
Table of Contents

Giriş	1.1
İnternet Nasıl Çalışır	1.2
HTML'e Giriş	1.3
Temel Etiketler	1.3.1
Derinlere İnelim	1.3.2
CSS'e Giriş	1.4
Başlayalım	1.5
HTML'i Hazırlayalım	1.5.1
Görünüşü Güzelleştirelim	1.5.2
Son Dokunuşlar	1.5.3
Sayfamızı Yayınlayalım	1.6

Giriş

Selamlar, bu döküman ile birlikte internet sitelerinin ve tarayıcıların vacgeçilmezi olan HTML/CSS ikilisi hakkında bilgi sahibi olacaksınız. Eğer dökümanı başarılı bir şekilde tamamlarsanız kendi kişisel websitenizin temellerini atmış olacaksınız. Dökümanı tamamlamak yaklaşık 3 saat kadar sürebilir. Eğer bu dökümanı herhangi bir workshopta uyguluyorsanız karşılaştığınız sorunlarda ve ya anlamadığınız noktalarda etkinlik alanındaki mentörden yardım istemekten çekinmeyin!

Bu dökümanda karşılaştığınız eksikleri, yazım yanlışlarını ve ya şöyle olsa daha anlaşılabilir olurmuş dediğiniz yerleri bizlere twitter [@kodlamag](#) ve [@bugraisguzar](#) hesaplarından bildirebilirsiniz. Aynı zamanda bu döküman github üzerinde barınıyor. Dolayısıyla github üzerinden düzeltilmiş versiyonunu gönderebilir, dökümana katkıda bulunabilirsiniz.

Hazırlık

Bu dökümanı sürdürürken ihtiyaç duyacağınız bazı şeyler var. Bunları başlamadan önce hazırlamanız gerekiyor.

- Bir kod editörü (*notepad++*, *sublime text* ve *atom* isimli editörlerden birini indirebilirsiniz)
- github.com hesabı (Ücretsiz, hesabınız yok ise lütfen şimdiden oluşturun)
- Kendinize ait bir adet fotoğraf (Eğer fotoğrafınız yoksa [bu görseli](#) kullanabilirsiniz, indirip kayıt edin)

Kaynaklar

Bu döküman hazırlanırken yararlanılan kaynaklar listesi.

- [W3Schools](#)
- [Midori Koçak'ın tweet floodu](#)



İnternet Nasıl Çalışır?

Bir websitesi yapacaksak öncelikle internetin nasıl çalıştığı konusunda en azından fikir sahibi olmamız gerekiyor. Öncelikle internetin bir çok bilgisayarı (kişisel bilgisayar, sunucu, süper bilgisayarlar) birbirine bağladığını söyleyebiliriz. Bu bilgisayarlar internet üzerinden sürekli birbirlerine anlamlı/anlamsız (bizim için anlamsız, uygulamalar için anlamlı) veriler gönderiyorlar. Örneğin Twitter'i açtığınızda gördüğünüz sayfa sizin için anlamlı ancak o sayfa ile birlikte tarayıcınıza iletilen diğer veriler (çerezler, headerler, kurabiyeler) sizin için bir anlam ifade etmese de tarayıcınız ve Twitter için çok anlam ifade edebiliyor. Bu dökümanda sadece HTML ve CSS dilleri hakkında giriş seviyesinde bilgi aktaracağımız için internetin çalışma prensibine çok derinden inmeyeceğiz.



Bu görselde internetin nasıl çalıştığı ve ana karakterlerin neler olduğu basitçe karikatürize edilmiş. Bu aşamada dikkat etmemiz gereken karakterler User (Kullanıcı), Browser (Tarayıcı) ve Hosting Server (Sunucu). Bunların ne olduğunu biraz detaylı inceleyelim.

User (Kullanıcı)

User, arama çubuğuna twitter.com yazıp siteye giren bizler oluyoruz.

