

# CSS

Cascading Style Sheet

# Css nedir?

HTML elemanlarının stil özelliklerini belirleyen bir işaretleme dilidir. CSS sayesinde tek bir tanımlama ile birçok elementin özelliği aynı anda değiştirilebilir. HTML ile sayfanın iskeleti ve içeriği tanımlanırken, CSS ile biçimlendirmesi yapılır.

CSS eklemek için ilgili tagın içine style attribute' u eklenir. Buna inline stil denir.

```
<p style="color: red">Merhaba</p>
```

Property  
(Özellik)

Value  
(Değer)

Birden fazla stil ekleneceği zaman, stil özelliklerinin arasına ; (noktalı virgül) konulur.

```
<div class="color:red; text-align:center"></div>
```



Inline stiller sadece bulundukları tagın içinde etkilidirler.

# Temel CSS özelliklerি

font-size

Yazının büyüğünü değiştirmek için kullanılır. Kullanılabilecek birimler: rem, em, px

```
<a href="sayfa1.html" style="font-size: 2.1rem">click</a>
```

text-align

Yazının bulunduğu tag içindeki yataydaki hizasını belirlemek için kullanılır.  
Kullanılabilecek değerler: left, right, center, justify

```
<div style="color:#1255FA">Sitemize hoşgeldiniz</div>
```

# Temel CSS özelliklerি

color

Yazının rengini değiştirmek için kullanılır. Kullanılabilecek değerler: hazır renk tanımları( red, green, yellow ...), hexadecimal system (#12AF45), rgb, rgba

```
<p style="color:#1255FA">Merhaba</p>
```

background  
-color

Html elemanın zemin rengini değiştirmek için kullanılır. Kullanılabilecek değerler color ile aynıdır.

```
<p style="background-color:#1255FA">Merhaba</p>
```

# Renk Sistemi

Css ile renk tanımlamak için kullanılabilecek bir çok yöntem bulunmaktadır. Bunlardan en çok 3 tanesi tercih edilmektedir.

Hazır renkler

Red, green, blue, beige, maroon, violet, darkviolet, lightviolet...

# Renk Sistemi

Css ile renk tanımlamak için kullanılabilecek bir çok yöntem bulunmaktadır. Bunlardan en çok 3 tanesi tercih edilmektedir.

## Hex (16'lık sistem)

Red, Green ve Blue renk karışımı ile elde edilir.

Renk değerleri 00 – FF arasında değişmektedir. 00, o renkten hiç katılmamasın, FF ise max seviyede katılsın anlamına gelir.

Hex sayı sistemi: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

# EF 54 A5  
Kırmızı Yeşil Mavi

#000000	#ef0033
#ab33ff	#35de10



[https://www.w3schools.com/colors/colors\\_picker.asp](https://www.w3schools.com/colors/colors_picker.asp)

# Renk Sistemi

## RGB

Red, Green ve Blue renk karışımı ile elde edilir.

Renk değerleri 0 – 255 arasında değişmektedir. 0, o renkten hiç katılmamasın, 255 ise max seviyede katılsın anlamına gelir.

rgb( EF , 54 , A5 )  
Kırmızı      Yeşil      Mavi

rgb(0,0,0)

rgb(239, 0, 51)

rgb(171, 51, 255)

rgb(53,222,16)

# Renk Sistemi

## RGBA

Red, Green ve Blue renk karışımı ile elde edilir. 4.parametre şeffaflık değeridir. 0 ile 1 arasında bir değer verilir. 0 tam şeffaf, 1 ise tam opaktır.

Renk değerleri 0 – 255 arasında değişmektedir. 0, o renkten hiç katılmamasın, 255 ise max seviyede katılsın anlamına gelir.

rgb( EF , 54 , A5 , 1 )  
Kırmızı      Yeşil      Mavi      Şeffaflık

rgb(0,0,0, 0.4)

rgb(239, 0, 51, 1)

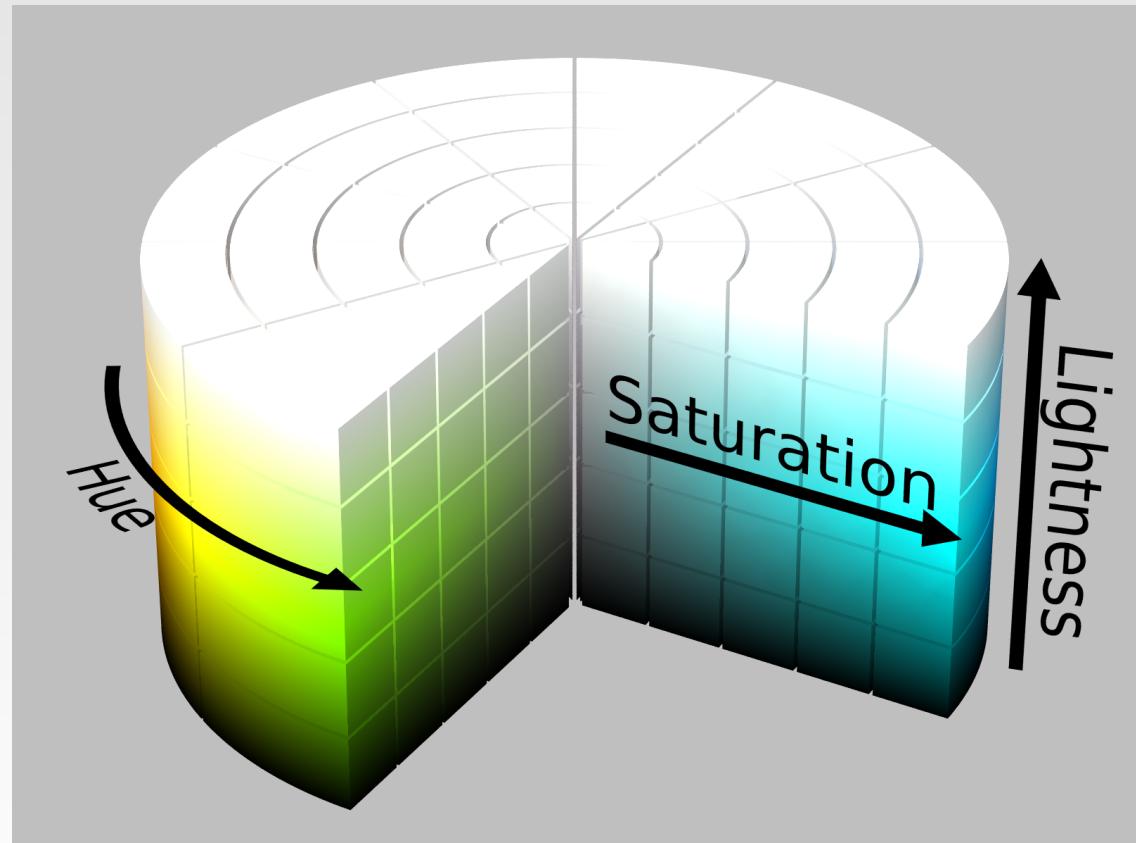
rgb(171, 51, 255, 0.3)

rgb(53,222,16, 0.8)

# Renk Sistemi

## HSL

- Hue, Saturation, Lightness (HSL):
- **Hue**, renk tekerinin 0-360 arasındaki derecesini temsil eder. 0 **Kırmızı**, 120 **Yeşil** ve 240 **Mavi**'dir.
- **Saturation** bir rengin şiddeti olarak adlandırılabilir. 0% Rengin başlangıcını, 100% ise tam rengi gösterir. Ara değerler ise rengin tonlarıdır.
- **Lightness** da % olarak gösterilir. 0% Siyah, 50% is gri, 100% ise Beyazdır.
- **HSLA(hue, saturation, lightness, alpha)**
  - Alpha, 0 ile 1 arasında bir sayıdır. (0 – tam şeffaf, 1 – şeffaf değil)



# Renk Sistemi

HTML

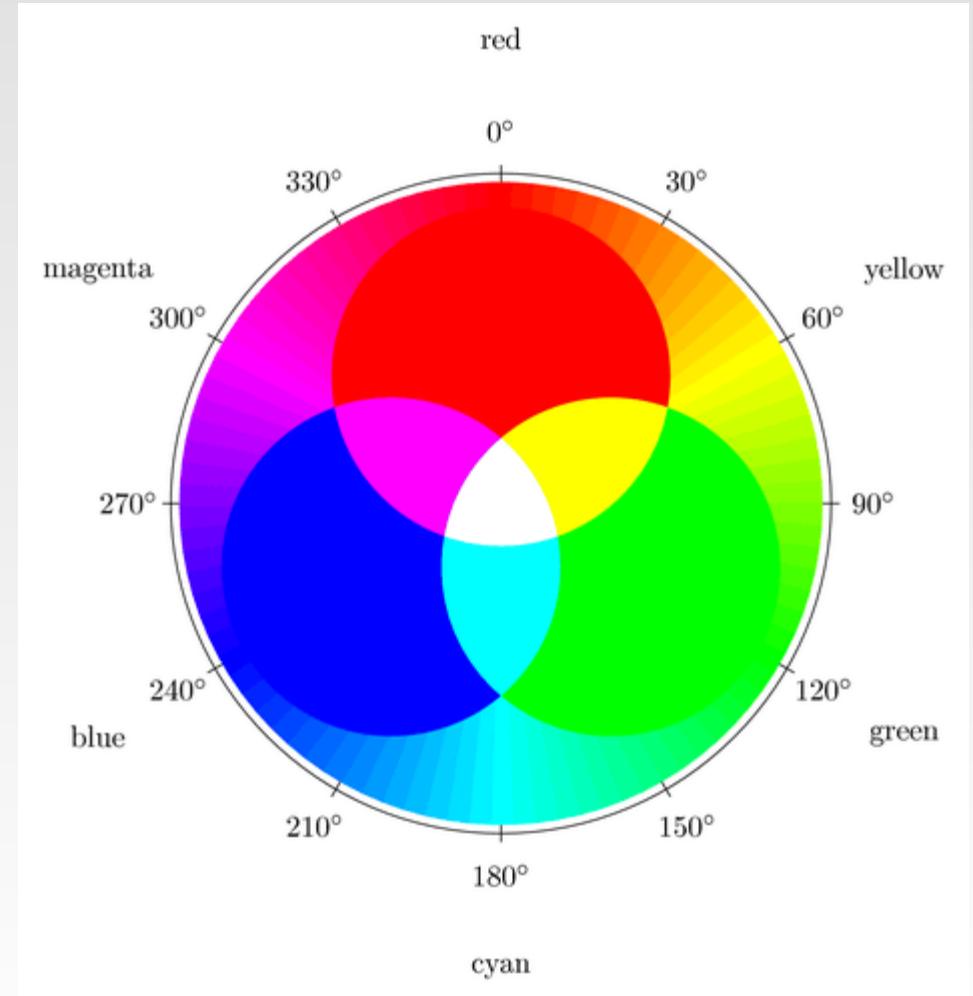
```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1>HSL Values</h1>
5     <p id="par1">blue</p>
6     <p id="par2">green</p>
7     <p id="par3">white</p>
8     <p id="par4">yellow</p>
9     <p id="par5">red</p>
10   </body>
11 </html>
```

CSS

```
1 #par1 {background-color:hsl(240, 100%, 50%);}
2 #par2 {background-color:hsl(120, 100%, 25%);}
3 #par3 {background-color:hsl(0, 0%, 100%);}
4 #par4 {background-color:hsl(60, 100%, 50%);}
5 #par5 {background-color:hsl(0, 100%, 50%);}
```

## HSL Values

blue  
green  
white  
yellow  
red



# Practise

Sola hizalı paragraf,  
yazı büyüklüğü 1rem

Arkaplan rengi  
Yazı rengi  
Metin ortası

## Placeholder Text (Lorem Ipsum)

Placeholder text (Lorem Ipsum) dolor sit amet consectetur adipisicing elit. Molestiae magnam quod officia saepe, veritatis aliquam voluptatem sapiente eaque quos pariatur non neque voluptatibus animi laborum voluptatum corporis, ipsam nostrum dolorem?

Ortaya hizalı  
paragraf,  
yazı büyüklüğü 2rem

Placeholder text (Lorem Ipsum) dolor sit amet consectetur adipisicing elit. Molestiae magnam officia saepe, veritatis **aliquam voluptatem sapiente** eaque quos neque voluptatibus animi laborum voluptatum corporis, ipsam nostrum dolorem?

Sağda hizalı  
paragraf,  
yazı büyüklüğü 1rem

Placeholder text (Lorem Ipsum) dolor sit amet consectetur adipisicing elit. Molestiae magnam quod officia saepe, veritatis aliquam voluptatem sapiente eaque quos neque voluptatibus animi laborum voluptatum corporis, ipsam nostrum dolorem?

# Temel CSS özellikleri

width

Html taglarının genişliğini (yatayda kapladığı alanı) ayarlamak için kullanılır.

height

Html taglarının yüksekliğini (dikeyde kapladığı alanı) ayarlamak için kullanılır.



Width ve height için %, px, rem, vh, vw birim olarak kullanılabilir.



Inline elementlere width ve height uygulanamaz

# Sayfa seviyesinde style (internal)

Tüm sayfada etkili olabilecek stil tanımlamak için `<style>` tagı kullanılır. `<style>` tagı genellikle head tagı içinde tanımlanır.

```
<head>
  <style>
    p {
      color: red;
      text-align: center;
    }
  </style>
</head>
```

# Site seviyesinde style (external)

Site içindeki tüm sayfalarda etkili olabilecek site tanımlaması yapmak için harici stil sayfaları kullanılır.

Bunun için css uzantılı bir dosya oluşturulup tük stil işlemleri bunun içinde yapılır.

Oluşturulan bu stil dosyası ile web sayfaları `<link>` komutu ile ilişkilendirilir. `<link>` tagı head tagı içinde tanımlanır.

```
<link href="css/stil.css" rel="stylesheet">
```

# Seçiciler (Selectors)

Sadece istenilen belli html elemanlarına stil uygulamak için selector lerin doğru kullanılması gereklidir.

1

Tüm html elemanlarını seçmek için \* işaretini kullanılır.

```
* {  
    text-align: center;  
    color: blue;  
}
```

2

Belli türdeki html elemanlarını seçmek için elemanın ismi seçici olarak kullanılır.

```
h1 {  
    text-align: right;  
    font-weight: 300;  
}
```

# Seçiciler (Selectors)

3

Bazı html elemanlarını seçmek için class tanımlaması yapılır.

CSS

```
.ortala {  
    text-align: center;  
    color: blue;  
}
```

HTML

```
<p class="ortala">Merhaba</p>
```

# Comment

CSS içine açıklama eklemek için /\* \*/ işaretleri kullanılır. Comment satırları ile CSS dokumanlarına içindeler ve bölümler oluşturulur.

```
/* Bu bir açıklama satırıdır */
p.aktif{
    text-decoration: none;
}
```

# Etkin stiller

```
<html>
  <head>
    <link rel="stylesheet" href="stil.css">
    <style>
      .baslik{
        color:blue;
      }
    </style>
  </head>
  <body>
    <h1 class="baslik" style="color:red">
      Benim rengim nedir?
    </h1>
  </body>
</html>
```

style.css

```
.baslik{
  color:green;
}
```

Aynı element üzerine farklı noktalardan etki eden **aynı seçicili** stillerde geçerli olan tarayıcının en son rastladığı yani en alta tanımlanan stildir. Inline stiller hep en son değerlendirilirler.

# Etkin stiller

```
<html>
  <head>
    <link rel="stylesheet" href="stil.css">
    <style>
      h1 {
        color:blue;
      }
    </style>
  </head>
  <body>
    <h1 class="baslik">
      Benim rengim nedir?
    </h1>
  </body>
</html>
```

```
.baslik{
  color:red;
}
* {
  color:green;
}
```

Aynı element üzerine farklı noktalardan etki eden **farklı seçicili** stillerde, geçerli olan en özel stildir.



# Practise

Ana Sayfa

Ürünler

Blog

İletişim

External css kullanarak yandaki menüyü oluşturunuz.

- Menünün arka plan rengi gri
- Yazılar ortalı
- Ayrıca aktif isimli bir class oluşturunuz. Hangi menü elemanına bu class ismi verilirse o elemenin arka plan rengi koyu gri ve yazı rengi beyaz olsun

# Font kullanımı

font-family

Yazı tipini değiştirmek için kullanılır. Birden fazla font ismi tanımlanabilir. Ziyaretçinin cihazında yüklü olan ilk font kullanılır.

```
body {  
    font-family: tahoma, verdana  
}
```



Eğer hiçbirini yüklü değilse tarayıcının default fontu kullanılır.

Tasarım dünyasında kullanılan çeşitli yazı tipi sınıfları bulunmaktadır.

serif

Sans-serif

fantasy

monospace

cursive

Times N R  
Georgia  
Garamond

**Verdana**  
Tahoma  
Arial

**COPPER**  
Papyrus  
Forte

Courier  
Lucida C  
Consolas

*Brush script*  
Lucida H.  
**Harlow**

Bunlar içinde web sayfalarında kullanılması tavsiye edilen font sınıfı okunması daha kolay olduğu için sans-serif tür.

# Google fontları

Kullanılan fontların düzgün gözükmesi için o fontun kullanıcının bilgisayarında yüklü olması gereklidir. Kullanıcının bilgisayarında hangi fontların yüklü olduğunu bilmemeyeceğimiz için bunu garantilemek için google fontlarını kullanabiliriz.

