

CSS3

CSS3



```
body { color: #6000;  
      font: 16px Times, serif;  
}  
  
h1 { color: #f00;  
     font-size: 24px;  
}
```

CSS3 Nedir ?

CSS3 (Cascading Style Sheets, sürüm 3) bir HTML belgesinde görünümün ve biçimlendirilmesinin tarif edilmesi için kullanılan stil sayfasıdır.

CSS3, CSS'nin en son sürümüdür ve web sayfalarının görünümünü tasarlamak için güçlü ve esnek stil seçenekleri sağlar.

Güncel web geliştirmeye için önemli bir araçtır ve modern web tarayıcılar tarafından geniş ölçüde desteklenir.

Yerleşim ve düzenleme alanında geliştirilmiş destek, yeni yazı tipi seçenekleri, metin gölgeleri, gradientler, şeffaf renkler kullanma ve ekranın boyutuna göre farklı stil uygulama imkanı sunan media sorguları sayesinde birçok cihaza uyumlu tasarım gibi yeni özellikler ve iyileştirmeler içerir.

CSS3 ile Yeni eklenen Özellikler

1. Yeni seçimciler: Bu seçimciler, daha spesifik elementlere stil atama imkanı sağlar. Örneğin, ".last-child" seçicisi ile bir elementin listenin son elemeni olduğu durumlarda stil atayabilirsiniz.
2. Geçişler ve animasyonlar: CSS3 sayesinde, elementlerin görünümünü değiştirdikten sonra kullanabilirsiniz. Bu, elementlerin görünümünü daha dinamik hale getirir ve kullanıcı deneyimini iyileştirir.
3. Media sorguları: Bu özellik sayesinde, web sayfalarının görünümünü farklı cihazlar ve ekran çözünürlüklerine göre özelleştirebilirsiniz. Örneğin, bir web sayfasının mobil cihazlarda farklı bir görünümü olabilir.

4. Çoklu arka plan resimleri: CSS3 ile, bir elemente birden fazla arka plan resmi atayabilirsiniz. Bu, elementlerin görünümünü daha zengin hale getirir.
5. Sınır çerçeveleri: Bu özellik sayesinde, elementlerin çerçevelerini özelleştirebilirsiniz. Örneğin, çerçevenin köşelerini yuvarlatabilir veya farklı renklerde çerçeveler oluşturabilirsiniz.
6. Yeni renkler: CSS3 ile, yeni renkler kullanabiliyorsunuz. Örneğin, "rgba()" ve "hsl()" gibi renk değerleri sayesinde transparan renkler ve renk değişimleri oluşturabilirsiniz.
7. Flexbox: Bu özellik, elementleri birbirlerine göre hizalama ve dağıtma işlevlerini kolaylaştırır. Örneğin, bir div elementinin içindeki öğeleri dikey olarak ortala'yabilir veya bir kutunun içindeki öğeleri yatay olarak dağıtabilirsiniz.

8. Grid: Bu Özellik, elementleri bir gridde sıralama ve düzenlemeye imkanı sağlar. Örneğin, bir div elementinin içindeki öğeleri bir tablo gibi düzenleyebilir ve hücrelerine stil atayabilirsiniz.

8. Ölçeklenebilir vektör grafikleri (SVG): Bu Özellik, web sayfalarında ölçeklenebilir vektör grafikleri kullanımı sağlar. Bu grafikler, web sayfalarında yüksek çözünürlüklu resimler kullanımını azaltır ve daha küçük dosya boyutlarına sahiptir.

8. Ölçeklenebilir birimler: CSS3 ile, birimleri ölçeklendirme imkanı sağlar. Örneğin, "vh" ve "vw" gibi birimler sayesinde elementlerin genişliğini ve yüksekliğini ekranın genişliğine ve yüksekliğine göre ayarlayabilirsiniz.

Yeni Seçiciler

CSS3 ile birlikte gelen yeni seçenekler, web sayfalarınızda daha spesifik öğeleri seçmenize yardımcı olur. Bu seçenekler, mevcut HTML etiketlerine ve özelliklerine ek olarak, daha fazla seçim imkanı sunar.

Örneğin, ":nth-child" seçicisi sayesinde, bir elementin çocuklarından belirli bir öğeyi seçebilirsiniz. Bu, örneğin bir tablo içinde belirli bir sütunu veya satırı özelleştirmek için kullanılabilir.

```
tr td:nth-child(3) {  
    color: red;  
}
```

Örnekte, "td" etiketlerinin 3. sıradaki öğelerine kırmızı renk verilmiştir:

```
:not(input) {  
    background-color: blue;  
}
```

Örnekte, "input" etiketlerinin dışındaki tüm etiketlerin arka plan rengi mavi olmuştur:

Seçicilerin listesi

".root": HTML dosyasının en üst düzeyde bulunan "html" elementini seçer. Bu seçici, örneğin HTML dosyasının genel stilini belirlemek için kullanılabilir.

".empty": İçeriği olmayan elementleri seçer. Örneğin, bir div elementi içinde hiç yazı veya öğe bulunmaması durumunda, bu seçici ile bu div elementini seçebilirsiniz.

".target": Bir URL içinde "#" işaretiyile belirtilen hedef elementi seçer. Örneğin, bir sayfada "#section2" olarak belirtilen bir hedef elementi varsa, bu seçici ile bu elementi seçebilirsiniz.

".nth-last-child": Bir elementin çocukların tersine sayılış belirli bir sıradaki öğeyi seçer. Örneğin, bir tablo içinde son sütunu veya satırı özelleştirmek için kullanılabilir.

".enabled" ve ".disabled": Bu seçenekler, etkin ve devre dışı form elementlarını seçer. Örneğin, bir giriş alanının devre dışı olduğunu belirtmek için ".disabled" seçicisini kullanabilirsiniz.

".checked": Seçili olarak işaretlenmiş radio veya checkbox elementlerini seçer. Örneğin, bir form içinde seçili olarak işaretlenmiş bir checkbox'ın stilini değiştirmek için kullanılabilir.

".first-child" ve ".last-child": Bu seçenekler, bir elementin ilk ve son çocuğunu seçer. Örneğin, bir div elementi içındaki ilk p elementini veya son p elementini özelleştirmek için kullanılabilir.

".first-of-type" ve ".last-of-type": Bu seçenekler, bir elementin aynı türde olan ilk ve son çocuğuunu seçer. Örneğin, bir div elementi içindeki ilk p elementini veya son p elementini özelleştirmek için kullanılabilir.

".dir()": Belirtilen yönü olan elementleri seçer. Örneğin, ".dir rtl" seçicisiyle, sağdan sola yazı yönü olan elementleri seçebilirsiniz.

":nth-last-of-type": Bir elementin aynı türde olan çocuklarından tersine sayılış belirli bir sıradaki öğeyi seçer. Örneğin, bir div elementi içinde son p elementini özelleştirmek için kullanılabilir.

