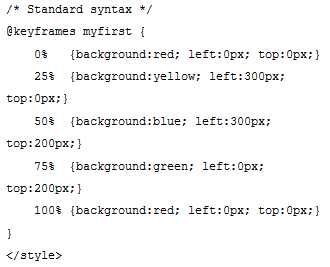


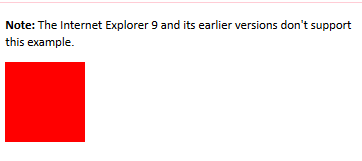




<style>







liikkuu vasemmasta nurkasta oikeaan nurkkaan jne…

punaisesta – oranssiksi

oranssista -siniseksi

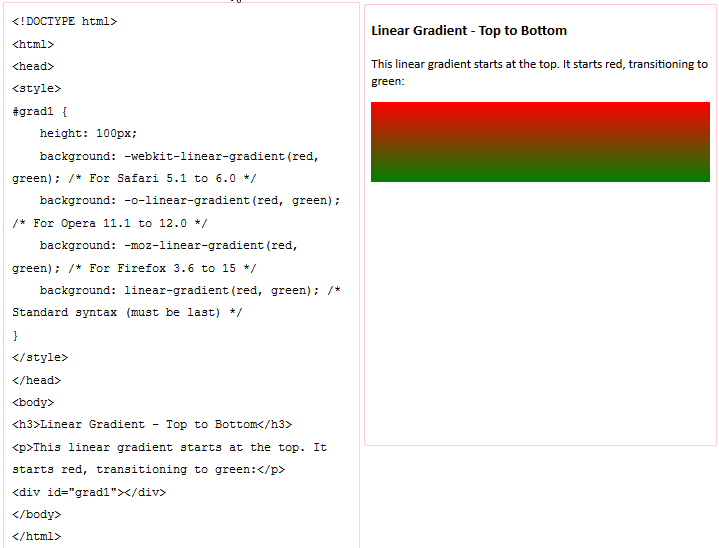
sinisestä – vihreäksi

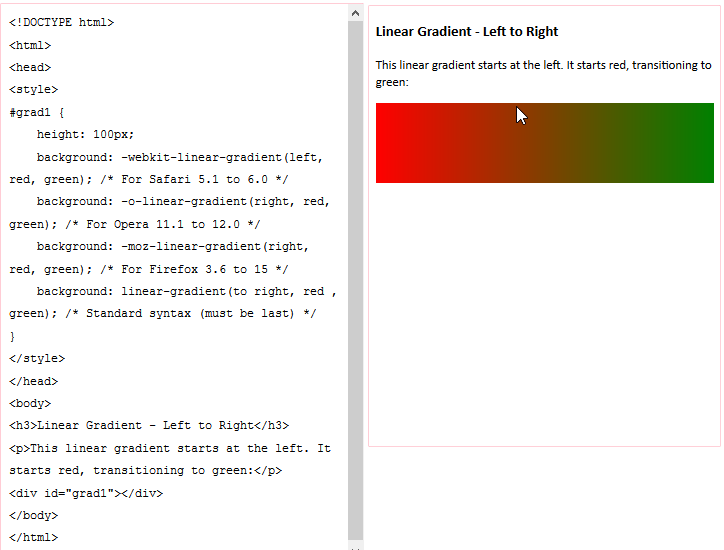
**CSS Gradient**

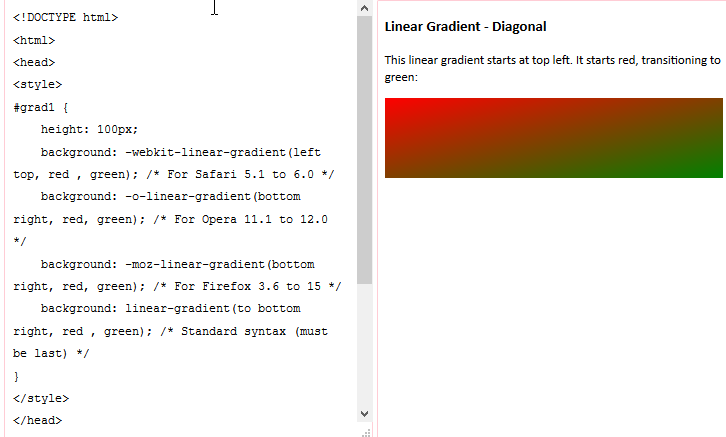
CSS-gradienttia käytetään näyttämään tasaista siirtymää kahden tai useamman määritetyn värin sisällä.  
  