<https://fonts.googleapis.com/>

```
<link rel="preconnect" href="https://fonts.gstatic.com">
<link href="https://fonts.googleapis.com/css2?
family=Roboto:ital,wght@0,100;1,100&display=swap" rel="stylesheet">
```

HTML

```
body{
  font-family: Roboto;
}
```

CSS

# Metin Stilleri

Text-decoration

Metnin altına, üstüne veya üzerine çizgi çekmek için kullanılır. Alabileceğimiz değerler underline, overline, line-through, none

```
a {  
    text-decoration: none;  
}
```

Merhaba

~~Merhaba~~

font-style

Yazıyı italik hale getirmek için kullanılır. Alabileceğimiz değerler normal, italic

```
font-style : italic;
```

font-weight

Yazıyı kalınlaştırmak için kullanılır. Alabileceğimiz değerler 100 – 900, bold, bolder, lighter

```
font-weight : 500;
```

# Metin Stilleri

letter-spacing

Harfler arasındaki boşluğu ayarlamak için kullanılır.

```
letter-spacing: .2rem;
```

word-spacing

Kelimeler arasındaki boşluğu ayarlamak için kullanılır.

```
word-spacing: .2rem;
```

line-height

Satırlar arası boşluğu belirlemek için kullanılır.

```
line-height: .2rem;
```

# Link stilleri

a: link

Ziyaret edilmemiş linkler üzerinde stil belirlenebilir

```
a:link{color: #000000 }
```

a:visited

Ziyaret edilmiş linkler üzerinde stil belirlenebilir

```
a:link{color: #000000 }
```

a:hover

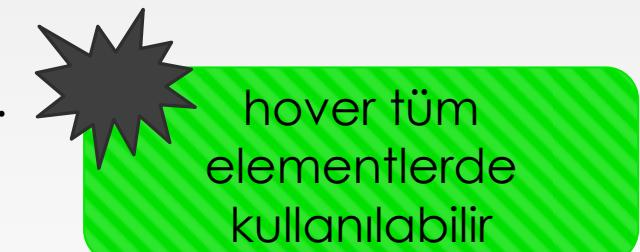
Fare ile linkin üzerine gelindiği andaki stil belirlenebilir.

```
a:link{color: #000000 }
```

a:active

Fare ile linkin üzerine tıklandığı andaki stil belirlenebilir.

```
a:link{color: #000000 }
```



# Liste stilleri

list-style-type

Listelerin türünü değiştirmek için kullanılır. Alabileceğiniz değerlerden bazıları: disc, square, circle, **none**

```
list-style-type: disc;
```

```
list-style-type: '...',
```



Emojipedia.com sitesinden  
icon kopyalanabilir

list-style-image

İşaretli listelerin başındaki simgeyi değiştirmek için kullanılır.

```
list-style-image: url (assets/img/bullet.png);
```



flaticon.com dan icon  
indirebilirsiniz

list-style-position

İşaretin pozisyonunu beliler. Alabileceğiniz değerler: inside | outside

```
list-style-position: inside;
```

# Practise

◆ Ana Sayfa

◆ Ürünler

◆ Blog

◆ İletişim

Daha önce oluşturduğunuz menüyü liste kullanarak tekrar yapınız.

- Yazılara link özelliği kazandırınız.
- Font ailesi: google roboto olsun
- Menü elemenaları sola hizalı olsun.
- Linklerin üzerine gelindiğinde arka plan rengini ve metin rengini değiştiriniz.
- Linklerin alt çizgisini kaldırınız
- Linklerin kalınlığı 400 olsun
- Ziyaret edilmiş linkler ve normal link renkleri siyah olsun.

İPUCU

Menü elementlerinin sol baş taraflarında olacak boşluğu kaldırmak için ul nin içine **padding:0** verebilirsiniz

# Chrome Geliştirici Araçları

Sayfanın HTML, CSS ve Javascript gibi bileşenlerini analiz etmek ve hataları gidermek için kullanılır.

Sağ üst köşedeki üç notaya tıklanıp «Diğer Araçlar» ve «Geliştirici Araçları» seçilerek ulaşılabilir. Ya da sayfadaki analiz edilecek elemana sağ tıklanıp incele (inspect) seçilerek de ulaşılabilir.

**Elements** bölümünden sayfanın HTML ve CSS incelemesi yapılabilir. **Performance** bölümünden ise sayfanın yükleneme performansı incelenerek hızlı yüklenmeyi etkileyen unsurlar tespit edilebilir.

# Seçiciler (selectors) devam...

4

Parent – child ilişkisi içinde stil tanımlanabilir.

```
nav ul li{  
    font-weight: 900;  
    color: red;  
}
```

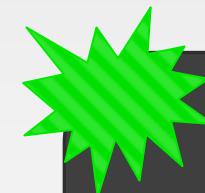
=

```
.navbar ul li{  
    font-weight: 900;  
    color: red;  
}
```

=

```
nav.navbar ul li{  
    font-weight: 900;  
    color: red;  
}
```

```
<nav class="navbar">  
    <ul>  
        <li><a href="home.html">Ana Sayfa</a></li>  
        <li><a href="products.html">Ürünler</a></li>  
        <li><a href="about.html">Hakkımızda</a></li>  
        <li><a href="blog.html">Blog</a></li>  
        <li><a href="contact.html">İletişim</a></li>  
    </ul>  
</nav>
```



Nav içindeki tüm ul  
ler içindeki tüm il  
elementlerini seçer.

# Practise

o 15dk

◆ Ana Sayfa

◆ Ürünler

◆ Blog

◆ İletişim

Daha önce oluşturduğunuz menüyü sadece ul tagına vereceğiniz tek bir class ile tekrar yapınız.

# Seçiciler (selectors) devam...

5

Aynı anda farklı elementlerin seçilmesi için virgül kullanılır.

```
.navbar ul,  
.navbar a{  
    padding:3rem;  
}
```

```
<nav class="navbar">  
    <ul>  
        <li><a href="home.html">Ana Sayfa</a></li>  
        <li><a href="products.html">Ürünler</a></li>  
        <li><a href="about.html">Hakkımızda</a></li>  
        <li><a href="blog.html">Blog</a></li>  
        <li><a href="contact.html">İletişim</a></li>  
    </ul>  
</nav>
```

# Seçiciler (selectors) devam...

6

Child elementlerden sadece belli bir tanesi de seçilebilir. Bunun için **pseudo** sınıfları kullanılır. Bunlar **first-child**, **last-child**, **nth-child** sınıflarıdır.

```
.navbar ul li:first-child{  
    color: blue;  
}
```

first-child

İlk elemenani seçer

last-child

Son elemenani seçer

nth-child(n)

İstenilen sıradaki elemeni seçer. ( n ) yerine sıra numarası, odd veya even verilebilir.

```
<nav class="navbar">  
    <ul>  
        <li><a href="home.html">Ana Sayfa</a></li>  
        <li><a href="products.html">Ürünler</a></li>  
        <li><a href="about.html">Hakkımızda</a></li>  
        <li><a href="blog.html">Blog</a></li>  
        <li><a href="contact.html">İletişim</a></li>  
    </ul>  
</nav>
```

**odd:** tek sıradaki elemanlar  
**even:** Çift sıradaki elemanlar

# Pseudo sınıfları

- :active → Üzerine tıklanan linkleri seçer
- :hover → Üzerine fare ile geline linkleri seçer
- :visited → Ziyaret edilmiş linkleri seçer
- :focus → Aktif durumda olan form elementlarını seçer
- :checked → İşaretlenmiş checkbox ve radio butonları seçer
- :first-child → İlk çocuk elemanı seçer
- :last-child → Son çocuk elemanı seçer
- :nth-child(n) → n. çocuk elemanı seçer

::after

Bir elementin hemen sonrasında bir içerik veya stil eklemek için kullanılır

::before

Bir elementin hemen öncesine bir içerik veya stil eklemek için kullanılır

# Icon Kütüphanesi (Fontawesome)

Web sayfalarına ikon eklemek için çeşitli hazır kütüphaneler bulunmaktadır. Bunlardan en popüler olanı fontawesome kütüphanesidir. Fontawesome kütüphanesi indirilerek site içine yerleştirilebildiği gibi CDN üzerinden de kullanılabilir.

[fontawesome.com](https://fontawesome.com)

Lokal

```
<link rel="stylesheet" href="assets/lib/fontawesome.css">
```

CDN

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/5.15.3/css/all.min.css">
```

# Icon Kütüphanesi (Fontawesome)

Fontawesome sayfaya bağlandıktan sonra html içinde i tagları ile ekleme yapılabilir.

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



Fas: solid (içi dolu iconlar)  
Fab: brand (markalar)

Icon

## Pseudo olarak eklemek için

```
ul li::before {  
    font-family: "Font Awesome 5 Free";  
    font-weight: 900;  
    content: "\f054";  
}
```

# Alışveriş Sepeti

Ürün Kodu	Ürün Adı	Miktar	Birim Fiyat	Ara Toplam	
154654	Notebook	1	5.400 ₺	5.400₺	 
456466	Klavye	2	80₺	160₺	 
564646	Monitör	1	700₺	700₺	 
123598	Mouse	2	50₺	100₺	 
613448	Play Station	1	2.500₺	2.500₺	 
789799	Usb Bellek	3	120₺	360₺	 
<b>TOPLAM</b>				<b>9.220₺</b>	



Alışverişe Devam



Ödemeye Geç

# Practise

Ana Sayfa

:! Ürünler

:! Blog

:! İletişim

1

Menü elemanlarının baş tarafına pseudo classları kullanarak simge ekleyiniz. Fare ile üzerine gelindiğinde pseudo elemanın rengini de değiştirin.

2

Fontawesome ikonlarını pseudo olarak atayınız

# Practise

◀   [Eylül](#) | [Ekim](#) | [Kasım](#) | [Aralık](#) | [Ocak](#) | [Şubat](#)   ▶

# Seçiciler (selectors) devam...

7

> işaretti kullanılarak seçimler daha özelleştirilebilir.

>

Belirtilen parent'in childlarını seçer. Örnekte parent'i div olan tüm a elementlerini seçer

```
div > a {     .... }
```

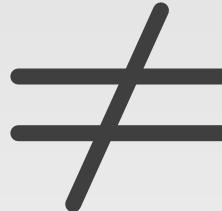
# Seçiciler (selectors) devam...

7

> işaretin kullanılarak seçimler daha özelleştirilebilir.

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

kutu içindeki tüm p elementlerini seçer



```
.kutu > p{  
    color: blue;  
}
```

Parenti kutu olan tüm p elementlerini seçer

```
<div class="kutu">  
    <h2>Notebook</h2>  
    <p>Lorem ipsum .....</p>  
    <div><p>145₺</p></div>  
</div>
```

# Seçiciler (selectors) devam...

8

+ ve ~ (tilt) işaretleri kullanılarak seçimler daha özelleştirilebilir.

+

Bir elementten hemen sonra gelen ilk elemeni seçer. Örnekte div elementi kapandıktan sonra gelen ilk a elementini seçer. Eğer divden hemen sonra a elementi yoksa hiçbir şey seçmez.

```
div + a { .... }
```

~

Bir elementten hemen sonra gelen elementleri seçer. Örnekte active classına sahip a elementinden sonra gelen tüm a elementlerini seçer.

```
a.active ~ a { .... }
```

# Seçiciler

Aşağıda belirtilen elementler için seçicileri yazınız

1. Tüm elementler
2. Sayfadaki tüm h2 ler
3. Sayfadaki tüm section-header isimli elementler
4. Sayfadaki tüm section-header isimli elementlerin içindeki h2 ler
5. Body içindeki 3.section
6. top-bar ismindeki section içindeki img
7. top-bar içindeki nav içindeki linkler
8. parent i sosyal-medya olan linkler
9. parent i section olan navların içindeki li ler
10. top-bar dan sonraki section
11. parenti kutu olan img lerden sonraki span lar
12. top-bar içindeki nav içindeki ilk li den sonraki li ler
13. referanslar içindeki kutuların üzerine gelindiğinde img ler
14. hizmetler içindeki parenti kutu olan h3 lerden önce içerik eklenecek
15. top-bar içinde 3. li nin üzerine geldiğinde a dan sonraya içerik eklenecek
16. top-bar içindeki ilk li nin üzerine geldiğinde, bu li den sonraki li içindeki a nın sonuna içerik eklenecek

# Ölçü birimleri

Css te genişlik ve yükseklik tanımlamalarında kullanılan çeşitli ölçü birimleri mevcuttur. Bunlardan en çok bilineni ve eskiden beri kullanılan px (piksel) dir. Ancak her zaman px kullanmak doğru değildir. Ekran çözünürlüklerinin çok çeşitlendiği günümüzde tüm ekranlarda sorunsuz bir görünüm elde etmek için daha çok göreceli ölçü birimlerini de kullanmak gereklidir.

## Sabit ölçü birimleri

- Cm
- Mm
- İnc
- Px\*

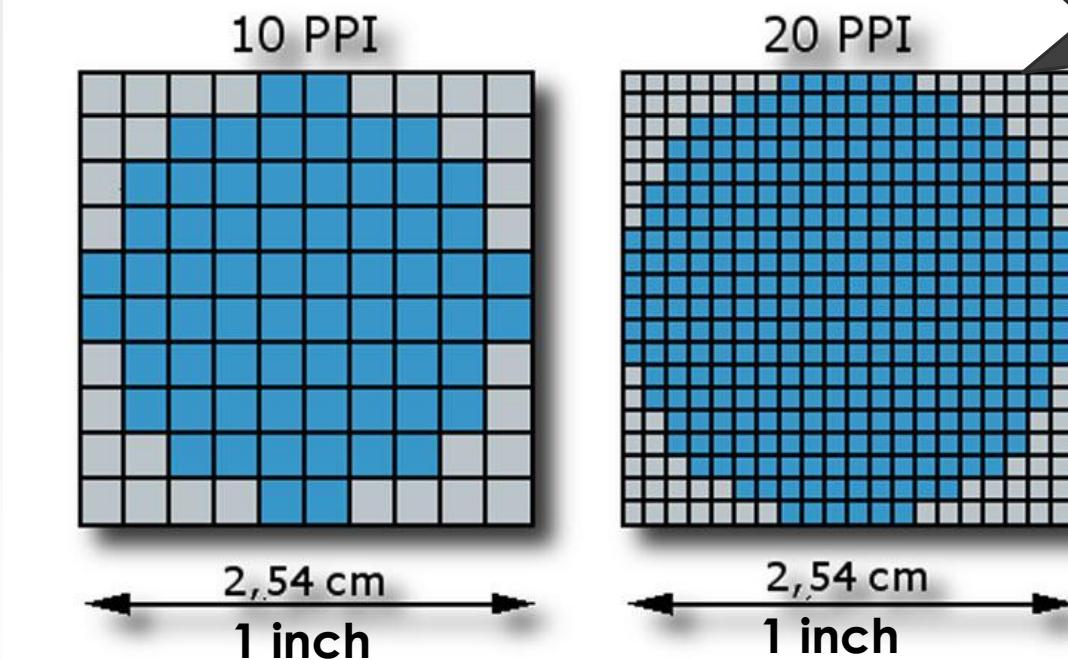
## Göreceli ölçü birimleri

- %
- em
- rem
- vh
- vw

# Sabit ölçü birimleri

Cm, mm ve inc özellikle web sayfası üzerinden çıktı alınması gereken sayfalarda kullanılır. Raporlar, biletler vs. Sabit ölçüler her ekranada aynı boyda gözükmür. Px (piksel) in özel bir durumu vardır. Dikkat edilmezse farklı cihazlarda istenmeyen görüntüler elde edilebilir.

PPI: Pixel per Inch 1 inch  
içindeki pixel sayısı  
cihazdan cihaza  
değişmektedir. Bu sebeple  
100px büyüklüğündeki bir  
html elementi her cihazda  
farklı büyülükte  
gözükecektir.

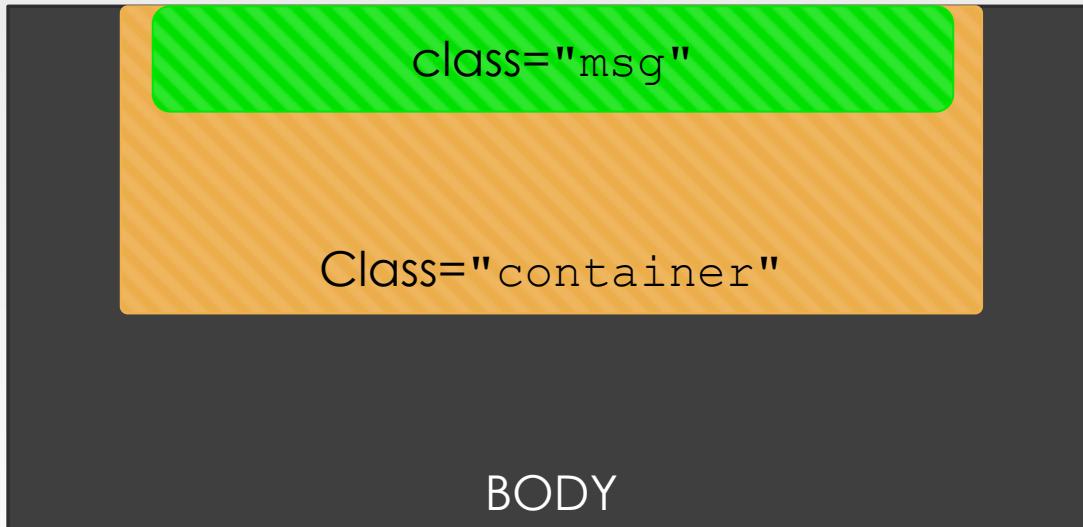


# Göreceli ölçü birimleri

Kullanıcının ekran çözünürlüğünden bağımsız bir şekilde kullanılan ölçü birimleridir.

%

Parent elementin büyülüüğünü esas alır. 100% olarak verilen büyülüük parent elementi kaplar. Parent elementin boyutu değiştiğinde kendisi de bundan etkilenir.

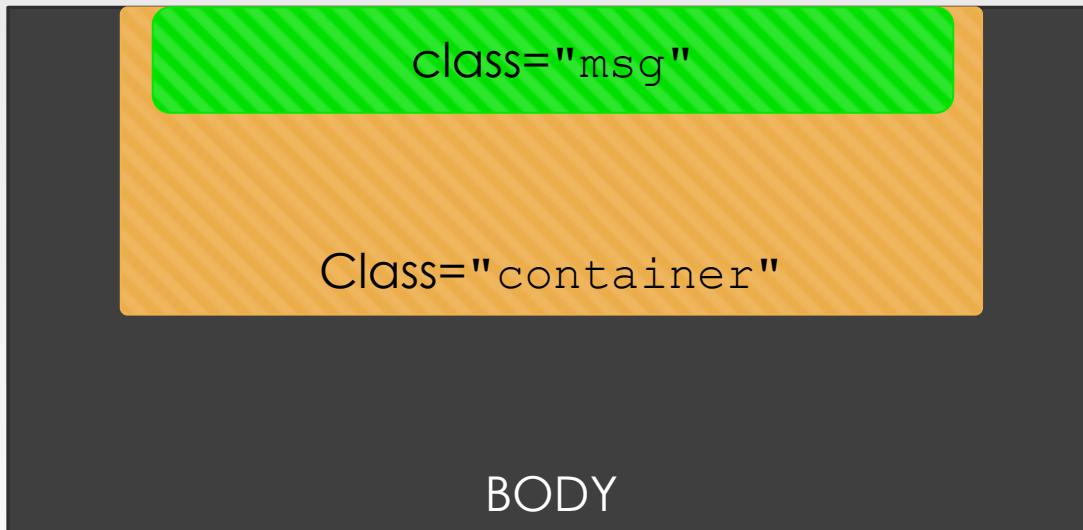