":nth-of-type": Bir elementin aynı türde olan belirli bir sıradaki çocuğunu seçer. Örneğin, bir div elementi içindeki ikinci p elementini özelleştirmek için kullanılabilir.

":only-child": Bir elementin tek çocuğu olduğu durumda o elementi seçer. Örneğin, bir div elementi içinde tek bir p elementi varsa, bu seçici ile o p elementini özelleştirebilirsiniz.

":only-of-type": Bir elementin aynı türde olan tek çocuğu olduğu durumda o elementi seçer. Örneğin, bir div elementi içinde tek bir p elementi varsa, bu seçici ile o p elementini özelleştirebilirsiniz.

Yeni Geçişler ve Animasyonlar

Yeni geçişler ve animasyonlar, web sayfalarınızda görsel efektler oluşturmamanıza yardımcı olur. Bu Özellikler sayesinde, öğelerin görünümünü zaman'a göre değiştirebilir veya animasyon oluşturabilirsiniz.

GEÇİŞLER

Geçişler, bir öğenin görünümünün bir andan diğerine nasıl değiştiğini belirtir.

Örneğin, bir div elementinin arka plan renginin mouse imlecinin üzerinde olduğu sırada değişmesi gibi. Aşağıdaki örnekte, bir div elementinin arka plan rengi mouseover olayı sırasında 2 saniye içinde yeşil rengine dönüşür:

```
div {  
background-color: blue;  
transition: background-color 2s;  
}  
  
div:hover {  
background-color: green;  
}
```

ANİMASYONLAR

Animasyonlar ise, bir öğenin görünümünün zamana göre nasıl değiştiğini belirtir.

Örneğin, bir div elementinin opaklılığınımouseover(fare imlecinin elementin üstünde olduğu durum) olayı sırasında değiştmesi gibi.

```
div {  
    opacity: 1;  
    background-color: red;  
    animation: fadeOut 2s;  
}  
  
@keyframes fadeOut {  
    from {opacity: 1}  
    to {opacity: 0.5}  
}  
  
div:hover {  
    animation-play-state: paused;  
}
```

Örnekte, bir div elementinin opaklısı sayfa yüklenliğinde 2 saniye içinde yüzde 50'ye düşürülür:

Eğer animasyon sırasında imleç elementin üzerine gelirse animasyon durur ve imleç üzerinden gidene kadar devam etmez

Giriş ve animasyon özellikleri

"transition-property": Geçişin hangi özelliğin (örneğin, arka plan rengi, genişlik, yükseklik gibi) değişeceğini belirtir.

"transition-duration": Geçişin kaç saniye süreceğini belirtir.

"transition-timing-function": Geçişin hızının nasıl değişeceğini belirtir. Örneğin, "ease" değeriyle geçiş yavaşlar ve hızını artırır, "linear" değeriyle ise geçiş sabit hızda gerçekleşir.

"animation-name": Animasyonun hangi adımları (keyframes) izleyeceğini belirtir.

"animation-duration": Animasyonun kaç saniye süreceğini belirtir.

"animation-timing-function": Animasyonun hızının nasıl değişeceğini belirtir. Örneğin, "ease" değeriyle animasyon yavaş başlar ve hızını artırır, "linear" değeriyle ise animasyon sabit hızda gerçekleşir.

"animation-delay": Bu özellik, animasyonun ne kadar süre sonra başlayacağını belirtir.

Media Sorguları

CSS3 ile birlikte gelen media sorguları, web sayfalarınızın görünümünü cihaz türü, ekran boyutu ve diğer özelliklere göre değiştirmeye yardımcı olur. Bu sayede, bir web sayfasının farklı cihazlarda farklı görünmesini sağlayabiliyorsunuz.

```
@media screen and (max-width: 768px) {  
    div {  
        background-color: red;  
    }  
}
```

Media sorguları, @media rule'u ile kullanılır.

Örneğin, yandaki örnekte, ekran genişliği 768px'ten düşük olan cihazlarda (örneğin, tabletler ve telefonlar) div elementinin arka plan rengi kırmızı olur:

Media sorgularında kullanabileceğiniz çok sayıda özellik vardır. Örneğin:

"min-width" ve "max-width" özellikleri: Ekran genişliğinin minimum ve maksimum değerlerini belirtir.

"min-height" ve "max-height" özellikleri: Ekran yüksekliğinin minimum ve maksimum değerlerini belirtir.

"orientation" özelliği: Cihazın yatay veya dikey pozisyonunu belirtir.

"aspect-ratio" ve "device-aspect-ratio" özellikleri: Ekranın en-boy oranını belirtir.

"resolution" özelliği: Ekran çözünürlüğünü belirtir. Örneğin, "resolution: 300dpi" ile cihazların 300 dpi çözünürlükte olması gereken stil özelliklerini tanımlayabilsiniz.

Sınır Çerçeveleri (border-radius)

CSS3 ile birlikte gelen sınır çerçeveleri (border-radius), web sayfalarınızdaki öğelerin köşelerinin yuvarlatılmasını sağlar. Bu özellik sayesinde, dikdörtgenlerden oluşan öğelerin yerine, yuvarlak köşeli öğeler kullanabilirsiniz.

```
div {  
    border-radius: 50px;  
}
```

Sınır çerçevelerini kullanmak için, "border-radius" özelliğini kullanırsınız.

Örneğin, üstteki örnekte, div elementinin tüm köşeleri 50px yuvarlatılmıştır:

Sınır çerçevelerini daha spesifik olarak kullanmak için, aşağıdaki gibi köşe bazlı özelliklerini de kullanabilirsiniz:

- "border-top-left-radius"
- "border-top-right-radius"
- "border-bottom-left-radius"
- "border-bottom-right-radius"

```
div {  
    border-top-left-radius: 20px;  
}
```

Örnekte, div elementinin sol üst köşesi 20px yuvarlatılmıştır.

Yeni Renkler

CSS3 ile birlikte gelen yeni renkler, web sayfalarınızda daha fazla renk seçeneği sunar. Önceki sürümlerinde, renkleri sadece RGB ve HSL değerleriyle belirtme imkanı vardı.

