

# Bootstrap





# Bootstrap nedir?



Responsive web sayfaları oluşturmak için kullanılan, içinde hazır html, css ve javascript kütüphaneleri içeren ücretsiz bir framework tür.

Bootstrap çok yaygın olarak kullanılmaktadır.

<https://w3techs.com/>

Bootstrap ile hazırlanan bir web sayfası çalıştığı tarayıcıya uyum sağlar.

<https://getbootstrap.com/>

Siteyi açıp incele



# Bootstrap kurulumu

Bootstrap bir web sayfasına 2 şekilde eklenebilir.

## CDN kullanım

- **CDN sunuculara referans verilerek kullanılır**
- Sitenin kaynak kullanımı azaltılmış olur
- Daha kolay eklenir.

## Lokal kullanım

- **Web sitesinin içine yerleştirerek kullanılır**
- Tasarım aşamasında offline çalışılabilir
- Dosyaların içinde değişiklik yapılabilir



# CDN bağlantısı

Sayfanın head bölümüne bootstrap css bağlantısı yapılır. Js bağlantısı ise asyfaının altında body etiketi kapanmadan hemen önce gerçekleştirilir.

```
<html>
<head>

</head>
<body>

</body>
</html>
```

Starter template kullanılarak pratik şekilde bootstrap sayfa şablonu eklenebilir.



# Bootstrap kullanımı

- Bootstrap in hazır class ları kullanılarak web sayfasında biçimlendirme işlemleri yapılabilir.

```
<h1 class="display-1">Merhaba</div>
```

```
<div class="text-center">Merhaba</div>
```

```
<p class="bg-danger">...</p>
```

```
<table class="table">...</table>
```




# Grid

Bootstrap te flex kullanılarak hazırlanmış 12 lik ızgara yapısı bulunmaktadır. Responsive tasarımlar bu ızgara üzerine inşa edilir.

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

```
<div class="container">
  <div class="row">
    <div class="col">1</div>
    <div class="col">2</div>
    <div class="col">3</div>
    ...
    <div class="col">12</div>
  </div>
</div>
```



Col class ı ile bir satıra sığdığı kadar hücre eklenebilir. Hücre genişlikleri birbirine eşit olur.



# Grid Container

## container

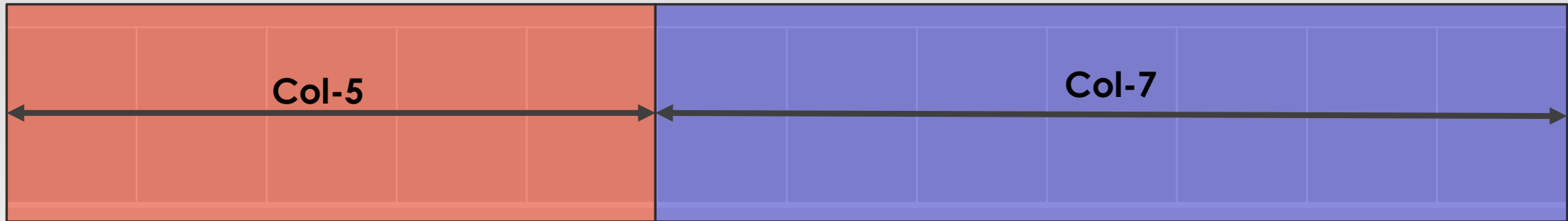
Sol ve sağ kenardan belli oranlarda boşluk bırakacak şekilde bir taşıyıcı oluşturur.

## container-fluid

Soldan ve sağdan boşluk kalmayacak şekilde bir taşıyıcı oluşturur.



# Grid



```
<div class="container">  
  <div class="row">  
    <div class="col-5">...</div>  
    <div class="col-7">...</div>  
  </div>  
</div>
```

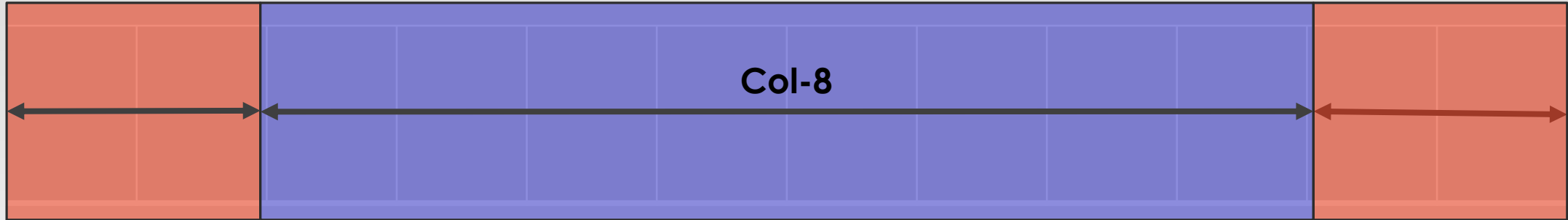
Col-x ifadesi ile oluşturulan hücrelerin ne kadar yer kaplayacağına karar verilebilir

Col-auto ile hücre genişliği, hücredeki içeriğin boyutuna eşitlenir.





