

CSS DOKÜMAN



Abdullah OĞUZ



www.linkedin.com/in/abdullah-oğuz

CSS DOKÜMAN

CSS nedir ? : Sayfa üzerindeki bütün html etiketlerine uygulayabileceğimiz stillere css denir.

Görsel olarak düzenleme yapılmasını sağlar.

Inline CSS ? : Her bir HTML etiketinin üzerine tek tek uygulanması gereken bir yöntemdir.

<style> etiketi html etiketlerinin içine yazabildiğimiz ve etiket içindeki veriye renk boyut vb değişiklikler yapabildiğimiz bir özelliktir.

```
<p style="color: blue ; font-size: 16px">Sıfırdan Web Developer Kursu</p>
```

renk düzenleme fontunu değiştirme

Noktalı virgülü birden fazla özellik ekleyecekken kullanıyoruz.

```
<h1 style="color: red">Web Programlama</h1>

<h1 style="color: red ; font-size: 250%">Web Programlama</h1>
```

Web Programlama → H1 ETİKETİ

Web Programlama → H1 ETİKETİNİN %250 DAHA BÜYÜK HALİ

İkinci örnekte uygulanan değer normal h1 etiketinin font değerinin %250 daha büyük halidir yani normal h1 etiketinden 2,5 kat daha büyük bir yazı oluşturduk.

```
<p style="text-align: right">Web Api Eğitimi : Javascript Uygulamaları</p>
```

metin hizalama

Eğer metni sağa doğru hizalama style özelliği

Internal CSS : aynı etikete sahip html sayfasındaki ne kadar içerik olursa olsun aynı özellikleri uygulamak.

Internal css <head> etiketi içine tanımlanır.

```
<style type="text/css">
  p {
    color: darkslategrey;
    font-size: 20px;
    text-align: center;
  }

  h1 {
    color: brown;
    font-size: 36px;
    text-align: justify;
  }
</style>
```

Body etiketi içerisinde bulunan <p> ve <h1> etiketlerinin hepsine süslü parantez içindeki özelleştirmeleri uygular.

Sıfırdan Web Developer Kursu

Web Programlama

Web Programlama

Web Programlama Eğitimi : Asp.net MVC

Web Api Eğitimi : Javascript Uygulamaları

Eğer bu yazılanların içinden farklı özellik uygulamak istediklerimiz varsa inline css yöntemini kullanmamız gerekecek.

Sıfırdan Web Developer Kursu

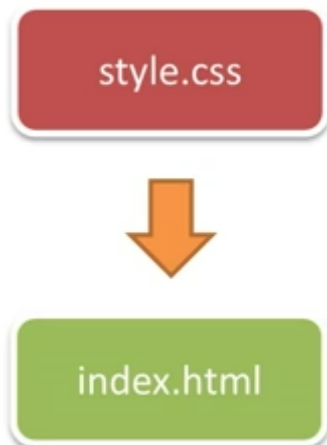
Web Programlama

Web Programlama — inline css kullanılarak yapıldı

Web Programlama Eğitimi : Asp.net MVC

Web Api Eğitimi : Javascript Uygulamaları

External CSS: Bütün css kodlarını ayrı bir css uzantılı bir dosyada toplamaktır.



Birçok html dosyasına ait css kodları da tek bir css dosyası altında toplanabilir.

style.css

referans.html

index.html

urun.html

____.css uzantılı dosyanın içinde yazdığımız css kodlarını html uzantılı dosyamızda uygulamak istersek

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

Eğer birden çok css uzantılı dosyamız varsa href içindeki bölümde belirtmemiz gerekiyor.

Her sayfaya özgü de css dosyası yazılabilir.

Eğer biz tanımlanmış bir özelliği başka etiketler içinde kullanmak istersek “class” kullanarak css yazmış oluruz.

CLASS:

- İlk <style> etiketi içinde class'ın özellikleri yazılır.
- Sonra kullanılacak etiket içine class ismi yazılarak kullanılır.

Belli bir özellik tanımlanır ve bu özelliği başka etiketlere vererek o etiketin üzerinde görsel olarak değişiklik yapmış oluruz.

<style> etiketinin içinde tanımlanır.

Bu şekilde tanımlanır:

```
<style type="text/css">
    .red{
        color: red;
    }
</style>
```

Bu şekilde etiket içinde kullanılır:

```
<p class="red">Web Geliştirme Eğitimi</p>
```

İki adet class tanımlaması yapılacaksa şu şekilde olur

```
<p class="red large">Web Geliştirme Eğitimi</p>
```

ID:

- İlk etiket içinde id'nin ismi tanımlanır.
- Sonra <style> etiketi içinde id'ye verilecek özellikler yazılır.

Etikete belli bir id verilerek o id üzerinden <style> etiketi içinde #id_ismi olacak şekilde belli özellikleri id'ye atayabiliriz.

Id'yi etiket içinde kullanma:

```
<h1 id="header">Ana Başlık</h1>
```

Id'yi <style> etiketi içinde tanımlama:

```
#header{  
  font-size: 300%;  
  border-bottom: 1px solid black;  
  color: red;  
}
```

Eğer biz belli bir etiketin yarısında farklı bir görsel düzenleme yapmak istersek ?

```
<p class="large">Sıfırdan WEB <span class="red underline">developer kursu</span></p>
```

Farklı özellik uygulamak istediğimiz bölgede span etiketi kullanıp içerisine class açarsak istediğimizi yapmış oluruz.

SELLECTORS:

Elements: element selector'u uygulayacağımız etiketin ismini yazıp etiket üzerinde yapmak istediğimiz değişikliği süslü parantezler içine yazılarak gerçekleştirilir.

```
/* Elements */  
  
li{  
  border: solid 1px;  
}
```

Class:

HTML etiketlerine verilen özelliktir. Sınıflara bölünür. ".class_adı" şeklinde tanımlanıp süslü parantezlerin içine yazmak istediğimiz kodlar yazılır.

<h3 class “active”> şeklinde tanımlanır.

```
/* Class */  
  
.active{  
    background-color:  
    aqua;  
}
```

ID: Belli bir etikete id tanımlanır ve o id üzerinde style.css dosyası üzerinde özellik verilir.

işareti ile tanımlama yapılır.

ID sadece 1 etiket üzerinde kullanılabilir.

Class’taki gibi bir çok etiket üzerinde kullanılamaz. Class’a göre daha özel.

```
/* id */  
  
#header{  
    font-size: 200%;  
}
```

Descendant:

```
/* Descendant */  
  
h3 a{  
    color: red;  
}
```

Eğer sadece a etiketi içindeki renk değerler düzenlemeleri yapılırsa tüm a etiketlerine uygulanır. Descendant tipi selectorun amacı h3 içindeki a etiketlerine css kodlarını uygulamak.

Aynı şekilde class içindeki p etiketine de ulaşabiliriz.

```
.container p{  
  
    color: blue;  
}
```

Ve class içindeki başka bir class’a da ulaşabiliriz

```
.container .box{  
    color: yellow;  
}
```

BURADA ŞU AYRIM ÖNEMLİ :

```
.container div.box{  
    border: 1px solid;  
    color: crimson;  
}
```

container olan div lerden classı box olanları al

```
.container div .box{  
    border: 4px solid;  
    color: darkgreen;  
}
```

Container olan divlerin içindeki etiketlerin classı box olanları al

```
div,p{  
    color: aqua;  
}
```

Sayfadaki bütün div ve p etiketlerine aşağıdaki özellikleri uygular.

```
div>p{  
    color: bisque;  
}
```

P etiketinin bir üst elemanındaki divleri seçer.

Attiribute sellectors:


```
p[title]{  
  
    border: 1px solid;  
    color: crimson;  
  
}
```

p etiketindeki attribute özelliği title olanlara yazılan özellikleri uygular.

```
p[title="second"]{  
    background-color: darkblue;  
}
```

p etiketinin içindeki title ismi “second” olan etiketlere verilen özellikleri uygular.

```
a[href="https://www.example.net"][target="_blank"]{  
  
    background-color: gold;  
}
```

Yukarıdaki attribute değerlerinin ikiside sağlarsa verilen özelliği geliştirir.

Pseudo Selectors:

```
p:first-line {  
    font-weight: bold;  
    word-spacing: 2px;  
    letter-spacing: 1px;  
    font-size: 14px;  
    color: aqua;  
}
```

p etiketlerinin **ilk satırı** için parantezler içindeki özellikleri gerçekleştirir.

```
p:first-letter {  
    font-size: 48px;  
}
```

P etiketleri içindeki **ilk harfi** için geçerli özellikleri yapar.

```
p.about:first-line {  
  
    color: red;  
}
```

P etiketlerine uygulanan “about” classının ilk satırı için geçerli özellikleri yapar.

