Responsive Web Tasarım Nedir? Nasıl Yapılır

Responsive web tasarım nedir?

Responsive web design veya Türkçe karşılığı **duyarlı web tasarımı** tasarlanan sayfanın tüm ekranlarda iyi görünmesi sağlayan tekniktir.

Responsive web tasarım veya duyarlı web tasarım tüm cihazlarda sayfanın iyi görünmesidir.

Responsive web tasarım sayfayı yeniden boyutlandırırken, küçültürken veya büyültürken tüm cihazlarda sayfanın iyi görünmesini sağlamaktır.

Nasıl yapılır?

Responsive web tasarım HTML ve CSS kullanarak yapılır.

Responsive web tasarım için herhangi bir programa veya yazılıma ihtiyaç yoktur.

Neden responsive web tasarım yapmalıyız?

Web sayfaları farklı ekran boyutlarında bilgisayar, tablet veya telefon ile ziyaret edilir.

Responsive web tasarım yapılmazsa sitemize cep telefonundan veya tabletten giren ziyaretçiler tarafından kötü görünecek ve siteden hemen çıkmasına sebep olacaktır.

Tasarlamış olduğunuz web sayfaları her ekran boyutunda iyi görünmeli ve kullanımı kolay olmalıdır.

Web sayfaları sadece bilgisayar kullanıcıları için tasarlanmamalı web sayfası her ekran boyutunda iyi görünecek şekilde yapılmalıdır.

Responsive web tasarım ekran boyutuna göre CSS ile HTML elemanlarının yeniden boyutlandırılması, gizlenmesi, küçültülüp, büyütülmesi veya başka bir alana taşınmasıdır.

meta viewport nedir?

Viewport veya Türkçe karşılığı görüş alanı web sayfasının kullanıcı tarafından görünen alanıdır.

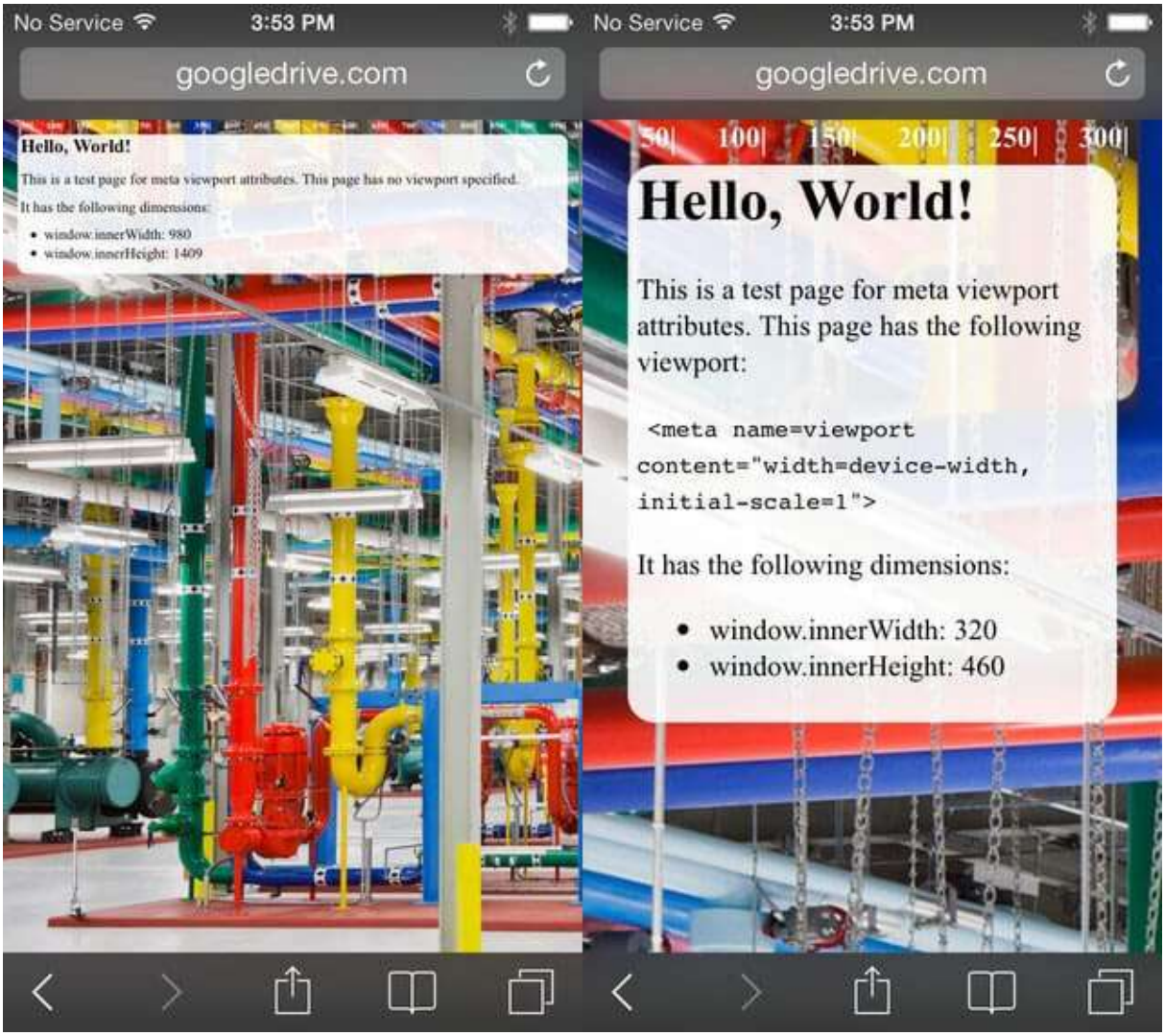
Web sayfasının görünüm alanı kullanılan cihaza göre değişir.