# Grid



```
<div class="container">  
  <div class="row">  
    <div class="col">...</div>  
    <div class="col-8">...</div>  
    <div class="col">...</div>  
  </div>  
</div>
```

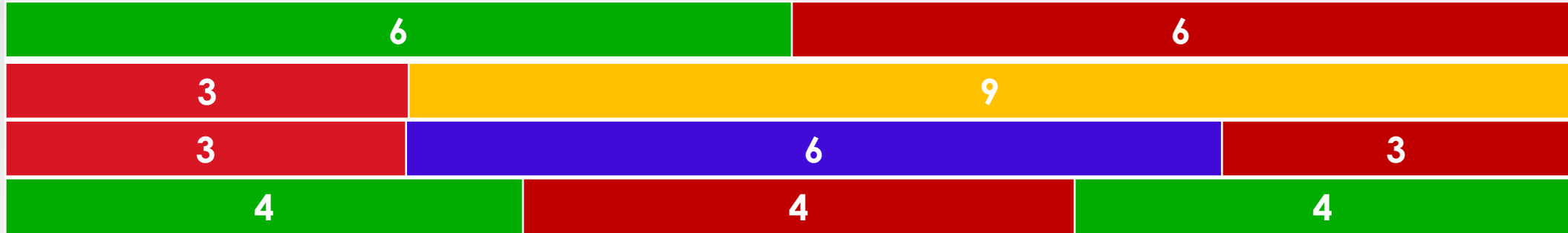
Büyüklik ifadesi kullanılan hücreler haricindeki hücreler eşit bir şekilde satıra yayılırlar



# Practice

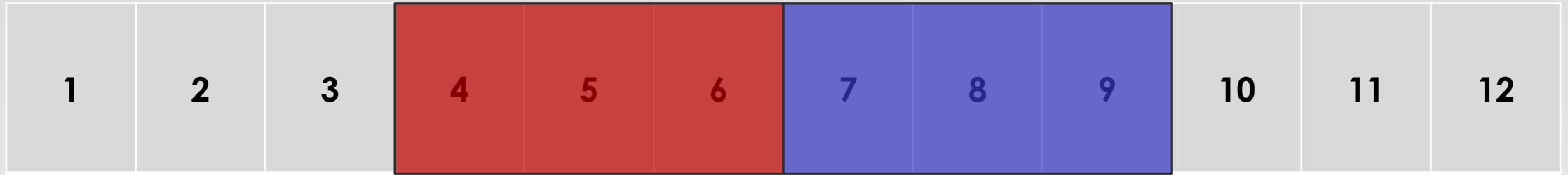
10 dk

Aşağıdaki ızgara yapısını oluşturunuz





# Grid



```
<div class="container">
  <div class="row justify-content-center">
    <div class="col-4">...</div>
    <div class="col-4">...</div>
  </div>
</div>
```

justify-content-**X**

Start  
Center  
End  
Around  
Between



# Breakpoints

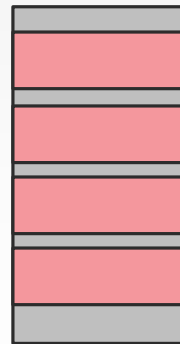
Breakpoint	Class infix	Dimensions
X-Small	None	< 576px
Small	sm	≥ 576px
Medium	md	≥ 768px
Large	lg	≥ 992px
Extra large	xl	≥ 1200px
Extra extra large	xxl	≥ 1400px



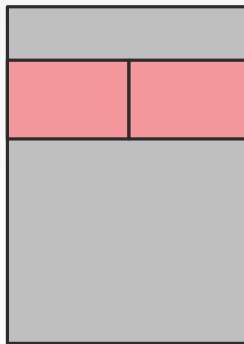
# Grid - Breakpoints

Farklı çözünürlüklere göre ızgaramızın yerleşimini ayarlayabiliriz. Bunun için breakpoint lerden faydalanırız.

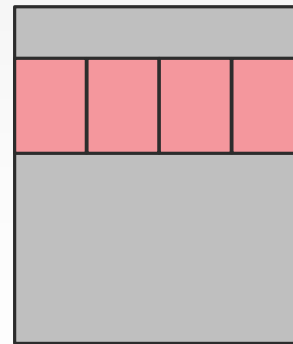
```
<div class="container">
  <div class="row">
    <div class="col-12 col-sm-6 col-md-3">...</div>
    <div class="col-12 col-sm-6 col-md-3 ">...</div>
    <div class="col-12 col-sm-6 col-md-3 ">...</div>
    <div class="col-12 col-sm-6 col-md-3 ">...</div>
  </div>
</div>
```