- Sadece p etiketi olması şart değil div,span vb. etiketler içinde de yapılabilir.

```
ul li:first-child {  
    background-color: palevioletred;  
}
```

Ul etiketinin içindeki li etiketlerinin **ilk elemanı** için geçerli özellikleri yapar.

```
ul li:last-child {  
    background-color: green;  
}
```

Ul etiketinin içindeki li etiketlerinin **son elemanı** için geçerli özellikleri yapar.

```
ul li:nth-last-of-type(odd) {  
    border: 1px solid;  
    color: dodgerblue;  
}
```

Ul etiketinin içindeki li etiketindeki TEK sayılara bu özelliği uygular.

```
ul li:nth-last-of-type(even) {  
    border: 1px solid;  
    color: goldenrod;  
}
```

Ul etiketinin içindeki li etiketindeki ÇİFT sayılara bu özelliği uygular.

```
a:link{  
    color: red;  
}
```

a etiketindeki linke tıklandıktan sonra link rengini kırmızı yapar.

```
a:visited{  
    color: blue;  
}
```

A etiketindeki linke tıklandıktan sonra daha önce tıklandığını göstermek amacıyla linkteki yazıyı maviye boyar.

```
a:hover{
  color:white;
  background-color: green;
}
```

Fare işaretçisi ile sayfa üzerinde gezinirken a etiketinin tanımlandığı yerin üzerine fare işaretçisi ile gelindiğinde o etiket üzerinde gösterilecek değişimler ortaya çıkar.

```
a:active{
  color: black;
  background-color: gold;
}
```

A etiketi üzerindeki linke tıklandığı zaman o anda gösterilecek değişimler ortaya çıkar.

```
p:after{
  content: " - deneme "
}
```

CONTENT İÇİNE YAZILAN İFADELER SAYFA ÜZERİNDE KOPYALANAMAZ. BİR NESNE GÖREVİ GÖRÜYOR

P etiketinin content(içerik) “-deneme” yazarsak p etiketinde yazan ifadeden sonra gelir.

```
p:before{
  content: " - deneme ";
  display: block;
}
```

- deneme
Lorem ipsum dolor.

Before yaparsak “ - deneme” contenti’ni bu sefer baş tarafa ekler.

Display:block dersek “ – deneme” nesnesini p etiketinden önce konumlandırır. Ve o nesneyi blok satırı yapar. O satırı tamamen kaplar. Ve p etiketi bir alt satırdan devam eder.

Star:

Star selectoru uygulanan css kodunu sayfanın tümünde uygular.

```
/* Star */  
  
*{  
    border: 1px solid #eee;  
}
```

1px boyutunda tüm sayfayı çizgilere böl

Komple Uygulamalı Web Geliştirme Eğitimi

Front-end Development

- Html
 - [Bölüm Tanıtımı](#)
 - [İlk Web Sayfam](#)
- Css
- Javascript
- Bootstrap4
 - Bölüm Tanıtımı
 - Grid System
 - a. Ders 1
 - b. Ders 2
- JQuery

Back-end Development

1. [Asp.net MVC Başlangıç](#)
2. [C# Programlama](#)
3. [Asp.net MVC Orta Seviye](#)
4. [Veritabanı Temelleri](#)
5. [Entity Framework](#)
6. [Linq](#)
7. [Asp.net MVC ile Entity Framework](#)
8. [Asp.net Identity](#)
9. [Asp.net Web Api](#)

Attiribute:

Attiribute değerlerinde değişiklik yapmak istersek şu şekilde yapılır.

```
a[href="http://www.udemy.com"]{  
    color: red;  
}  
a[href="http://www.edurive.com"]{  
    color: olive;  
}
```

A etiketinin href attirbute özelliğinin rengini kırmızı yapar.

```
input[type="checkbox"]  
{  
    width: 100px;  
    height: 100px;  
}
```

Form nesnelerinin özellikleri içinde de düzenleme yapabiliriz.

NTH:

Sıralanmış özelliklerin sadece 1 tanesini seçmek için kullanılır.

```
ol li:nth-child(3){  
    background-color: olive;  
}
```

Ol sıralı etiketinin liste elemanını 3. Elemanını arka plan rengini zeytin yap.

Eğer 3 yerine “ even “ kelimesi gelirse çift elemanları alır.

Sellectors Öncelikleri:

Genelden özele gidilirken özel olan sellctors aktif oluyor diğeri pasif oluyor.

Eğer bir css koduna “ !important “ yazarsak o kod her zaman etkin olur sellectors üzerinden değişime uğramaz.

Body>ul>li>.active(class) > #passive (ID)

Tanımlanan css değerleri

```
body
{
  color: blue;
}

ul
{
  color: red;
}

li
{
  color:purple;
}

.active
{
  color: orange;
}

#passive
{
  color: gray;
}
```

Uygulanan css kodları:

```
<ul>
  <li id="passive" class="active">Html5</li>
  <li>Css</li>
  <li class="active">Javascript</li>
  <li>Jquery</li>
</ul>
```

Html5 yazan kısımda hem class hemde id sellectorleri uygulanmış fakat genelden özele doğru gidişat olduğu için id'nin özellikleri geçerli olur.

INHERİT

<a> etiketi sadece kendisine verilen css işlemlerini gösterir bir üst etiketteki özellikleri kendine almaz.

```

<ul>
  <li>Html5</li>
  <li>Css</li>
  <li>Javascript</li>
  <li>
    <a href="#">Jquery</a>
  </li>
</ul>

```

Eğer biz etiketine bir özellik atarsak <style> etiketi içinde bu özellik etiketleri içinde yazan Html5 vb. gibi mestinsel ifadelerle yansır fakat <a> etiketine yansımaz bu a etiketinin bir özelliğidir.

Eğer bir a etiketine bir özellik vermek istersek.

Css leri bir üst etiketten bir alt etikete aktarabiliriz

Eğer <a> etiketi için ayrı bir kod yazmak istemiyorsak a etiketi alacağı özelliklerin aynısını bir üst etiketten alsın istiyorsak.

```

li{
  color:firebrick;
}

a{
  color: inherit;
}

```

Bu şekilde yapmamız gerekiyor.

Eğer bir css özelliğini class , id gibi sıralamaların (bir etiketin diğer etiketteki özelliği ezmesi) önüne çekmek istersek “ !important ” özelliğini kullanmalıyız.

```

.orange{
  color: orange !important;
}

```

Etiket sıralaması olarak ezilse bile “ .orange ”class’ı etkin olur.

DİV'S

Sayfa tasarımlarında bir bölme oluşturmak istediğimizde <div> etiketlerini kullanıyoruz.

Etiketleri bölümllemek istediğimizde kullanabiliriz.

```
<div>|
  <p>Lorem ipsum dolor sit amet.</p>
  <p>Lorem ipsum dolor sit amet.</p>
  <p>Lorem ipsum dolor sit amet.</p>
  <p>Lorem ipsum dolor sit amet.</p>
  <p>Lorem ipsum dolor sit amet.</p>
</div>
```

```
<p>Lorem ipsum dolor sit amet.</p>
<p>Lorem ipsum dolor sit amet.</p>
<p>Lorem ipsum dolor sit amet.</p>
<p>Lorem ipsum dolor sit amet.</p>
<p>Lorem ipsum dolor sit amet.</p>
```

İki <div> etiketine ayrı ayrı ulaşmak istersek class veya id vererek ulaşabiliriz. Fakat id'yi class gibi her etikete veremeyiz. Sadece bir etikete verilebilir.

```
#selection-one{
  color: blue;
  background-color: aquamarine;
  width: 50%;
  height: 100px;
```

Buradaki kodların anlamı :

Rengi mavi yap

Arkaplan rengini aquamarine yap.

Genişlik sayfa boyutunu %50 si kadar olsun.

Yükseklik 100px yap.

Eğer arka planın yüksekliğe bir değer verdiğimizde ve o arkaplan rengi yazı boyutundan kısa kalıyorsa

Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet.

Aquamarine rengi 100px fakat o div içindeki p etiketleri arka plan rengi dışında kalmış.

Eğer yine arka plan rengi 100px de kalsın fakat div içindeki yazılarda da taşma olmasın diye bir css özelliği ("overflow") kullanabiliriz.

```
#selection-one{
  color: blue;
  background-color: aquamarine;
  width: 50%;
  height: 100px;
  overflow:scroll;
}
```

O div içindeki taşan yazıları scroll bar içine alır ve hem arka plan rengi 100px te kalış olur hem de yazılarda taşma olmaz.

Eğer

overflow : hidden ;

yapılırsa taşan yazılar kaybedilir.

Eğer

Overflow= auto;

Yapılırsa eğer taşma varsa scrollbar ile taşma engellenir. Taşma durumu yoksa hiçbir şey yapılmadan bırakılır.

COLOR'S

1-Biz belli bir metine , kenarlığa , butona belirli renkler veririz. Ve bu renklerin oluşumunda rgb renkleri kullanılır (rgb= Right ,Green ,Blue) bu renk değerlerini css kodu içinde de verebiliriz.

```
h2{
  color: rgb(180, 107, 100);
}
```

2- İstedğimiz renkleri hexadecimal olarak ta yazarsak istediğimiz rengi elde etmiş oluruz.

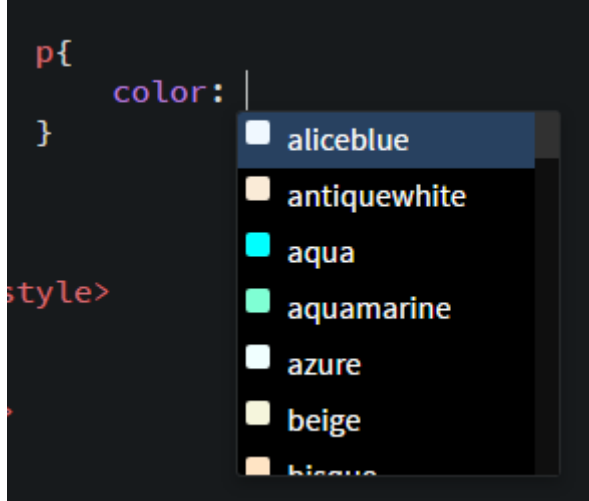

```
h2{  
    color: #b46b64;  
}
```

3- Rgba:

Rgba rgb yi zaten biliyoruz. RGBA ise sonunda bulunan a opaklığı temsil eder .

a değeri 0-1 arasında değer alır.