Örneğin; cep telefonu ve tabletin görünüm alanı masaüstü bilgisayarın görünüm alanından küçüktür.

HTML meta viewport özelliği web sayfasının mobil cihazda nasıl görüntüleneceğini belirler.

Viewport kullanılmadığında mobil cihazlar sayfayı masaüstü ekran genişliğinde ve ekrana sığacak biçimde ölçeklenmiş olarak görüntüleyecektir.

Aşağıdaki örneğin sağ tarafında kullanılmış, sol tarafında kullanılmıştır.



Viewport kullanımı

Görünüm alanını belirlemek için kullanılan viewport özelliği <head> </head> etiketi arasına yazılan <meta /> etiketiyle kullanılır.

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

Genel olarak viewport özelliği **width** ve **initial-scale** özelliklerine **device-width** ve **1.0** değeri verilerek kullanılmaktadır.

width=device-width – Piksel cinsinden görünüm alanı genişliğini belirtir. **device-width** değeri görünüm alanı genişliğinin cihaza bağlı olarak değişiklik göstereceğini ifade eder.

initial-scale=1.0 – Web sayfası açıldığında yakınlaştırma oranını belirmek için kullanılır. **0** ila **10.0** arasında değer alır.

Responsive web tasarım için viewport özelliğinin kullanılması gerekmektedir.

Responsive web tasarım nasıl yapılır?

Web siteler genel olarak yukarıdan aşağıya doğru bir tasarım ile yapılır.

Çünkü kullanıcılar için web sitelerini yukarıdan aşağıya doğru gezmek daha kolaydır.

Kullanıcılar tasarlamış olduğumuz sitede gezinmek için yatay kaydırma, sayfayı yakınlaştırma veya uzaklaştırma yapmak zorunda bırakılırsa kullanıcılar için iyi bir site olmayacak ve hemen siteden çıkacaklardır.

Sayfa taşmasının önlemek ve **Responsive Web Tasarım** için yapılması gerekenler;

Site bölümleri sabit genişlikte veya çok geniş olmasın

Web sayfamızda kullandığımız herhangi bir resim ekran boyutundan büyük olursa kullanıcı resmi görebilmek için sayfayı yatay olarak sağa sola kaydırmak zorunda kalır. Sitede kullanılan resimlerin veya bölümlerin genişliği ekran boyutu kadar veya ekran boyutundan daha küçük olması gerektiğini unutmayın.

Site genişliğinin herhangi bir cihazda iyi görünmesine aldanmayın

Kullanmış olduğumuz genişlik bilgisayar ve telefonda iyi görünebilir ancak tabletlerde iyi bir görüntü vermeyebilir. Bunun için bir cihazda iyi görünen genişlik değeri ve ölçü birimini kullanmayın.

Farklı ekran boyutları için farklı stil tanımlaması yapın

Ekran boyutuna özel genişlik ve ölçü birimi için CSS medya sorguları kullanın.