Ancak CSS3 ile birlikte, daha çeşitli renkleri "name", "hex" ve "rgba" gibi farklı yollarla belirtme imkanına sahipsiniz.

```
div {  
background-color: aliceblue;  
}
```

Örnekte, div elementinin arka plan rengi "aliceblue" olarak belirtilmiştir:

Bu örnekte ise, div elementinin arka plan rengi "hex" değerini kullanarak belirtimiştir:

```
div {  
background-color: #F0F8FF;  
}
```

Ve aşağıda örnekte ise, div elementinin arka plan rengi "rgba" değerini kullanarak belirtimiştir:

```
div {  
background-color: rgba(240, 248, 255, 0.5);  
}
```

Flexbox

CSS3 ile birlikte gelen yeni Flexbox (Flexible Box), web sayfalarınızda daha esnek ve daha kolay yönetilebilir bir düzenleme sağlar. Önceki sürümlerinde, düzenleme için sadece "float", "position" ve "display" gibi özellikleri kullanabiliyordunuz. Ancak Flexbox ile, öğelerinizi daha kolayca yatay ve dikey olarak sıralayabilir, genişliklerini otomatik olarak ayarlayabilir ve diğer özelliklerini daha kolayca değiştirebilirsiniz.

Flexbox'ı kullanmak için, "display" özelliğini "flex" olarak ayarlayın.

Aşağıdaki örnekte, div elementinin Flexbox olarak ayarlandığı gösterilmiştir:

```
div {  
  display: flex;  
}
```

Daha sonra, Flexbox öğelerinizi yönetmek için "flex-direction", "flex-wrap", "justify-content" ve "align-items" gibi özellikleri kullanabilirsiniz. Örneğin, aşağıdaki örnekte, div elementinin içindeki öğeler yatay olarak sıralanmış ve ortalanmıştır:

```
div {  
  display: flex;  
  flex-direction: row;  
  justify-content: center;  
}
```

"align-self" Özelliği: Bu özellik, tek tek Flexbox öğelerinin hizalamasını değiştirmeye yardımcı olur. Örneğin, aşağıdaki örnekte, div elementinin içindeki tek tek öğeler sağ hizalı olarak gösterilmiştir:

```
div {  
  display: flex;  
  flex-direction: row;  
  justify-content: center;  
  align-items: center;  
}  
div span {  
  align-self: flex-end;  
}
```

A Flexbox Cheatsheet

İşlek Vönü

You can change the overall content flow (column or row), and you can even change the arrangement of content.

```
.container {  
    display: flex;  
    align-items: center;  
}  
  
1 2 3 4 5
```

```
.container {  
    display: flex;  
    justify-content: center;  
}  
  
1 2 3 4 5
```

```
.container {  
    display: flex;  
    align-items: flex-start;  
}  
  
1 2 3 4 5
```

```
.container {  
    display: flex;  
    align-items: flex-end;  
}  
  
1 2 3 4 5
```

```
.container {  
    display: flex;  
    justify-content: space-around;  
}  
  
1 2 3 4 5
```

Yatay Hizalama

You can align items horizontally as a group or individually.

```
.container {  
    display: flex;  
    justify-content: center;  
}  
  
1 2 3 4 5
```

```
.container {  
    display: flex;  
    align-items: flex-start;  
}  
  
1 2 3 4 5
```

```
.container {  
    display: flex;  
    align-items: flex-end;  
}  
  
1 2 3 4 5
```

```
.container {  
    display: flex;  
    justify-content: space-around;  
}  
  
1 2 3 4 5
```

Dikey Hizalama

You can align items horizontally as a group or individually.

```
.container {  
    display: flex;  
    align-items: center;  
}  
  
1 2 3 4 5
```

```
.container {  
    display: flex;  
    justify-content: center;  
}  
  
1 2 3 4 5
```

```
.container {  
    display: flex;  
    align-items: flex-start;  
}  
  
1 2 3 4 5
```

```
.container {  
    display: flex;  
    align-items: flex-end;  
}  
  
1 2 3 4 5
```

Taşınmaları kontrol etme

By default, all items are put on a single line.

```
.container {  
    display: flex;  
}  
  
1 2 3 4 5 6 7 8 9 10 11
```

```
.container {  
    display: flex;  
    flex-direction: row-reverse;  
}  
  
5 4 3 2 1
```

```
.container {  
    display: flex;  
    flex-direction: column;  
}  
  
1 2 3 4 5
```

```
.container {  
    display: flex;  
    flex-direction: column-reverse;  
}  
  
11 12 13 14 15
```

Default Özelliği

The default behavior of Flexbox will...

- Treat the container as block (full width)
- Left align all items
- Stretch each item's height to fit the container

```
.container {  
    display: flex;  
}  
  
1 2 3 4 5
```

Yatay-Dikey Hizalama

You can combine selectors to get your desired layout. Perfect centering is a breeze with Flexbox.

```
.container {  
    display: flex;  
    justify-content: space-between;  
}  
  
1 2 3 4 5
```

```
.container {  
    display: flex;  
    align-items: center;  
    justify-content: center;  
}  
  
1 2 3 4 5
```

Grid

CSS3 ile birlikte gelen yeni Grid (Düzenli), web sayfalarınızda daha esnek ve daha kolay yönetilebilir bir düzenleme sağlar. Grid ile, bir web sayfasının tüm alanını kolayca düzenleyebilir, öğelerinizi daha kolayca sıralayabilir ve diğer özelliklerini daha kolayca değiştirebilirisiniz.

Grid'ı kullanmak için, "display" özelliğini "grid" olarak ayarlayın.

Örneğin, aşağıdaki örnekte, div elementinin Grid olarak ayarlandığı gösterilmiştir:

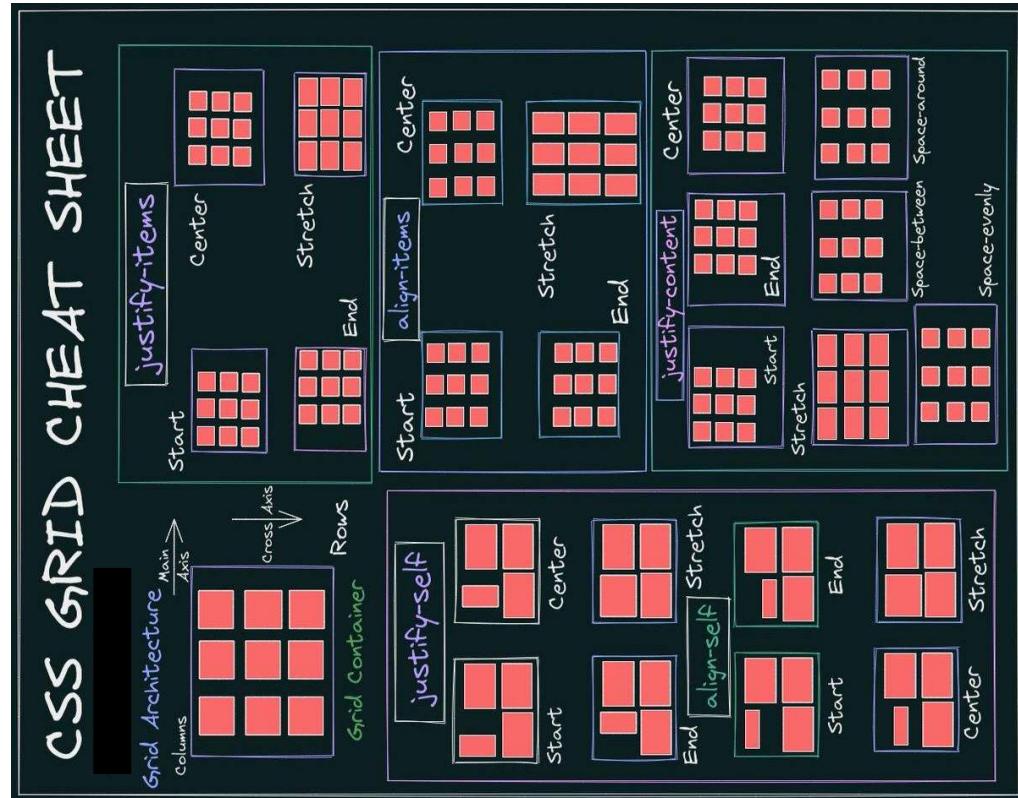
```
div {  
    display: grid;  
}
```

CSS GRID CHEAT SHEET

Daha sonra, Grid öğelerinizi yönetmek için "grid-template-columns", "grid-template-rows", "grid-template-areas" ve "grid-gap" gibi Özellikleri kullanabilirsiniz.

Örneğin, aşağıdaki örnekte, div elementinin içindeki öğeler iki sütun ve iki satır olarak sıralanmıştır:

```
div {  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
  grid-template-rows: 1fr 1fr;  
}
```



Box Shadow

CSS3 ile birlikte gelen "box-shadow" özelliği, bir elemente gölgeme ye yardımcı olur. Bu özellik sayesinde, elementlerinizin görünümünü daha dinamik hale getirebilir ve daha fazla ön plana çıkarmaya yardımcı olabilirsiniz.

"box-shadow" özelliğini kullanmak için, aşağıdaki şekilde belirtim:

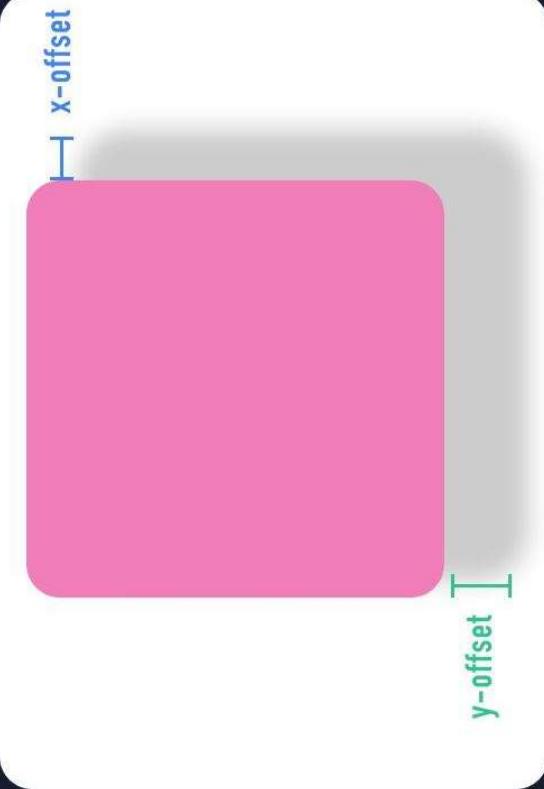
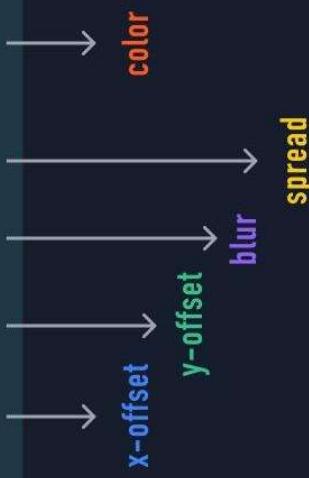
```
element {  
  box-shadow: [x-ekseni-uzaklı ğı] [y-ekseni-uzaklı ğı] [blur-çapı] [dağınlık] [renk];  
}
```

Box Shadow

The `box-shadow` property enables you to cast a drop shadow from the frame of almost any element.



```
box-shadow: 10px 20px 8px 5px #333333;
```



Box Sizing

Varsayılan olarak HTML elementlerinin genişlik ve yüksekliği şu şekilde hesaplanır.

`width + padding + border = HTML elementinin genişliği`
`height + padding + border = HTML elementinin yüksekliği`

Bir elementin genişlik ve yüksekliğini belirlediğinizde element genellikle belirlediğinizden daha büyük görünür. (yükseklik ile genişliğine kenarlık (border) ve iç boşluk (padding) eklendiğinden dolayı)

```
.div1 {  
    width: 300px;  
    height: 100px;  
    border: 1px solid blue;  
}
```

```
.div2 {  
    width: 300px;  
    height: 100px;  
    padding: 50px;  
    border: 1px solid red;  
}
```

Yandaki iki <div> elementinin boyutları aynı belirtilmesine rağmen farklı olacaktır. (div2 elementine iç boşluk eklendiğinden)

Bu sorunu genişlik ile yükseklikten iç boşluk ve kenarlık değerlerini çıkartarak çözüyorduk.

CSS3 box-sizing özelliğiyle artık kutu hesaplamasını biz belirleyebiliyoruz.

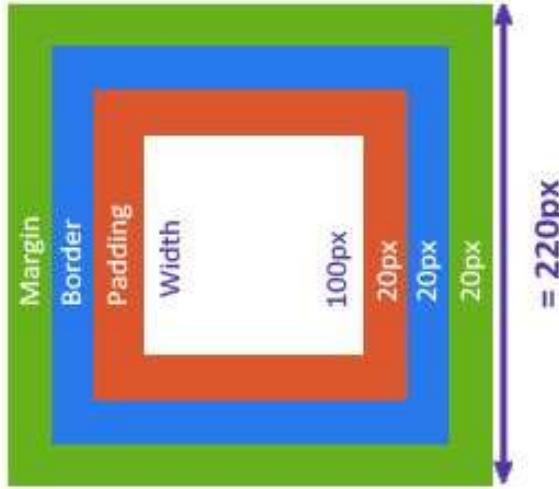
```
.div1 {  
    width: 300px;  
    height: 100px;  
    border: 1px solid blue;  
    box-sizing: border-box;  
}  
  
.div2 {  
    width: 300px;  
    height: 100px;  
    padding: 50px;  
    border: 1px solid red;  
    box-sizing: border-box;  
}
```

Örnekte kullanılan box-sizing:border-box; özelliği sayesinde kutucuklara verilen farklı padding boyutlarına rağmen ikisinde genişlik ve yükseklik değeri belirtilenden farklı olmayacaktır.