4-Son olarakta text editörün kendisinde kayıtlı olan renklerden de seçim yapabiliriz



Renk kartelası ve hexadecimal kodları için;

https://www.w3schools.com/colors/colors_picker.asp

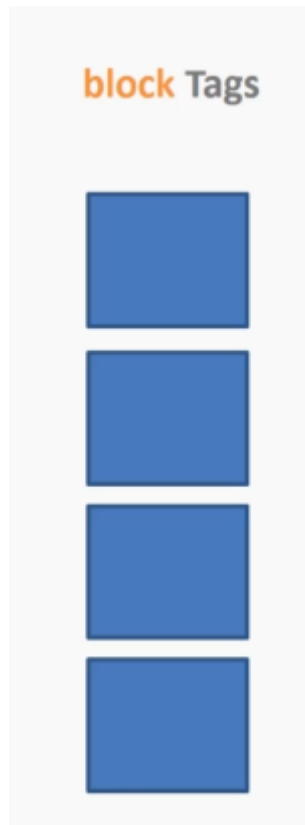
https://www.w3schools.com/cssref/css_colors.asp

FLOATİNG İMAGES

İnline tags dediğimiz etiketler sayfada şu şekilde kullanılır. (<a> , etiketleri)



Block etiketler ise şu şekilde konumlanır. (<p> , <div> etiketleri)



Herhangi bir nesneyi sayfa üzerindeki konumunu değiştirmek istersek bunu floating kavramı ile yapıyoruz.

Float vererek resmin konumunu belirleyebiliyoruz.



```
<p>  
    
  Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quos aliquam repellat soluta rerum  
  inventore dolorem dolorum qui aliquid cumque. Minus aliquid reiciendis ducimus, suscipit veritatis  
  sequi voluptate et accusamus explicabo.  
  Lorem ipsum dolor sit amet, consectetur adipisicing elit. Soluta nisi magnam debitis eum provident!  
  Consequatur, placeat asperiores esse atque! Officiis et debitis sint nam recusandae ipsum adipisci  
  beatae vitae atque.  
  Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quam dignissimos illo placeat, repellat  
  sunt aliquid ab, saepe quaerat natus deleniti culpa numquam totam, enim molestiae alias unde  
  officia quisquam veniam.  
</p>
```

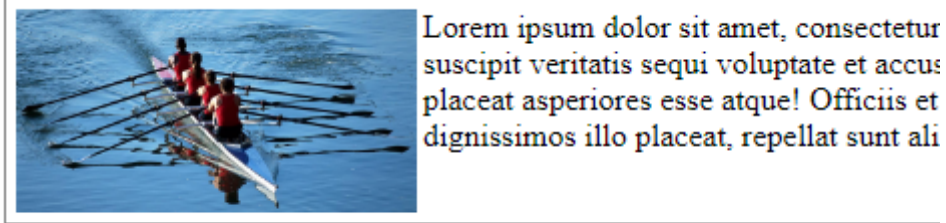
Kodu bu şekilde aynı p etiketi içinde kullandık

Fakat resim dışarı taştı float kullandığımız için eğer kullanmasak resim üst tarafta kalırdı.

Biz resimin taşmasını istemezsek div konusunda da bahsettiğimiz overflow özelliğini kullanacağız

```
<p class="close">
```

Close class'ına `overflow=auto;` dersek taşmalar olmaz çerçeve resimin altına gelir.



FLOATING DIV'S

Resimlere sayfa üzerinde konumlama yapabildiğimiz gibi div etiketlerine de konumlamalar yapabiliriz.

Örneğin:

Divlerin içinde 170px genişlik ve 170 px yükseklik değişik arka plan renklerinde 3 adet kutu ekledik ve bu kutular ile floating div's kavramını anlamaya çalışacağız.

Kodları görelim

```

<style>
  #container {
    background-color: gray;
  }

  .box {
    width: 170px;
    height: 170px;
    margin: 5px;
  }

  #first {
    background-color: red;
    float: left;
  }

  #second {
    background-color: blueviolet;
  }

  #third {
    background-color: brown;
  }

</style>

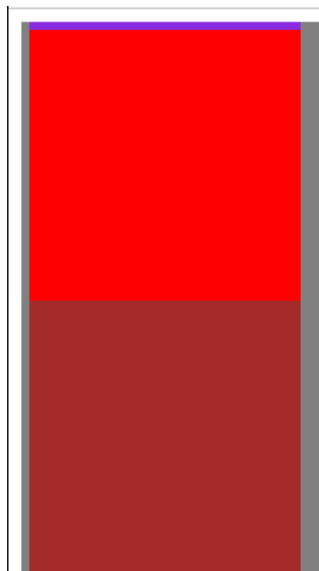
</head>

<body>
  <div id="container">
    <div id="first" class="box"></div>
    <div id="second" class="box"></div>
    <div id="third" class="box"></div>
  </div>

```

#first id sini uygularsak float değerini left yaptık ve ilk kutu sol tarafa konumlandı.

Hemen kırmızı rengin arkasına mor rengi ve onun altına da kahve rengi konumlandı.



Eğer biz #first id is ile birlikte #second id sine de floating : left; yaparsak

Kırmızı ve mor rengi yan yana dizilir. Kırmızının da arkasına kahve rengi konumlanır

1-Neden kahve kırmızının arkasına konumlandı ?

Çünkü biz iki id'ye floating özelliği ile sol tarafa doğru konumladık fakat #third id'si için hiçbir şey vermedik bu yüzden #third olan kutucuk herhangi bir konumlama özelliği olmadığı için arka tarafta durdu.

Eğer 3 id ye de floating:left; dersek üçünü de sola yaslamış olur.



Biz floating işlemini devreye soktuktan sonra arka plan gri işlemi kayboldu. Dış “container” id’li div in yüksekliği o div’in içinde bulunan divlerin yüksekliği ile alakalı değil. İçeride bulunan divlerin akışı başka bunu tekrardan devreye sokmak için daha öncede gördüğümüz overflow işlemini yapacağız.

```
<body>
  <div id="container" class="clear" >
    <div id="first" class="box"></div>
    <div id="second" class="box"></div>
    <div id="third" class="box"></div>
  </div>
```

Container id’sinin içinde ki clear class’ı (overflow = auto)

Container id sine overflow vererek arka planın gri olmasını sağladık.

Arka planın gri olmasını bir başka yolla da yapabiliriz.

```
.clearfix:after{
  content: "";
  display: block;
  clear: both;
}
```

Bu kodları büyük div içindeki class a verirsek yine arka planı gri yapmış oluruz.

```
<div id="container" class="clearfix">

  <div id="first" class="box"></div>
  <div id="second" class="box"></div>
  <div id="third" class="box"></div>

</div>
```

My first blog post

10.10.2018

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Doloribus ea voluptate qui aliquid est, illum quaerat aut! Dolore qui doloremque culpa suscipit harum molestiae commodi fugiat. Natus hic quos consequuntur.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Doloribus ea voluptate qui aliquid est, illum quaerat aut! Dolore qui doloremque culpa suscipit harum molestiae commodi fugiat. Natus hic quos consequuntur.



Caleb Fields

Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet.

Sayfa içerisinde kırmızı çerçeve ile ilgili bir görüntü bozukluğu var bunu düzeltmek için ekstra bir div ekleyip clear özelliğine both verememiz gerekiyor. Clear : both : hem sağa hem de sola yaslar.

My first blog post

10.10.2018

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Doloribus ea voluptate qui aliquid est, illum quaerat aut! Dolore qui doloremque culpa suscipit harum molestiae commodi fugiat. Natus hic quos consequuntur.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Doloribus ea voluptate qui aliquid est, illum quaerat aut! Dolore qui doloremque culpa suscipit harum molestiae commodi fugiat. Natus hic quos consequuntur.



Caleb Fields

Lorem ipsum dolor sit amet.

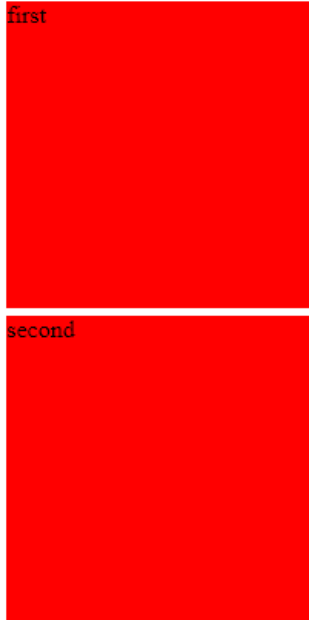
Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet.

DISPLAY

Blok etiketler o satırı kaplayan etiketlerdir. Birden çok blok etiket eklendiğinde her eklenen etiket için bir alt satıra geçer. (<div> ,...)



İnline etiketler ise blok etiketler gibi satırı kaplamaz. Birden çok inline etiket eklendiğinde aynı satırdan devam eder. (<a> , )

deneme 1 deneme 2 deneme 3 ➔

[Home](#) [Galery](#) [Contact](#) [About](#) ➔ <a>

İnline etiketlerde yazının çevrelediği kısım kadar arka planı rengi (span etiketine background-color verilmiştir) olur.

Span etiketine width: 500px dersek inline etiket olduğu için sayfaya etki etmez.