Miksi CSS Gradient  
  
Nämä ovat seuraavat syyt käyttää CSS-gradienttia.  
  
     Sinun ei tarvitse käyttää kuvia siirtymisen vaikutusten näyttämiseen.  
     Latausaikaa ja kaistanleveyden käyttöä voidaan myös vähentää.  
     Se tarjoaa paremman kuvan elementtiin zoomaamalla, koska selaimen luoma gradientti.  
  
CSS3: ssa on kahdenlaisia kaltevuustyyppejä.  
  
     Lineaariset kaltevuudet  
     Radiaaliset kaltevuudet  
  
1) CSS Lineaarinen gradientti  
  
CSS3-lineaarinen gradientti menee ylös / alas / vasemmalle / oikealle ja diagonaalisesti. Jotta voit luoda CSS3-lineaarisen gradientin, sinun on määritettävä kaksi tai useampia värin pysähdyksiä. Värirajoitukset ovat värejä, joita käytetään luomaan sujuva siirtymä. Lähtöpiste ja suunta voidaan lisätä myös gradienttiefektin kanssa.

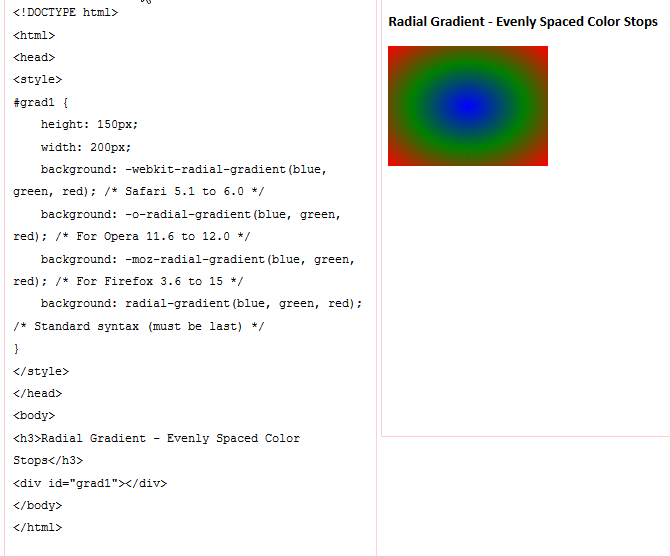


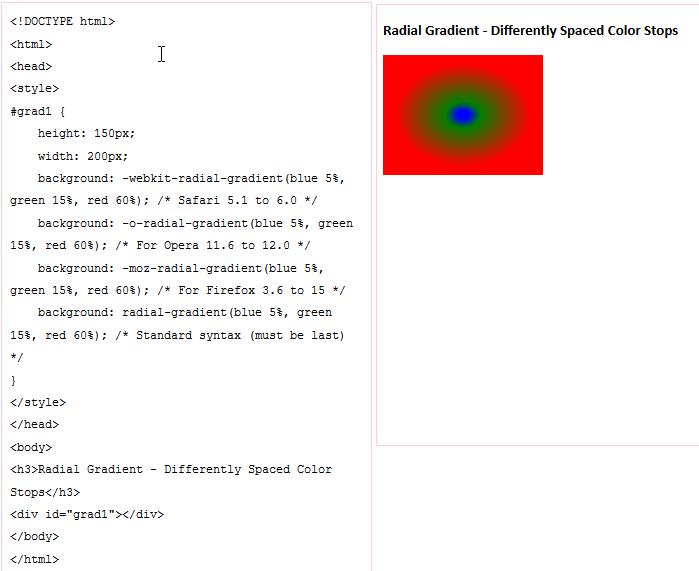
CSS Lineaarinen gradientti: (ylhäältä alas)  
  
Ylhäältä alas Lineaarinen gradientti on oletus lineaarinen gradientti. Otetaan esimerkki lineaarisesta gradientista, joka alkaa ylhäältä. Se alkaa punaiseksi ja siirtyy vihreiksi.