Browser (Tarayıcı)

Browser ise Chrome, Firefox, Opera, Vivaldi, Safari gibi tarayıcılar oluyor. Browser, user(kullanıcı) arama çubuğuna twitter.com yazıp gönderdiği zaman *Hosting Server*'e User'in twitter.com'a girmek istediğini bildiriyor. Hosting server gerekli kontrolleri yaptıktan sonra (eğer kontrol katmanları varsa) twitter.com sitesinin HTML ve CSS dosyalarını tekrar Tarayıcıya gönderiyor. Tarayıcımız da kendisine sunucudan ulaşan bilgileri kullanıcının anlayabileceği formata çevirip, biçimlendirip ekrana yansıtır.

Hosting Server (Sunucu)

Aslında web sitelerinin kalbi sunucular diyebiliriz. Web sitelerinin temelde HTML/CSS dosyalarından oluştuğunu biliyoruz. Ve bu HTML/CSS dosyalarının kullanıcının anlayabileceği görsel formata *tarayıcılar* tarafından çevirildiğini de az önce öğrenmiştik. Dolayısıyla bu HTML/CSS dosyalarının bir yerde durması ve gerektiği/istendiği zaman tarayıcıya gönderilmesi gerekiyor. İşte bu işlemleri sunucular gerçekleştiriyor.

Sunucuları şu an bu dökümanı okuduğunuz bilgisayarınızın/telefonunuzun daha gelişmiş ve sadece dosya tutmat/göndermek için özelleşmiş çeşidi olarak düşünebilirsiniz. Biz bu dökümanın son aşamasında web sitemizi yayına alırken sunucu olarak Github'u kullanacağız.

HTML Nedir?

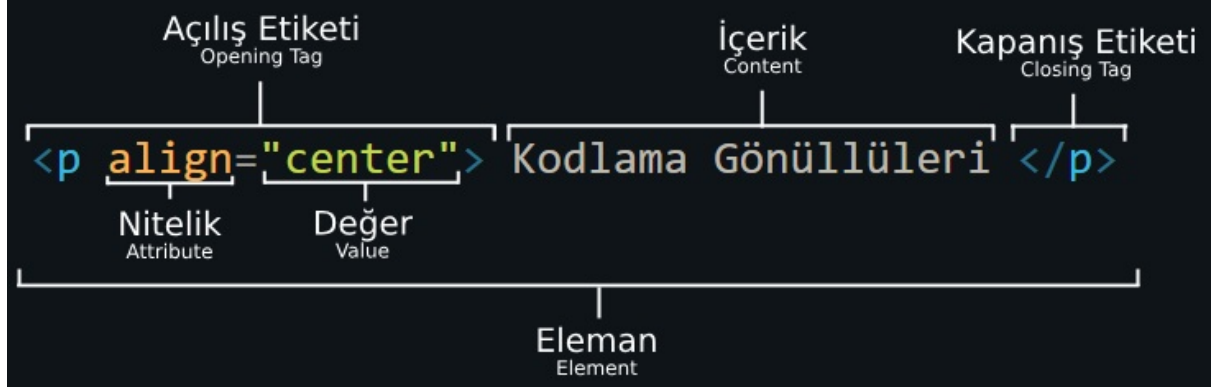
HTML (Hypertext Markup Language) bir işaretleme dilidir. Çok genel tabir olarak bu şekilde tanımlayabiliriz. Peki neyi işaretler? Sitenizde bulunan bir yazının tarayıcının neresinde duracağını, kalın mı italik mi olacağını hatta belki her iki özelliği de taşıyacağını, ekleyeceğiniz resmin konumunu, tablolarınızın kaç satır/sütundan oluşacağını işaretler. Tarayıcıların daha sonrasında bu dosyaları işleyerek kullanıcının okuyabileceği formata çevirdiğinden bir önceki kısımda bahsetmiştik.

HTML'in bir programlama dili olmadığını bilmeliyiz. Yani HTML ile mantıksal yapılar kuramaz, kullanıcı girişi/yazı ekleme gibi kompleks sayfaların işlevsel kısmını HTML ile hazırlayamayız. Ancak bu sayfaların ön yüzünü hazırlarken her zaman HTML kullanıyoruz. Günümüzde tarayıcılar sadece HTML/CSS/JS üçlüsünü tanıyorlar, dolayısıyla web sitesi hazırlarken son çıktı olarak bu üç dilden farklı bir şey kullanamıyoruz.