Eğer **inline** etiketlere **genişlik ve yükseklik** vermek istersek “ display “ özelliğini kullanacağız.

display : block ; eğer birden fazla etiket içinde yazılan değer varsa her birini bir satıra yerleştirir.

```
<span>deneme 1</span>  
<span>deneme 2</span>  
<span>deneme 3</span>
```

Normalde span etiketlerine gelen ifadeler inline olduğu için sayfa üzerinde tek bir satırda gösterilirken span etiketlerine `display = block` ; değerini verirsek

```
deneme 1  
deneme 2  
deneme 3
```

Resimdeki hali alır ve deneme1 in yazdığı yer tüm satırı kaplar. Display block diyerek display uygulanan etiketlere genişlik ve yükseklik verebiliriz.

Eğer biz inline etiketler gibi aynı satır içinde yer kaplasın dersek ama biz bu etiketlere genişlik ve yükseklik verelim dersek. Display = inline-block ; özelliğini kullanacağız.

```
<span>deneme 1</span>  
<span>deneme 2</span>  
<span>deneme 3</span>
```

Span etiketlerinin `display=inline-block`; özelliğini uyguladığımızda birden fazla eklenen span etiketleri hem aynı satır içinde bu etiketlere genişlik ve yükseklik verilebilecek.

```
deneme 1deneme 2deneme 3
```

Gerekli düzenlemeler yapıldığında bu şekilde bir görüntü sağlanabilir.

Span etiketleri için söylediğimiz display özelliklerinin aynısı `<div>` etiketleri içinde geçerlidir.

- Genişlik ve yükseklik verdiğimiz bir div etiketine display özelliğini inline yaparsak içindeki metnin genişliği kadar yer kaplar.
- Display = inline-block; yine aynı satır içinde bulunurlar fakat genişlik ve yükseklik verilebilir.
- Div etiketlerinde tarayıcı varsayılanı olarak display = block olarak gelir. Genişlik ve yükseklik verilebilir.

Positioning

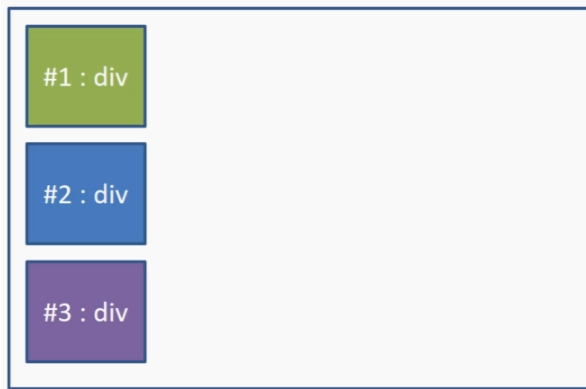
İstedığımız bir nesnenin sayfa üzerindeki konumunu belirtebiliyoruz. Float işleminde nesneleri sağa ya da sola hizalayabiliyorduk fakat nesneler arasındaki mesafeyi ayarlayamıyorduk.

Position kavramı ile nesneler arasındaki mesafeyi de ayarlayabiliyoruz.

Position'un 5 adet kavramı vardır bunlar;

- Static
- Relative
- Fixed
- Absolute
- Sticky

Position: static:



3 div sayfada birbiri ardına gelir ve sayfada bu şekilde konumlanır. Çünkü blok elemanlar blok elemanlar alt alta gelirler.

Biz bunların pozisyonlarıyla oynamak istersek ilk özelliğimiz relative :

Position :relative :

Örneğin css position özelliğine relative özelliğini verdiğimizde o nesne için 4 köşeden hizalama yapabiliyoruz.

- Left : soldan
- Right : sağdan
- Top : üstten
- Bottom : alttan



3 adet div var ve ben “ #2 “ div’i nesnesinin pozisyonu 30px yukarıdan 30px soldan hizalayacağız.

```
#second{  
  background-color: dodgerblue;  
  position: relative;  
  left: 30px;  
  top: 30px;  
}
```

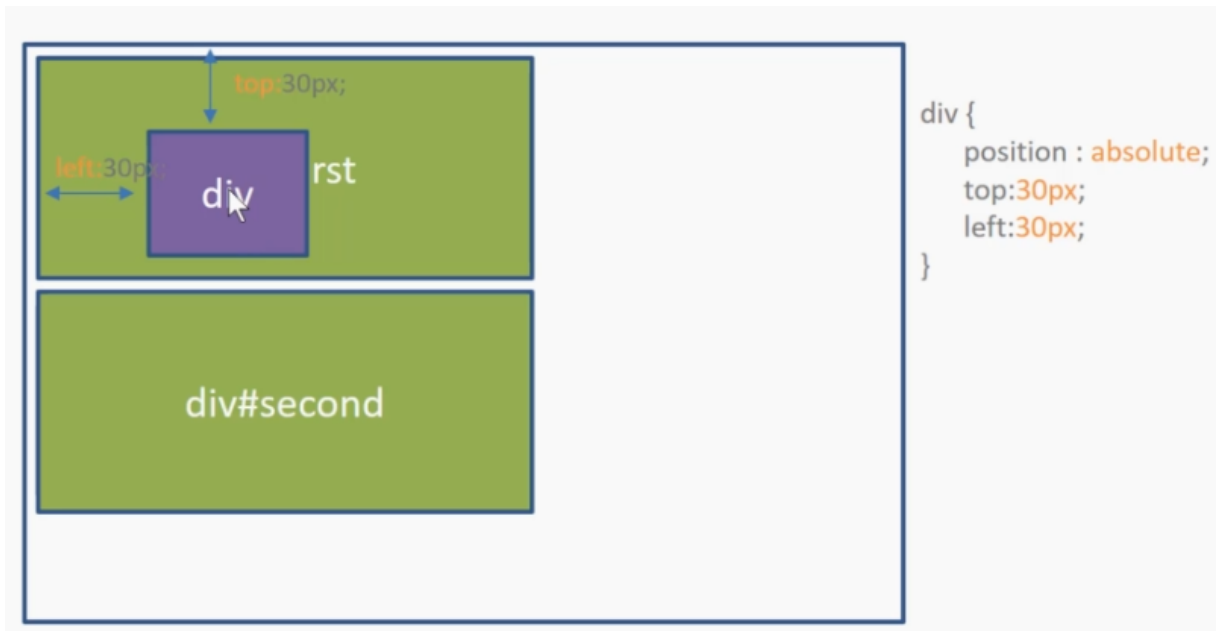
Position’un cinsini yazıldıktan sonra sağ sol hizalamaları altına yazıyoruz.

Relative özelliğinde normal akışın üstünden düzenleme yapabiliyoruz.

Position : absolute :

Position : absolute özelliğini uyguladığımız div normal akışın dışına çıkar ve sayfanın en üstünden hizalanmaya başlar.

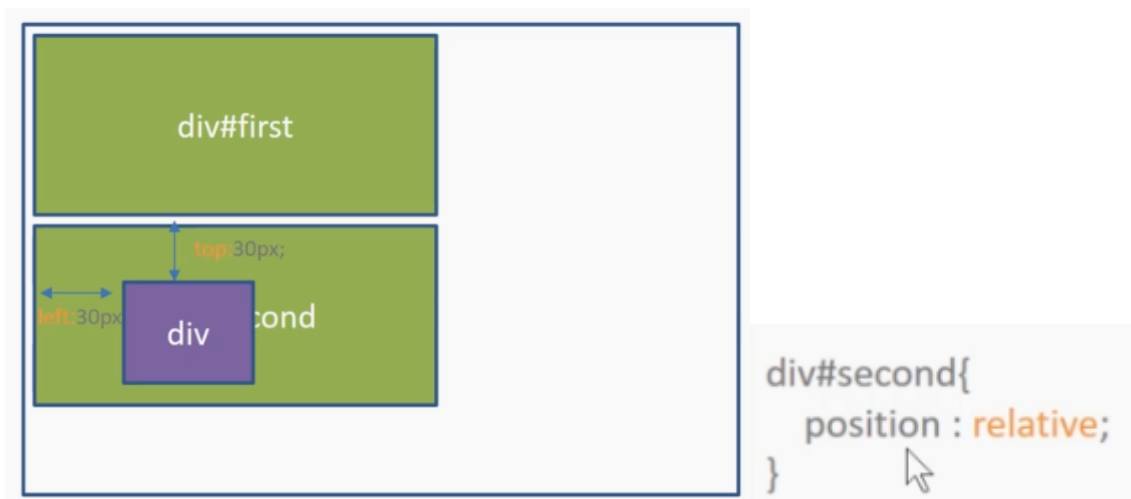




Şu şekilde iki adet div olsun ve 1.div'in içine bir div açtık ve bu dive yandaki position değerlerini uyguladığımızda div çerçevesindeki sınırlar üzerinden değil sayfanın sınırları ile işlem yapar.

Aynı nesne 2. Div içine de konumlanırsa yine 2. Divin sınırları içinde değil sayfa sınırları ile işlem yapılır.