CSS3 Box Model

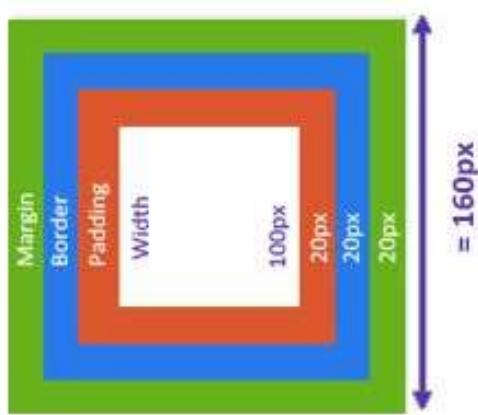
box-sizing : content-box;

Verilen genişliğe padding, border, dahil değildir yani onlar sonradan eklenir



box-sizing : border-box;

Verilen genişliğe padding, border, dahildir yani kutucuk boyutu belirttiğiniz genişliği geçemez



Column

CSS3 "column" özelliği, bir elementin içerisindeki içeriğin birden fazla sütuna bölünmesine yardımcı olur. Bu özellik sayesinde, elementlerinizin içeriğini daha okunabilir hale getirebilir ve daha fazla yer kaplamamasını sağlayabilsiniz.

"column" özelliğini kullanmak için, aşağıdaki şekilde belirtin:

```
element {  
    column-count: [value];  
    column-gap: [value];  
}
```

"column-count": Bu değer, sütun sayısını belirtir. Örneğin, "column-count: 3" değeri, içeriğin 3 sütuna bölünmesini sağlar.

"column-gap": Bu değer, sütunlar arasındaki boşluk miktarını belirtir. Örneğin, "column-gap: 20px" değeri, sütunlar arasında 20px boşluk bırakır.

Div elementinin içeriğinin 3 sütuna bölünmesi ve sütunlar arasında 20px boşluk bırakılması için aşağıdaki CSS kodunu kullanabilirsiniz:

```
div {  
    column-count: 3;  
    column-gap: 20px;  
}
```

Bu sadece birkaç örnekti ve "column" Özelliğinin diğer Özellikleri de mevcuttur. Örneğin, "column-width" Özelliği sayesinde, sütunların minimum genişliğini belirleyebilir, "column-rule" Özelliği sayesinde sütunlar arasında çizgi tasarımını değiştirebilir ve "column-span" Özelliği sayesinde bir ögenin tüm sütunları kapsamasını sağlayabiliyorsınız.

```
div {  
    column-count: 3;  
    column-gap: 20px;  
    column-width: 200px;  
    column-rule: 3px solid gray;  
}
```

Örnekte, div elementinin içeriğinin 3 sütuna bölünmesi ve sütunlar arasında 20px boşluk bırakılması dışında, sütunların minimum genişliği 200px olarak ayarlanmış ve sütunlar arasında 3px kalınlığında, gri renkte çizgi eklenmiştir:

CSS MULTIPLE COLUMNS

COLUMN-COUNT: 2;

Defines number of columns to be divided

CONTENT 1 **CONTENT 2**

`count1` `count2`

COLUMN-WIDTH: 100PX;

Defines width of each columns

CONTENT 1 **CONTENT 2**

`width= 100px` `width= 100px`

COLUMN-GAP: 50PX;

Defines gap between two columns

CONTENT 1 → **CONTENT 2**

`Gap`

COLUMN-RULE-STYLE: SOLID;

Sets column between two columns

CONTENT 1 | **CONTENT 2**

`Solid style`

COLUMN-RULE-WIDTH: 25PX;

Defines column width

CONTENT 1 | **CONTENT 2**

`25px width`

COLUMN-RULE-COLOR: PINK;

Defines column color

CONTENT 1 | **CONTENT 2**

`pink color`

Text Shadow

CSS3 ile birlikte gelen "text-shadow" özelliği, bir elementin içerisindeki yazının gölgесini eklemeye yardımcı olur. Bu özellik sayesinde, elementlerinizin yazısını daha dinamik hale getirebilir ve daha fazla ön plana çıkarmaya yardımcı olabilirsiniz.

"text-shadow" özelliğini kullanmak için, aşağıdaki şekilde belirtin:

```
element {  
    text-shadow: [offset-x] [offset-y] [blur-radius] [color];  
}
```

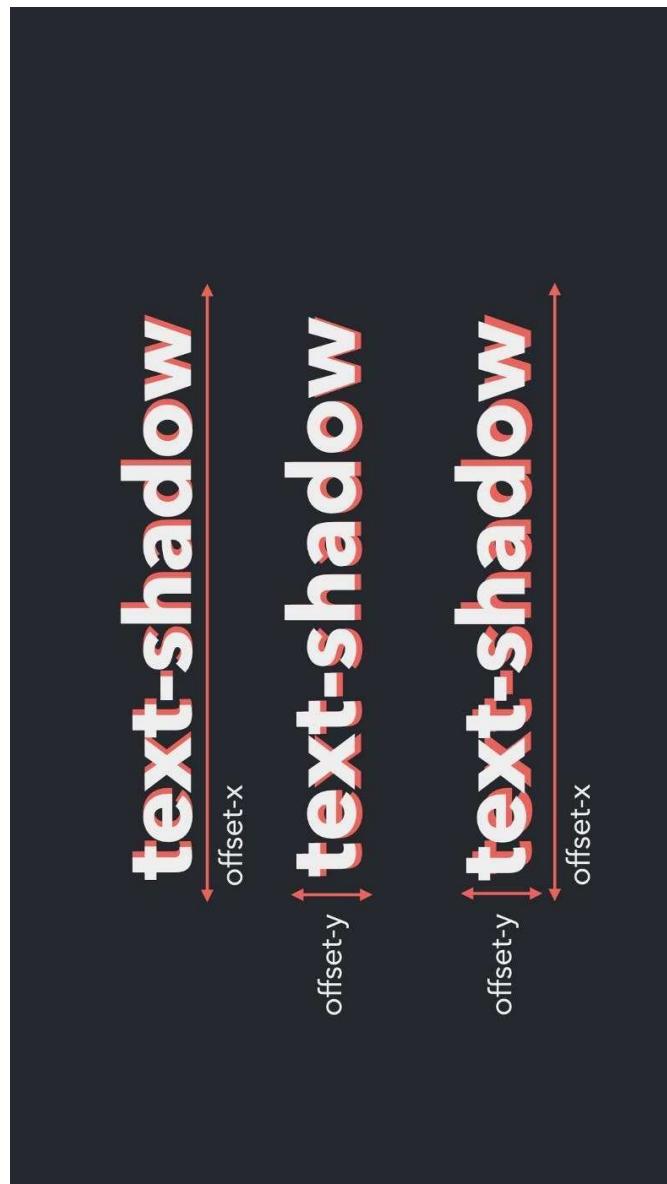
"offset-x" ve "offset-y": Bu değerler, gölgenin yatay ve dikey olarak pozisyonunu belirtir. Örneğin, "offset-x: 10px" değeri, gölgeyi 10px sağa kaydırır.