Mutlak genişlik yerine göreceli genişlik değeri kullanın

Ekranı büyük cihazlar için hazırlanan siteyi responsive site haline getirirken veya yeni responsive site tasarımı yaparken “width: 960px” yerine “width: 90%” gibi göreceli genişlik değeri kullanmaya özen gösterin.

Genişlik sınırı vermek için CSS min- ve max- ön eklerini kullanın

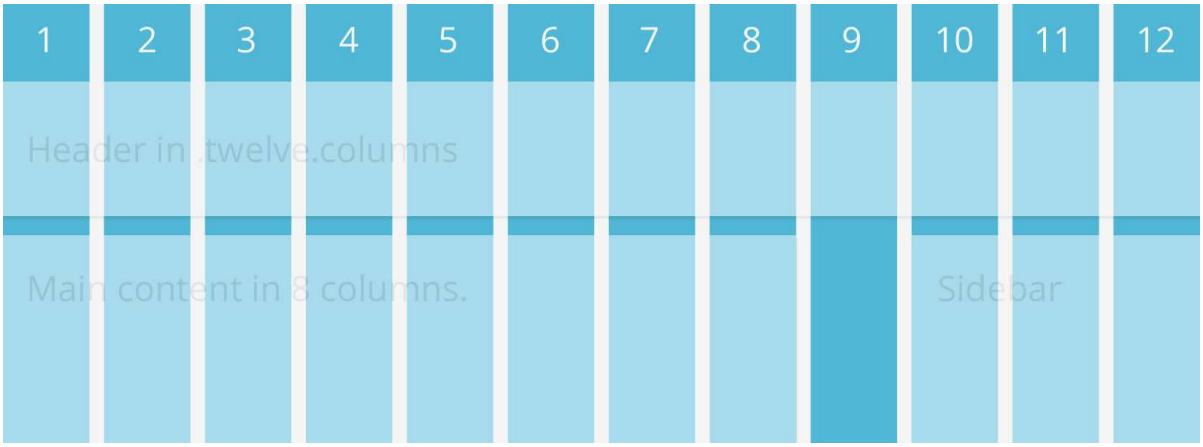
Genişlik sınırı vermek için CSS min- ve max- ön eklerini kullanın. Genişlik sınırı verildiğinde bozulmalar en aza inecektir.

Responsive web tasarım grid sistemi

Web sayfası tasarlarken çoğunlukla grid (kılavuz) sistemi kullanılır.

Grid sistemi sayfayı sütunlara bölme tekniğidir.

Web sayfası tasarlarken grid sistemi kullanmak sayfadaki bölümlerin yerleşimini kolaylaştırır.



Responsive web tasarım yaparken genellikle toplam genişliği 100% olan ve 12 sütuna ayrılmış grid sistemi kullanılır.

Göreceli genişlik kullanıldığından farklı ekran boyutlarında genişlik küçülür veya genişler.

Responsive grid sistemi oluşturmak

Grid sistemi oluşturmaya başlamadan önce tüm tarayıcılarda genişliğe kenarlık (border) ve iç boşluk (padding) değerinin de katılarak hesaplanması için **box-sizing** özelliğine **border-box** değeri veriyoruz.

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

Responsive web tasarım yaparken sayfadaki bölümleri daha iyi konumlandırmak için 12 sütunlu grid sistemi kullanılmaktadır.

Grid sisteminde bir sütun genişliği şöyle hesaplanır: 100% / Toplam Sütun Sayısı = 1 sütun genişliği

Kullanacağımız grid sistemindeki Toplam Sütun Sayısı = 12 olduğundan, bir sütunun genişliği: 100% / 12 = 8.33%

Daha sonra sütun genişliğini belirten ön eki **col-** veya istediğimiz bir takma isim ekleyerek Toplam Sütun Sayısı kadar sınıf tanımlarız.

```
.col-1  {width: 8.33%;       }

.col-2  {width: 16.66%;      }

.col-3  {width: 25%;  }

.col-4  {width: 33.33%;      }

.col-5  {width: 41.66%;      }

.col-6  {width: 50%;  }

.col-7  {width: 58.33%;      }

.col-8  {width: 66.66%;      }

.col-9  {width: 75%;  }

.col-10 {width: 83.33%;      }

.col-11 {width: 91.66%;      }

.col-12 {width: 100%; }
```

Sütunların yan yana görünmesi için tüm sütunları **float** özelliğiyle sol tarafa yaslıyoruz.

```
[class*="col-"] {

    float: left;

}
```

Sütunlara **float** uygulandığından her eklediğimiz sütun sol taraftan devam edecektir.