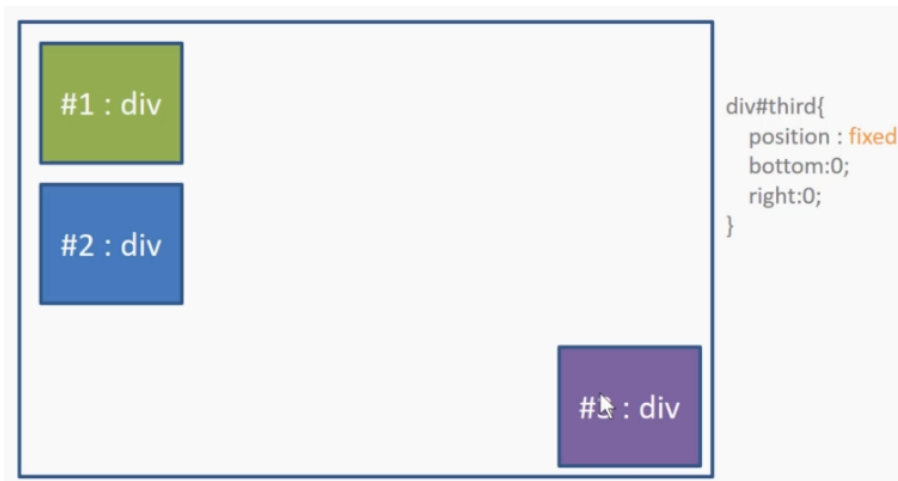
Position : relative :



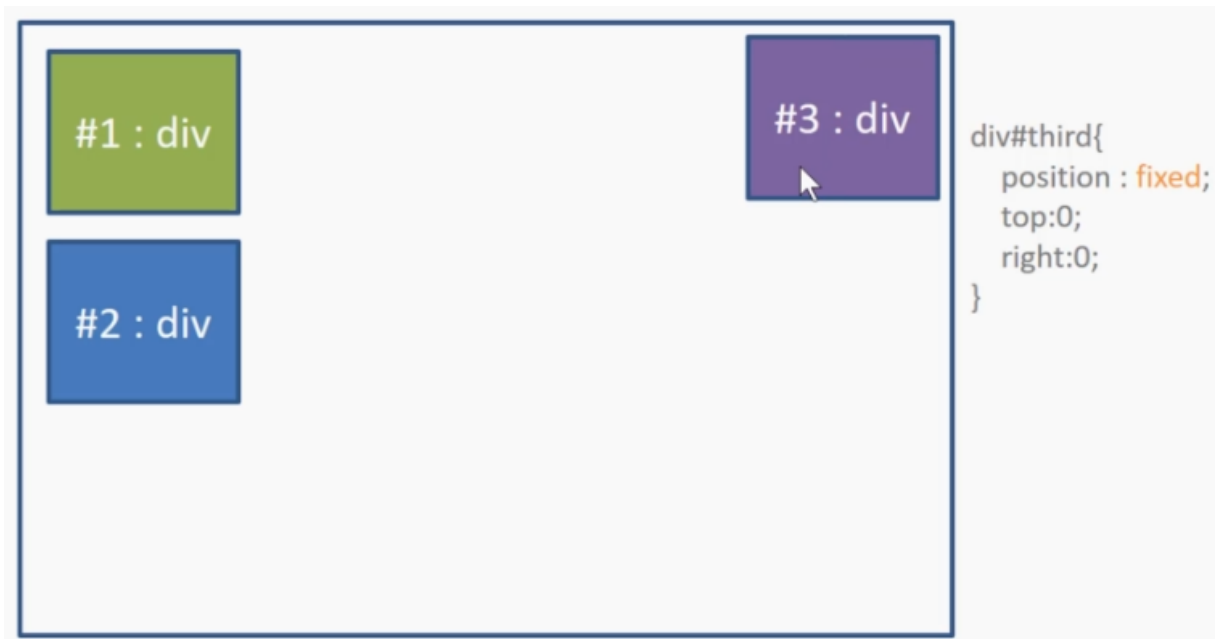
Eğer biz bu durumda position özelliğine relative verirsek. İkinci div'in sınırları içinden işlem yapar.

Position : fixed : position fixed olduğunda yine 4 köşeden hizalama yapılabilir.

En önemli özelliği scroll bar ile aşağıya indirdiğimizde sayfa üzerinde hala görünmeye devam eder.

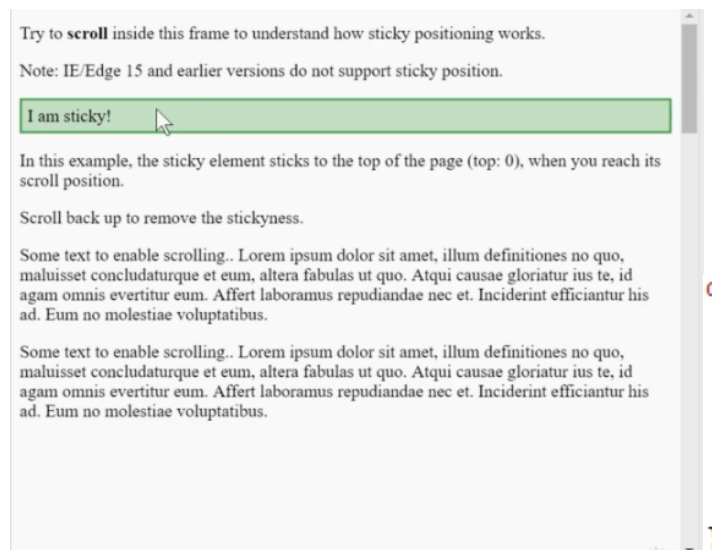


Verilen özellikler uygulandığında bottom : 0 ve right 0 olduğunda sayfanın sağ köşesine yapışır ve scroll bar ile sayfayı aşağıya indirsek bile scroll bar ile sürekli görünür hale gelir.



Top: 0 ve right : 0 olduğunda sağ üst köşeye yapışık olarak bir div gelir ekrana ve scroll bar ile kaydığımızda o div sürekli sağ üst köşede görünür.

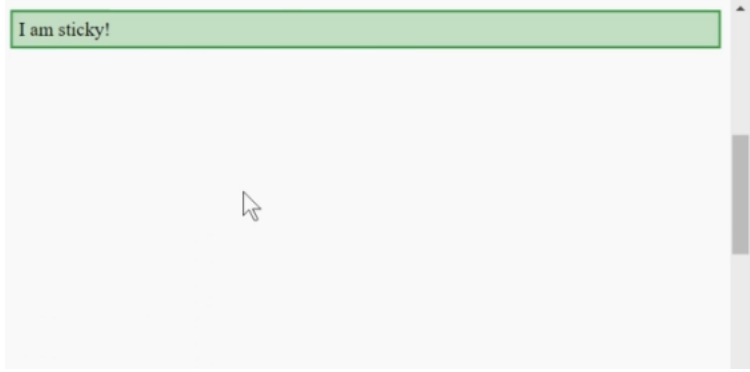
Position : sticky :



Bu ekranda “ı am sticky “yazan div scroll bar aşağıya indiğinde de birlikte aşağıya iner.Üst sayfada verilen özelliklerde top : 0 yani scroll bar indirilirken sticky kısmı yukarıdan 0px olduğu anda scroll bar ile birlikte aşağıya iner ve sayfanın üstünde kalmaya devam eder.

Eğer top : 20px olsaydı. Scroll bar ile indirirken sticky nin uygulandığı div sayfanın en üstü ile div arasında 20px lik bir mesafeden sonra hep sayfanın üstü ile div arasında 20 px lik bir mesafe farkı ile aşağıya iner.

Sticky özelliğini daha çok üst menü yapımında kullanılıyor.



Örnek :

My first blog post

10.10.2018

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Doloribus ea voluptate qui aliquid est, illum quaerat aut! Dolore qui doloremque culpa suscipit harum molestiae commodi fugiat. Natus hic quos consequuntur.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Doloribus ea voluptate qui aliquid est, illum quaerat aut! Dolore qui doloremque culpa suscipit harum molestiae commodi fugiat. Natus hic quos consequuntur.

My first blog post

10.10.2018

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Doloribus ea voluptate qui aliquid est, illum quaerat aut! Dolore qui doloremque culpa suscipit harum molestiae commodi fugiat. Natus hic quos consequuntur.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Doloribus ea voluptate qui aliquid est, illum quaerat aut! Dolore qui doloremque culpa suscipit harum molestiae commodi fugiat. Natus hic quos consequuntur.



Caleb Fields

Şekilde verilen tarihleri kendi div'i içindeki en sağa hizalayalım

My first blog post

10.10.2018

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Doloribus ea voluptate qui aliquid est, illum quaerat aut! Dolore qui doloremque culpa suscipit harum molestiae commodi fugiat. Natus hic quos consequuntur.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Doloribus ea voluptate qui aliquid est, illum quaerat aut! Dolore qui doloremque culpa suscipit harum molestiae commodi fugiat. Natus hic quos consequuntur.

div.blog-post | 855x170, 91

Div sınırları bunlar.

Bizim bu işlemi yapabilmemiz için iki şeye ihtiyacımız var

1-) tarihlere bir class açıp onları absolute olarak hizaladım top: 0 ve right : 0 olmak üzere.

2-) ve son olarak bu iki yazının yazıyı çevreleyen blog post a bir class oluşturduk ve bu class'a position özelliği olarak relative verdik

Tarihlere position olarak absolute ve blog post'a ise relative özelliğini vermemizin sebebi

İlk tarihler sayfanın sınırlarına göre hizalanır sonra biz bu sınırlamayı blog posta verdiğimiz relative özelliği bile blog post div'i içinde yaparız

Sonuç olarak tarihler blog post divi referans alınarak hizalanır.

My first blog post

10.10.2018

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Doloribus ea voluptate qui aliquid est, illum quaerat aut! Dolore qui doloremque culpa suscipit harum molestiae commodi fugiat. Natus hic quos consequuntur.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Doloribus ea voluptate qui aliquid est, illum quaerat aut! Dolore qui doloremque culpa suscipit harum molestiae commodi fugiat. Natus hic quos consequuntur.