xs



sm



Md+

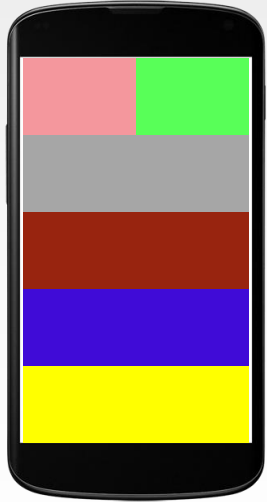


# Practise

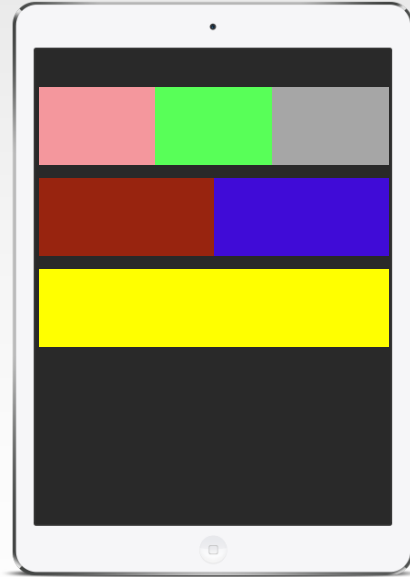
15dk

Xs ve xlg çözünürlükte görünüm nasıl olur?

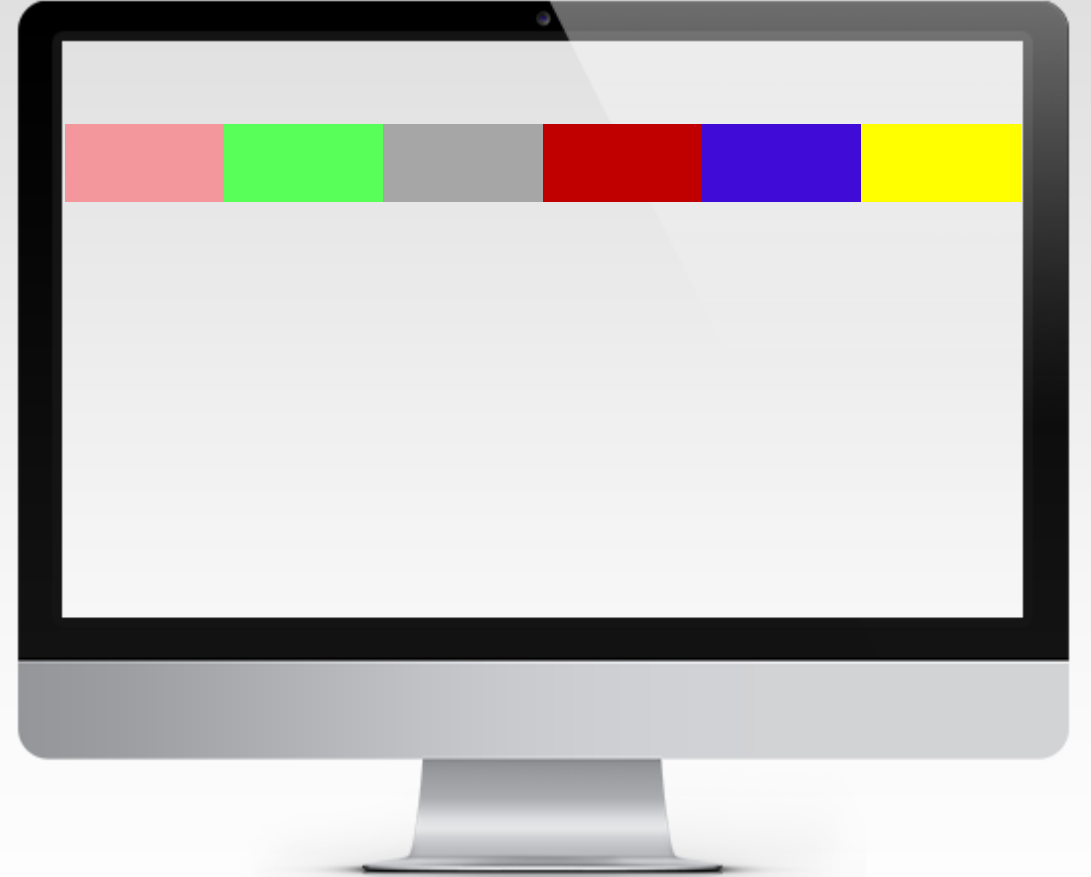
Aşağıdaki ızgara yapılarını istenilen çözünürlüklere göre ayarlayınız.



sm



md



lg



# Renkler

primary

secondary

success

danger

warning

info

light

dark

white

Yazı rengi değiřtirmek için « text-**renk** » řablonu kullanır

Çerçeve rengi değiřtirmek için « border-**renk** » řablonu kullanır

Zemin rengi değiřtirmek için « bg-**renk** » řablonu kullanır

Zemin rengine gradient vermek için « bg-**renk** bg-gradient » řablonu kullanır

Link rengi değiřtirmek için « link-**renk** » řablonu kullanır



# Display

d-none

Elementi gizler

d-block

Elemente block özelliği kazandırır

d-inline

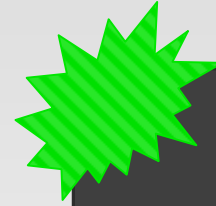
Elemente inline özelliği kazandırır

d-inline-block

Elemente inline-block özelliği kazandırır

d-flex

Elemente flex özelliği kazandırır



Breakpoint değerleri kullanarak istenilen çözünürlükte display özelliği verilebilir.  
Örnek: d-xs-none