HTML Temelleri

HTML temel olarak ilerleyen zamanlarda **tag** olarak da anacağımız *etiketler*den oluşuyor. HTML'de açılan her etiket aynı sıra ile kapatılmak zorundadır. Yani her başlangıcın bir sonu olmalıdır. Örneğin paragraf içinde bir bölümü kalın yazdırmak istiyorsanız bu bölümün bir başlangıcı bir de sonu olmalı. Bazı etiketlerin içine içerik girebiliyoruz. Mesela paragraf oluşturmak için kullandığımız **p** etiketinin içine paragrafın kendisini yazıyoruz. Bu kısma *content* yani *içerik* diyoruz. Ve açılış etiketi, içerik, kapanış etiketi üçlüsünün oluşturduğu bütüne ise *element* diyoruz.

Ayrıca bazı HTML etiketleri özel nitelikler alabilirler. Her etiketin alabileceği nitelikler farklı olabilir. Nitelikleri açılış kısmında **nitelik="değer"** şeklinde belirtiyoruz. Çok soyut olduğu için tam olarak anlayamamış olabilirsiniz. Görsel üzerinden daha oturaklı olacaktır.



HTML Etiketleri

HTML'in ne olduđu ve yapısı hakkında biraz bilgi sahibi olduk. Şimdi bazı HTML etiketlerini öğreneceğiz. Bu etiketleri dökümanın devamında bolca kullanacağımız için anlamamız önem taşıyor. Daha sonrasında buraya geri dönüp tekrar bakmaktan çekinmeyin.

HTML İskeleti

HTML sayfalarının değışmez bir iskeleti var, bu iskelette belli başlı değışmez kısımlar bulunuyor. Öncelikle bu etiketi oluşturalım daha sonrasında üstünde konuşalım.

```
<html>
<head>
  <title></title>
</head>
<body>

</body>
</html>
```

Bu iskeleti açıklamaya başlamadan önce lütfen etiketlere bir göz atın. Her etiket açıldığı sırayla kapatılıyor gördüğünüz gibi. Bu HTML'in temel prensiplerinden biri. En başta *html* etiketi açılmış ve daha sonrasında *head* etiketi takip etmiş. Dolayısıyla bu aşamada önce *head* etiketi kapatılmalıdır. Eğer *head* etiketinden önce *html* etiketini kapatırsanız istenmeyen sonuçlarla karşılaşmanız kaçınılmaz olacaktır.

Şimdi iskelette kullanılan etiketlere göz atalım.

<html> etiketi

Bu etiketimiz tarayıcıımıza bu dökümanın bir HTML dökümanı olduğunu ve ona göre yorumlaması gerektiğini bildirir. Ayrıca diğerk tüm etiketler bu etiketin içine yazılır. Genel kapsayıcıdır diyebiliriz.

<head> etiketi

Adından da anlaşılacağı üzere baş kısmı ilgilendiren kodların yazıldığı bölümdür. İleride stil (CSS) dosyalarımızı bu kısımda içe aktaracağız.

<title> etiketi

Bu etiketimiz sayfamızın (sitemizin) başlığını barındırıyor. Bu etikete içerik olarak verdiğimiz değerk tarayıcıımızın başlık kısmında görünecek. Modern tarayıcılarda sekme adı olarak da düşünebilirsiniz.



<body> etiketi

Bu etiketimiz ise yazacağımız tüm HTML kodlarını içinde barındıracak. Yani HTML sayfamızın gövdesi olacak, sayfayı oluştururken etiketlerimizi bu etiketin içine yine usulüne uygun yazacağız.

Temel HTML şablonumuzdaki etiketlerin artık ne olduklarını ve ne işe yaradıklarını biliyoruz. Tekrar hatırlatmakta fayda görüyorum ki bu etiketlerin hepsi ilgili örnekte de gördüğünüz üzere açıldıktan sonra kapatılıyor. Daha da önemlisi açıldıkları sıra ile kapatılıyor. Önce açılan önce kapatılmalı. Bir sonraki aşamaya geçip sayfamızı oluştururken kullanacağımız diğer etiketleri de öğrenelim.