My first blog post

10.10.2018

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Doloribus ea voluptate qui aliquid est, illum quaerat aut! Dolore qui doloremque culpa suscipit harum molestiae commodi fugiat. Natus hic quos consequuntur.

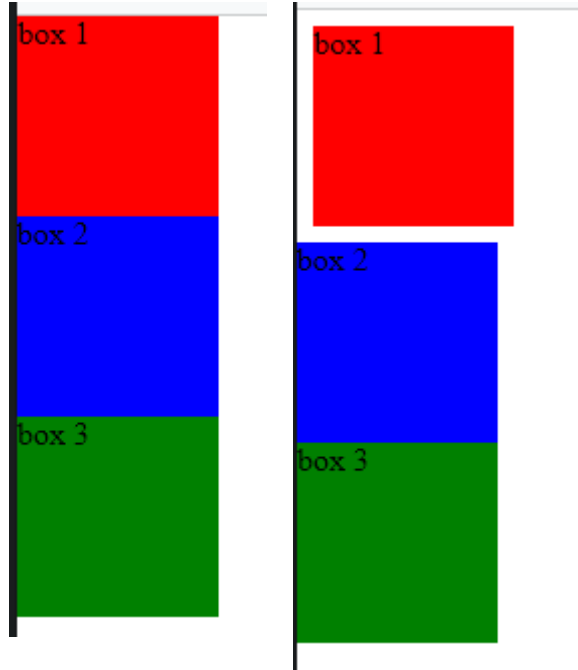
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Doloribus ea voluptate qui aliquid est, illum quaerat aut! Dolore qui doloremque culpa suscipit harum molestiae commodi fugiat. Natus hic quos consequuntur.



Caleb Fields

MARGIN

Margin kutular ya da div'ler arasındaki mesafe ayarlamak istediğimizde kullanacağımız anahtar kelimedir.



Margin 0

Margin 8px(her 4 köşeden 8px hizalama yaptı)

```
margin: 8px 10px 5px 150px;
```

Bu kullanımda saat yönünden başlayarak hizalama yapılır. İlk üst sonra sağ sonra alttan sonra soldan hizalama yapılır.

Ya da bir diğer kullanım şu şekildedir.

```
margin-top: 8px;  
margin-right: 10px;  
margin-bottom: 5px;  
margin-left: 150px;
```

```
margin: 100px 200px;
```

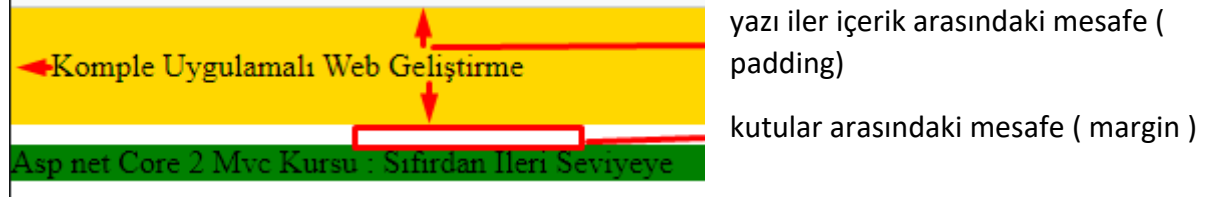
üst alt 100px ve sol 200px olarak ayarlıyor.

Span elemanlara margin sağdan soldan etki eder aşağıdan yukarıdan etki etmez. Eğer etki etmesini istersek display : inline-block yapabiliriz.

Margin auto : margin auto dediğimizde o nesne her zaman sayfanın en ortasına geldi ve sağdan soldan ne kadar pixel büyüyüp küçüleceği tarayıcı tarafından hesaplanır.

PADDING

İçerik ile köşeler arasındaki mesafeyi ayarlamak istediğimizde kullanılacak anahtar kelimedir.



padding-top : sadece üst taraftan hizalama yapılır.

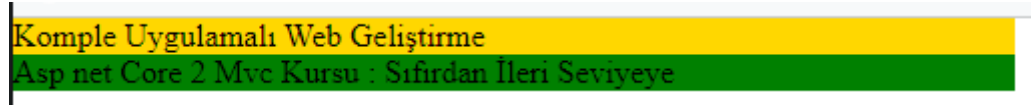
Padding-bottom : sadece alt taraftan hizalama yapılır.

Padding-left : sadece sol taraftan hizalama yapılır.

Padding-right : sadece sağ taraftan hizalama yapılır.

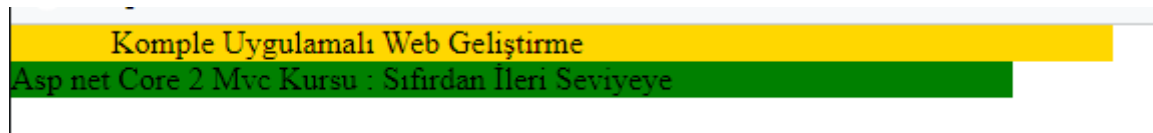
```
padding: 50px 100px 50px 100px;
```

Marginde olduğu gibi saat yönünde de değerler girilerek hizalama yapılabilir.



Biz sadece div'e genişlik verdik.

Eğer üstteki div'e bir padding verirsek üstteki div biraz sağa doğru kayar.



inline(, <a> , ... elemanlarda padding değeri 4 köşeden uygulanabilir.

BORDER

Border sayfa içerisinde kenarlık eklememize imkan sağlar.

Border oluşturken vereceğimiz 3 özellik vardır.

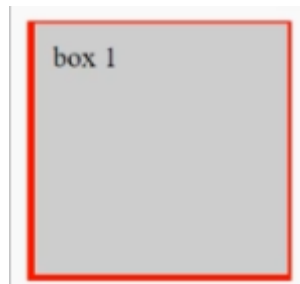
```
#first{  
  border: 1px solid red;  
}
```

Diagram showing the CSS border property with labels: 'kalınlık' (thickness) pointing to '1px', 'tipi' (type) pointing to 'solid', and 'rengi' (color) pointing to 'red'.


```
border-width: 1px;  
border-style: solid;  
border-color: red;
```

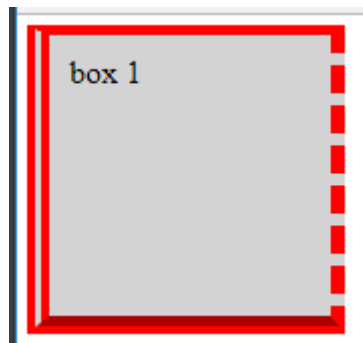
Border'in kalınlığı , rengi , tipi ayrı ayrı verilebilir.

```
border-top-width: 1px;  
border-right-width: 2px;  
border-bottom-width: 3px;  
border-left-width: 4px;
```



Her köşeye farklı bir kalınlık verebiliriz.

```
border-top-style: solid;  
border-right-style: dashed;  
border-bottom-style: groove;  
border-left-style: double;
```

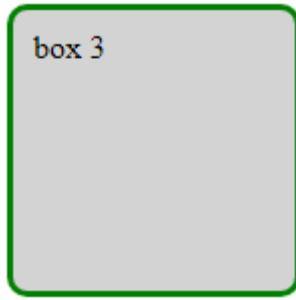


Her köşeye farklı bir çizgi tipi verilebilir.

```
border-top-color: red;  
border-right-color: aqua;  
border-bottom-color: bisque;  
border-left-color: blue;
```

Her köşeye farklı bir renk verebiliriz.

Border-radius ile kenarlıkları yumuşatabiliriz.

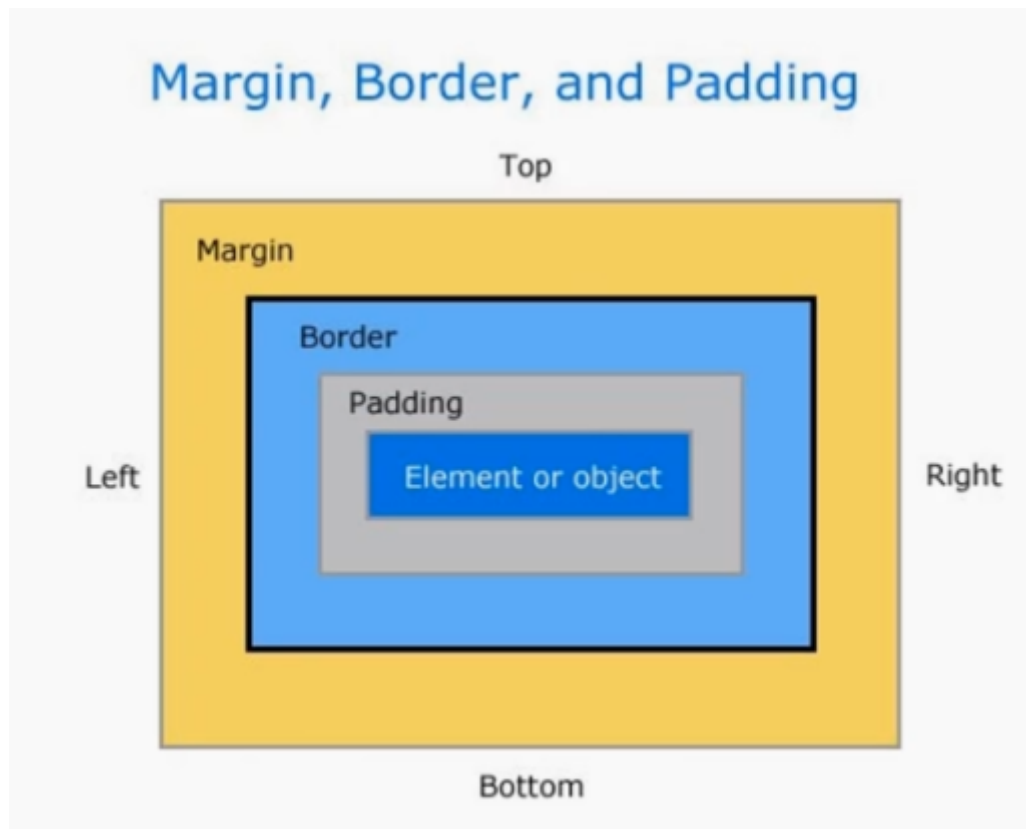


Eğer tamamen yuvarlak yapmak istersek border-radius'a %50 değer verebiliriz. Genelde bir profil resimini tam yuvarlak yapmak istediğimizde kullanıyoruz.

```
border-top-left-radius: 15px;  
border-top-right-radius: 20px;  
border-bottom-left-radius: 30px;  
border-bottom-right-radius: 40px;
```

Her köşeyi farklı farklı yuvarlatabiliriz.