# Practise



- Sadece xs ekranlarda gözükecek bir element tasarlayınız.
- Sadece xs ekranlarda gözükmeyecek bir element tasarlayınız.
- Sadece md ekranlarda gözükecek bir element tasarlayınız.
- Sadece md ekranlarda gözükmeyecek bir element tasarlayınız.
- Sadece md+ ekranlarda gözükecek bir element tasarlayınız.



# Display

d-flex

Elemente flex özelliği kazandırdıktan sonra aşağıdaki özellikler kullanılabilir.

## **Yatay hizalama için:**

Justify-content- [ start | end | center | between | around ]

## **Dikey hizalamak için:**

Align-items-[ start | end | center ]



Breakpoint değerleri kullanarak istenilen çözünürlükte hizalama yapılabilir.



# Practise

 15dk

LARGE



+90 555 555 55 55



+90 555 555 55 55



#techproed

MEDIUM



+90 555 555 55 55



+90 555 555 55 55



SMALL





# Padding & Margin

p: padding  
m: margin

t: top  
b: bottom  
s: start  
e: end  
x: start & end  
y: top & bottom

0-6: mesafe  
auto: margin  
için

## Örnekler:

m-4: dört kenardan 4 birim margin verir  
mt-2: üst taraftan 2 birim margin verir  
mx-auto: sol ve sağ marginleri auto yapar.  
pe-2: sağ taraftan 2 birim padding verir.  
py-0: üst ve alt marginleri sıfırlar.



# Position

**fixed-top**

Elementi üst tarafa sabitler

**fixed-bottom**

Elementi alt tarafa sabitler

**sticky-top**

Elementi üst taraf için yapışkan hale getirir.



Sticky-top, breakpointlerle istenilen çözünürlük için ayarlanabilir.



# Resimler

## img-fluid

Responsive resim oluşturmak için kullanılır.

```

```

## img-thumbnail

Thumbnail küçük resim oluşturmak için kullanılır.

```

```

## rounded

Resimlerin köşelerini yuvarlaklaştırmak için kullanılır

```

```



# Button

**btn**

Buton oluşturmak için btn class ı kullanılır. Mutlaka kullanılmalıdır.

**btn-[ renk ]**

Butona renk vermek için kullanılır

```
<button type="button" class="btn btn-primary">  
Click  
</button>
```

Click

**btn-outline [ renk ]**

Butona çerçeve rengi vermek için kullanılır

```
<button type="button" class="btn btn-outline-  
primary">Primary</button>
```

Click



# Button

## btn-[ boyut ]

Buton boyutunu ayarlamak için kullanılır. sm, md veya lg değeri alabilir.

```
<a href="#"« class="btn btn-secondary btn-lg">Click</a>
```

Click

## W-100

Butonu bulunduğu yere yatayda tam sığdırmak için kullanılırç





# Button Group

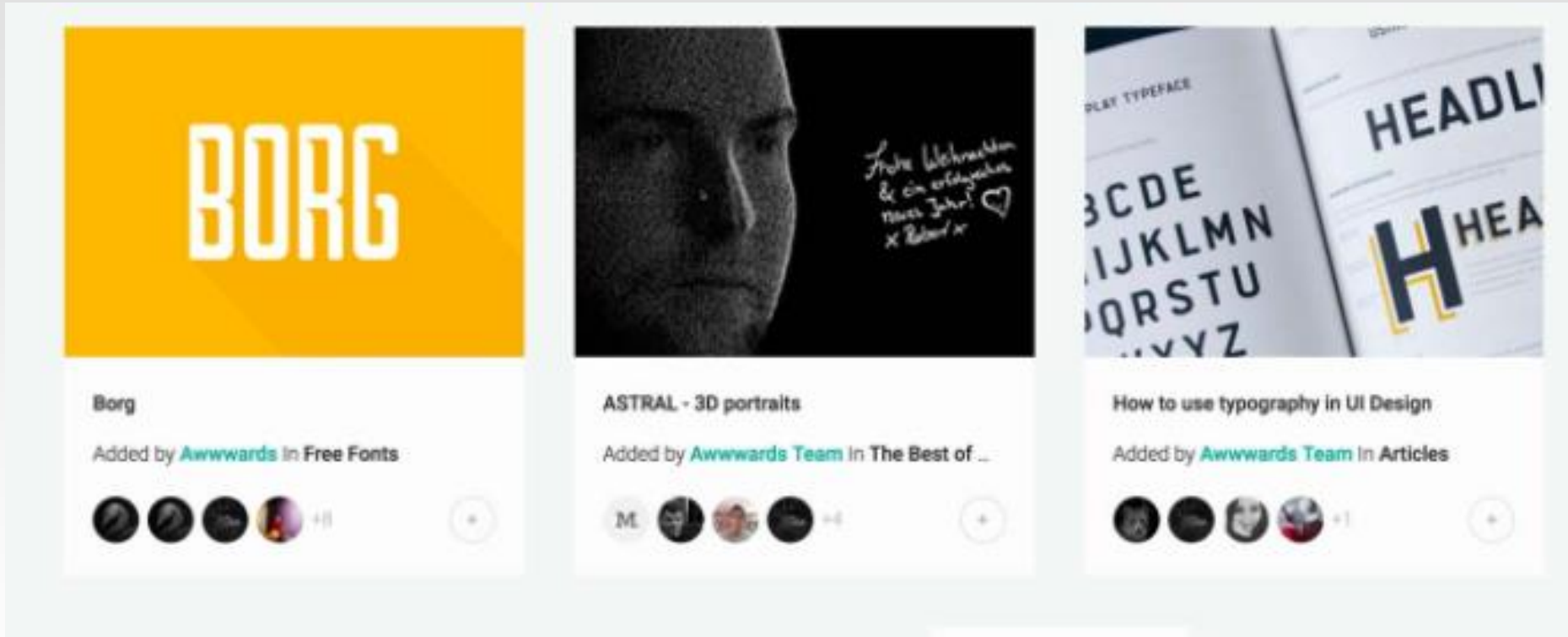
Birden fazla butonu birarada göstermek için btn-group kullanılır.

```
<div class="btn-group">  
  <a href="#" class="btn btn-danger">Left</a>  
  <a href="#" class="btn btn-warning">Middle</a>  
  <a href="#" class="btn btn-success">Right</a>  
</div>
```