"blur-radius": Bu değer, gölgenin bulanıklık seviyesini belirtir. Örneğin, "blur-radius: 5px" değeri, gölgeyi 5px bulanık hale getirir.

"color": Bu değer, gölgenin rengini belirtir. Örneğin, "color: red" değeri, gölgeyi kırmızı renge dönüştürür.

Aşağıdaki örnekte, h1 elementine 10px sağa, 20px aşağı kaydırılmış, 5px bulanık hale getirilmiş ve kirmizi renkte bir gölge eklenmiştir:

```
h1 { text-shadow: 10px 20px 5px red; }
```



Opacity

CSS3 ile birlikte gelen "opacity" özelliği, bir elementin saydamlık seviyesini ayarlamaya yardımcı olur. Bu Özellik sayesinde, elementlerinizin görünümünü değiştirebilir ve daha fazla ön plana çıkarmaya yardımcı olabilirsiniz.

"opacity" özelliğini kullanmak için, aşağıdaki şekilde belirtin:

```
element {  
    opacity: [değer];  
}
```

"value": Bu değer, elementin saydamlık seviyesini belirtir. Değerler 0 ile 1 arasında olabilir, bu da elementin tamamen saydam (0) ve tamamen opak (1) olmasını sağlar.

Örneğin, "opacity: 0.5" değeri, elementin yarısı kadar saydam olmasını sağlar.

Aşağıdaki örnekte, div elementinin saydamlık seviyesi yarısına indirilmiştir:

```
div {  
    opacity: 0.5;  
}
```

Bu özellik sayesinde, elementlerinizin görünümünü değiştirebilir ve daha fazla ön plana çıkarmaya yardımcı olabilirsiniz. Ancak unutmayın, "opacity" özelliği sadece elementlerin saydamlık seviyesini değiştirir ve arka plan renkleri veya görselleri etkilemez. Bu nedenle, arka plan rengi veya görseli değiştirmek istiyorsanız, "background-color" veya "background-image" özelliklerini kullanmalısınız



Transform

CSS3 ile birlikte gelen "transform" özelliği, bir elementin görünümünü değiştirmeye yardımcı olur. Bu Özellik sayesinde, elementlerinizi Ölçeklendirme, döndürme, eğme ve benzeri işlemleri uygulayabilirsiniz.

"transform" Özelliğini kullanmak için, aşağıdaki şekilde belirtin:

```
element {  
    transform: [function] ([value]) [function] ([value]) ...;  
}
```

"function": Bu değer, uygulanacak işlemi belirtir. Örneğin, "scale" fonksiyonu elementi ölçeklendirir, "rotate" fonksiyonu elementi döndürür ve "skew" fonksiyonu elementi eğer.

"value": Bu değer, işlemin nasıl uygulanacağını belirtir. Örneğin, "scale(2)" değeri elementi 2 katına ölçeklendirir, "rotate(45deg)" değeri elementi 45 derece döndürür ve "skew(30deg, 20deg)" değeri elementi 30 derece yatay eksende ve 20 derece dikey eksende eğer.

Translate

Translate yöntemi uygulanan elemanı bulunduğu konumdan aldığı değere göre (X ekseni, Y ekseni) başka konuma taşımak için kullanılır.

```
div {  
    transform: translate(50px, 100px);  
}
```

Rotate

Rotate yöntemi uygulanan elemanı saat yönünde veya saat yönünün tersi yönünde döndürür.

```
div {  
    transform: rotate(20deg);  
}
```

Scale

Scale yöntemi uygulanan elemanı yeniden ölçeklendirmek (küçültmek, büyütmek) için kullanılır.

```
div {  
    transform: scale(2);  
}
```

Skew

Skew yöntemi uygulanan elemanı X ve Y ekseninde eğmek (sağ, sola, yukarı, aşağı) için kullanılır.

```
div {  
    transform: skew(20deg, 30deg);  
}
```

Font Face

CSS3 ile birlikte gelen "@font-face" kuralı, web sayfalarında kendi özel yazı tiplerini kullanmayı mümkün hale getirir. Bu kural sayesinde, standart yazı tiplerinin dışında, dilediğiniz yazı tipini web sayfalarınızda kullanabilirsiniz.

"@font-face" kuralını kullanmak için, aşağıdaki şekilde belirtin:

```
@font-face {  
    font-family: 'Roboto';  
    src: url('fonts/Roboto-Regular.ttf');  
}
```

Bu yazı tipini kullanmak için, aşağıdaki gibi bir CSS kuralı kullanılır:

```
body {  
    font-family: 'Roboto', sans-serif;  
}
```

Blend Mode

CSS3 ile birlikte gelen "blend-mode" Özelliği, bir elementin arka planındaki renklerle birleştirilmesine yardımcı olur. Bu Özelliğin sayesinde, elementlerinizin görünümünü değiştirebilir ve daha farklı görünümler oluşturabilirsiniz.

"blend-mode" Özelliğini kullanmak için, aşağıdaki şekilde belirtin:

```
element {  
  blend-mode: [value];  
}
```

"value": Bu değer, elementin arka planındaki renklerle nasıl birleştirileceğini belirtir.

Örneğin, "normal" değeri elementin arka planındaki renklerle hiçbir şekilde birleşirmez.

Text Overflow

CSS3 ile birlikte gelen "text-overflow" özelliği, bir elementin içinde yer alan yazının sınırları dışına çıkması durumunda, yazının nasıl gösterileceğini belirtir. Bu özellik sayesinde, elementlerinizin içinde yer alan yazının sınırları dışına çıktığında, yazının nasıl gösterileceğine kontrol edebilirsiniz.

"text-overflow" özelliğini kullanmak için, aşağıdaki şekilde belirtin:

```
element {  
    text-overflow: [value];  
}
```

"value": Bu değer, yazının sınırları dışına çıktığında yazının nasıl gösterileceğini belirtir.

Örneğin, "clip" değeri yazının sınırları dışında kalan kısmını keser ve göstermez, "ellipsis" değeri yazının sınırları dışında kalan kısmını "... gibi bir nokta üçlüsüyle gösterir.

"text-overflow" Özelliğinin "value" değeri için yazabilecekleriniz;

"clip": Bu değer, yazının sınırları dışında kalan kısmını keser ve göstermez. Bu değer, varsayılan değerdir ve yazının sınırları dışında kalan kısmını göstermez.

"ellipsis": Bu değer, yazının sınırları dışında kalan kısmını "..." gibi bir nokta üçlüsüyle gösterir. Bu değer sayesinde, yazının sınırları dışında kalan kısmı gösterilir ancak yazının gerçek uzunluğu değiştirilmez.