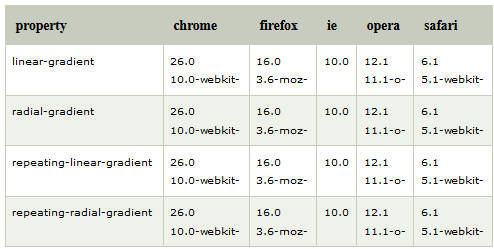








Tuetut selaimet



**CSS Transition**

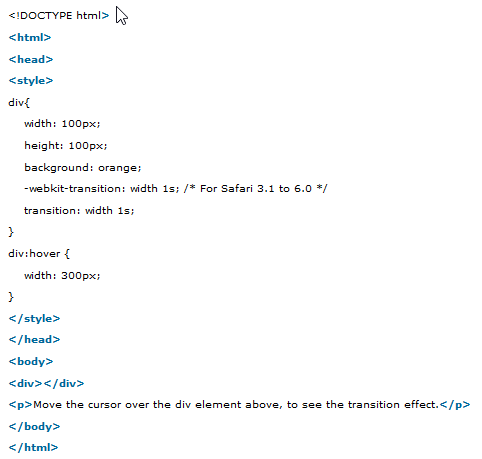
CSS-siirtymät ovat vaikutuksia, joita lisätään elementin vaihtamiseksi vähitellen tyyliltään, ilman Flashia tai JavaScriptia.

Sinun pitäisi määrittää kaksi asiaa CSS-siirtymän luomiseen.

* CSS-ominaisuus, johon haluat lisätä vaikutuksen.
* Vaikutuksen kestoaika.

Otetaan esimerkki, joka määrittää siirtymävaikutuksen leveysominaisuuksiin ja kestoon 3 sekuntia.

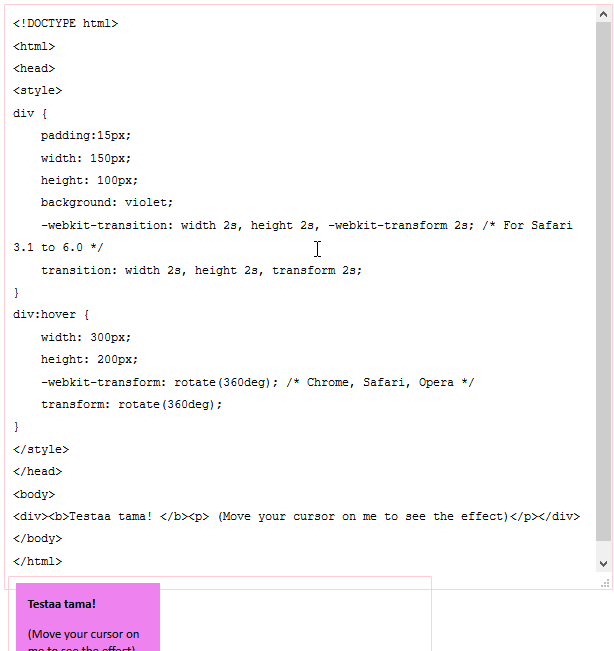
Huomautus: Jos et määritä kesto-osaa, siirtymällä ei ole vaikutusta, koska sen oletusarvo on 0. Siirtymätehoste alkaa, kun siirrät kohdistimen elementille.  
Huomautus: IE9 ja aiempi versio eivät tue siirtymisominaisuutta.



Huomaa: kun otat hiiren osoittimen elementistä, se saa alkuperäisen tyylinsä vähitellen.

## CSS Multiple Transition Effect (monisiirtymä)

Sitä käytetään lisäämään siirtymisvaikutusta useammalle kuin yhdelle CSS-ominaisuudelle. Jos haluat lisätä siirtymisvaikutuksen useampaan kuin yhteen ominaisuuteen, erota nämä ominaisuudet pilkulla.  
  
