

Kutu Modeli - Görünüm Efektı Özellikleri

Bu aralar CSS kitabına(e-kitap) ağırlık verdim. Kitaptaki eksik bölümleri tamamlıyorum. İnşallah en yakın zamanda bitecek. Bu konuda onlardan biri.

Kutu elementlerin görünümüne çeşitli etkiler yapan özellikleri göreceğiz bu makalede çoğu CSS2 ile birlikte gelen bu özellikler aşağıda sıralanmıştır.

- **overflow** - taşma
- **clip** - kırpm
- **visibility** - görünürlük

Overflow

Yapısı: overflow: <değer>

Aldığı Değerler: visible | hidden | scroll | auto | inherit

Başlangıç değeri: visible

Uygulanabilen elementler: blok-level elementler ve replacement elementlere

Kalıtıllık: Yok

Türkçeye taşmak(overflow) olarak çevirdiğimiz bu özellik sınırlarını belirlediğimiz kutu elementinin içeriğinin sınırları aşması durumunda nasıl davranacağını bildiren bir özelliktir. Özelliğe atanan değerleri tek tek incelersek

- inherit - kalıtıllık üst elementin değerini alabilir
- visible - kutu sınırları dışına taşan alanları göster
- hidden - kutu sınırları dışına taşan alanları gizle.
- scroll - kutu sınırlarını aşan kısmı kaydırma çubuğu ile ulaşmamızı sağlar.
- auto - kutu sınırlarını aşması durumunda scroll ile aynı etkiyi yapar.

Browser Uyumu:

Internet Explorer 6+

Mozilla 1.7+

Opera 7.5+

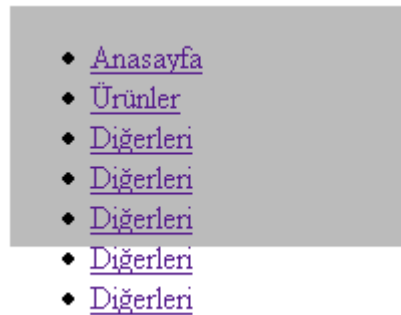
Safari 1.2

W3C's CSS Level 2+

CSS Profile 2.0

overflow:visible Overflow'ın başlangıç değeri visible'dır. Bu değer taşma olan kısımların gösterilmesini sağlar.

```
1. div#solMenu {  
2. position: absolute;  
3. top: 0;  
4. left: 0;  
5. width: 200px;  
6. height: 120px;  
7. background: #BBB;  
8. overflow: visible;  
9. }
```

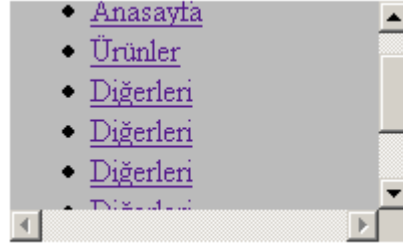


Kodun yeni nesil tarayıcılardaki görünümü

overflow özelliğinin değeri visible olduğu için içerik taşmasına rağmen taşan bölümler gösterilmiştir.

overflow:scroll Eğer içeriğin taşmasını engellemek istiyorsak ama yinede taşan kısımlara ulaşılması için kaydırma çubuğu çıkması için **scroll** özelliği ekleriz.

```
1. div#solMenu {  
2. position: absolute;  
3. top: 0;  
4. left: 0;  
5. width: 200px;  
6. height: 120px;  
7. background: #BBB;  
8. overflow: scroll;  
9. }
```

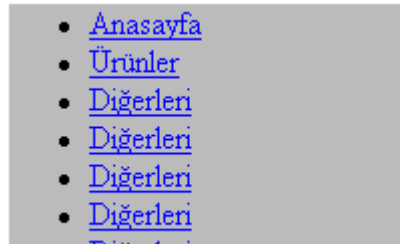


overflow:scroll olma durumu

overflow:hidden Eğer overflow:hidden değeri atarsak. Sınırladığımız kutunun dışına taşan kısımlar gösterilmeyecektir.

```
1. div#solMenu {  
2. position: absolute;  
3. top: 0;  
4. left: 0;  
5. width: 200px;  
6. height: 120px;  
7. background: #BBB;  
8. overflow: hidden;  
9. }
```

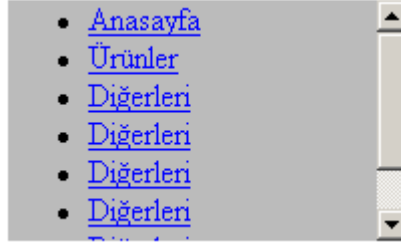
kodu bu şekilde değiştirdiğimizde görüntü aşağıdaki gibi olacaktır.



overflow:auto Son olarakta otomatik değeri vardır. Bu değeri atayınca kutu içeriğine göre otomatik işlem yapacaktır. Eğer taşma olmuyorsa kaydırma çubuğu çıkmayacaktır. Ama eğer içerik taşıyorsa kaydırma çubuğu çıkacaktır.

```
1. div#solMenu {  
2. position: absolute;  
3. top: 0;  
4. left: 0;  
5. width: 200px;
```

6. height: 120px;
7. background: #BBB;
8. **overflow: hidden;**
9. }



Dikkat ederseniz scroll'dan farklı olarak sadece dikey kaydırma çubuğu çıkmıştır. Yatayda bir taşma olmadığı için yatay kaydırma çubuğu çıkmamıştır.