"string": Bu değer, yazının sınırları dışında kalan kısmını belirtilen bir dizeyeyle gösterir. Örneğin, "string: '...more'" şeklinde kullanılırsa, yazının sınırları dışında kalan kısmı "more" dizesiyle gösterilir.



Transition

CSS3 ile birlikte gelen "blend-mode" Özelliği, bir elementin arka planındaki renklerle birleştirilmesine yardımcı olur. Bu Özellik sayesinde, elementlerinizin görünümünü değiştirebilir ve daha farklı görünümler oluşturabilirsiniz.

"blend-mode" Özelliğini kullanmak için, aşağıdaki şekilde belirtin:

```
element {  
  transition: [property] [duration] [timing-function] [delay];  
}
```

"property" Özelliği: Geçişin hangi özelliğin (örneğin, arka plan rengi, genişlik, yükseklik gibi) değişeceğini belirtir.

"duration": Bu değer, değişimin ne kadar süre içinde gerçekleşeceğini belirtir. Örneğin, "2s" değeri değişimin 2 saniye içinde gerçekleşeceğini belirtir.

"timing-function": Bu değer, değişimin ne şekilde gerçekleşeceğini belirtir. Örneğin, "ease" değeri değişimin yavaş başlayıp yavaş biteceğini belirtir.

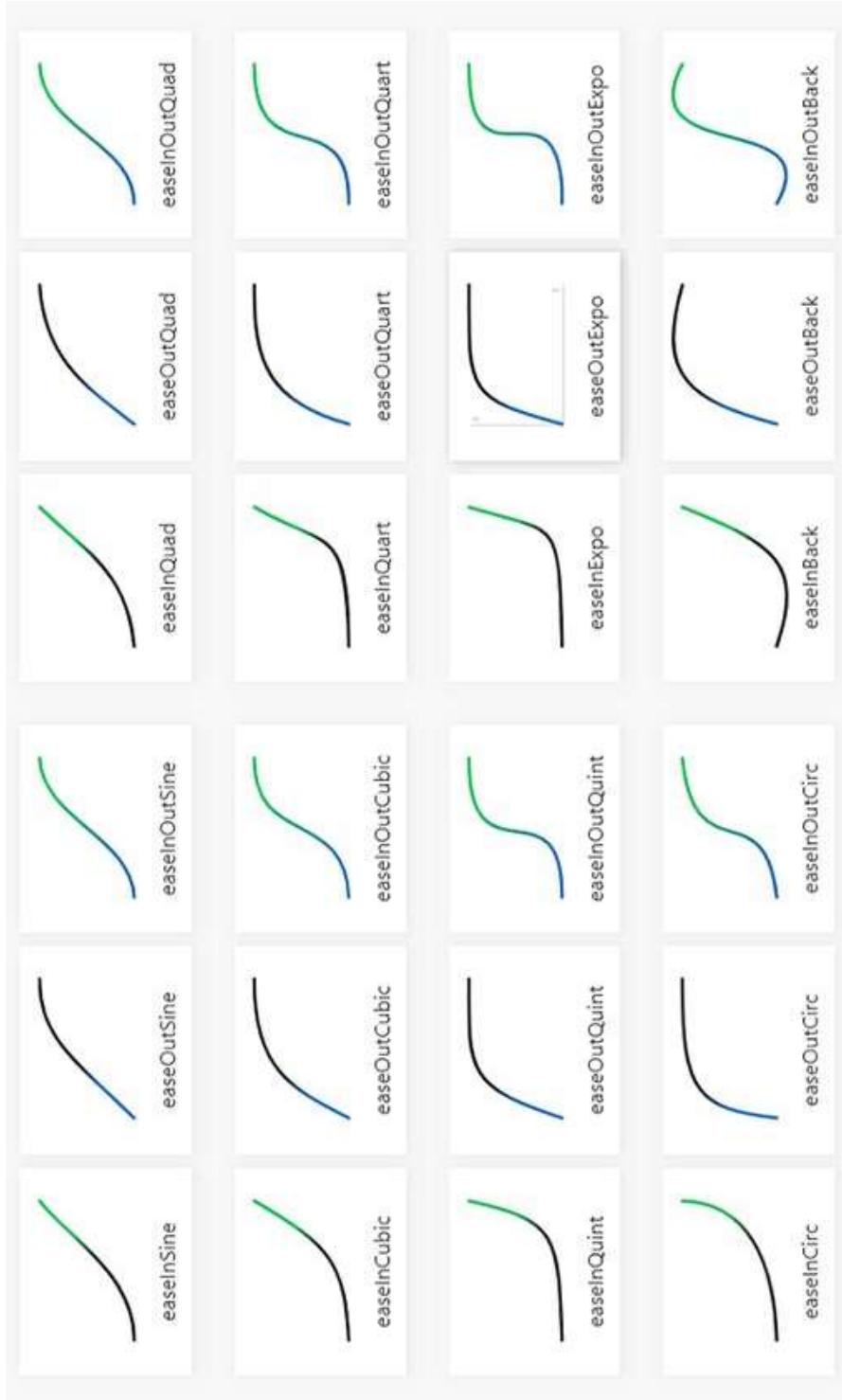
"delay": Bu değer, değişimin ne kadar süre sonra gerçekleşeceğini belirtir

Aşağıdaki örnekte, div elementinin genişliğinin değişimi sırasında, değişimin 2 saniye içinde, yavaş başlayıp yavaş bitecek şekilde gerçekleşeceği gösterilmektedir:

```
div { width: 100px; transition: width 2s ease; }

div:hover { width: 200px; }
```

timing-function in alabileceği değerler ve grafikleri



Linear Gradient

CSS3 ile birlikte gelen "linear-gradient" özelliği, bir elementin arka planına doğrusal bir gradyan eklemenizi sağlar. Bu özellik sayesinde, elementlerinizin arka planında doğrusal bir gradyan oluşturabilir ve bu sayede daha estetik bir görünüm sağlayabilirsiniz.

"linear-gradient" özelliğini kullanmak için, aşağıdaki şekilde belirtin:

```
element {  
background: linear-gradient ([direction [, color-stop1] [, color-stop2] [,  
...]);  
}
```

"direction": Bu değer, gradyanın hangi yönde olacağını belirtir. Örneğin, "to right" değeri gradyanın sağdan sola doğru olacağını belirtir.

"to left": Bu değer, gradyanın soldan sağa doğru olacağını belirtir.

"to right": Bu değer, gradyanın sağdan sola doğru olacağını belirtir.

"to top": Bu değer, gradyanın yukarıdan aşağıya doğru olacağını belirtir.

"to bottom": Bu değer, gradyanın aşağıdan yukarıya doğru olacağını belirtir.