HTML Etiketleri (devam)

Bir önceki kısımda gördüğümüz etiketler kullanmamız gereken, bir HTML sayfasının temel taşlarını oluşturan etiketlerdi. Ancak bu sayfada göreceğimiz tüm etiketler bizim hayal gücümüzü tarayıcıya aktarmamızı sağlayacak etiketler. Yani bu etiketleri istediğimiz kombinasyonlarda kullanarak istediğimiz web sayfasını oluşturabiliriz. Ancak tahmin edebileceğiniz üzere HTML etiketleri buradakiler ile sınırlı değil. Biz burada sadece dökümanın ilerleyen aşamalarında işimize yarayacak olanları göreceğiz. Diğer tüm HTML etiketleri için [w3schools/tags \(ingilizce\)](https://www.w3schools.com/html/html_tags.asp) sayfasını ziyaret edebilirsiniz.

Yazı Etiketleri

Web sayfaları oluştururken her zaman büyüğü işler yapan kodlar yazmıyoruz. Bazen de düz yazılar yazmamız gerekiyor ki çoğu zaman sayfaların temellerini bu düz yazılar oluşturuyor. Ve biz yazı yazarken bazı noktaları kalınlaştırmak, başlıklar oluşturmak isteyebiliyoruz. Tam bu noktada şimdi göreceğimiz etiketlerden yardım alıyoruz. Ancak önemli bir konuyu hatırlamamız gerekiyor. Dökümanın HTML'e Giriş kısmında HTML etiketlerinin yapısından bahsetmiştik. Bazı etiketlerin açılması ile kapanması arasına değerler girebildiğimizi ve bunlara içerik (content) dediğimizi hatırlıyor olmamız gerekiyor.

Başlıklar

Aslında başlık oluşturmak HTML'de çok basit. Başlık olmasını istediğimiz yazıyı içerik olarak `<h1>` etiketine verdiğimiz zaman o bizim için her şeyi yapıyor. Ancak bazen farklı boyutlarda başlıklar oluşturmamız gerekebiliyor. Tam bu noktada `<hX>` şeklinde kullanabiliyoruz bu etiketi. X yerine 1 ile 6 dahil olmak üzere 1 ile 6 arasında bir rakam girebiliyor ve başlığın boyutunu belirleyebiliyoruz. Bilmemiz gereken nokta ise rakam ne kadar büyür ise (en fazla 6 olabiliyor) başlığın o oranda küçüleceği. Yani ters orantı var.

```
<h1>Ben h1 ile oluşturulmuş bir başlığım</h1>
<h2>Ben h2 ile oluşturulmuş bir başlığım</h2>
<h3>Ben h3 ile oluşturulmuş bir başlığım</h3>
<h4>Ben h4 ile oluşturulmuş bir başlığım</h4>
<h5>Ben h5 ile oluşturulmuş bir başlığım</h5>
<h6>Ben h6 ile oluşturulmuş bir başlığım</h6>
```

Bu örneğimizin tarayıcıda verdiği çıktıya hep beraber bakalım.

Ben h1 ile oluşturulmuş bir başlığım

Ben h2 ile oluşturulmuş bir başlığım

Ben h3 ile oluşturulmuş bir başlığım

Ben h4 ile oluşturulmuş bir başlığım

Ben h5 ile oluşturulmuş bir başlığım

Ben h6 ile oluşturulmuş bir başlığım

Kalın Yazı

`` etiketine içerik olarak verdiğimiz yazı tarayıcılarımız tarafından normal yazılara kıyasla daha kalın yazdırılacaktır.

Paragraflar

Günlük hayatta bile yazılar yazarken (özellikle de uzun yazılar) okunabilirliği arttırabilmek adına paragraflar halinde yazıyoruz. HTML'de de paragraflar oluşturmak için `<p>` etiketinden yararlanıyoruz. Her paragrafı **p** etiketi içinde yazacağız. Böylelikler daha sonra CSS ile biçimlendirirken işimiz kolay olacak.