Otetaan esimerkki. Tässä siirtymävaikutukset leveyteen, korkeuteen ja muunnokseen. Suorakulmio kääntyy…

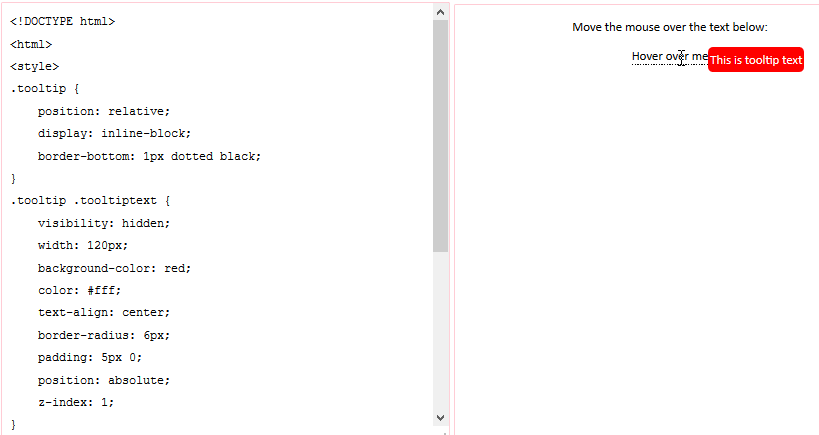


**Css Tooltips**

CSS-työkaluvihjeet ovat erinomainen tapa näyttää lisätietoja, kun käyttäjä siirtää hiiren osoittimen elementtiin.

Työkaluvihjeet voidaan sijoittaa ylös, alas, vasemmalle ja oikealle.

**Tooltip Animation**



<https://www.javatpoint.com/css-tooltips> testaa kaikki!

<https://www.javatpoint.com/oprweb/test.jsp?filename=css-tooltip-animation> häilyvä vihje

**CSS Arrow**

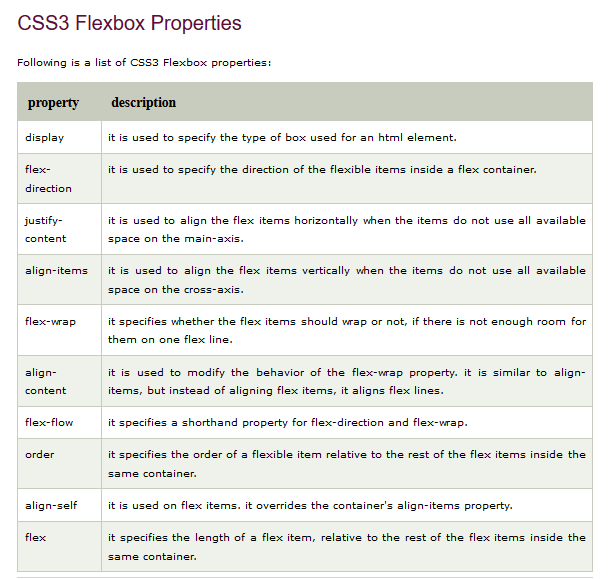
<https://www.javatpoint.com/oprweb/test.jsp?filename=css-arrow1> löydät nuoli vihjetyökalun.

**CSS Flexbox**

CSS3 Joustavat laatikot, jotka tunnetaan myös nimellä CSS Flexbox, ovat CSS3: n uusi ulkoasu.  
CSS3-flexboxia käytetään elementtien ennakoituvaan käyttäytymiseen, kun niitä käytetään erilaisilla näytön kokoilla ja erilaisilla näyttölaitteilla. Se tarjoaa tehokkaamman tavan jakaa, kohdistaa ja jakaa tilaa kontin sisältämien tuotteiden välillä.  
  
CSS3 pystyy muuttamaan kohteen leveyttä ja korkeutta parhaiten sopivaksi kaikille käytettävissä oleville tiloille. Se on edullista lohkomallin yli.  
  
CSS3-flexboxissa on joustavia säiliöitä ja joustavia esineitä.  
  
Flex-säiliö: Flex-säiliö määrittää emon ominaisuuksia. Se julistetaan asettamalla elementin näyttöominaisuus joko taipumaan tai inline-flexiin.  
  
Flex-kohteet: Flex-kohteet määrittelevät lasten ominaisuuksia. Flex-säiliössä voi olla yksi tai useampi taipuisa.