```
div.container{  
    width:80%;  
    height:50%;  
}  
  
div.msg{  
    width:99%;  
}
```

# Göreceli ölçü birimleri

em

Parent elementin font-size değerine bağlıdır.



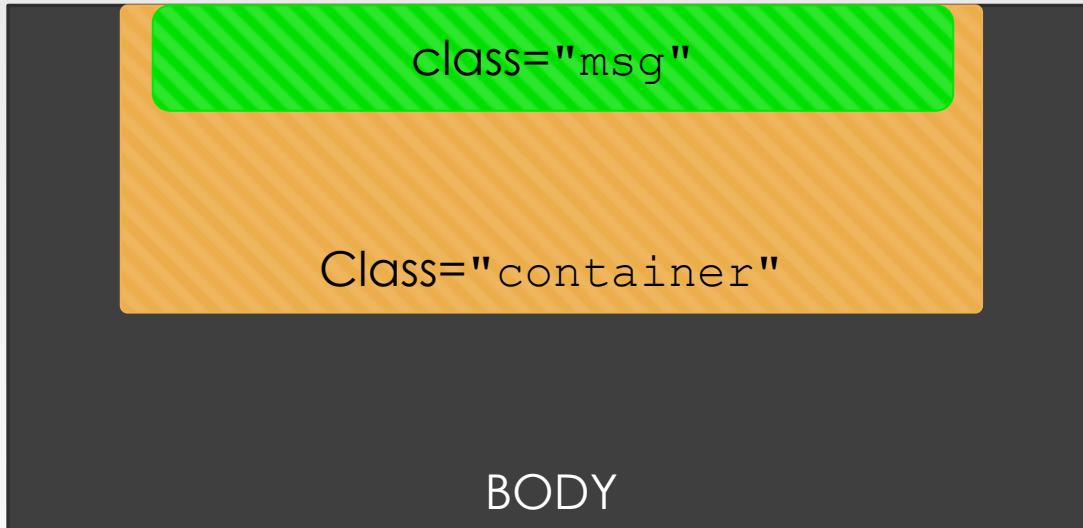
```
div.container{
    font-size:2em;
}
```

```
div.msg{
    font-size:2em;
}
```

# Göreceli ölçü birimleri

rem

En parent (root) elementin font-size değerine bağlıdır



```
body{  
    font-size:1.1rem;  
}  
  
div.msg{  
    width:2rem;  
}
```

# Relative ölçü birimleri

vw

Tarayıcı penceresinin viewport genişliğine bağlıdır. Pencere genişliği değişikçe bu değere bağlı css ler etkilenir. **Parent genişliğine bakmaz.**



```
div.msg {  
    width: 2vw;  
}
```

# Relative ölçü birimleri

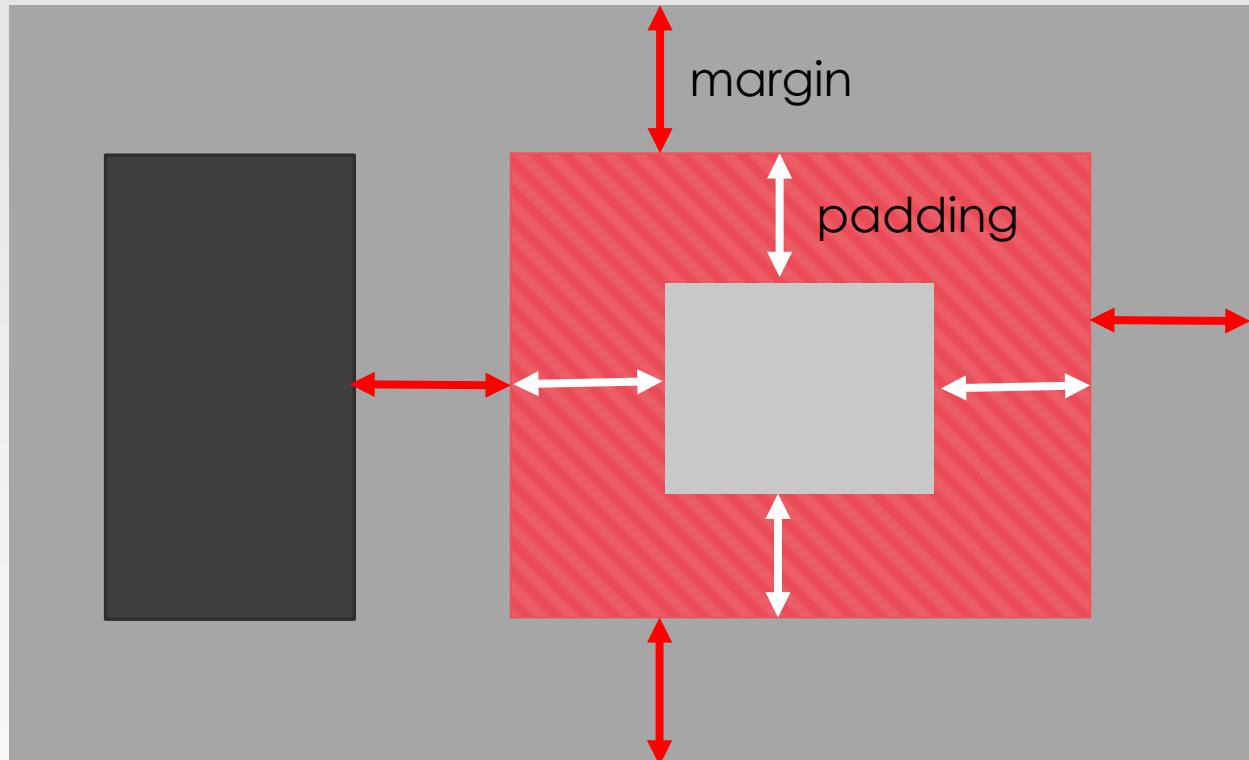
vh

Tarayıcı penceresinin viewport yüksekliğine bağlıdır. Pencere yüksekliği değişikçe bu değere bağlı css ler etkilenir. **Parent yüksekliğine bakmaz.**



```
div.msg{  
    height: 2vh;  
}
```

# Padding & Margin



padding

Elementin içindeki diğer elementlerin kenarlardan olan uzaklığını belirler.

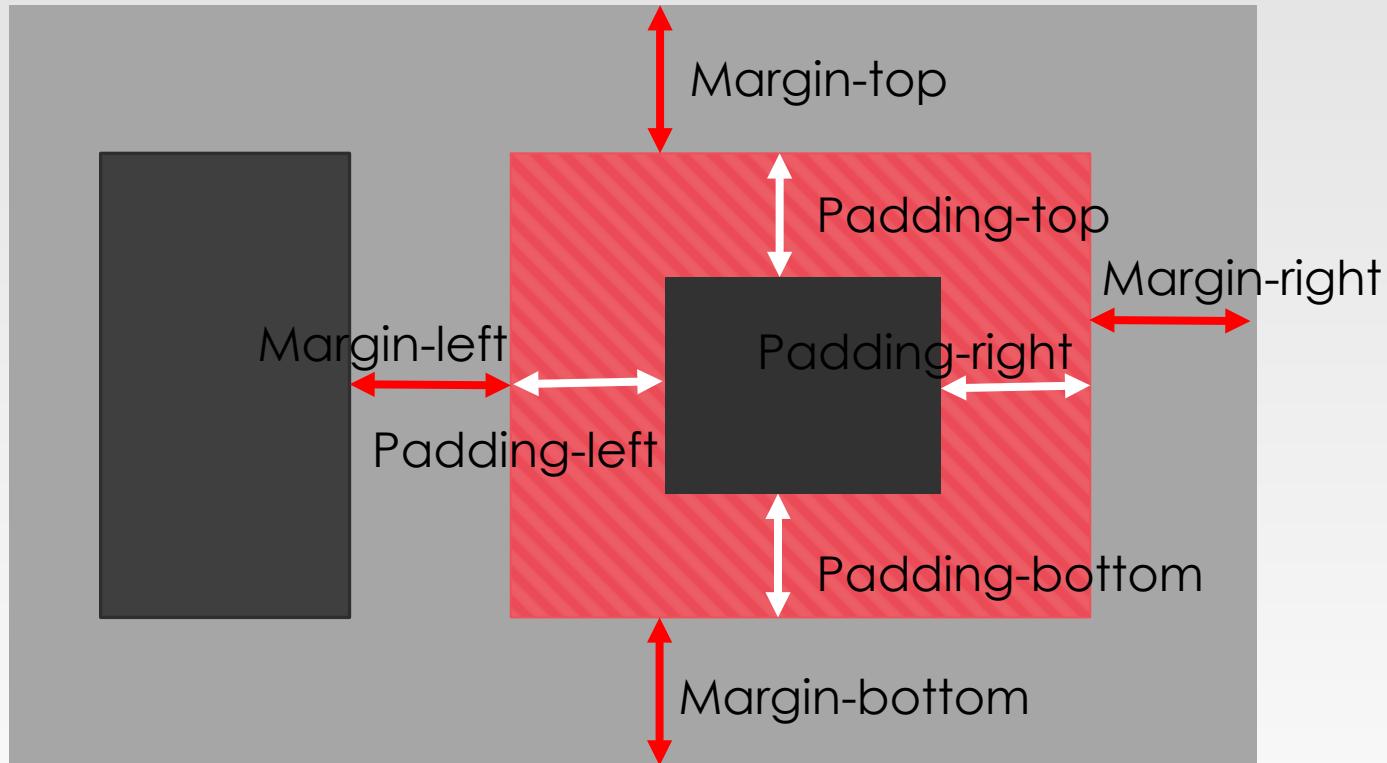
margin

Elementin dıştaki elementlerden olan uzaklığını belirler

```
<body>
  <div> </div>
  <div class="box">
    <div>
    </div>
  </div>
</body>
```

```
.box{
  padding: 3rem;
  margin: 5rem;
}
```

# Padding & Margin



Margin ve padding değerleri ayrı ayrı tanımlanabilir.

```
margin-top: 10px;  
padding-left: 20px;
```

**margin: üst sağ alt sol** şeklinde de tanımlanabilir

```
margin: 5px 3px 7px 8px;  
padding: 4rem 5rem 12rem 0;
```

**margin: dikey yatay** şeklinde de tanımlanabilir

```
margin: 5px 3px;  
padding: 4rem 5rem;
```

**Inline elementlere sadece yatay margin uygulanabilir. Padding her yönden verilebilir.**

# Dikey Hizalama (1.yöntem)

- Bir elementi başka bir element içinde hizalamak için alt üst padding kullanılabilir.

Merhaba Dünya

```
div{  
    padding: 5rem 0;  
}
```

```
<div>Merhaba Dünya</div>
```

# Float kullanımı

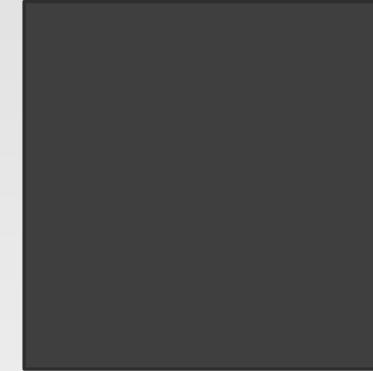
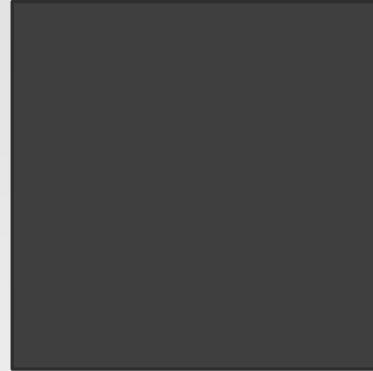
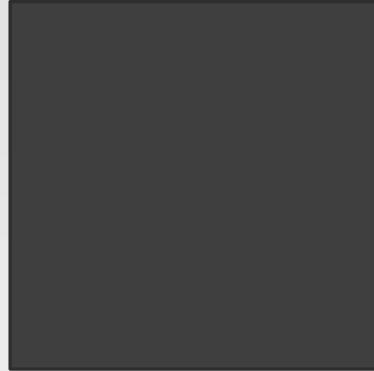
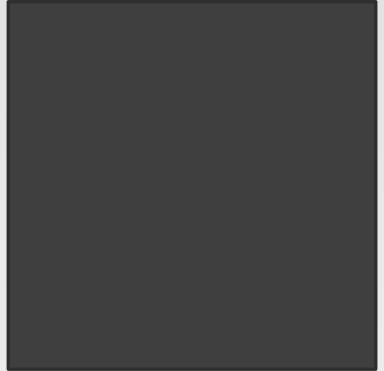


Elementleri yanyana getirebilmek için kullanılırlar. Float özelliği kazandırılan bir element html in normal akışından çıkararak konumlanır. Sonrasında gelen elementler onun yanından itibaren konumlanırlar.

```
  
<p> Elementleri yanyana getirebilmek  
için kullanılırlar. Float özelliği  
kazandırılan bir element ...</p>
```

```
img{  
    float:left;  
}
```

# Float kullanımı



```
<div class="kutu"></div>
<div class="kutu"></div>
<div class="kutu"></div>
<div class="kutu"></div>
```

```
.kutu{
    width:100px;
    height:100px;
    background-color:#222222;
    float:left;
    margin-right:1rem;
}
```

# Float kullanımı

Float özelliği, kendisinden sonra gelen elemanların da dizilimini etkilemektedir. Float özelliğini devre dışı bırakmak için clearfix diye isimlendirilen bir hack yöntemi uygulanır.

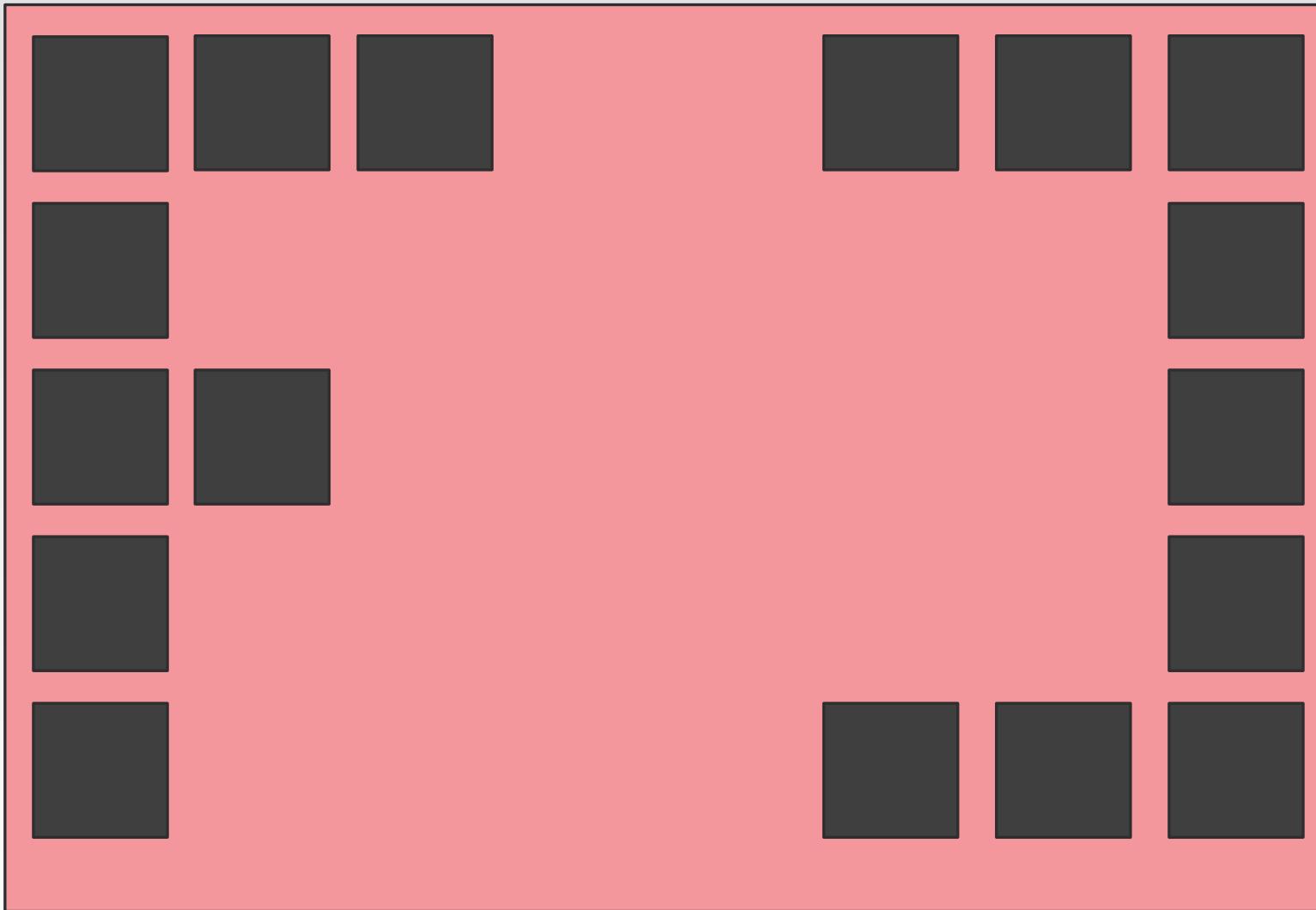