BOX SIZİNG



Box Model

Box sizing ikiye ayrılır;

1-) Content Box

2-) Border-Box

Content Box:

Biz bir elemente padding , border, margin gibi değerler verdiğimizde elementin kaplayacağı alan artıyor.

Örneğin elementin genişliği 300px padding , margin , border gibi değerlerin toplamı 50px ise bu div'in sayfada kaplayacağı alan 350px dir.

Bir elementin sayfada ne kadar yer kaplayacağını hesaplamak için padding, margin , border gibi özelliklerinde hesaba katılması gerekiyor.

Border Box : Biz bir div elementine genişlik ve yükseklik verdiğimizde bu hesaplamanın içinde padding ve border değerleride oluyor.

Margin bu hesaplamanın içine dahil olmuyor.

```
#box1 {  
  box-sizing: content-box;  
  width: 300px;  
  height: 100px;  
  background: green;  
  margin-bottom: 10px;  
  padding: 10px;  
  border: 10px solid black;  
}  
  
#box2 {  
  box-sizing: border-box;  
  width: 300px;  
  height: 100px;  
  background: gold;  
  padding: 10px;  
  border: 10px solid black;  
}
```



Sayfada hiçbir deęişiklik yapılmazsa Google Chrome tarayıcısının varsılan font deęeri 16px dir.

Eęer body ye bir font deęeri verirsek css kısmında o font deęeri tüm alt etiketlere etki eder

Eęer bu şekilde yaparsak body'e hangi font deęerini verirsek p etiketine o deęerin iki katı büyüklüğünde bir yazı çıkar karşımız

Font büyüklüğü :

```
body{
    font-size: 15px;
}
p{
    font-size: 200%;
}
```

Font büyüklüklerini pixel olarak ayaralayabiliriz.

Eęer resimde görüldüğü gibi font büyüklüğüne yüzde olarak deęer verirsek o yüzde deęerine göre işlem yapar ilgili etikette

Örneğin resimde tüm body'deki yazıların font büyüklüğü 15px ve p etiketlerine ise %200 olarak ayarlanmış (yani iki katı) p etiketlerinde font büyüklüğü 30px olarak gerçekleşir.

```
body{
    font-size: 15px;
}
p{
    font-size: 2em;
}
```

2em yazmak ile 200% yazmak aynı şeydir , tarayıcı aynı işlemi yapar.

Font türleri :

Chrome tarayıcısı için aksi belirtilmedikçe uygulanan font türü " Times New Roman" dır.

```
body{
    font-family: "Times New Roman", Times, serif;
    font-size: 16px;
}
```

Font böyle veriliyor eęer o font türünü tarayıcı desteklemiyorsa font kütüphanesini de ekleyeceğiz burada eęer tarayıcı "Times New Roman" türünü desteklemezse , "Times" font türünü uygular, eęer onu da desteklemezse "serif" türünü destekler.

Başlıca font türleri için kaynak :

<https://www.w3.org/Style/Examples/007/fonts.en.html>

https://www.w3schools.com/cssref/css_websafe_fonts.asp

(tarayıcıların genel olarak tanıdığı font türleri.

<https://fonts.google.com> (deęişik estetik font kütüphaneleri)

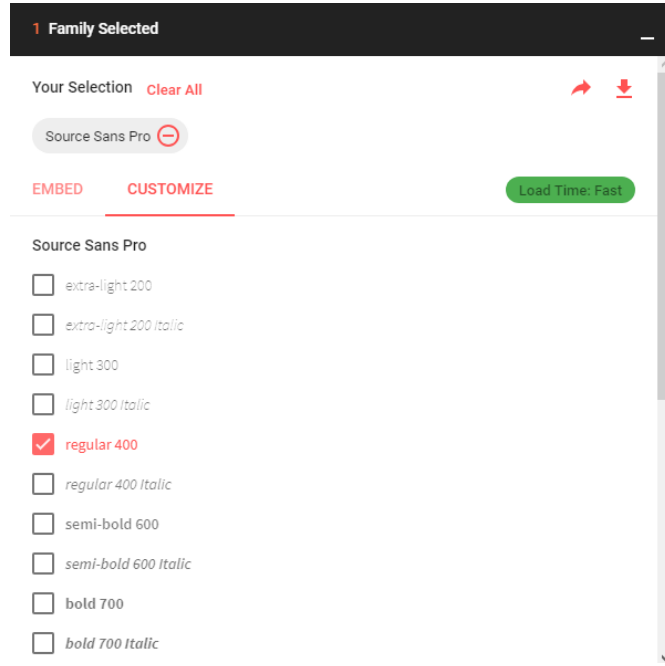
Font kütüphanesi eklemek : Google font's sitesi içinden beğendiğimiz bir font türünü eklemek istersek.

Source Sans Pro
Paul D. Hunt (12 styles)

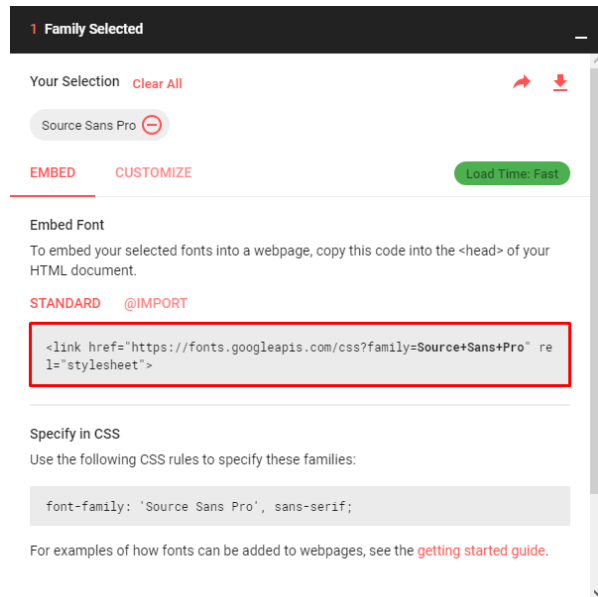


The face of the
moon was in
shadow.

Sağ üst köşedeki “ + ” simgesine basarak



İstedğimiz özellikleri ekledikten sonra



Buradaki linki kopyalayıp html dosyası içindeki <head> etiketinin içine yapıştırmak gerekiyor.

Bu font kütüphanesini ekledikten sonra kullanabilmek için gerekli olan kod aşağıda kırmızı çerçeve içine alınmıştır.

Embed Font

To embed your selected fonts into a webpage, copy this code into the <head> of your HTML document.

STANDARD @IMPORT

```
<link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro" rel="stylesheet">
```

Specify in CSS

Use the following CSS rules to specify these families:

```
font-family: 'Source Sans Pro', sans-serif;
```

For examples of how fonts can be added to webpages, see the [getting started guide](#).

Bu font türünü istediğimiz gibi kullanabiliriz istersek body'nin hepsine istersek belli bir etikete vererek kullanabiliriz.

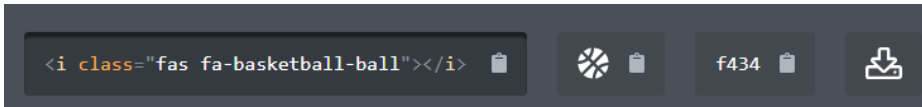
ICON'S

Web sayfalarımızı daha dinamik hale getirmek için belli iconlar kullanıyoruz.

Bu iconları eklemek için linkini verdiğim siteyi kullanabiliriz. (<https://fontawesome.com>).

Bu iconları kullanabilmek için verdiğim sitedeki link kütüphanesini kopyalayıp <head> etiketi içine yapıştırıp projemize dahil etmeliyiz.

Linkini verdiğim siteden seçtiğimiz bir iconun üstüne tıklayıp aşağıya geldiğimizde bu görsel ile karşılaşacağız



Burada etiket olarak ta Unicode olarak ta verilmiş hangi türü kullanmak istersek o türü kopyalayıp editörümüzün ilgili kısmına yapıştırmalıyız.