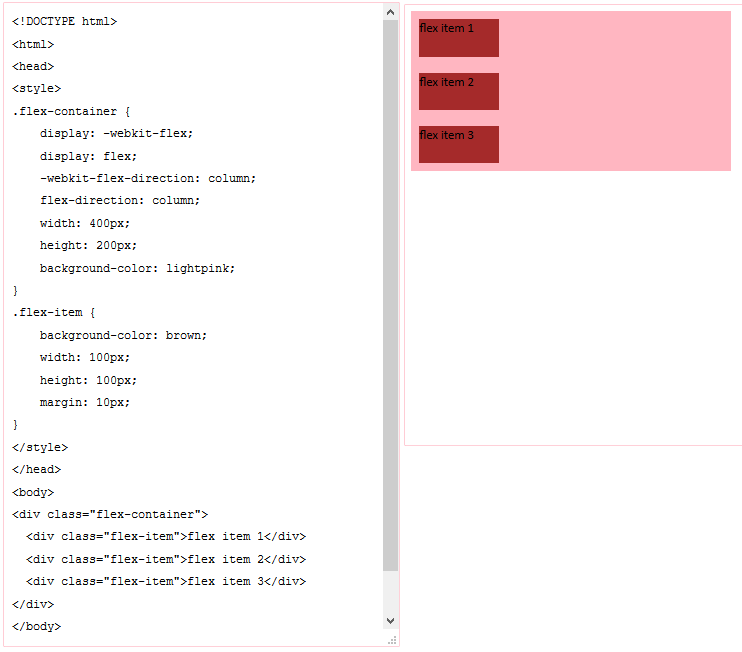
CSS Flexbox1  
  
Huomaa: Flexbox määrittää, kuinka flex-elementit asetetaan flex-kontin sisällä. Se asettaa flex-elementit flex-säiliöön flex-linjaa pitkin. Oletuksena on vain yksi flex-viiva kohti flex-säiliötä. Kaikki joustavasta säiliöstä ja flex-elementin sisällä pidetään tavallisena.  
  
Otetaan esimerkki näyttämään kolme joustavaa osaa flex-säiliössä. Oletusarvon mukaan ne on asetettu vaakasuoralle joustorille, vasemmalta oikealle:





Flex Direction Property  
  
Flex-suunta-ominaisuutta käytetään joustavien kohteiden suuntauksen säätämiseksi flex-säiliön sisällä. sen oletusarvo on rivi (vasemmalta oikealle, ylhäältä alas).  
  
Muut mahdolliset arvot ovat:  
  
     rivi-käänteinen  
     sarake  
     sarake-käänteinen

Käytä sarakkeen arvoa  
  
Otetaan esimerkki osoittamaan sarakkeen arvon käyttö.



[**https://www.javatpoint.com/css-flex-justify-content**](https://www.javatpoint.com/css-flex-justify-content) **Css flex justify**

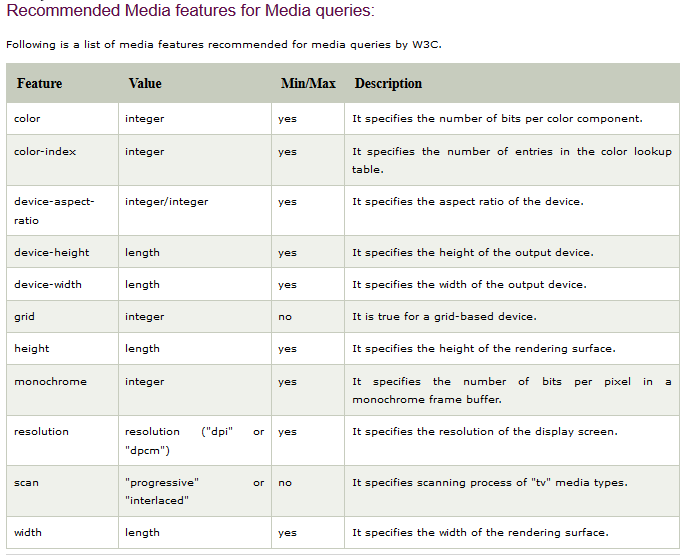
[**https://www.javatpoint.com/css-flex-align-items**](https://www.javatpoint.com/css-flex-align-items) **Css flex align items**

[**https://www.javatpoint.com/css-flex-wrap**](https://www.javatpoint.com/css-flex-wrap) **Css flex wrap**

[**https://www.javatpoint.com/css-flex-align-content**](https://www.javatpoint.com/css-flex-align-content) **css flex align content**