```
<div class="kutu"></div>
<div class="kutu"></div>
<div class="kutu"></div>
<div class="clearfix"></div>
<div class="kutu"></div>
```

```
.kutu{
    width:100px;
    height:100px;
    background-color:#222222;
    float:left;
    margin-right:1rem;
}

.clearfix{
    clear:both;
}
```

# Practise



Özdeş kutulardan  
oluşan sayfanın sol  
tarafına yaslanmış F  
harfi ile sayfanın sağ  
tarafına yaslanmış ters  
C harfi yapınız.

# Practise



External css kullanarak yandaki yapıyı oluşturunuz.

# Display

Elementlerin inline ve block özelliklerini değiştirmek için display stili kullanılır. Display stili inline, block ve inline-block değerleri alabilir.

inline

Bir elemanı inline hale getirir. Böylece sadece ihtiyacı kadar yer kaplar. Width, height, margin-top, margin-bottom değeri alamazlar.

block

Bir elemanı block hale getirir. Böylece eleman yeni satırdan başlar ve tüm satırı kaplar.

inline-block

Hem block hem inline özellikler gösteren bir element haline gelir. Sadece ihtiyacı kadar yer kaplarken, width, height, dikey margin gibi block elementlere mahsus özellikler de alabilir.

none

Elemanı gizler

Aşağıdaki kodları şekildeki menüye çevirecek css kodlarını yazınız

Ana Sayfa

Ürünler

Hakkımızda

Blog

İletişim

```
<nav>
  <ul>
    <li><a href="#">Ana Sayfa</a></li>
    <li><a href="#">Ürünler</a></li>
    <li><a href="#">Hakkımızda</a></li>
    <li><a href="#">Blog</a></li>
    <li><a href="#">İletişim</a></li>
  </ul>
</nav>
```

Fare ile  
Üzerine  
gelindiğinde  
buton zemini  
yeşil, yazı rengi  
siyah  
olacaktır.

# Practise

Üzerine gelindiğinde açılacak menü tasarlayınız.

## Ürünler

- Notebook
- Klavye
- Fare
- Monitör

# Yatay hizalama

Block veya inline-block elementlerin içindeki elementleri yatayda konumlandırmak için, parent element içinde text-align stili kullanılır

Merhaba

```
p {  
    text-align: center;  
}
```

```
<p>  
    <b>Merhaba</b>  
</p>
```

# Yatay hizalama

Blok bir elementi başka bir element içinde yatayda konumlandırmak için



Merhaba

```
<div>
  <div>Merhaba</div>
</div>
```

**Sola hizalamak için (geçerli değer)**

```
margin-left:0;
margin-right: auto;
```

**Sağ'a hizalamak için**

```
margin-left: auto;
margin-right:0;
```

**Ortalamak için**

```
margin-right:auto;
margin-left:auto;

/* Kısa hali */
margin:auto;

/*veya*/
margin:5rem auto;
```

# Çerçeveler (Borders)

Elementlere çerçeve vermek için border stili uygulanır.

```
border: genişlik tip renk;
```

```
border: 3px solid red;
```

```
border: 3px dashed blue;
```

```
border: 5px dotted gray;
```

# Çerçeveler (Borders)

Tip için verilebilecek değerler:  
**dotted, dashed, solid, double, groove,  
ridge, inset, outset, none**

**This is a dotted box**

**This is a dashed box**

**This is a solid box**

**This is a double box**

**This is a groove box**

**This is a ridge box**

**This is a inset box**

# Çerçeveler

Elementlerin **4 tarafına** border verilebildiği gibi sadece istenilen tarafına da verilebilir.

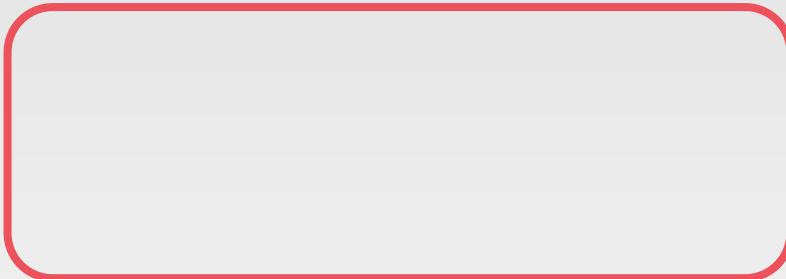
```
border-left | border-right | border-bottom | border-top
```



```
border-top: 3px solid red;
```

# Çerçeveler

Çerçevelerin köşelerini yumuşatmak için **border-radius** stili kullanılır.



```
div{  
    border:.2rem solid red;  
    border-radius: .5rem;  
}
```



```
div{  
    background-color:red;  
    border-radius: .5rem;  
}
```

# Practise

- Aşağıdaki butonu tasarılayınız. Butonun üzerine gelince hover efekti uygulayınız.



Kaydet

# Outlines

- Outline, bir elementin çerçevelerinin de dışında olan bir çizgidir. Outline'lar Çerçevelelerden farklıdır ve CSS 5 farklı outline özelliğine sahiptir.
  - **outline-style**
  - **outline-color**
  - **outline-width**
  - **outline-offset**
  - **outline**

# Outline Styles

- dotted
- dashed
- solid
- double
- groove
- ridge
- inset
- outset
- none
- hidden

A dotted outline.

A dashed outline.

A solid outline.

A double outline.

A groove outline. The effect depends on the outline-color value.

A ridge outline. The effect depends on the outline-color value.

An inset outline. The effect depends on the outline-color value.

An outset outline. The effect depends on the outline-color value.

# Outlines

The image shows a code editor interface with two tabs: "HTML" and "CSS".

**HTML Tab:**

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <h1>CSS Outline</h1>
5 <h2>CSS Outline</h2>
```

**CSS Tab:**

```
1 h1{
2   border: 1px dotted black;
3   outline: 5px double black;
4 }
5 h2{
6   outline-style: dashed ;
7   outline-color: red;
8   outline-width: 15px;
9 }
```

To the right of the code editor, there is a preview window titled "CSS Outline". The preview shows two heading elements: an 

# and an . The has a black dotted border and a thick black double outline. The has a red dashed outline with a width of 15px.

# Outline Width ve Color

The `outline-width` property specifies the width of the outline, and can have one of the following values:

- thin (typically 1px)
- medium (typically 3px)
- thick (typically 5px)
- A specific size (in px, pt, cm, em, etc)

The `outline-color` property is used to set the color of the outline.

A thin outline.

A medium outline.

A thick outline.

A 4px thick outline.

A solid red outline.

A dotted blue outline.

An outset grey outline.

# Örnek

```
<!DOCTYPE html>
<html>
<head>
<style>
p.ex1 {outline: dashed;}
p.ex2 {outline: dotted red;}
p.ex3 {outline: 5px solid yellow;}
p.ex4 {outline: thick ridge pink;}
</style>
</head>
<body>
```

## The outline Property

```
<p class="ex1">A dashed outline.</p>
<p class="ex2">A dotted red outline.</p>
<p class="ex3">A 5px solid yellow outline.</p>
<p class="ex4">A thick ridge pink outline.</p>

</body>
</html>
```

The **outline** property is a shorthand property for setting the following individual outline properties:

- **outline-width**
- **outline-style (required)**
- **outline-color**

A dashed outline.

A dotted red outline.

A 5px solid yellow outline.

A thick ridge pink outline.

# Outline offset

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  margin: 30px;
  border: 1px solid black;
  outline: 1px solid red;
  outline-offset: 15px;
}
</style>
</head>
<body>

<h2>The outline-offset Property</h2>

<p>This paragraph has an outline 15px outside the border edge.</p>

</body>
</html>
```

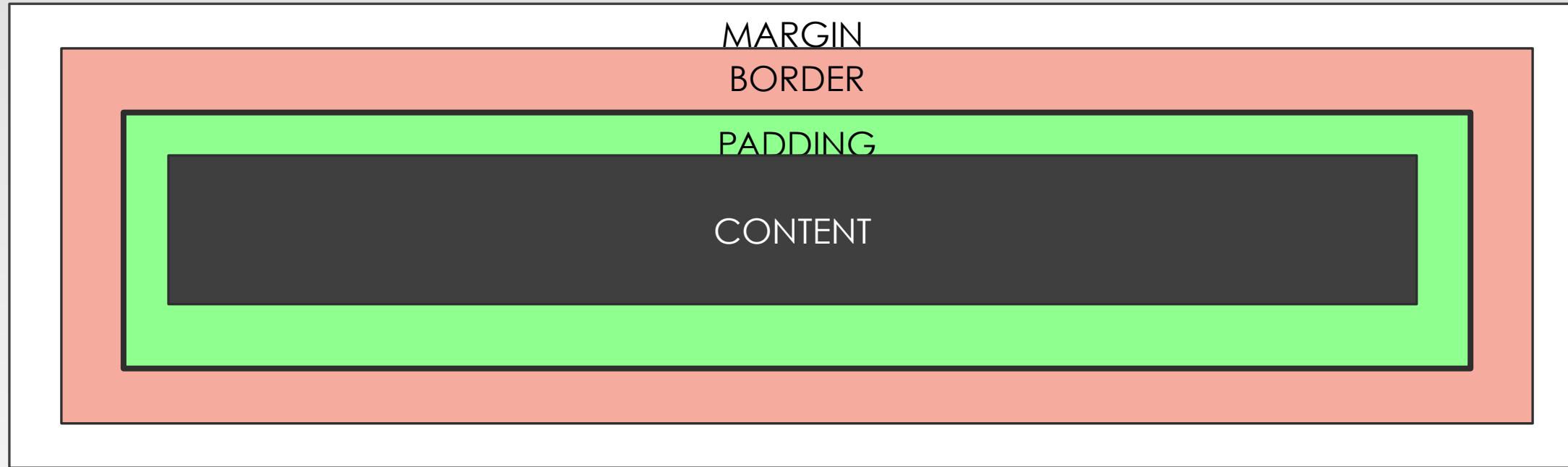
The **outline-offset** property adds space between an outline and the edge/border of an element.

The space between an element and its outline is **transparent**.

## The outline-offset Property

This paragraph has an outline 15px outside the border edge.

# Box Model



```
div {  
  width: 300px;  
  border: 15px solid green;  
  padding: 50px;  
  margin: 20px;  
}
```

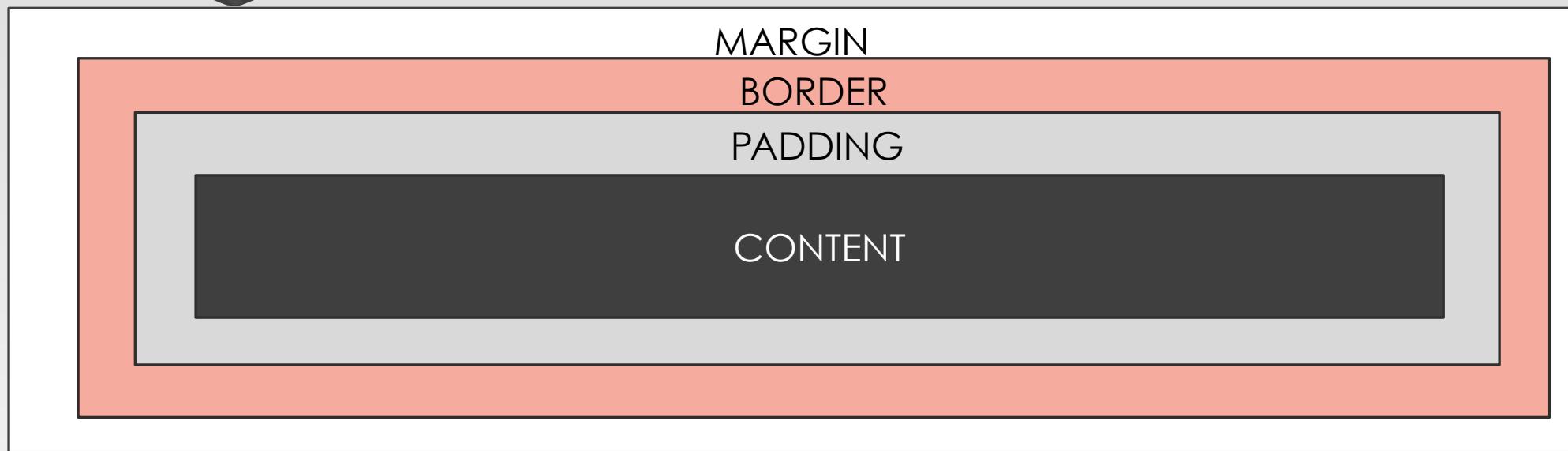
470px

GENİŞLİK = width + border + padding + margin

GENİŞLİK =  $300 + (15 \times 2) + (50 \times 2) + (20 \times 2)$

GENİŞLİK = 470 px

# Box Model



```
div {  
    width: 300px;  
    border: 15px solid green;  
    padding: 50px;  
    margin: 20px;  
}
```

box-sizing: border-box;

**340px**

GENİŞLİK = width + border + padding + margin

GENİŞLİK =  $170 + (15 \times 2) + (50 \times 2) + (20 \times 2)$

GENİŞLİK = 340 px



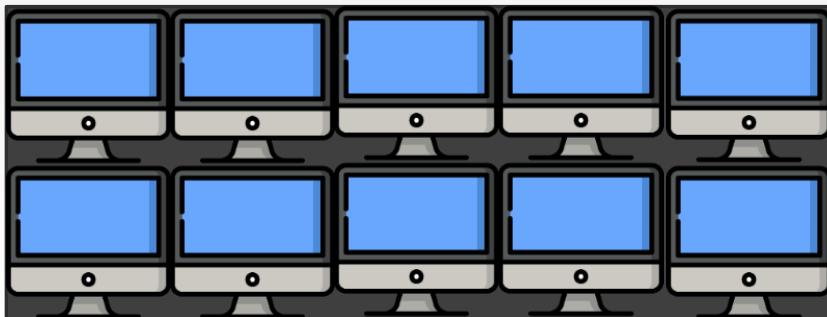
box-sizing, border-box olarak ayarlandığında elementin toplam genişlik ve yüksekliği padding ve border değerlerini de içerir.

# Zemin fotoğrafı

- Elementlere arkaplan fotoğrafı vermek için background-image stili uygulanır. Dosya adresini tanımlamak için url kelimesi kullanılır.

```
div{  
    background-image: url('img/bg.png');  
}
```

- Arkaplanın default davranışı tekrarlanarak döşenmesidir. Bu davranışı değiştirmek için **background-repeat** stili kullanılır.



Repeat: default davranış  
No-repeat: Tek bir tane yerleştirir.  
Repeat-x: yatay eksende tek sıra döşer  
Repeat-y: düşey eksende tek sıra döşer

# Zemin fotoğrafı

- Arkaplanın pozisyonunu değiştirmek için **background-position** stili kullanılır. Bu stile değer vermek için yatay için left, right, center; dikey için top, bottom, center ifadelerinin birleşimi kullanılır.

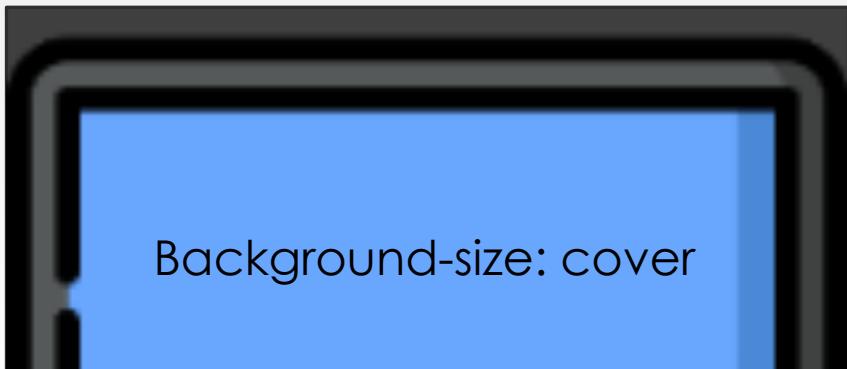
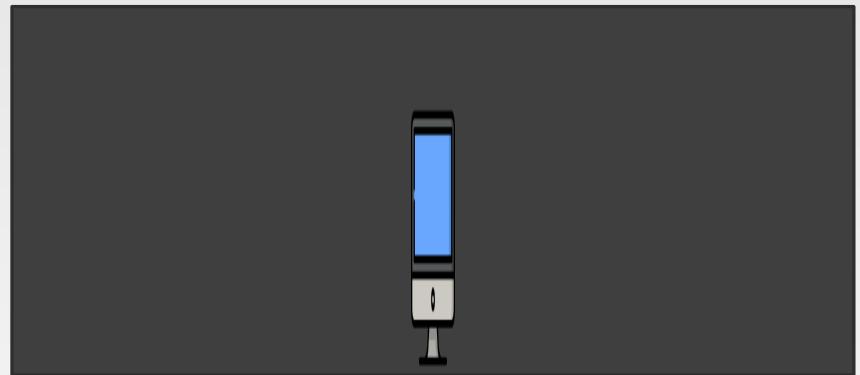
```
div{  
    background-image: url('img/bg.png');  
    background-repeat: no-repeat;  
    background-position: center bottom;  
}
```



# Zemin fotoğrafı

- Arkaplanın boyutunu değiştirmek için **background-size** stili kullanılır. Boyut için belli bir değer verilebileceği gibi cover veya contain değerlerinden biri de varilebilir.