Eğer font büyüklüklerini etiket içinde değiştirmek istersek

Class	Size	Other Notes
<code>fa-xs</code>	.75em	
<code>fa-sm</code>	.875em	
<code>fa-lg</code>	1.33em	Also applies <code>vertical-align: -25%</code>
<code>fa-2x</code>	2em	
<code>fa-3x</code>	3em	
<code>fa-4x</code>	4em	
<code>fa-5x</code>	5em	
<code>fa-6x</code>	6em	
<code>fa-7x</code>	7em	
<code>fa-8x</code>	8em	
<code>fa-9x</code>	9em	

```
<div style="font-size: 24px;">
  <i class="fas fa-stroopwafel fa-xs"></i>
  <i class="fas fa-stroopwafel fa-sm"></i>
  <i class="fas fa-stroopwafel fa-lg"></i>
  <i class="fas fa-stroopwafel fa-2x"></i>
  <i class="fas fa-stroopwafel fa-3x"></i>
  <i class="fas fa-stroopwafel fa-5x"></i>
  <i class="fas fa-stroopwafel fa-7x"></i>
  <i class="fas fa-stroopwafel fa-10x"></i>
</div>
```

Bu şekilde kullanılabilir.

```
<i class="fab fa-app-store fa-10x">
```

Tarayıcı varsayılana kaç px ise (örnek hiçbir şey yazmazsak 16px dir.) Bu değerin 10 katını ekranda bize gösterir. Mesela body ye uygulanan font değeri 25px ise bu etiketin içinde kullandığımız “ fa-10x” ifadesi ile 250px e taşıyabiliyoruz.

Hareketli iconlar eklemek için

```
<i class="fas fa-spinner "></i>
```



Bu kodu eklediğimizde bu şekilde bir görsel karşımıza gelir

Eğer bu görseli hareketlendirmek istediğimizde

```
<i class="fas fa-spinner fa-spin "></i>|
```

Bu kod satırını eklersek aynı icon hareketli olarak dönmeye başlar.

```
<i class="fas fa-spinner fa-pulse "></i>
```

Bu kod satırını eklediğimizde ise daha farklı bir animasyon ile icon karşımıza gelir.

Animasyonlu linkler ile ilgili gerekli link parantez içindedir. (<https://fontawesome.com/how-to-use/on-the-web/styling/animating-icons>)

ALIGNING TEXT

Column-count özelliği :

```
column-count: 2;
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Veritatis, eius, neque. Consequuntur aut eius ex eaque maiores blanditiis alias aperiam modi architecto, itaque quidem? Lorem ipsum dolor sit amet, consectetur

adipisicing elit. Eaque voluptatum maiores culpa, placeat harum inventore illo dolorem in, ipsam earum facere error sunt quidem aperiam enim provident eveniet nesciunt pariatur!

Bu şekilde metni iki parçaya böler.

```
p#second{  
    column-count: 3;  
    column-rule: 1px solid gray;  
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Fuga quae explicabo, doloribus

magnum doloremque incidunt. Odit impedit, veritatis facilis consectetur dolorem nam

accusantium doloremque sequi, mollitia harum repudiandae, ipsum voluptates!

Bu şekilde sayfayı 3'e bölüp aralarına çizgi koyar.

```
column-count: 3;  
-webkit-column-count: 3; /*opera chrome safari */  
-moz-column-count: 3; /*mozilla*/  
  
column-rule: 1px solid rgb(0,0,0,0.4);  
-webkit-column-rule: 1px solid rgb(0,0,0,0.4);  
-moz-column-rule: 1px solid rgb(0,0,0,0.4);|
```

Tarayıcı ve sürüm farklılığından dolayı eklediğimiz içeriğin gözükmemesi durumunda resimde eklenen “-webkit , -moz “ gibi anahtar kelimeler ile belirtilen tarayıcılarda sürüm farkı olmaksızın gösterilir.

STYLING LINKS

İconları buton içine koyma :

Fontawesome sitesinden aldığımız iconları button içine yerleştirmek istediğimizde şu şekilde yapabiliriz.

```
<a href="/link1" class="btn primary ">
<i class="fas fa-home"></i>
Link 1
</a>
```

BOX SHADOW

Nesnelere gölgelendirme eklemek için kullanılır.



Bu şekilde kullanılır. Sağdan 10 px yukarıdan 20px dediğimizde arkada verdiğimiz değer pixeli kadar bir kutu oluşur.

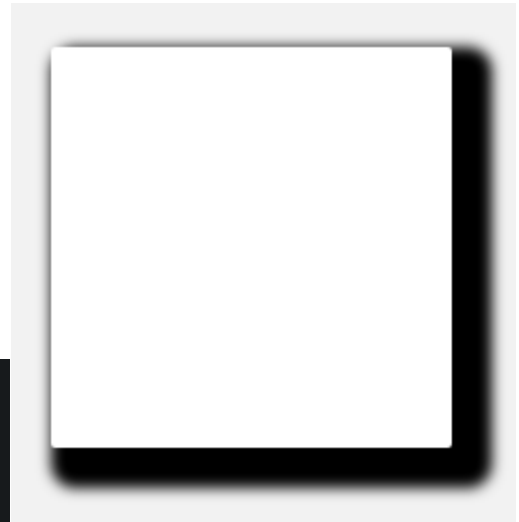
2. tip box shadow:



3.tip box shadow:

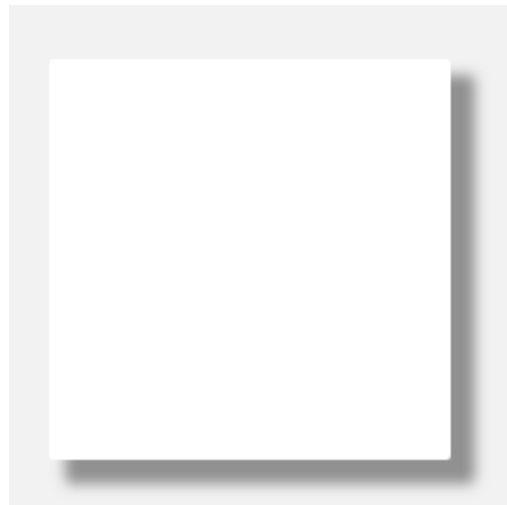
```
.card-3{  
  box-shadow: 10px 10px 10px 10px;  
}
```

gölgelendirmenin dağıtılması ile ilgili



Kutu içine alınan özellikler 2.tip box shadowda açıklanan özelliklerdir.

```
.card-3{  
  box-shadow: 10px 10px 10px 2px  
  rgba(0,0,0,0.4);  
}
```



Gölgelendirmeye renk de verebiliriz.

```
.card-2{  
  box-shadow: 10px 10px 10px;  
  -webkit-box-shadow: 10px 10px 10px;  
  -moz-box-shadow: 10px 10px 10px;  
}
```

-webkit-(chrome , safari) ve -moz- (mozilla) anahtar sözcüklerini kullanmamızdaki sebep parantez içinde verilen tarayıcılarda da her hangi bir uyum sorunu yaşamadan çalışması içindir.

```
.card-5{
  box-shadow: 0px 3px 6px rgba(0,0,0,0.16),0px 3px 6px rgba(0,0,0,0.30);
}
```

Eklenmiş olan bir gölgelendirmeye , koyarak yeniden bir gölgelendirme ekleyip estetik açıdan bir değişim yaratabiliriz. İki tane gölgelendirme eklediğimizde biraz daha öne çıkan bir kutu görünümü oluyor.

TRANSITIONS

İmlecin buton üzerine geldiğinde belli bir süre içinde butonun genişlik ve yüksekliğinin değişmesine transitions denir.

Örnek: 1 adet 170px genişlik ve yükseklikte kutumuz olsun mouse ile kutunun üzerine geldiğimizde hover olayı olduğunda kutunun genişliği 170 px'den 250px'e çıkıyor fakat bu işlem keskin bir şekilde yapılıyor. Bu işlemi biraz yumuşatmak ya da biraz daha yavaşlatmak için transitions kavramını kullanıyoruz.

```
.card{
  width: 170px;
  height: 170px;
  background: blue;
  transition: width 2s;
}
.card:hover{
  width: 250px;
}
```

Transition da ilk değiştirmek istediğimiz özelliği sonra o özelliğin kaç saniye içinde değişeceğini ayarlıyoruz.

Transition olayının olması için bir hover olayının ve dolayısıyla hover css kodunun yazılması gerekmektedir.

```
.card{
  width: 170px;
  height: 170px;
  background: blue;
  transition: width
2s,height 1s;
}
.card:hover{
  width: 250px;
  height: 250px;
}
```

Bu şekilde yaparak da hem yüksekliğini hem de genişliğindeki saniye hesaplamalarını birlikte yapabiliriz.

```
.card{
  width: 170px;
  height: 170px;
  background: blue;
  transition: width
    2s,height 2s,background 1s;
}
.card:hover{
  width: 250px;
  height: 250px;
  background:red;
}
```

Hem rengini hem genişlik ve yüksekliğini belirli saniyeler atayarak gerçekleştirebiliriz.

Transitions'un tüm tarayıcılarda sorunsuz çalışması için

Chrome , safari“-webkit-”

Mozilla “-moz-”

Opera “-o-”

Anahtar kelimelerini ekleyerek tarayıcılardaki uyum sorununu atlatabiliriz.

Transitions özellikleri :

Ease: yavaş başlıyor bitmeye doğru hızlanıyor biterken tekrar yavaşlıyor.

Linear: hep aynı hızda ilerler.

Ease-in: yavaş başlıyor bitmeye doğru hızlanıyor.

Ease-out: hızlı gidiyor ve bitmeye yakın yavaşlıyor.

Ease-in-out: başlangıçta ve bitişte yavaş.

Transitionsların ne zaman sonra başlayacağını da ayarlayabiliyoruz .

- “transitions-delay : 2s; “