**CSS media Query** CSS-mediakyselyt

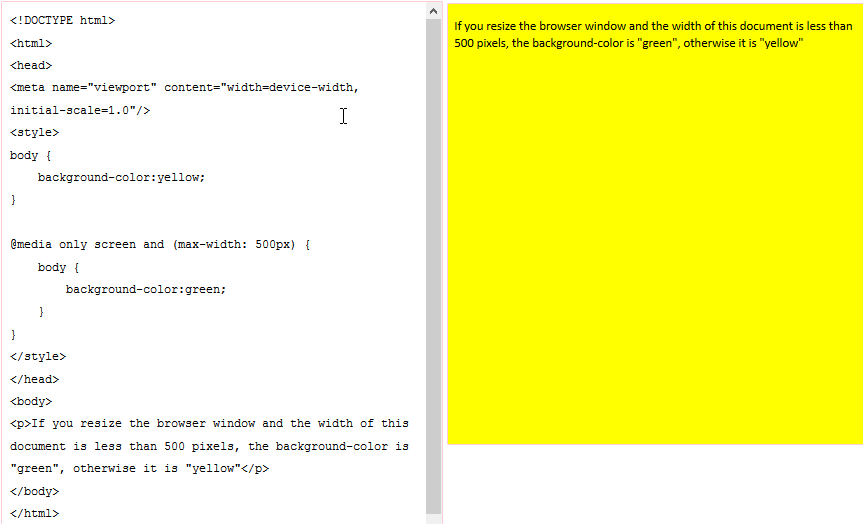
Mikä on Media Query?  
  
CSS Media-kysely on W3C-suositus ja CSS3-moduuli, jota käytetään sopeutumaan sellaisiin olosuhteisiin kuin näytön tarkkuus (esimerkiksi älypuhelimen näyttö ja tietokonenäyttö).  
  
     Media-kyselymenetelmä, jota käytetään CSS3: ssa.  
     Se on laajennusta mediasisällöistä tyylisivuista, joita käytetään erilaisissa mediatyypeissä (eli näytössä ja tulostuksessa), jotka löytyvät CSS2: sta.  
     Yleisimmin käytetty mediaominaisuus on "leveys".  
     Se käyttää @media-sääntöä sisällyttämään CSS-ominaisuuksien lohkon vain, jos tietty ehto on tosi.



**Mikä on responsiivinen web design**

Mikä on Responsive Web Design?  
  
Termi Responsible Web Design antoi Ethan Marcotte. Se helpottaa verkkojen, joustavien kuvien ja mediakyselyjen käyttöä verkkosivun lisäämiseksi asteittain erilaisille katseluympäristöille eli työpöydälle, älypuhelimelle, tablet-laitteelle jne.

Mitkä näytön resoluutiot käytät kuvakaappauksen aikana?  
  
Älypuhelin: 320px  
  
Tabletti: 768px  
  
Netbook: 1024px  
  
Työpöytä: 1600px  
  
Otetaan esimerkiksi esimerkki multimediakyselyn yksinkertaisesta käytöstä:  
  
Tässä esimerkissä määritetään, että jos pienennät ikkunasi alle 500px, taustan väri muuttuu.



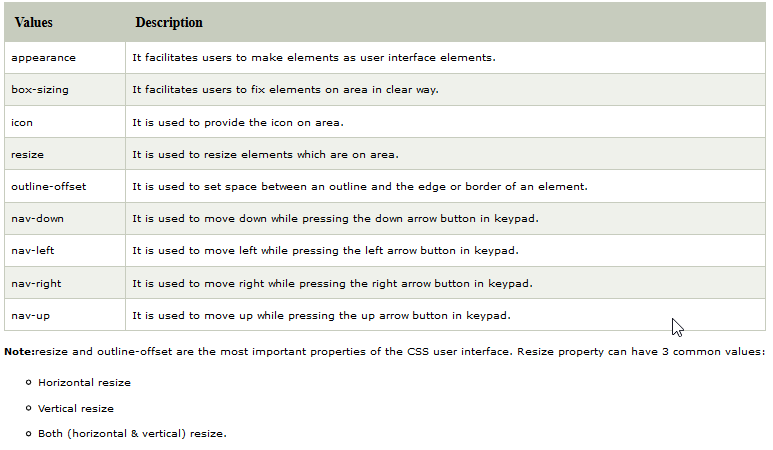
[**https://www.javatpoint.com/oprweb/test.jsp?filename=css-media-query2**](https://www.javatpoint.com/oprweb/test.jsp?filename=css-media-query2) **sivun responsiivisuus**