```
div{  
    background-image: url('img/bg.png');  
    background-repeat: no-repeat;  
    background-position: center bottom;  
    background-size: 10px 80px;  
}
```



Taşıyıcının tamamını kaplayacak şekilde resmi büyütür, gerekirse resmi kırpır



Resmin tamamı gözükecek şekilde resmi taşıyıcı sınırlarına doğru genişletir.

# Zemin Fotoğrafı

Cover stilinin resmi gösterme şekli



# Practise

Aşağıdaki gibi bir slider tasarlayınız



# Zemin fotoğrafı

- Zemin fotoğrafını ekran'a sabitlemek için background-attachment stili kullanılır. Bu sayede popüler olan **parallax** efekti uygulanabilir.

```
div{  
    background-image: url ('img/bg.png');  
    background-repeat: no-repeat;  
background-attachment: fixed;  
}
```



Parallax efekti ile zemin resmi sabit tutularak kendisinden sonra gelen içeriğin onun üzerinde kayması sağlanabilir.

<https://www.invisionapp.com/inside-design/10-examples-of-parallax-scrolling/>

<https://demo.tutorialzine.com/2015/09/6-practical-examples-for-building-parallax-websites/>

# Zemin renk geçisi (linear-gradient)

- Bir elementin zeminine düzgün renk geçisi verebilmek için de background-image ve linear-gradient stili kullanılır. İstenildiği kadar renk tanımlanabilir.

```
background-image: linear-gradient(yön, renk1 %oran, renk2 oran%, ...);
```

To right | to left | to top | to bottom | xdeg ←



```
background-image: linear-gradient(yellow, violet);
```



```
background-image: linear-gradient(to right, yellow, violet);
```

# Zemin renk geçisi (linear-gradient)



```
background-image: linear-gradient(60deg, yellow, violet,  
yellow);
```



```
background-image: linear-gradient(60deg, rgba(12,200,80,0.3),  
rgba(0,112,192,1) );
```



```
background-image: linear-gradient(60deg, rgba(12,200,80,0.3)  
20%, rgba(0,112,192,1) );
```

# Zemin renk geçisi (radial-gradient)

- Bir elementin zeminine yuvarlak renk geçisi verebilmek için de background-image ve radial-gradient stili kullanılır. İstenildiği kadar renk tanımlanabilir.

```
background-image: radial-gradient(şekil, renk1 oran%, renk2 oran, ...);
```

circle | ellipse ←



```
background-image: radial-gradient(yellow, violet);
```



```
background-image: radial-gradient(circle, yellow, violet);
```



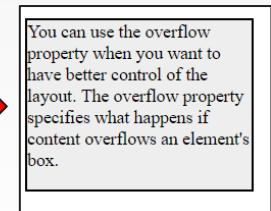
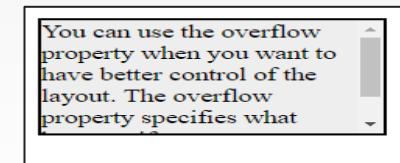
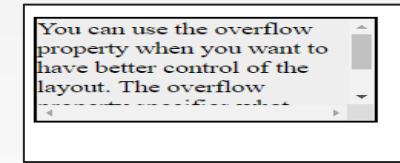
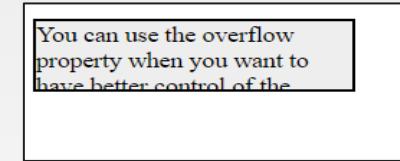
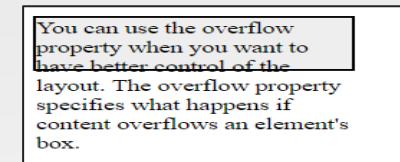
```
background-image: radial-gradient(circle, yellow 80%, violet);
```

# Overflow Property

The **overflow** property specifies whether to add scrollbars when the content of an element or to clip the content which doesn't fit in the specified area. The overflow property only works for block elements with a specified height.

The overflow property has the following values:

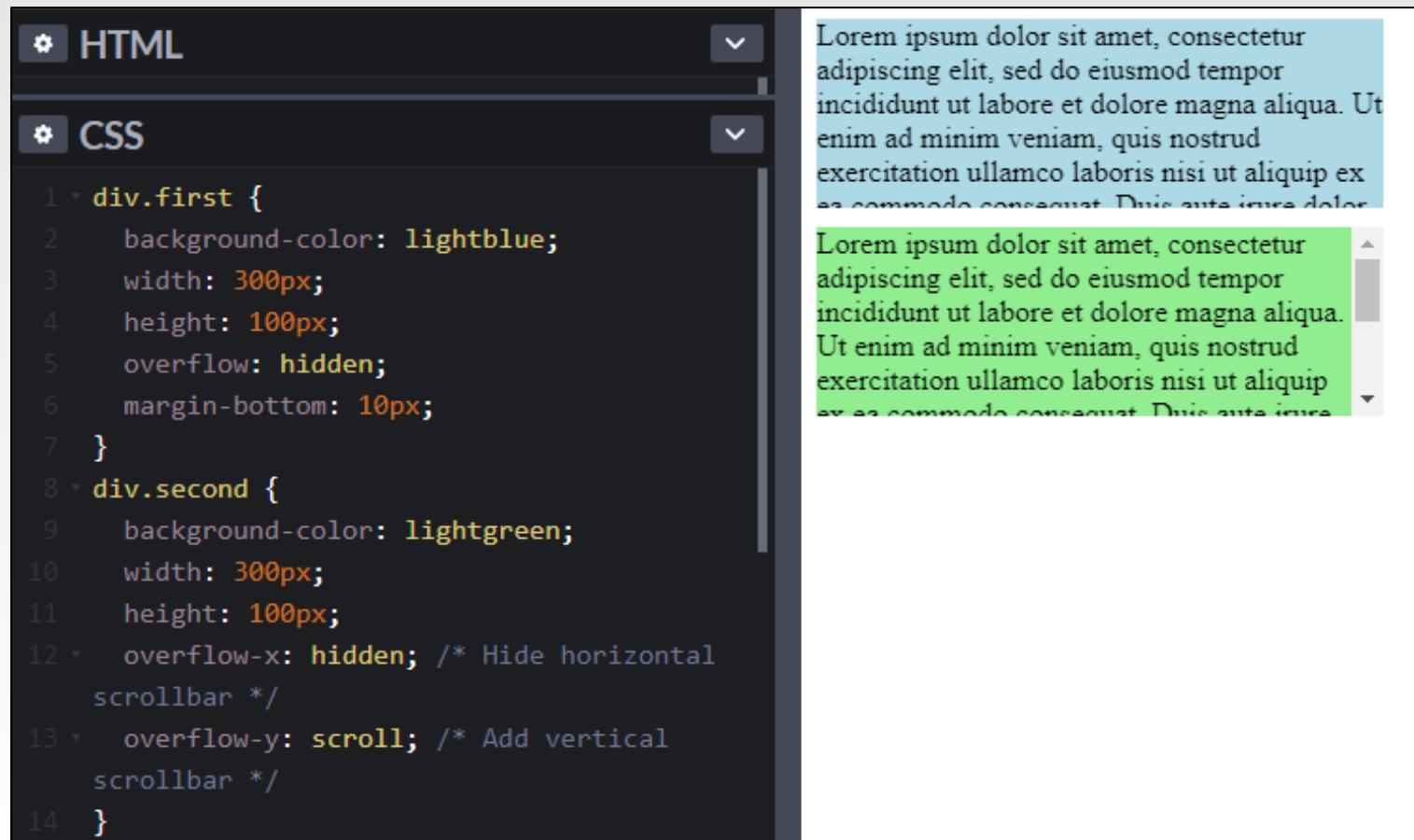
- **visible:** Default. The overflow is not clipped. The content
- **hidden:** The overflow is clipped, and the rest of the content
- **scroll:** The overflow is clipped, and a scrollbar is added to see the rest of the content.
- **auto:** Similar to scroll, but it adds scrollbars only when necessary.



# Overflow Property

overflow-x and  
overflow-y

- The overflow-x and overflow-y properties specify whether to change the overflow of content just horizontally or vertically (or both)



```
HTML
CSS
1 div.first {
2     background-color: lightblue;
3     width: 300px;
4     height: 100px;
5     overflow: hidden;
6     margin-bottom: 10px;
7 }
8 div.second {
9     background-color: lightgreen;
10    width: 300px;
11    height: 100px;
12    overflow-x: hidden; /* Hide horizontal scrollbar */
13    overflow-y: scroll; /* Add vertical scrollbar */
14 }
```

div.first {  
background-color: lightblue;  
width: 300px;  
height: 100px;  
overflow: hidden;  
margin-bottom: 10px;  
}  
  
div.second {  
background-color: lightgreen;  
width: 300px;  
height: 100px;  
overflow-x: hidden; /\* Hide horizontal scrollbar \*/  
overflow-y: scroll; /\* Add vertical scrollbar \*/  
}

div.first {  
background-color: lightblue;  
width: 300px;  
height: 100px;  
overflow: hidden;  
margin-bottom: 10px;  
}  
  
div.second {  
background-color: lightgreen;  
width: 300px;  
height: 100px;  
overflow-x: hidden; /\* Hide horizontal scrollbar \*/  
overflow-y: scroll; /\* Add vertical scrollbar \*/  
}

# Position

static

Html elemanları statik olarak, yani normal sayfa akışına göre konumlandırılırlar. Top, left, right, bottom özellikleri alamazlar. Default değer budur.

relative

Bulundukları konuma göreceli olarak top, left, right, bottom değerleri alabilirler.

fixed

Viewport büyüklüğüne göreceli olarak konumlanırlar.

absolute

En yakın parent a göre konumlanırlar. Ancak parent'i static olMAMALıdır.

sticky

Scroll pozisyonuna göre relative veya fixed olarak konumlanmadır. Top stili ile beraber kullanılır. Scroll pozisyonu top değerine gelince yapışır.

Static haricindeki pozisyonlarda elementler üst üste gelebilir. Bu durumda hangisinin üstte olacağı z-index özelliği ile belirlenir. Z-index değeri daha büyük olan üstte olur.

# Practise

Welcome

Login | Cart

# Practise

[Home](#) | [Products](#) | [About](#) | [Contact](#)

Navbar kısmı sticky olacak şekilde tasarlayınız

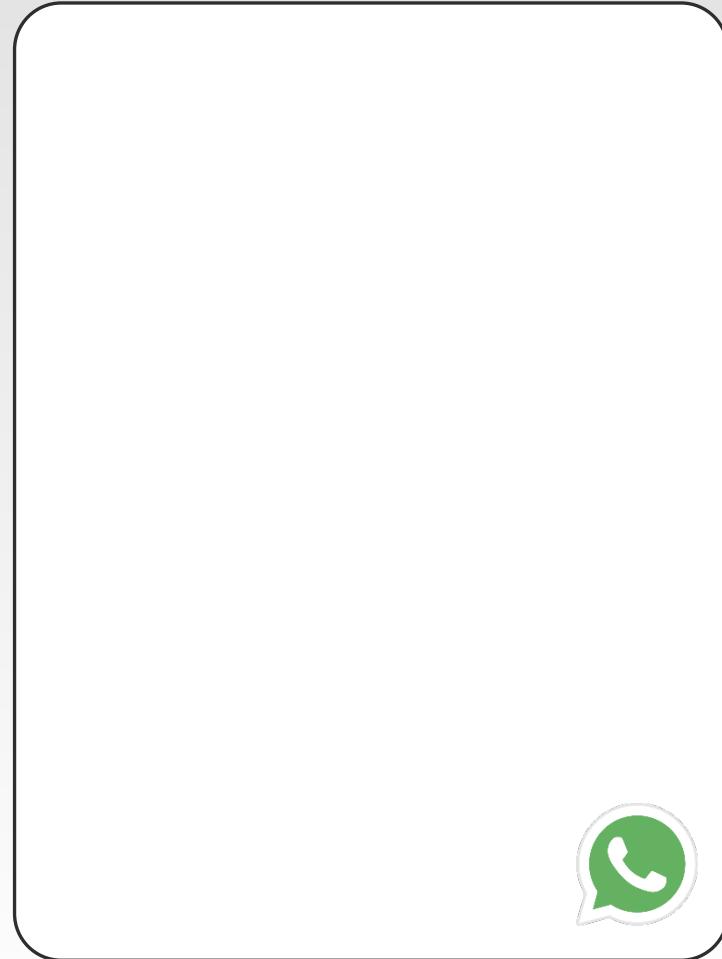
# Practise

Aşağıdaki metin kutularını yapınız.

  **Gönder**

# Practise

Yandaki gibi sayfanın sağ alt köşesine sayfa kaydırıldığında konumu değizmeyecek şekilde whatsapp ikonu yerleştiriniz.



# Dikey Hizalama (2.yöntem)

- Bir elementi başka bir element içinde aşağıya yada yukarıya hizalamak için position absolute kullanılır.

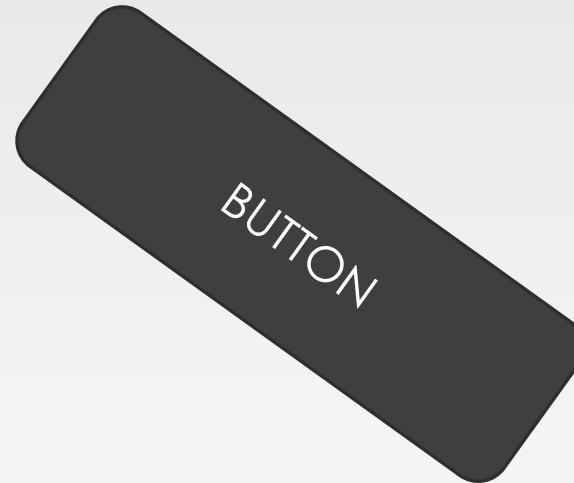
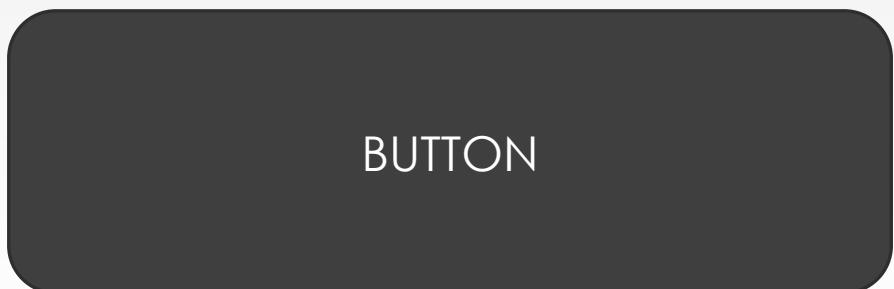
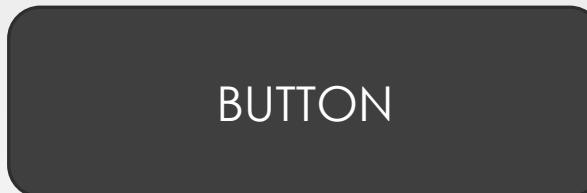


```
<div class="kutu1">  
    <div class="kutu2"></div>  
</div>
```

```
.kutu1{  
    position:relative;  
}  
  
.kutu2{  
    position:absolute;  
    bottom:0;  
    left:0;  
}
```

# Transform

**Transform** stilleri ile elementlerin konumu, boyutları değiştirilebilir ya da döndürülebilir. Genellikle **transition** ile beraber kullanılarak sayfaya bir hareket kazandırılır.



# Transform > Translate

transform: translateX(n)

Elementin **x eksenindeki** konumunu değiştirir.

transform: translateY(n)

Elementin **y eksenindeki** konumunu değiştirir.

transform: translate(x,y)

Elementin **x ve y eksenindeki** konumunu değiştirir.

# Transform > Rotate

`transform: rotateX(ndeg)`

Elementi x eksenini referans olarak döndürür.

`transform: rotateY(ndeg)`

Elementi y eksenini referans olarak döndürür.

`transform: rotate (ndeg)`

Elementi saat yönünde ya da tersi istikamette döndürür.

X eksenİ



Y eksenİ



# Transform > Scale

transform: scaleX(w)

Elementin genişliğini belirtilen oranda değiştirmek için kullanılır.

transform: scaleY(h)

Elementin yüksekliğini belirtilen oranda değiştirmek için kullanılır.

transform: scale(w, h)

Elementin ebatlarını belirtilen oranda değiştirmek için kullanılır.



Scale için verilen parametreler, oran değerlerindir.  
Örneğin 2 değeri verildiğinde mevcut büyüğün 2 katı anlamına gelir.

# Transition

- Elementlerin değişen css özelliklerini takip ederek eski değer ile yeni değer arasında yumuşak bir geçiş sağlar.

Transition 2 adımda uygulanır

1

Elementin geçerli özelliklerini tanımlamak ve transition uygulamak.