# Card

Web sayfalarında aşağıdaki gibi kartlar oluşturmak için kullanılır.





# Card

En basit kullanımında kenarları yuvarlaklaştırılmış, gri renkli çerçevesi ve padding değerleri ile bir kutu oluşturur.

```
<div class="card">  
  <div class="card-body">  
    This is some text within a card body.  
  </div>  
</div>
```

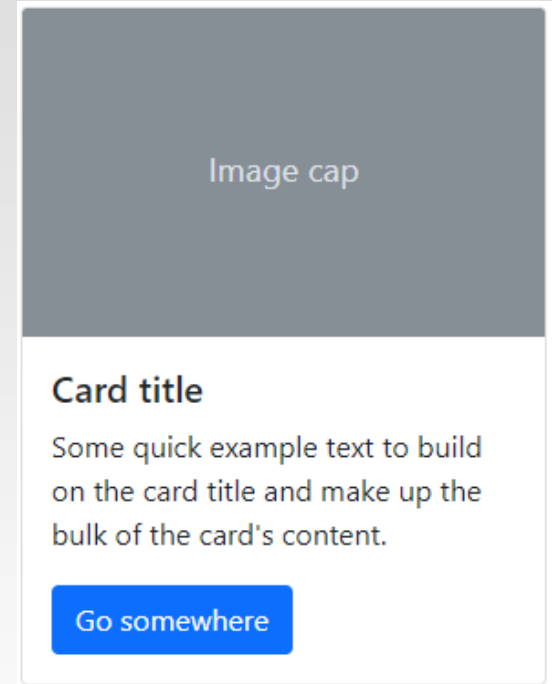


Detaylar [getbootstrap.com](https://getbootstrap.com) da



# Card

```
<div class="card" style="width: 18rem;">
  
  <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <p class="card-text">Some quick ...</p>
    <a href="#" class="btn btn-primary">Go somewhere</a>
  </div>
</div>
```



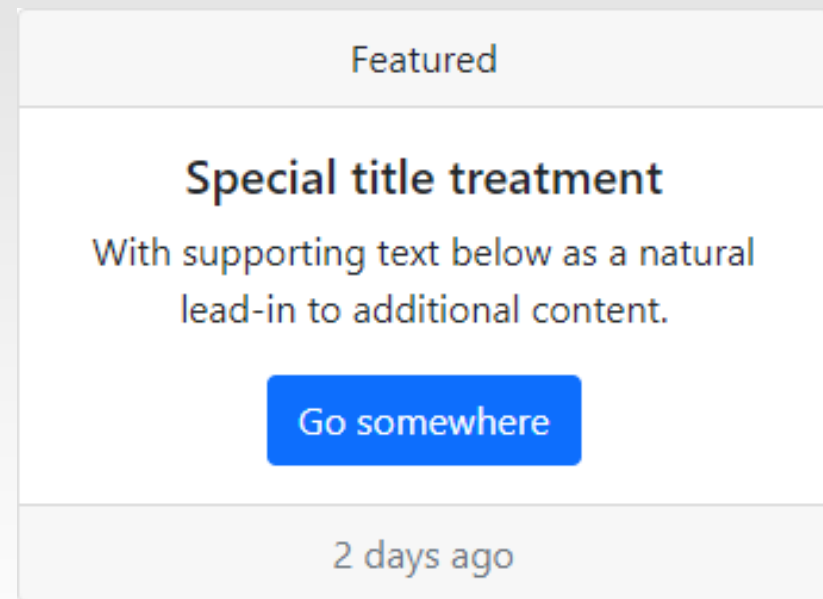
Fotoğrafı altta göstermek için, img elementi card-body nin bitimine yerleştirilir ve img ye card-img-bottom class ı verilir. Sola ya da sağa almak için row ve col oluşturulur.