Bunun önüne geçmek için sütunları satır taşıyıcısı içine eklemeli ve satır taşıyıcısı dışında eklenen öğelerin yeni satırda başlaması için **clear** özelliğiyle **float** işlemini temizlemeliyiz.

```
.row::after {

    content: "";

    clear: both;

    display: table;

}
```

Gerekli CSS tanımlarını yaptık.

```
<div class="row">

  <div class="col-3">...</div> <!-- 25% -->

  <div class="col-9">...</div> <!-- 75% -->

</div>
```

Satır taşıyıcısı içindeki **col-** ile belirtilen sütunların toplamı Toplam Sütun Sayısını geçmemesi gerekir.

12 sütunlu bir grid sistemi tasarladığımızdan col-3 + col-9 = col-12 olması gerekiyor.

Gerekli grid sistemi tanımlaması yaptıktan sonra artık sitemizi kolay bir şekilde bölümlere ayırabiliriz.

Grid sistemi örneği

```
* {

    box-sizing: border-box;

}

.col-1  {width: 8.33%;       }

.col-2  {width: 16.66%;      }

.col-3  {width: 25%;  }

.col-4  {width: 33.33%;      }

.col-5  {width: 41.66%;      }

.col-6  {width: 50%;  }

.col-7  {width: 58.33%;      }

.col-8  {width: 66.66%;      }

.col-9  {width: 75%;  }

.col-10 {width: 83.33%;      }

.col-11 {width: 91.66%;      }

.col-12 {width: 100%; }

[class*="col-"] {

    float: left;

    border: 1px solid red;

}
```

```
.row::after {

    content: "";

    clear: both;

    display: table;

}

<header class="row">

  <div class="col-12"><h1>Site başlığı</h1></div>

</header>


<main class="row">

  <div class="col-3">Menü</div>

  <div class="col-9">İçerik</div>

</main>


<footer class="row">

  <div class="col-12">Alt kısım</div>

</footer>
```

Örnekte görüldüğü gibi grid sistemi kullanmak web sayfasındaki bölümleri konumlandırmak için kolaylık sağlıyor.

Örnekteki menü ve içerik bölümlerinin yerlerini sayfa içerisinde değiştirmek için sadece HTML kodlarındaki yerlerini değiştirmek yeterli olacaktır.

Tasarlamış olduğumuz grid sistemi tarayıcıyı küçülttüğümüzde iyi görünmeyebilir.

Responsive web tasarım CSS medya sorguları

CSS medya sorguları ekran boyutuna göre stil tanımlamayı sağlar.

CSS **@media** ile belirlenmiş şartın doğru olması durumunda **@media** blokları arasındaki stil tanımları geçerli olur.

Aşağıdaki örnekte tarayıcı genişliği **600px** ve üzerinde ise arka plan rengi açık yeşil, tarayıcı genişliği **599px** ve altındaysa arka plan rengi açık mavi olacaktır.

```
body {

    background-color: lightblue;
```

```
}

@media screen and (min-width: 600px) {

    body {

        background-color: lightgreen;

    }

}
```

Responsive Web Tasarım Grid Sistemi başlığında grid sistemini oluşturduk.

Fakat tarayıcıyı küçülttüğümüzde iyi görünmüyordu.

CSS medya sorguları ile ekran boyutuna göre stil tanımları uygulayarak cep telefonu, tablet gibi cihazlarda sitenin daha iyi görünmesini sağlayabiliriz.

Grid sistemi örneğine aşağıdaki CSS kodları eklendiğinde tarayıcı genişliği 768px ve altında olduğunda her bir sütun genişliği 100% olacaktır.

```
@media only screen and (max-width: 768px) {

    [class*="col-"] {

        width: 100%;

        float: left;

    }

}
```

Tarayıcı penceresini küçülttüğümüzde veya ekran boyutu küçük cihazlardan web sayfasına baktığımızda sayfa daha iyi görünecektir.

Önceliği küçük ekranlara vermek

Responsive web tasarım yaparken önceliği ekran boyutu küçük olan cihazlara verirsek ekran boyutu küçük cihazlarda sayfanın daha hızlı açılmasını sağlarız.