**CSS 2d Transforms**

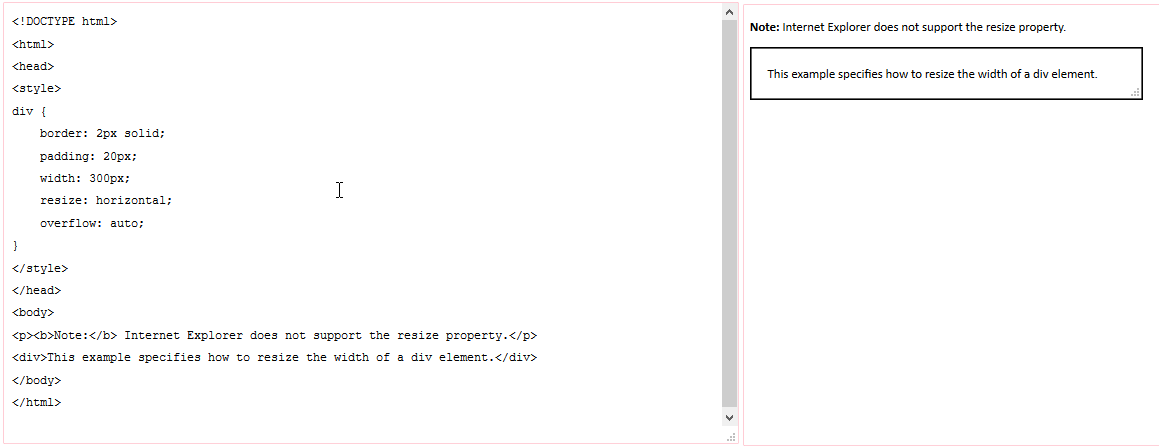
[**https://www.javatpoint.com/css-2d-transforms**](https://www.javatpoint.com/css-2d-transforms) **Css 2d transforms**

**CSS 3d Transforms**

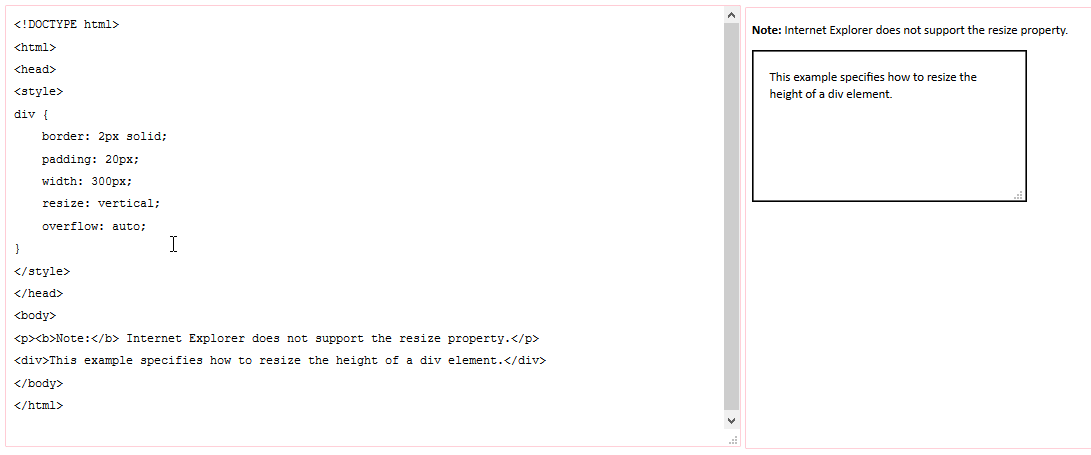
[**https://www.javatpoint.com/css-3d-transforms**](https://www.javatpoint.com/css-3d-transforms) **Css 3d transforms**

**Css User Interface** CSS-käyttöliittymä  
  
CSS tarjoaa monia käyttöliittymän ominaisuuksia, kuten resize-elementtejä, ääriviivoja ja laatikkokohtia.  
  