```
button{  
background-color:red;  
transition: background-color 1s;  
}
```

2

Elemente yeni css özellikler vermek

```
button: hover{  
background-color:blue;  
}
```

# Practise

Daha önce yaptığımız açılır menüyü transition ile açılır hale getirelim

## Ürünler

- Notebook
- Klavye
- Fare
- Monitör

# Transition

Transition tanımlanırken aşağıdaki kalıp kullanılır.

transition: **stil\_ismi** **süre** **tür gecikme**



Color  
Width  
Height  
Padding  
Margin  
....

Transition  
işlemi  
kaç  
saniyede  
tamaml  
anacak

**ease**: yavaş başlar, hızlanır, yavaş bitirir. (default)  
**linear**: aynı hız  
**ease-in**: yavaş başlar  
**ease-out**: yavaş biter  
**ease-in-out**: yavaş başlar ve biter

# Transition

```
transition: color 1s;  
transition: width .2s linear;  
transition: padding .5s ease-out 1s;  
transition: color 1s, background-color .5s;  
transition: padding .3s ease-in-out, color .4s linear .2s;  
transition: all .3s;
```



Transition tanımlarken aralarına virgül koyarak istenildiği kadar transition tanımlanırken, all parametresi ile tüm transition lar için ortak bir tanımlama da yapılabilir.

# Transition

```
<div>
    <div class="transform x">X
=> 50px</div>
    <div class="transform y">Y
=> 50px</div>
    <div class="transform
xy">X => 50px, Y =>-50px</div>
</div>
```

```
.transform {
    width: 100px;
    height: 50px;
    background-color: rgb(221, 100,
237);
    color: white;
    text-align: center;
    margin: 50px;
    padding: 10px;
    transition: transform 2s;
}
.x:hover {
    transform: translateX(50px);
}
.y:hover {
    transform: translateY(50px);
}
.xy:hover {
    transform: translate(50px, -50px);
}
```

# Practise

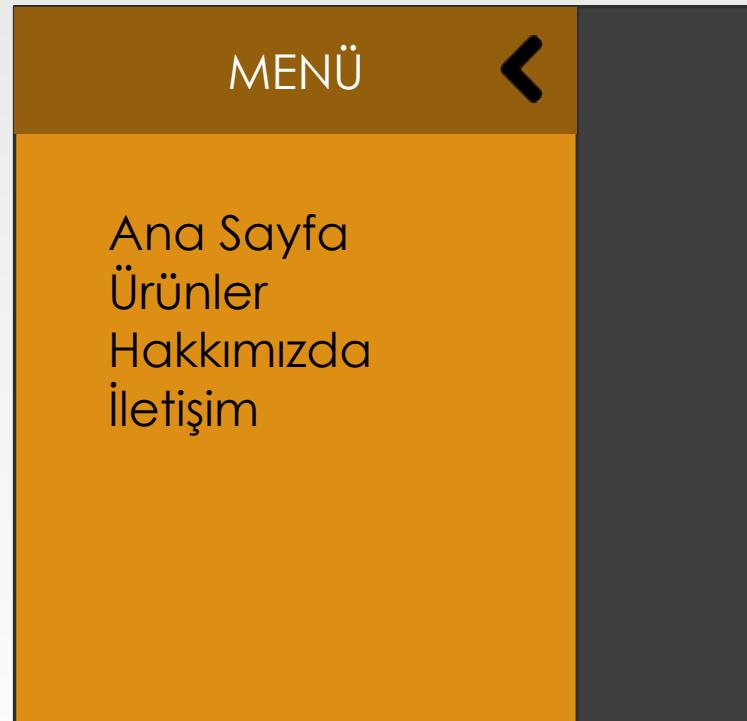
Aşağıdaki şekilde transition efektli buton tasarılayınız. Hover olduğunda arka plan rengini ve yazı rengini transition ile değiştiriniz.

Ana Sayfa

Ana Sayfa

# Practise

Soldan transition ile açılan menü tasarılayınız



# Shadow (Gölge)

Yazılara gölge vermek için

**text-shadow**

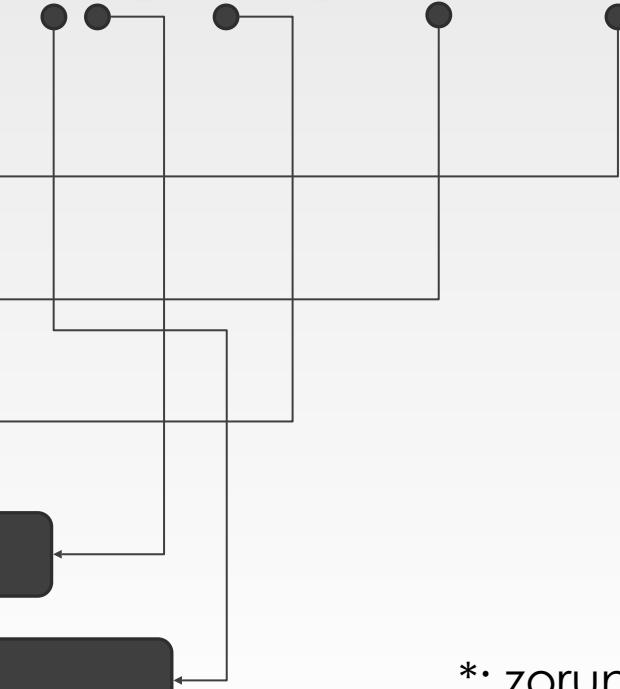
Elementlere gölge vermek için

**box-shadow**

**text-shadow: *x y blur spread color;***



**box-shadow: *x y blur spread color;***



\*: zorunlu

# Shadow (Gölge)

Yazılara gölge vermek için

**text-shadow**

```
text-shadow: 2px 3px 3px #ff0000;
```

Aralarına virgül konularak birden fazla gölge verilebilir.

```
text-shadow: 2px 3px 3px #ff0000, -2px  
-3px 3px #00ff00
```

Elementlere gölge vermek için

**box-shadow**

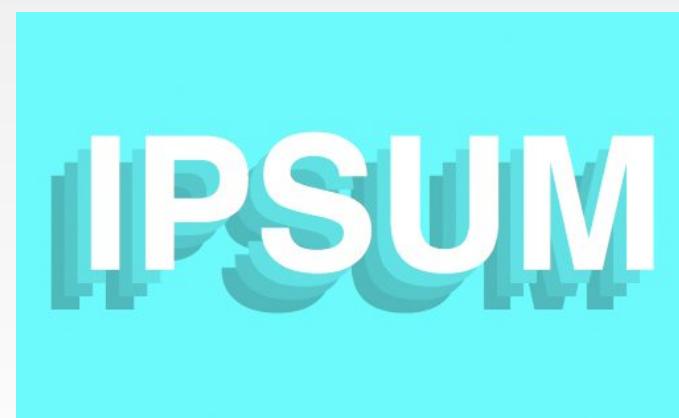
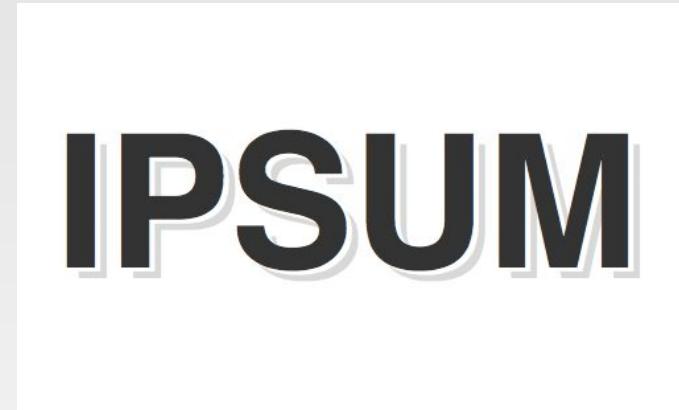
```
box-shadow: 4px 2px 3px #12ab12;
```

Aralarına virgül konularak birden fazla gölge verilebilir.

```
box-shadow: 4px 2px 3px #12ab12, 0px 0px  
5px #red;
```

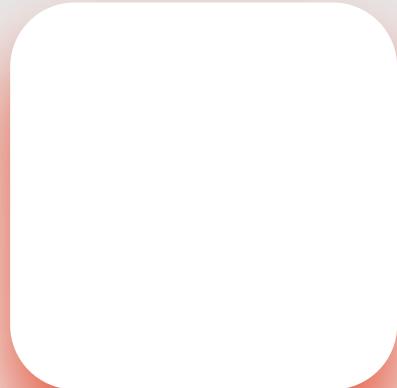
# Practise

H1 tagına aşağıdaki gibi gölgeler veriniz



# Practise

Aşağıdaki şekilde kutulara gölge veriniz.



# Değişkenler

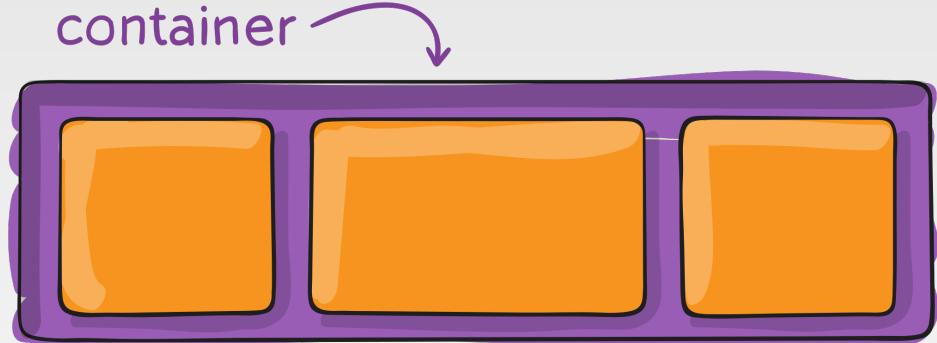
- Değişken oluşturularak css içindeki değerler tek noktadan kolaylıkla değiştirilebilir. Değişken oluşturmak için root kullanılır.

```
:root {  
    --blue: #1e90ff;  
    --white: #ffffff;  
}
```

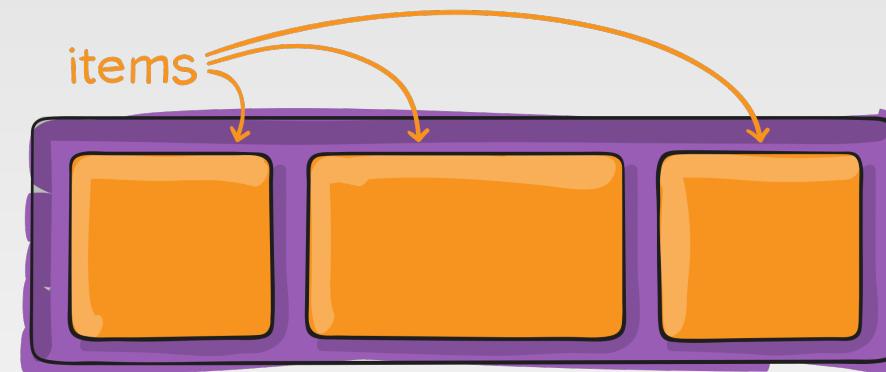
```
body {  
    background-color: var(--blue);  
}  
  
h2 {  
    border-bottom: 2px solid var(--blue);  
}  
  
.container {  
    color: var(--blue);  
    background-color: var(--white);  
    padding: 15px;  
}
```

# Flexbox

**Float** ya da **position** kullanmaksızın HTML elementlerini normal akışının dışına çıkararak konumlandırmaya yarayan bir özelliktir. **Responsive** tasarımlar oluşturmak için oldukça elverişlidir.



```
<div class="container">  
  <div></div>  
  <div></div>  
  <div></div>  
  <div></div>  
</div>
```



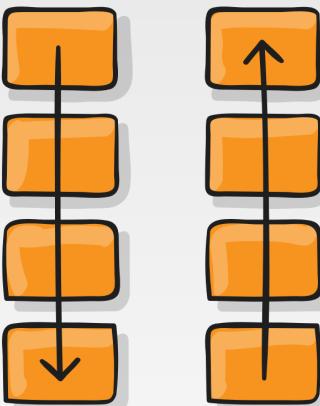
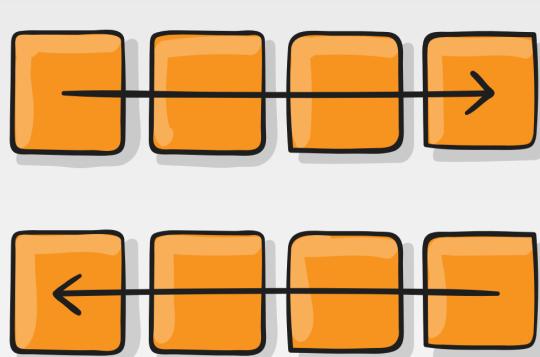
```
.container{  
  display: flex  
}  
.container div{  
  width:10rem;  
  height: 10rem;  
  background-color: #13aa45;  
  padding:2rem;  
}
```

# Flexbox (flex-direction)

flex-direction

Flex-direction container içerisindeki elementlerin dizilimin yönünü belirlemek için kullanılır.

- **Row** => yan yana, **column** => (Default değer: row)



```
.container{  
    display: flex;  
    flex-direction: column;  
}  
.container div{  
    width:10rem;  
    height: 10rem;  
    background-color: #13aa45;  
    padding:2rem;  
}
```

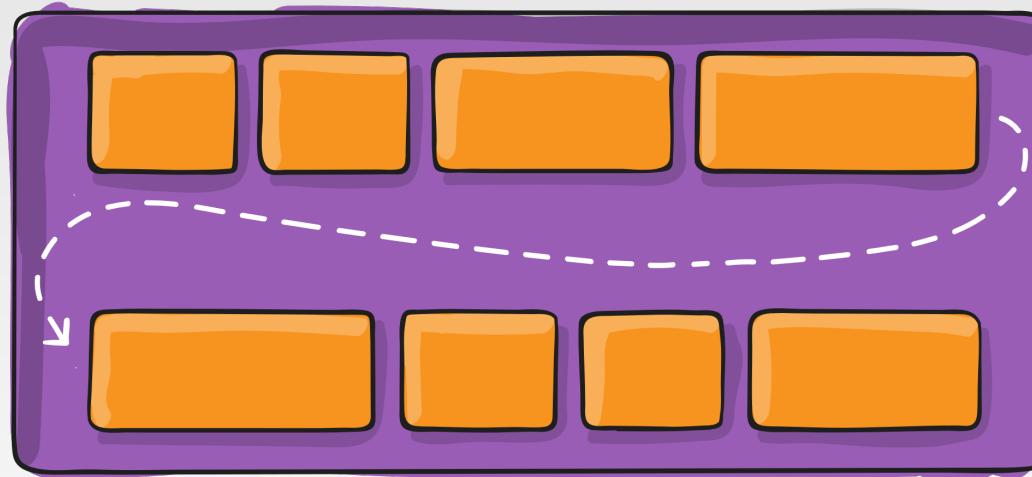
**flex-direction:** row | column | row-reverse |  
column-reverse

```
<div class="container">  
    <div></div>  
    <div></div>  
    </div>
```

# Flexbox (wrap)

flex-wrap

**flex-wrap** ile elementlerin tek satırda sıyrılmaması veya sıyrılmayanların alt satırda kaydırılması sağlanabilir. Wrap olarak verilirse sıyrılmayanlar alt satırda kaydırılır, no-wrap olursa tek satırda sıyrılr.