# Card

Kartlara header ve footer da eklenebilir.

```
<div class="card text-center">
  <div class="card-header">
    Featured
  </div>
  <div class="card-body">
    <h5 class="card-title">Special title treatment</h5>
    <p class="card-text">With supporting ....</p>
    <a href="#" class="btn btn-primary">Go somewhere</a>
  </div>
  <div class="card-footer text-muted">
    2 days ago
  </div>
</div>
```





# Navbar

Navbar Home Link Dropdown ▾ Disabled

Search

Search

Responsive bir menü eklemek için kullanılır.



# Navbar classes

navbar-expand-lg

Navbar normalde daraltılmış halde durur. Navbar-expand ile hangi çözünürlükten itibaren genişletileceğini belirler. navbar-expand-md yazılırsa md ekranlardan itibaren genişletilir.

navbar-light

Metin renklerinin koyu mu yoksa açık mı olacağını belirler. Navbar ın zemin renginin koyu mu açık mı olduğu belirtilir. Metin renkleri bunun zıttı olarak koyu ya da açık olarak ayarlanır. Navbar-dark veya navbar-light değeri alır.

bg-dark

Zemin rengini belirler



# Navbar classes

container-fluid

Navbar ın tam ekran olup olmayacağını belirler

navbar-brand

Logonun tanımlandığı yerdir. Img ile fotoğraf da konulabilir.

navbar-toggler

Hamburger ikonunun olduğu bölümdür.

me-auto

Solda gözüken menüyü sağ tarafa almak için ms-auto yapılır





# Navbar classes

nav-item

Her bir menü elemanıdır.

nav-link

Menü linkleridir

active

Hangi menü elemanının aktif olduğunu belirler

dropdown

Açılır menü elemanını tanımlar

dropdown-menu

Açılır menüyü tanımlar



# Navbar dropdown

- Navbar içinde alt menü kullanmak için ana menü li elemanına dropdown sınıfı, alt menü ul elemanına ise dropdown-menu sınıfı eklenir. Ayrıca ana menü a elemanına dropdown-toggle sınıfı ve data-bs-toggle="dropdown" attribute u eklenir.

```
<li class="nav-item dropdown">
  <a class="nav-link dropdown-toggle" href="#" data-bs-toggle="dropdown">
    Hizmetler
  </a>
  <ul class="dropdown-menu">
    <li><a class="dropdown-item" href="#">Action</a></li>
    <li><a class="dropdown-item" href="#">Another action</a></li>
    <li><hr class="dropdown-divider"></li>
    <li><a class="dropdown-item" href="#">Something else here</a></li>
  </ul>
</li>
```



# Navbar sticky & fixed

- Navbar elemanına sticky veya fixed özellikler kazandırılabilir. Bunun için sticky-top, fixed-top, fixed-bottom sınıfları kullanılabilir.

```
<nav class="navbar fixed-top navbar-light bg-light">  
  ....  
</nav>
```



# Practise

15 dk

Aşağıdaki şekilde bir Navbar yapınız.