Örneği görüntülemek için tıklayınız.

Safari'nin eski sürümlerinde sorun çıkartıyor Javascript ile bir çözüm var.

<http://blog.deconcept.com/2005/03/25/safari-and-links-to-elements-in-overflow-auto-content>

Operadaki sorun için ise sadece min-height ve min-width özelliklerini kullanmak yeterli.

<http://nontropo.org/test/Op7/overflowbug.html>

Bunların dışında Internet Explorer'un tescillediği ve CSS3 standartlarında yer alan iki adet özellik daha vardır. overflow-x ve overflow-y

overflow-x ve overflow-y: Internet Explorer5x ve üzeri sürümlerden itibaren desteklediği bu özellikler aslında overflow gibi çalışır. Buradaki x ve y'dende anlaşıldığı gibi overflow özelliğini yatay ve dikey olarak ikiye ayıran özelliklerdir. overflow-x özelliği yatayda taşma olması durumunda taşan kısımlara uygulanacak özellikleri(visible, scroll, hidden ve auto) atmamızı sağlar. overflow-y'de dikeyde aynı şeyi yapar. Firefox'unda 1.1 sürümünden itibaren bu özellikleri desteklemesi kullanılabilirliğini arttırmıştır. Tabi biraz daha beklemekte yarar var.

Örnekler için tıklayınız. (brunildo.org'dan)

Browser Uyumu:

Internet Explorer 5x+

Mozilla 1.8+
Opera 7.5+
Safari 3
W3C's CSS Level 3+
CSS Profile 3.0

clip

Yapısı : clip: rect(<üst>,<sağ>,<alt>,<sol>)

Aldığı Değerler : rect(top, right, bottom, left) | auto | inherit

Başlangıç değeri: auto

Uygulanabilen elementler: mutlak konumlandır uygulanmış elementler (CSS2'den itibaren blok-level elementler ve replacement elementlere)

Kalıtılabilirlik: Yok

Browser Uyumu:

Internet Explorer 4+
Mozilla 1+
Opera 3+
Safari 1
W3C's CSS Level 2+
CSS Profile 2.0

clip, elementin içeriğinin belli bir bölümünün gösterilmesi sağlar. Başlangıç değerinin **auto** atanmış olması herhangi bir kırpmanın olmayacağını gösterir.

Resim düzenleme araçlarında yaptığımız kırpma işini CSS yardımıyla yapmamızı sağlayan bu özelliği Javascript ile birlikte kullanarak çok güzel uygulamalar yapabiliriz.

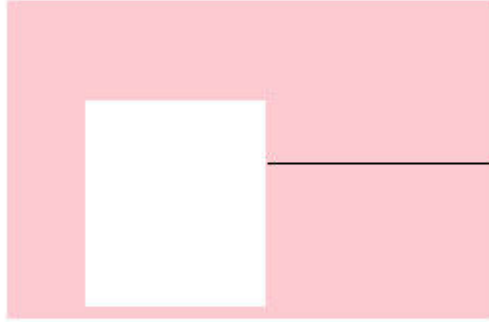
Clip'e atanan değerler **em** cinsinden verilerek esnek sonuçlar elde edilebilir. Ayrıca negatif değerlerde alabilir. Negatif değer atandığında element kutusunun dışına doğru kırpma yapacaktır.

Not

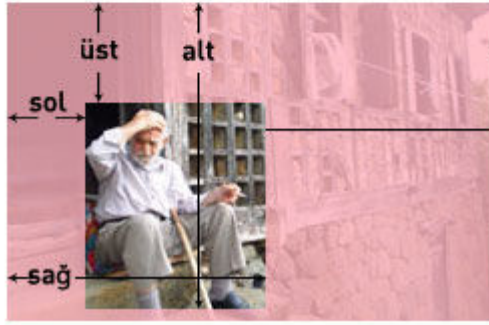
W3C standartları belirlerken konumlandırmalar arasında virgül konmasını öneriyor. Ancak garip bir şekilde IE virgül kullanılan örnekleri normal modda farklı yorumlamaktadır. Acayip modda ise doğru yorumlamaktadır. Ne yapalım bizde tüm tarayıcılara uygun olması için virgül kullanamayacağız



Ahmet emicenin resmi



Uygulanan kırpma alanı



Element

Kırılan Alan

Bir örnek yapalım.