```
<div class="container">  
  <div></div>  
  <div></div>  
  <div></div>  
  <div></div>  
</div>
```

```
.container{  
  display: flex;  
  flex-wrap: wrap;  
}  
  
.container div{  
  width:10rem;  
  height: 10rem;  
  background-color: #13aa45;  
  padding:2rem;  
}
```

# Flexbox (justify-content)

justify-content

Flex içindeki elemanları yatayda hizalamak için kullanılır

**justify-content:** flex-start | flex-end | center | space-between | space-around

```
.container{  
    display: flex;  
    justify-content: flex-start;  
}
```

```
<div class="container">  
    <div></div>  
</div>
```

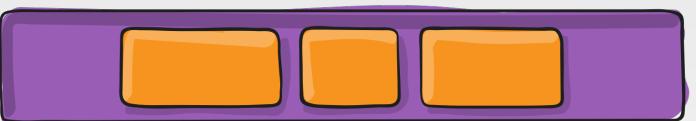
flex-start



flex-end



center



space-between



space-around



space-evenly



# Flexbox (align-items)

align-items

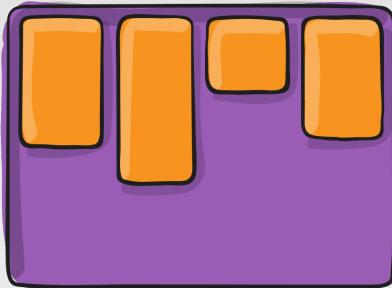
Flexbox içerisindeki elemanları **dikye**de hizalamak için kullanılır

align-items: stretch | center | flex-start | flex-end | baseline

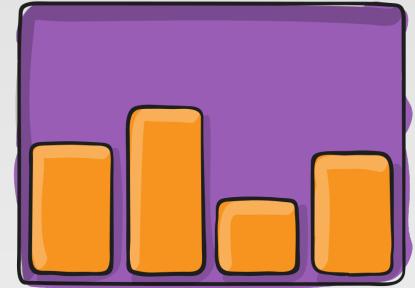
```
.container{  
    display: flex;  
    align-items : center;  
}
```

```
<div class="container">  
    <div></div>  
</div>
```

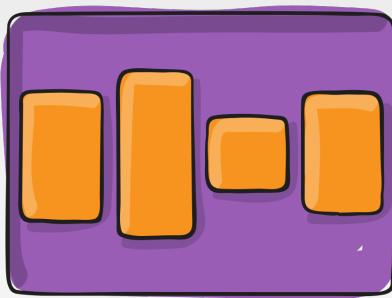
flex-start



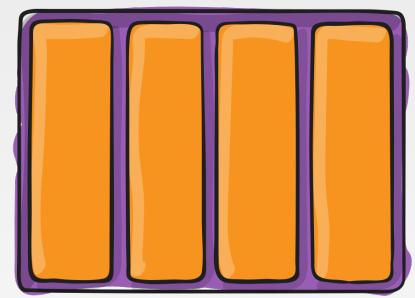
flex-end



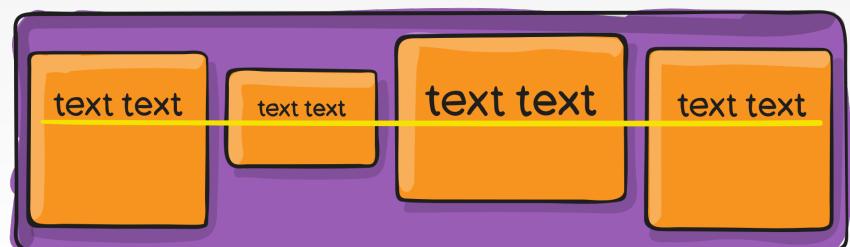
center



stretch



baseline



# Flexbox (align-content)

align-content

Flexbox içerisindeki **çok satırlı** elemanları grup olarak **dikeyde** hizalamak için kullanılır

align-content: stretch | center | flex-start | flex-end | space-between | space-around

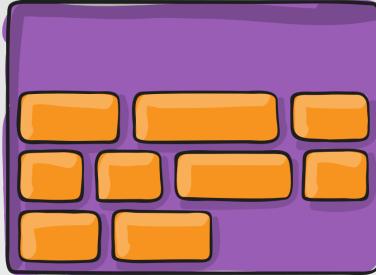
```
.container{  
    display: flex;  
    align-content : center;  
}
```

```
<div class="container">  
    <div></div>  
</div>
```

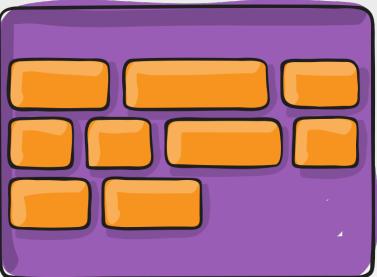
flex-start



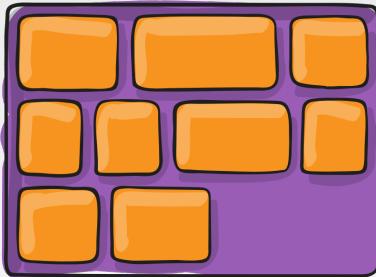
flex-end



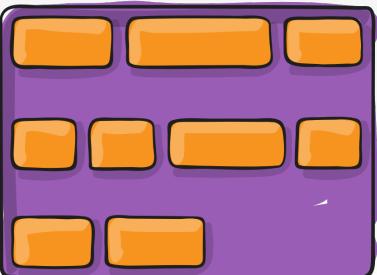
center



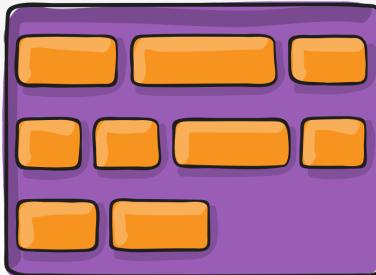
stretch



space-between



space-around

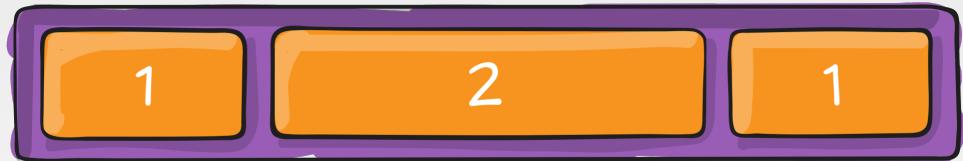


# Flexbox (flex-grow)

flex-grow

Aynı flexbox içerisindeki elemanların diğer elemanlara göre **büyütme** oranını belirler. Büyümenin gerçekleşmesi için satırda yer açılması gereklidir.

**flex-grow: 2**



```
.container{  
    display: flex;  
    flex-grow: 1;  
}  
  
.container div:nth-child(2) {  
    flex-grow: 2  
}
```

```
<div class="container">  
    <div></div><div></div>  
    <div></div>  
</div>
```

# Flexbox (flex-shrink)

flex-shrink

Satır içerisinde yeterli alan olmaması durumunda bir elementin diğerlerine oranla **küçültme** miktarını belirler. Default değeri 1'dir.

flex-shrink : 2



```
.container{  
    display: flex;  
}  
  
.container div:nth-child(2) {  
    flex-shrink: 2  
}
```

```
<div class="container">  
    <div></div><div></div>  
    <div></div>  
</div>
```

# Flexbox (flex-basis)

flex-basis

Büyüme  
oranı kapalı

Küçülme  
oranı kapalı

Satır içerisindeki var olan  
genişliği ihtiyacı olana  
elemanlara göre  
otomatik paylaştır.

- Flexbox içerisindeki satırda yeterli alan olması durumunda öğelerin alabileceği maksimum genişliği belirlemek için kullanılır.
- Değer olarak tüm uzunluk birimleri (px, pt, em vb.) kullanılabilir. Bu şekilde bir değer verilirse bir elemanın alabileceği uzunluk bu rakam kadar olacaktır.
- Varsayılan olarak değeri **auto**'dur. Eğer auto olarak kalırsa var olan alan, ihtiyacı olana elemana tahsis edilir.
- Değer 0 belirlenirse, her bir elemanın büyütüklüğü sadece içeriğinin büyütüklüğü kadar olur.

```
div{  
    flex-grow: 0;  
    flex-shrink: 1;  
    flex-basis: auto;  
}
```

```
div{  
    flex: 0 1 auto;  
}
```

# Flexbox (gap)

gap

flexbox içerisindeki elemanların arasındaki boşlukları belirlemek için kullanılır.

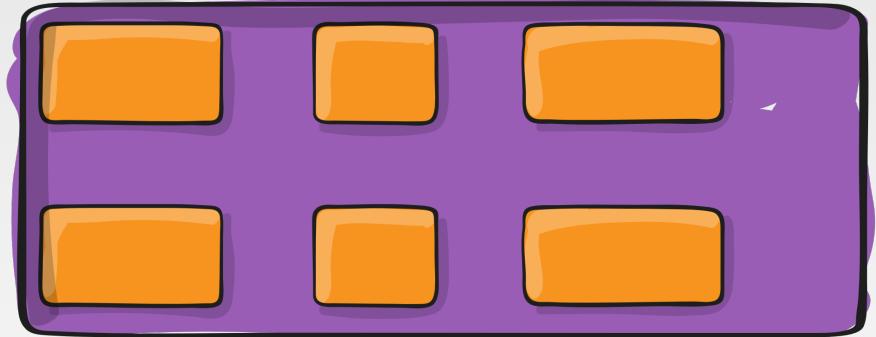
```
.container{  
    display: flex;  
    gap :10px 30px  
}
```

```
<div class="container">  
    <div></div><div></div>  
    <div></div>  
</div>
```

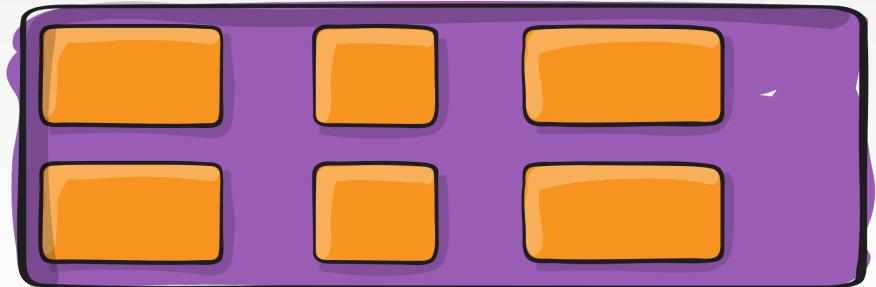
gap: 10px



gap: 30px



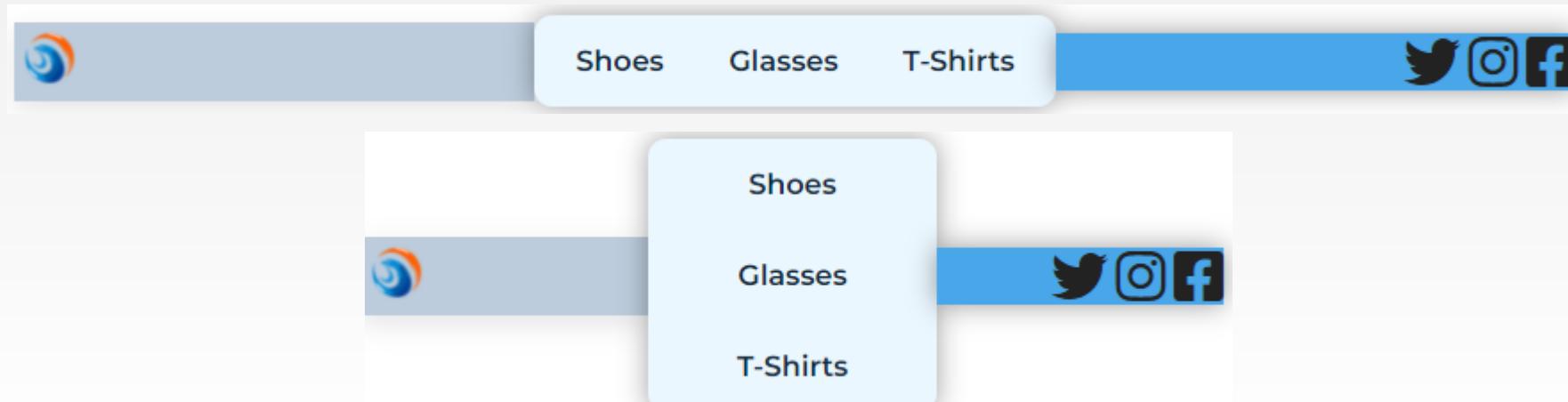
gap: 10px 30px



# Practise

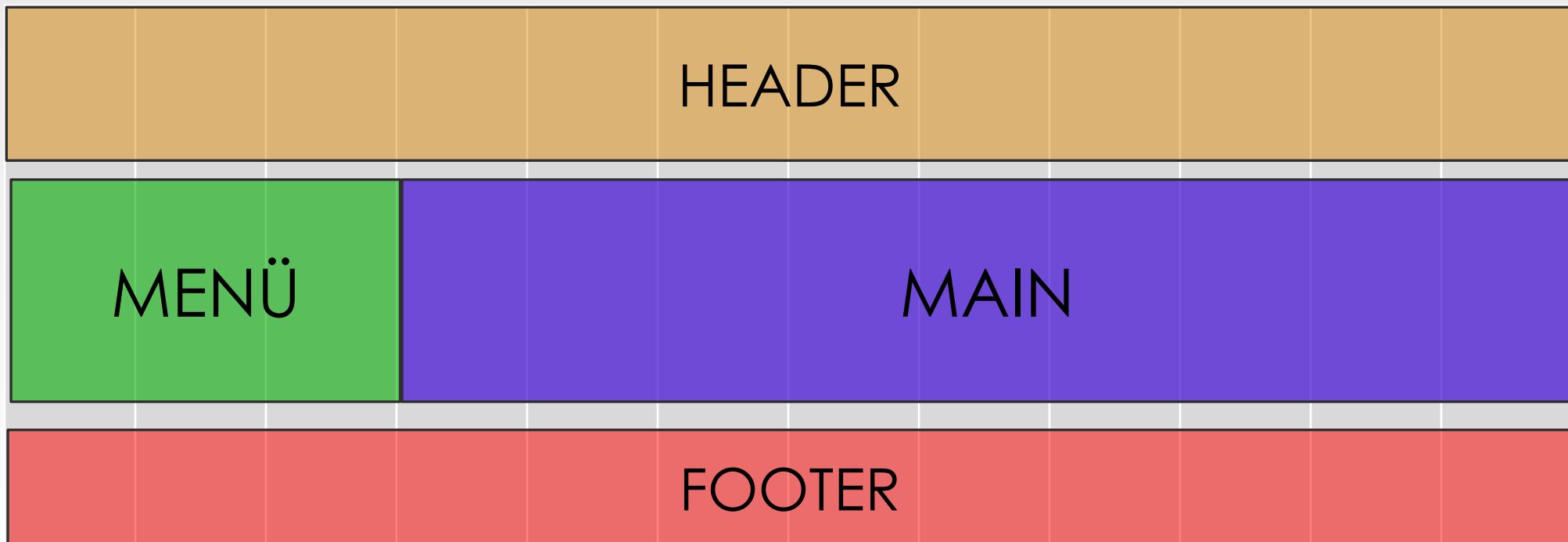
Flexbox kullanarak aşağıdaki gibi bir menü tasarlaymentınız.

- Ekran küçüldüğünde Ortadaki Menü Elemanları 2.şekilde görüldüğü gibi **aşağıya** kayabilmelidir.
- Orta menüdeki linklerin üzerine gelince **hover ve transition** ile **yazı** ve **arka-plan** rengi değişimelidir.
- Sağdaki ikonlara gelindiğinde ise yazı karakteri yeterli ölçüde **büyütücek** bir animasyon yapılmalıdır.
- Uygulamadaki renkler için daha önceki derste yaptığımız **renk teması** css'i kullanılmalıdır.



# Flex Grid

- Web sayfalarında istenilen elemanı istenilen yere koymak için izgara (grid) yöntemi kullanılır. Grid yöntemi ile sayfa yatayda **12 eş alana** bölünür. Bu alanların genişlikleri toplamı %100 olmalıdır. Viewport değişikçe bunların ebatları daralır ya da genişler.



# Flex Grid

1

Tüm elementler için box-sizing özelliği border-box yapılır.

```
* {  
    box-sizing: border-box;  
}
```

# Flex Grid

2

Satırları bir arada tutacak  
container classı tanımlanır.

```
.container{  
    width:100%  
}
```

3

Hücleri bir arada tutacak  
row classı tanımlanır.

```
.row {  
    display: flex;  
    flex-wrap: wrap;  
}
```

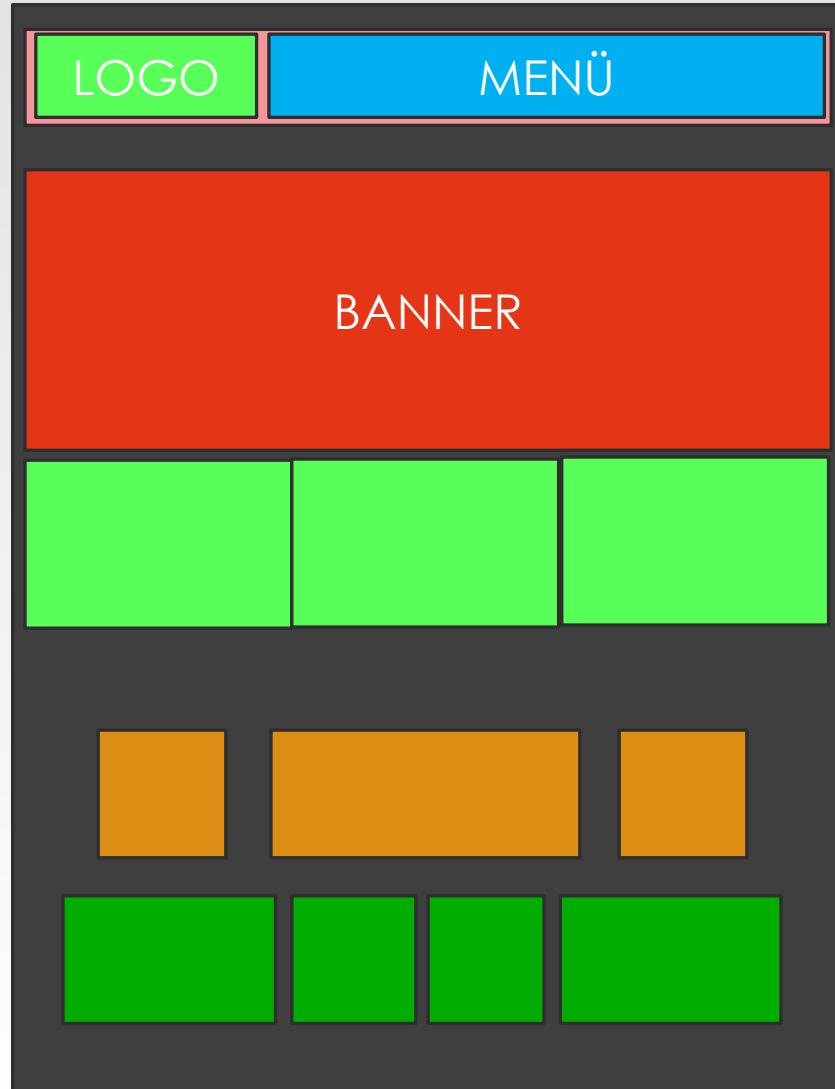
# Flex Grid

4

12 tane hücre tanımlaması yapılır.  
Bu hücrelere bir genişlik verilir.  
Genişlik değerleri  $100 / 12$   
hesaplaması sonucu elde edilir. 1'lik  
hücrenin genişliği 8.33 olarak  
ayarlanır.