Ekran boyutu küçük cihazlara öncelik vermek için CSS kodlarımızda çeşitli değişiklik yapmamız gerekiyor.

```
/* Ekran boyutu küçük cihazlar */

[class*="col-"] {

    float: left;

    width: 100%;

}
```

```
@media only screen and (min-width: 768px) {
    /* Ekran boyutu büyük cihazlar */
    .col-1 {width: 8.33%;}
    .col-2 {width: 16.66%;}
    .col-3 {width: 25%;}
    .col-4 {width: 33.33%;}
    .col-5 {width: 41.66%;}
    .col-6 {width: 50%;}
    .col-7 {width: 58.33%;}
    .col-8 {width: 66.66%;}
    .col-9 {width: 75%;}
    .col-10 {width: 83.33%;}
    .col-11 {width: 91.66%;}
    .col-12 {width: 100%;}
}
```

Ekran boyutana özel sütun genişliği

Her ekran boyutu için farklı sütun genişliği belirleyebiliriz.

```
/* Ekran boyutu küçük cihazlar */
[class*="col-"] {
    float: left;
    width: 100%;
}
@media only screen and (min-width: 600px) {
    /* Tablet cihazlar */
    .col-m-1 {width: 8.33%;}
    .col-m-2 {width: 16.66%;}
    .col-m-3 {width: 25%;}
    .col-m-4 {width: 33.33%;}
    .col-m-5 {width: 41.66%;}
    .col-m-6 {width: 50%;}
```

```
    .col-m-7 {width: 58.33%;}

    .col-m-8 {width: 66.66%;}

    .col-m-9 {width: 75%;}

    .col-m-10 {width: 83.33%;}

    .col-m-11 {width: 91.66%;}

    .col-m-12 {width: 100%;}

}

@media only screen and (min-width: 768px) {

    /* Ekran boyutu büyük cihazlar */

    .col-1 {width: 8.33%;}

    .col-2 {width: 16.66%;}

    .col-3 {width: 25%;}

    .col-4 {width: 33.33%;}

    .col-5 {width: 41.66%;}

    .col-6 {width: 50%;}

    .col-7 {width: 58.33%;}

    .col-8 {width: 66.66%;}

    .col-9 {width: 75%;}

    .col-10 {width: 83.33%;}

    .col-11 {width: 91.66%;}

    .col-12 {width: 100%;}

}
```

Ekran boyutuna özel sütun genişliği kullanmak sayfadaki bölümlerin farklı ekran boyutunda daha iyi görünmesini sağlar.

Yukardaki CSS kodları ekli olan web sayfasına aşağıdaki HTML kodları eklendiğinde ekran boyutuna özel sütun genişliği kullanımı daha iyi anlaşılacaktır.

```
<div class="row">

<div class="col-3 col-m-3">...</div>

<div class="col-6 col-m-9">...</div>

<div class="col-3 col-m-12">...</div>
```

```
</div>
```

Yukardaki örneği tarayıcı genişliğini küçülterek veya büyüterek denediğimizde;

Ekran genişliği 768px ve üzerinde sayfa bölümleri 3 sütun – 6 sütun – 3 sütun olarak bölünecek,

Ekran genişliği 600px ve 768px arasında sayfa bölümleri 3 sütun – 9 sütun – 12 sütun olarak bölünecek,

Ekran genişliği 600px ve altında her bir sütun sayfa genişliğinde olacaktır.

Resimler

Responsive web tasarım yaparken resimlere göreceli değer (width: 100%) vererek resimleri responsive yapabiliriz.

```
img {
    width: 100%;
    height: auto;
}
<img src="https://unsplash.it/500" alt="Grrr" />
```

Tarayıcı genişliği resim genişliğinden geniş olursa resim daha da büyüyecek ve resimde bozulmalar olacaktır.

Resmin fazla büyümesi ve bozulmasını önlemek için **max-width** özelliği kullanılır.

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

Ekran boyutuna özel resim

Büyük bir resim ekranı büyük cihazda güzel görünür.

Ancak ekranı küçük cihazlarda resim fazla küçüldüğünden resim güzel görünmeyebilir.

Ekran boyutuna özel resim kullanarak daha iyi görünüm sağlanmış olur.

HTML5 <picture> etiketi

HTML5 **<picture>** etiketi ekran boyutuna özel resim kullanılmasını sağlar.