"to top left": Bu değer, gradyanın sol üstten sağ alta doğru olacağını belirtir.

"to top right": Bu değer, gradyanın sağ üstten sol alta doğru olacağını belirtir.

"to bottom left": Bu değer, gradyanın sol altdan sağ üste doğru olacağını belirtir.

"to bottom right": Bu değer, gradyanın sağ altdan sol üste doğru olacağını belirt

"color-stop1": Bu değer, gradyanın hangi renklerle oluşacağını belirtir. Örneğin, "red" değeri gradyanın kırmızı renkle oluşacağını belirtir.

"color-stop2": Bu değer, gradyanın hangi noktada ne renge dönüştüğünü belirtir. Örneğin, "50%" değeri gradyanın yarısında ne renge dönüştüğünü belirtir.

Aşağıdaki örnekte, div elementinin arka planında sağdan sola doğru mavi ve yeşil renklerden oluşan gradyan oluşturduğu gösterilmektedir:

```
div {  
background: linear-gradient(to right, blue, green);  
}
```

Radial Gradient

CSS3 ile birlikte gelen "radial-gradient" özelliği, bir elementin arka planına radial (çember şeklinde) bir gradyan eklemenizi sağlar. Bu özellik sayesinde, elementlerinizin arka planında radial bir gradyan oluşturabilir ve bu sayede daha estetik bir görünüm sağlayabilirsiniz.

"radial-gradient" özelliğini kullanmak için, aşağıdaki şekilde belirtin:

```
element {  
background: radial-gradient ([shape] [, size] [, position] [, color-stop1] [,  
color-stop2] [, ...]);  
}
```

"**shape**": Bu değer, gradyanın hangi şekilde olacağını belirtir. Alabileceğimiz 2 değer `circle(daire)` ve `ellipse`.

"**size**": Bu değer, gradyanın hangi büyülükte olacağını belirtir. Örneğin, "`closest-side`" değeri gradyanın gradyanın çember şeklinde olacağı elemente en yakın tarafına göre büyülüğünü belirtir.

"position": Bu değer, gradyanın hangi noktadan başlayacağını belirtir. Örneğin, "center" değeri gradyanın ortadan başlayacağını belirtir.

"color-stop1": Bu değer, gradyanın hangi renklerle oluşacağını belirtir. Örneğin, "red" değeri gradyanın kırmızı renkle oluşacağını belirtir.

"color-stop2": Bu değer, gradyanın hangi noktada ne renge dönüşeceğini belirtir. Örneğin, "50%" değeri gradyanın yarısında ne renge dönüşeceğini belirtir.

Aşağıdaki Örnekte, div elementinin arka planında ortadan daire şeklinde sol üstten sağ alta doğru mavi ve yeşil renklerden oluşan gradyan oluşturduğu gösterilmektedir:

```
div {  
background: radial-gradient(circle, blue, green);  
}
```

Çoklu Arkaplan Resimleri

CSS3 ile birlikte gelen "multiple background images" özelliği, bir elementin arka planına birden fazla resim eklemenizi sağlar. Bu özellik sayesinde, elementlerinizin arka planında birden fazla resim oluşturabilir ve bu sayede daha estetik bir görünüm sağlayabilirsiniz.

"multiple background images" Özelliğini kullanmak için, aşağıdaki şekilde belirtin:

```
element {  
background-image: url(image1.jpg), url(image2.jpg);  
background-position: top left, bottom right;  
background-repeat: no-repeat, repeat-x;  
}
```

"background-image": Bu değer, elementin arka planına eklenecek resimlerin URL'lerini belirtir. Örneğin, "url(image1.jpg)" değeri elementin arka planına "image1.jpg" resmini ekler.

"background-position": Bu değer, eklenen resimlerin hangi pozisyonda olacağını belirtir. Örneğin, "top left" değeri resmin sol üstte olacağını belirtir.

"background-repeat": Bu değer, eklenen resimlerin tekrar edip etmeyeceğini belirtir. Örneğin, "no-repeat" değeri resmin tekrar etmeyeceğini belirtir.

Aşağıdaki Örnekte, div elementinin arka planına "image1.jpg" ve "image2.jpg" resimleri ekendiği ve "image1.jpg" resmi sol üstte, "image2.jpg" resmi ise sağ alta olacağı gösterilmektedir:

```
div {  
background-image: url(image1.jpg), url(image2.jpg);  
background-position: top left, bottom right;  
background-repeat: no-repeat, no-repeat;  
}
```



SVG

CSS3 ile birlikte gelen "SVG" (Scalable Vector Graphics) özelliği, vektörel grafikleri kullanarak elementlerinizin arka planını oluşturmanıza yardımcı olur. Bu özellik sayesinde, elementlerinizin arka planında daha kaliteli, yüksek çözünürlüklü ve büyük ekranlarda da net görünen grafikler oluşturabilirsiniz.

"SVG" özelliğini kullanmak için, aşağıdaki şekilde belirtin:

```
element {  
background-image: url(image.svg);  
}
```

"SVG" özelliği, elementlerinizin arka planına yüksek çözünürlüğü ve büyük ekranlarda da net görünen grafikler eklemenize yardımcı olur. Bu özelliği kullanarak, elementlerinizin arka planında daha kaliteli grafikler oluşturabilirsiniz.

Ölçeklenebilir Birimler (vw / vh)

CSS3 ile birlikte gelen "ölçeklenebilir birimler" sayesinde, elementlerinizin boyutlarını ve pozisyonlarını daha esnek bir şekilde ayarlayabilirsiniz. Bu Özelliğin sayesinde, elementlerinizin boyutları ve pozisyonları ekranaya göre otomatik olarak değiştirilebilir ve bu sayede daha esnek bir tasarım oluşturabilirsiniz.

CSS3'te kullanılabilecek ölçeklenebilir birimler şunlardır:

"**vw**" (**viewport width**): Bu birim, ekranın yatay çözünürlüğüne göre ölçeklendirilir. Örneğin, "100vw" değeri ekranın tam genişliğini ifade eder.

"**vh**" (**viewport height**): Bu birim, ekranın dikey çözünürlüğüne göre ölçeklendirilir. Örneğin, "100vh" değeri ekranın tam yüksekliğini ifade eder.

"vmin" (viewport minimum): Bu birim, ekranın dikey ve yatay çözünürlüklerinden düşük olanına göre ölçeklendirilir. Örneğin, "50vmin" değeri ekranın dikey ve yatay çözünürlüklerinden düşük olanının yarısını ifade eder.

"vmax" (viewport maximum): Bu birim, ekranın dikey ve yatay çözünürlüklerinden yüksek olanına göre ölçeklendirilir. Örneğin, "75vmax" değeri ekranın dikey ve yatay çözünürlüklerinden yüksek olanının yüzde yetmişbeşini ifade eder.