```
.col {  
    flex: 0 0 auto;  
    width: 8.333333%;  
}  
.col-2 {  
    flex: 0 0 auto;  
    width: 16.666667%;  
}  
  
.col-3 {  
    flex: 0 0 auto;  
    width: 25%;  
}  
...  
.col-12{  
    flex: 0 0 auto;  
    width: 100%;  
}  
  
...
```

# Practise

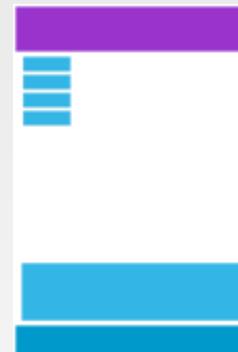


# Responsive

- Bir sitenin masaüstü, table ve telefon gibi cihazların tamamında güzel gözükmeye ve aynı işlevselligi sunmasına responsive denir. Bir sitenin responsive olması HTML ve CSS ile sağlanır.



Desktop



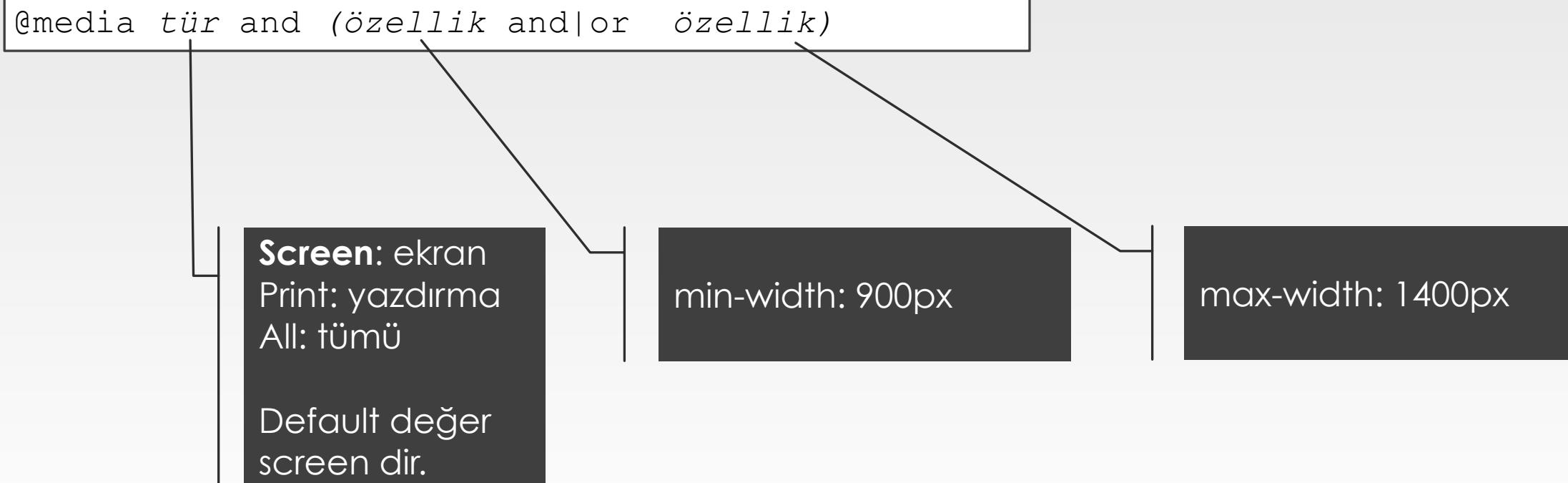
Tablet



Phone

# Media Sorguları

Responsive bir tasarım için medya sorguları kullanılır. Medya sorguları ile farklı büyüklükteki ekranlara özel CSS tanımlamaları yapılabilir.

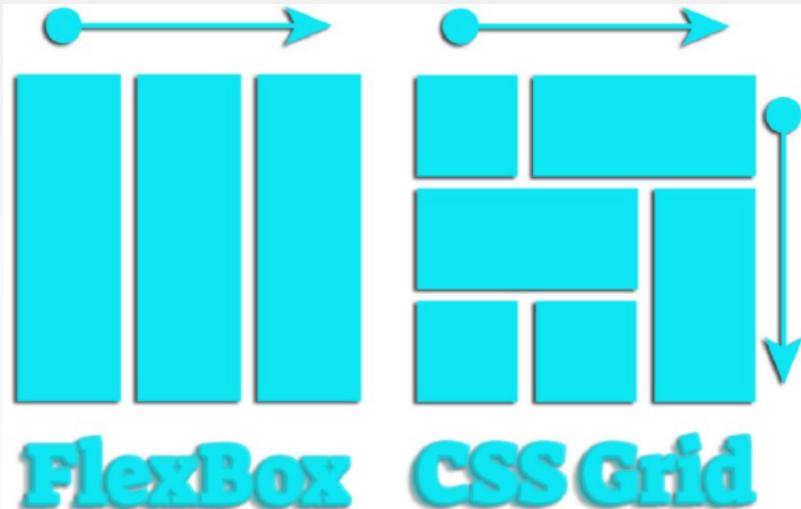


# Media Sorguları

```
//Çok küçük cihazlar → 3xküçük  
// Telefonlar @media (min-width: 376px) { ... } → sm  
// Tabletler @media (min-width: 768px) { ... } → md  
// Masaüstü @media (min-width: 992px) { ... } → lg  
// Masaüstü (Büyük) @media (min-width: 1200px) { ... } → xl  
// Masaüstü (Daha büyük) @media (min-width: 1400px) { ... } → xxl
```

# CSS Grid

- CSS Grid Layout, is **a two-dimensional** grid-based layout system.
- CSS has always been used to layout our web pages
  - **floats, positioning, inline-block**, but all of these methods were essentially hacks and left out a lot of important functionality (vertical centering, for instance).
  - Flexbox is also a very great layout tool, but its **one-directional** flow has different use cases — and they actually work together quite well!



For Details:

<https://css-tricks.com/snippets/css/complete-guide-grid/#aa-grid-item>

# CSS Grid-Terminology

Grid container

The element on which **display: grid** is applied. It's the direct parent of all the grid items. In this example container is the grid container.

```
HTML  

```

Grid Item

The children (i.e. direct descendants) of the grid container. Here the item elements are grid items, but sub-item isn't.

```
HTML  

```

# CSS Grid-Terminology

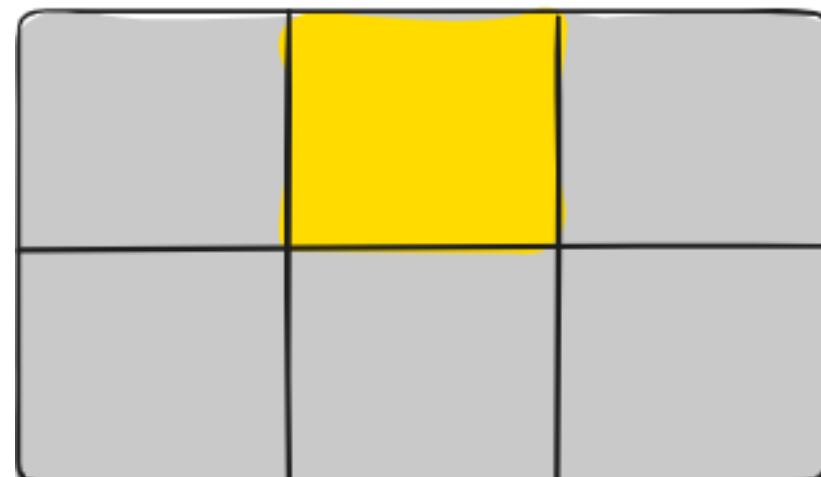
## Grid Line

The dividing lines that make up the structure of the grid. They can be either vertical ("column grid lines") or horizontal ("row grid lines") and reside on either side of a row or column. Here the yellow line is an example of a column grid line.



## Grid Cell

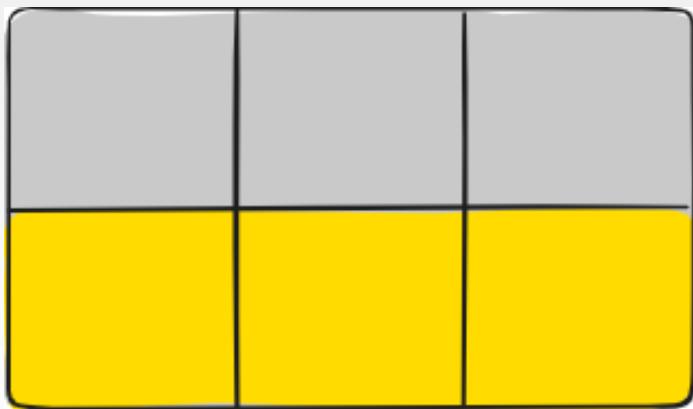
The space between two adjacent row and two adjacent column grid lines. It's a single "unit" of the grid. Here's the grid cell between row grid lines 1 and 2, and column grid lines 2 and 3.



# CSS Grid-Terminology

## Grid Track

The space between two adjacent grid lines. You can think of them as the columns or rows of the grid. Here's the grid track between the second and third-row grid lines.



## Grid Area

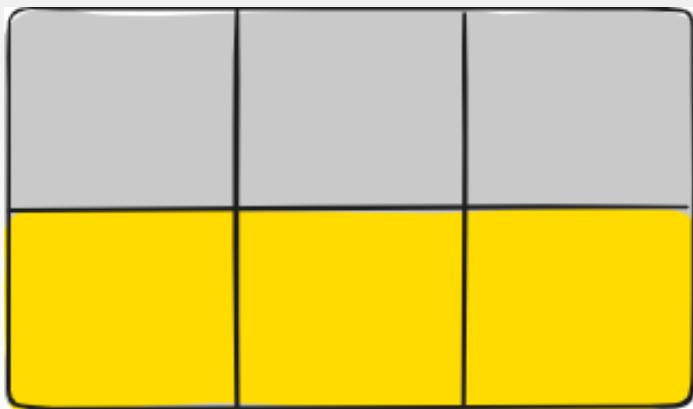
The total space surrounded by four grid lines. A grid area may be composed of any number of grid cells. Here's the grid area between row grid lines 1 and 3, and column grid lines 1 and 3.



# CSS Grid-Terminology

## Grid Track

The space between two adjacent grid lines. You can think of them as the columns or rows of the grid. Here's the grid track between the second and third-row grid lines.



## Grid Area

The total space surrounded by four grid lines. A grid area may be composed of any number of grid cells. Here's the grid area between row grid lines 1 and 3, and column grid lines 1 and 3.



# Grid Container (display:grid)

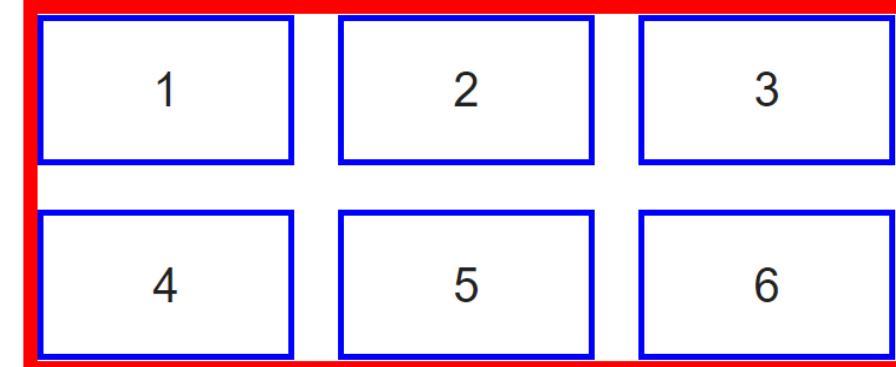
- Grid container : The container that holds the entire CSS grid

```
.container {  
  display: grid | inline-grid;  
}
```

css

- **grid** – generates a block-level grid
- **inline-grid** – generates an inline-level grid

```
.container {  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr;  
  gap: 15px;  
  
  margin: 50px auto;  
  max-width: 300px;  
  border: 5px red solid;  
  width: auto;  
}
```



# Grid Container (grid-template-column | rows)

Defines the **columns** and **rows** of the grid with a space-separated list of values. The values represent the track size, and the space between them represents the grid line.

```
css  
.container {  
    grid-template-columns: 1fr 1fr 1fr;  
}
```

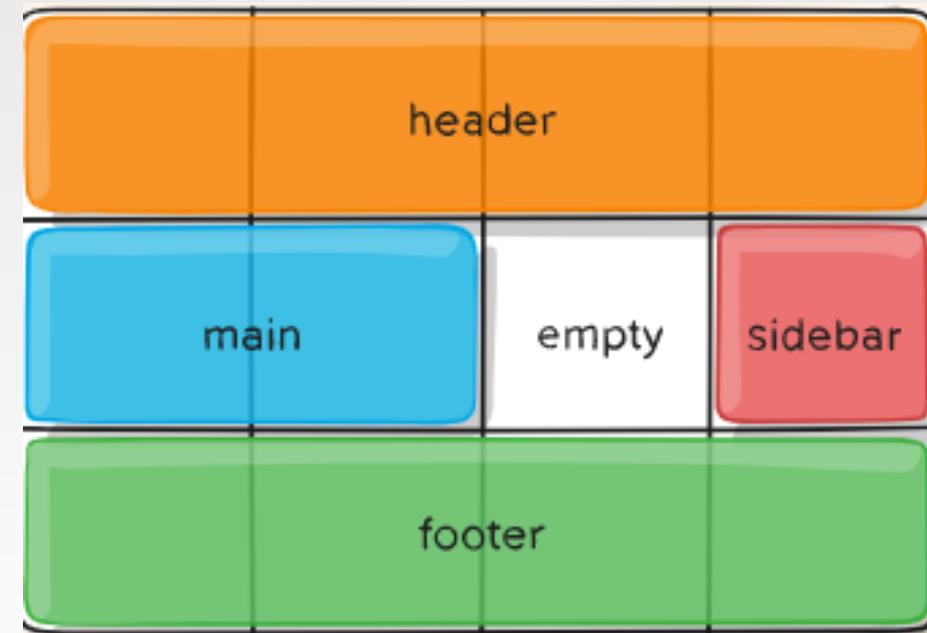
```
css  
.container {  
    grid-template-columns: 1fr 50px 1fr 1fr;  
}
```

```
css  
.container {  
    grid-template-columns: ... ...;  
    /* e.g.  
       1fr 1fr  
       minmax(10px, 1fr) 3fr  
       repeat(5, 1fr)  
       50px auto 100px 1fr  
    */  
    grid-template-rows: ... ...;  
    /* e.g.  
       min-content 1fr min-content  
       100px 1fr max-content  
    */  
}
```

# Grid Container (grid-template-areas)

Defines a grid template by referencing the names of the grid areas which are specified with the grid-area property. Repeating the name of a grid area causes the content to span those cells. A period signifies an empty cell. The syntax itself provides a visualization of the structure of the grid.

```
.item-a {  
  grid-area: header;  
}  
.item-b {  
  grid-area: main;  
}  
.item-c {  
  grid-area: sidebar;  
}  
.item-d {  
  grid-area: footer;  
}  
  
.container {  
  display: grid;  
  grid-template-columns: 50px 50px 50px 50px;  
  grid-template-rows: auto;  
  grid-template-areas:  
    "header header header header"  
    "main main . sidebar"  
    "footer footer footer footer";  
}
```



# Responsive

**VIEWPORT:** Bir web sayfasında kullanıcının görüntüleyebildiği alandır. Viewport'lar cihaza göre farklılık gösterir; mobil cihazlarda daha küçük ebatta iken, masaüstü bilgisayarlarda ise ebatı büyütür. Viewport ekran boyutu değildir, tarayıcının görünür alanıdır.

Eskiden sadece masaüstü bilgisayarlar vardı ve tasarımlar sadece ona göre yapıliyordu. Sonraları cep telefonlar ve tabletler de çıkışınca, web sitelerini düzgün göstermek için tarayıcılara içeriği sığdırma özelliği getirildi. Bu özellik web sitesini ölçekleyip daha küçük viewport larda aynı görüntüyü elde etmeyi amaçladı. Ancak bu sefer nesneler çok küçük kaldı, yazılar okunamamaya başladı.

# Responsive



# Responsive

1

Web sitelerinin responsive yapılması için ilk adım, tarayıcıların içerik sığdırma özelliklerinin devre dışı bırakılmasıdır. Bunun için HTML5 deki aşağıdaki ifade kullanılır.

```
<meta name="viewport"  
content="width=device-width, initial-scale=1.0">
```



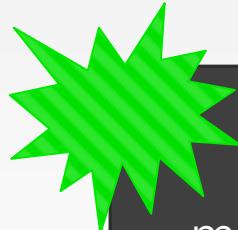
# Responsive

2

Fotoğraf ve videoların farklı viewport larda düzgün gözükmesi için default genişlik ve yükseklikleri göreceli olarak belirlenir.

```
img {  
    max-width: 100%;  
    height: auto;  
}
```

```
video {  
    max-width: 100%;  
    height: auto;  
}
```



Eğer içerik max-width ten büyük ise içerik max-width genişliğine eşitlenir. Eğer içerik max-width ten küçük ise mevcut büyüğünü ile gösterilir.

# Responsive

3

Farklı viewport lar için media query ler yazılır. Öncelikle grid için farklı viewport lara göre class lar oluşturulur.

Bunun için ilgili medya soruguları içinde col-sm, col-md, col-lg, col-xl, col-xxl sınıfları oluşturulur ve stiller tanımlanır.

# Responsive

```
@media (min-width: 576px) {  
    .col-sm-1 {  
        flex: 0 0 auto;  
        width: 8.333333333%;  
    }  
  
    .col-sm-2 {  
        flex: 0 0 auto;  
        width: 16.666666667%;  
    }  
    ....  
}
```

```
@media (min-width: 768px) {  
    .col-md-1 {  
        flex: 0 0 auto;  
        width: 8.333333333%;  
    }  
  
    .col-md-2 {  
        flex: 0 0 auto;  
        width: 16.666666667%;  
    }  
    ...  
}
```

```
@media (min-width:992px) {  
    .col-lg-1 {  
        flex: 0 0 auto;  
        width: 8.333333333%;  
    }  
  
    .col-lg-2 {  
        flex: 0 0 auto;  
        width: 16.666666667%;  
    }  
    ...  
}
```

# Practise

Lorem

Hakkında

Haberler

Kategori

Referanslar

Bağış