CSS kodu:

```
1. .kırpma{  
2.   position:absolute;  
3.   clip:rect(50px 130px 153px 40px);  
4. }
```

HTML kodu:

```
1. <div class="kırpma">  
2.     
3. </div>
```



örneđi görüntüsü

Örneđi görmek için tıklayınız.

visibility

Yapısı : visibility: <deger>

Aldıđı Deđerler : visible | hidden | collapse | inherit

Başlnagıç deđerı: visible

Uygulanabilen elementler: tüm elementlere

Kalıtsallık: Yok

Browser Uyumu:

Internet Explorer 4+

Mozilla 1+

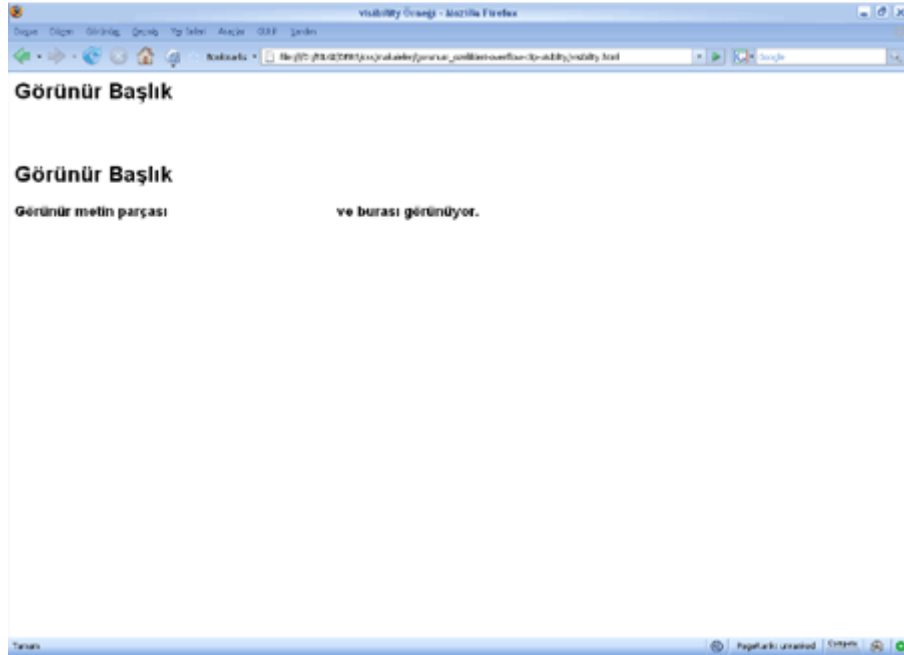
Opera 3.6+

W3C's CSS Level 2+

CSS Profile 2.0

Görünürlük(visibility) özelliđi nesnelerin görünümünü belirler. Deđerı **visible** ise element görünür. Eğer deđerı **hidden** ise element görünmezdir. Eğer bir elementin görünürlük deđerı hidden olarak atanmış ise element sayfa yapısındaki yeri korur ancak gözle görülmez.

1. <html>
2. <head>
3. <title>visibility Örneđi</title>
4. <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5. <style>
6. body {font-family: sans-serif; font-weight: bold; font-size: 14px}
7. </style>
8. </head>
9. <body>
10. <h1 style="visibility: visible">Görünür Başlık</h1>
11. <h1 style="visibility: hidden; background-color:#f00; border:1px solid #000; margin:20px 0; padding-top:10px;">Görünmez Başlık</h1>
12. <h1 style="visibility: visible">Görünür Başlık</h1>
13. <p style="font-size: large">Görünür metin parçası<b style="visibility: hidden">görünmez metin parçası ve burası görünüyor.</p>
14. </body>
15. </html>



Örneđi görmek için tıklayınız.

Örnekte gördüğümüz gibi element görünmediđi halde sayfadaki yerini korumaktadır. Ayrıca gizli nesneye atanan kenarlık ve zemin rengide görünmemektedir. Ancak elementin konumu etkileyen kenar boşluđu(margin) ve padding değeri görünmese de sayfayı etkilemektedir.

Ata elementi görünmez olan bir çocuk elemente eğer görünür özelliği atanmış ise ata element görünmez ancak çocuk element görünür. Bu ters bir durum olarak görünür ancak bu bize avantaj sağladığı durumlarda yok değildir. Örnek yaparsak

CSS kodu

1. p.gorunmez {visibility: hidden;}
2. p.gorunmez strong {visibility: visible;}

HTML kodu

1. <p class="gorunmez"> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras sem. In sagittis mauris eu turpis. Mauris ut lorem nec pede consequat suscipit. </p>



Örneği görmek için tıklayınız.

visibility:collapse özelliği vardır. collapse tablo elementler için hidden etkisi yapar. Ancak burada bir fark var gizlene elementler yerini korumaz. Ayrıca bu özelliği tüm tarayıcılar desteklemiyor, bunun yerine display:none özelliği kullanılır.

Kaynaklar

- <http://www.w3.org/TR/CSS21/visufx.html>
- <http://www.seifi.org/css/creating-thumbnails-using-the-css-clip-property.html>
- http://www.ibloomstudios.com/articles/misunderstood_css_clip/
- http://www.w3schools.com/css/pr_pos_clip.asp