Seuraavassa on luettelo CSS3-käyttöliittymän yleisistä ominaisuuksista: 

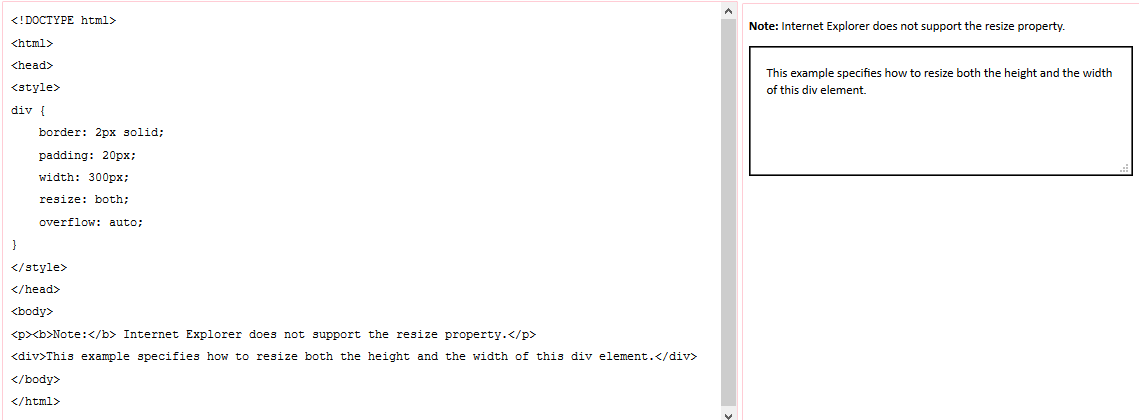
**Vaaka**

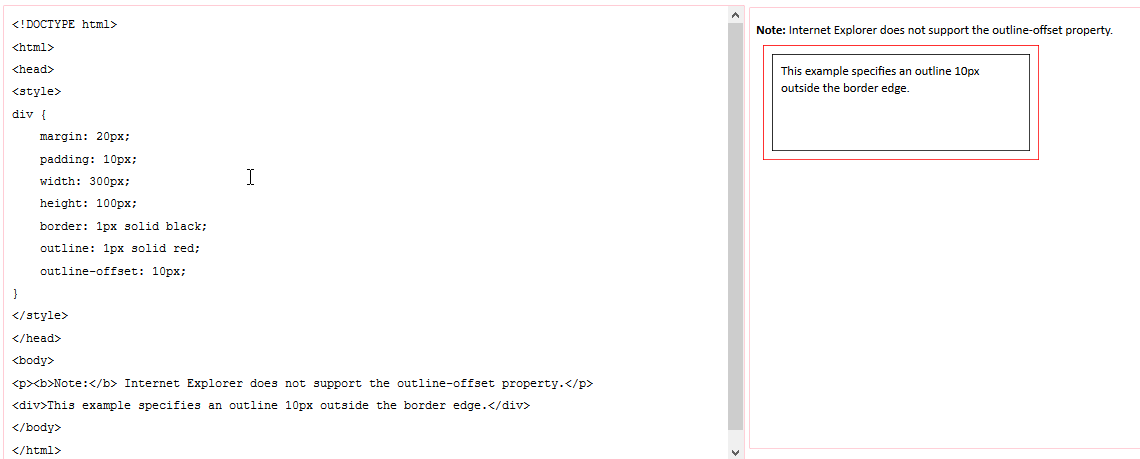


**Pysty**



**Molemmat reunat**





**CSS Pagination**

[**https://www.javatpoint.com/css-pagination**](https://www.javatpoint.com/css-pagination)kuin navigointi

**Css Layout** CSS -asettelu  
  
CSS-asettelu on helppo suunnitella  
Verkkosivun ulkoasun suunnitteluun on kolme tapaa:  
  
     HTML Div ja CSS: nopea ja laajasti käytetty.  
     HTML-taulukko: hidas ja vähemmän suositeltava.  
     HTML-kehys: vanhentunut.  
  
CSS-ulkoasussa voi olla otsikko, alatunniste, vasen paneeli, oikea paneeli ja bodyn osa. Katsotaanpa yksinkertainen esimerkki CSS-ulkoasusta.

**Css Table**

[**https://www.javatpoint.com/css-table**](https://www.javatpoint.com/css-table) **taulukon ominaisuuksia**