HTML5 **<picture>** etiketinin kullanımı <video> ve <audio> etiketine benzer.

```
<picture>

  <source srcset="https://unsplash.it/500?image=1" media="(max-width:
700px)">

  <source srcset="https://unsplash.it/1000?image=0">

  <img src="https://unsplash.it/1000?image=0" alt="Grr" />

</picture>
```

HTML5 **<picture>** etiketini desteklemeyen tarayıcılar için **<img>** etiketini kullanmayı unutmayın.

Videolar

Responsive web tasarım yaparken videolara göreceli değer (width: 100%) vererek videoları responsive yapabiliriz.

```
video {

    width: 100%;

    height: auto;

}

<video width="360" height="320" controls="controls">

  <source src="video.mp4" type="video/mp4" />

  <source src="video.ogg" type="video/ogg" />

  Tarayıcınız video etiketini desteklemiyor.

</video>
```

Tarayıcı genişliği video genişliğinden daha geniş olursa video daha da büyüyecek ve videoda bozulmalar olacaktır.

Videonun fazla büyümesi ve bozulmasını önlemek için **max-width** özelliği kullanılır.

```
video {

    max-width: 100%;

    height: auto;

}
```

Responsive Web Tasarım Framework Kullanmak

Responsive web tasarımı yaparken hazır stil tanım topluluğu olan CSS frameworklerini kullanabilirsiniz.

CSS frameworklerinde çoğunlukla Grid Sistemi, Menü Sistemi, Slayt Sistemi gibi stiller tanımlanmıştır.

Tasarlayacağınız siteye ve ihtiyacınıza göre CSS frameworklerinden birini seçebilirsiniz.

CSS framework seçiminde; tarayıcı desteği, responsive desteği, kullanıcı desteği, bileşen desteği gibi gereksinimlere dikkat edilmelidir.

CSS frameworkleri hazır stil tanımları yanında hazır renk tanımları ile gelmektedir.

CSS frameworkleri ücretsiz olduğundan çoğu web sitesi benzer renk tanımlarını kullanır CSS framework içindeki çeşitli stil ve renk tanımlarını değiştirmeniz faydalı olacaktır.

Çok kullanılan CSS frameworkleri; Bootstrap, Foundation, Bulma, Semantic UI, Materialize

Framework Kullanımı

```
<!DOCTYPE html>

<html lang="tr">

<head>

  <title>Framework kullanımı</title>

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />

</head>

<body>



<main class="container">

  <section class="jumbotron">

    <h1>Bootstrap Framework kullanımı</h1>

    <p>Etkiyi görmek için tarayıcıyı yeniden boyutlandırın!</p>

  </section>

  <div class="row">

    <article class="col-sm-4">

      <h3>Sütun 1</h3>

      <p>Lorem Ipsum, dizgi ve baskı endüstrisinde kullanılan
metinlerdir.</p>
```

```
      <p>Yinelenen bir sayfa içeriğinin okuyucunun dikkatini dağıttığı
bilinen bir gerçektir.</p>

    </article>

    <article class="col-sm-4">

      <h3>Sütun 2</h3>

      <p>Lorem Ipsum, dizgi ve baskı endüstrisinde kullanılan
metinlerdir.</p>

      <p>Yinelenen bir sayfa içeriğinin okuyucunun dikkatini dağıttığı
bilinen bir gerçektir.</p>

    </article>

    <article class="col-sm-4">

      <h3>Sütun 3</h3>

      <p>Lorem Ipsum, dizgi ve baskı endüstrisinde kullanılan
metinlerdir.</p>

      <p>Yinelenen bir sayfa içeriğinin okuyucunun dikkatini dağıttığı
bilinen bir gerçektir.</p>

    </article>

  </div>

</main>

</body>

</html>
```

Örnekte görüldüğü gibi CSS framework kullanımı son derece kolaydır.

İlk olarak gerekli dosyaları sayfamıza dahil ettik daha sonra dosya içerisindeki hazır stil tanımlarını sayfa içerisinde kullandık.

Örnekteki sütunların Grid Sistemi yazımdaki gibi 12 sütuna bölündüğüne ve satır taşıyıcısı içindeki sütun toplamının 12 olduğuna dikkat edin.

Framework içerisindeki diğer stil tanımları ve kullanımı hakkında bilgi almak için framework sitesindeki dokümana bakabilirsiniz.