TECHPROED

Search

Free Courses

Home

Programs ▾

FAQ

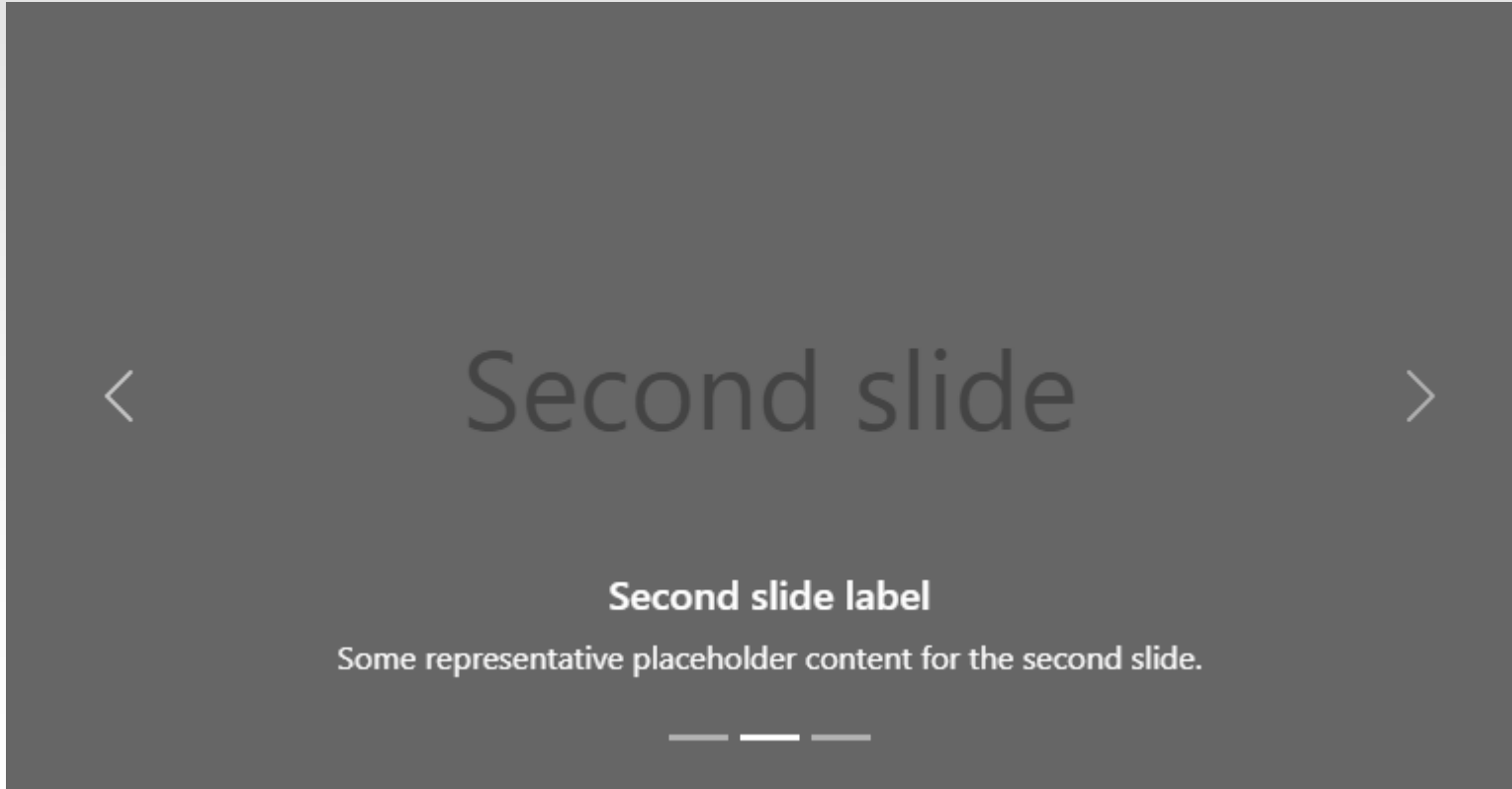
About Us

Contact Us



# Carousel

Özellikle ana sayfalarda banner bölümünde kullanılan slayt geçişleri için kullanılır.



Slaytlardan birine mutlaka active class ı verilmelidir.



Aynı sayfada birden fazla carousel kullanılacaksa mutlaka her bir carousel için farklı bir id verilmelidir.



# Carousel classes

carousel slide

Taşıyıcı elemente verilir ve mecburidir

data-bs-  
ride="carousel"

Otomatik olarak başlaması için gereklidir

carousel-inner

İçerik kısmı için bir taşıyıcıdır

carousel-item

Carousel içindeki her bir slaytı tanımlar. Bir tanesinin yanına mutlaka active konulmalıdır.

carousel-caption

Fotoğraf üzerindeki başlık ve açıklama için taşıyıcıdır



# Carousel classes

carousel-indicators

İstenilen slayta geçiş yapmayı sağlayan butonları içerir

data-bs-target="#carouse"

Ana taşıyıcı için verilen id değerini içermelidir

data-bs-slide-to="0"

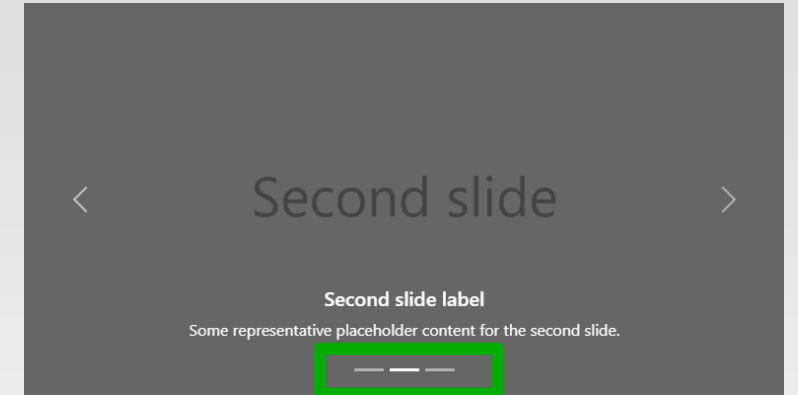
Tıklandığında kaçınıcı slayta gidileceğini belirtir

active

İlk gösterilecek slayta ait butona verilir.

aria-current="true"

İlk gösterilecek slayta ait butona verilir.





# Carousel classes

`carousel-control-  
prev`

Önceki buton için kullanılır

`data-bs-  
target="#carouse"`

Ana taşıyıcı için verilen id değerini içermelidir

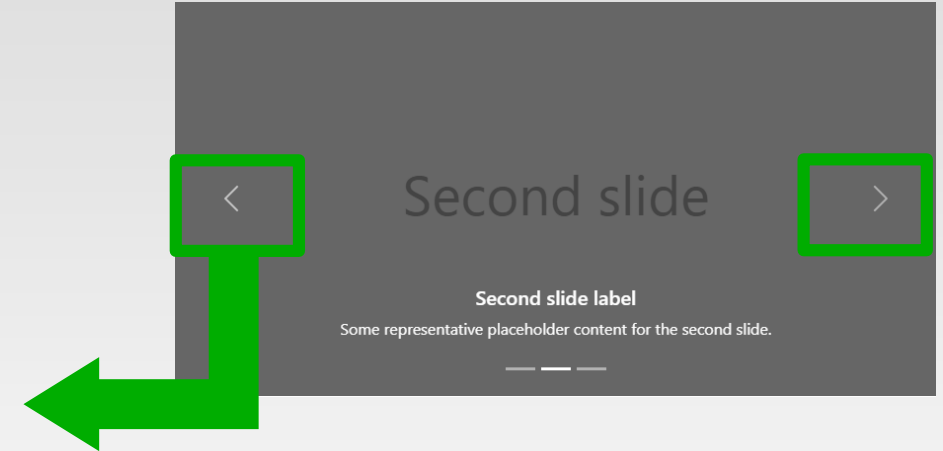
`data-bs-  
slide="prev"`

Tıklandığında gideceği slaytı belirler

`carousel-control-  
prev-icon`

İkon için kullanılır

Önceki slayt için prev, sonraki slayt için next kelimesi kullanılır.







# Carousel classes

`carousel-fade`

Fade efekti ile geçiş için ana taşıyıcıya eklenebilir

`data-bs-  
interval="3000"`

Hangi slaytta ne kadar bekleneyeceğini belirlemek için `carousel-item` elementine milisaniye cinsinden eklenebilir.

`carousel-dark`

Koyu renkli resimler kullanılacaksa üzerinde yazılar belirgin olsun diye kullanılabilir.



# Practise



Navbar ve carousel kullanarak bir tasarım yapınız. Carousel her ekranda tam yükseklik sağlamalı.



# Accordion

Uzun metin ifadelerinin tümünü aynı anda göstermek yerine gruplara bölünerek aynı anda sadece ilgili bölümün gösterilmesini sağlar.

Accordion Item #1	▼
Accordion Item #2	▼
Accordion Item #3	▼



# Accordion

accordion

Taşıyıcı elemente verilir ve mecburidir

accordion-item

Her bir bağımsız bölüm için taşıyıcıdır

accordion-header

Her bir bağımsız bölümün başlık ifadesi için kullanılır.

accordion-collapse

Header daki butona tıklandığında gösterilecek olan içerik için taşıyıcıdır.

accordion-body

Header daki butona tıklandığında gösterilecek olan içeriği barındırır



# Accordion

Header içindeki butona tıklandığında ilgili içeriğin gösterilmesi için dikkat edilmesi gereken durum:

Heading içinde bulunan butondaki `data-bs-target` attribute değeri ile `accordion-collapse` class ına sahip elemanın `id` değeri aynı olmalıdır.

Bir içerik default olarak gösterilmek isteniyorsa dikkat edilmesi gereken durum:

`accordion-collapse` class ına sahip elemetie `show` class ı eklenir. Ayrıca butondaki `collapsed` class ı kaldırılır.



Aşağıdaki gibi iki ayrı sıkça sorulan sorular bölümü oluşturunuz.

Accordion Item #1

**This is the first item's accordion body.** It is hidden by default, until the collapse plugin adds the appropriate classes that we use to style each element. These classes control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the `.accordion-body`, though the transition does limit overflow.

Accordion Item #2

Accordion Item #3

Accordion Item #1

**This is the first item's accordion body.** It is hidden by default, until the collapse plugin adds the appropriate classes that we use to style each element. These classes control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the `.accordion-body`, though the transition does limit overflow.

Accordion Item #2

Accordion Item #3



# Tabs

İçerikler gruplanarak tablolar içinde daha derli toplu gösterilebilir.



```
<ul class="nav nav-tabs">
  <li class="nav-item">
    <button
      class="nav-link active"
      id="home-tab"
      data-bs-toggle="tab"
      data-bs-target="#home"
      type="button"
      >Home</button>
  </li>
  ...
```

```
<div class="tab-content">
  <div class="tab-pane fade show active"
    id="home">...1</div>
  <div class="tab-pane fade"
    id="profile">...2</div>
  <div class="tab-pane fade"
    id="contact">...3</div>
</div>
```



nav-tabs içindeki data-bs-target ile tab-content içindeki id lerin eşit olması gerekir.



# Practise



Aşağıdaki gibi bir tasarım yapınız. Soldaki menü elementlarına tıklanınca o kategorideki fotoğraflar gösterilmelidir.