Resim/Görsel

Sayfamız sadece yazılardan oluşmuyor. Bir çok zaman resimlerden de yardım alıyoruz. Bu noktada `` etiketi bizim yardımımıza koşuyor. HTML'de etiketlerin kapatılması gerektiğinden bahsetmiştik. Ancak bazı istisna etiketlerimiz de yok değil. Örneğin **img** etiketimizi açıyoruz ancak kapatmıyoruz. Çünkü bu etiket içerik (content) olarak bir değer almıyor ve dolayısıyla kapatılmasına da gerek olmuyor. Yani bir kapsama alanı yok, kendisi bir işlemi gerçekleştiriyor sadece diyebiliriz.

HTML'e Giriş kısmında bazı etiketlerin nitelikler alabildiğinden bahsetmiştik. Bu etiketlerden biri `` etiketi. Bu etikete **src** adında bir nitelik ile değer olarak görselin adresini veriyoruz ve o görseli sayfamıza eklemiş oluyoruz. Adres olarak görselin internet üzerindeki bir adresini verebileceğimiz gibi eğer HTML sayfamız ile erişilecek bir konumdaysa dosya yolunu da verebiliriz. Örneğin test.html sayfamız ve test.png dosyamız aynı klasörde olsunlar. Biz HTML sayfamız içinde `` kodumuzu yazarsak yine görsel sayfamıza eklenmiş olacak.

`<a>` etiketi

`<a>` etiketi ile farklı bir etikete ya da bir yazıya bağlantı özelliği verebiliyoruz. HTML'de nitelik alabilen bir diğer etiket de **a** etiketimiz. Kendisine **href** niteliği ile birlikte değer olarak bağlantı adresini veriyoruz.

```
<a href="www.twitter.com/kodlamag"> @kodlamag </a>
```

Örneğimizi bir HTML dosyası olarak kaydedip herhangi bir tarayıcı ile açarsak mavi renkte **@kodlamag** yazdığını ve bu yazıya tıkladığımızda ilgili Twitter adresinin açıldığını görebilirsiniz. Ayrıca farklı etiketlere de bağlantı verebildiğimizden bahsetmiştik. **a** etiketine içerik olarak farklı bir etiket (örneğin **img** etiketi ile bir görsel) vererseniz verdiğiniz etiketin çıktısı da tıklanabilir olacaktır.

`
` etiketi

Bazı durumlarda bir alt satıra geçmemiz gerekebiliyor. Bu durumda `
` etiketini kullanıyoruz. Bu etiket de *img* etiketi gibi kapatılmaya gerek duyulmayan etiketlerden.

Bölümler

Web sayfalarımızı yekpare oluşturmuyoruz. Örneğin çoğu web sayfasının üst kısmında bir menü bölümü, onun altında içerik bölümü, (eğer var ise) yan kısmında ek detaylar bölümü ve en altta da bir altbilgi bölümü görüyoruz. Bunların her biri bir bölüm olarak değerlendirilebilir.

HTML'de bazı yapıları bölümlere ayırmamız gerekiyor. İleride CSS'ye giriş yaptığımızda daha iyi anlayacağız bu durumu. Tam bu noktada `<div>` etiketi yardımımıza koşuyor. **Div** etiketi bir konteyner oluşturuyor ve kendisine verdiğimiz içeriği bu konteynerin içine koyuyor diye düşünebilirsiniz. Kendisinin alabildiği niteliklere daha sonra değineceğiz.

Ancak günümüzde bu iş biraz daha karmaşık bir hâl aldı(!). **HTML5** ile birlikte gelen özelliklerden biri de isim olarak daha spesifik etiketler. Örneğin artık menü oluştururken *div* etiketi yerine *nav* etiketini kullanıyoruz. Kendisini yine bir div olarak düşünebiliriz. Ancak ismi daha özel ve sayfamızın kodlarını daha okunabilir yapıyor. Buna benzer HTML5 ile beraber eklenen diğer etiketleri de listeleyelim.

- `<article>`
- `<aside>`
- `<details>`
- `<figcaption>`
- `<figure>`
- `<footer>`
- `<header>`
- `<main>`
- `<mark>`
- `<nav>`
- `<section>`
- `<summary>`
- `<time>`

Özetle bu etiketler *div* etiketinin daha insancıl/okunabilir hali ve bir işi yapmak için özelleştirilmiş halidir diyebiliriz.

Listeleme

HTML'de liste oluşturmanın birkaç farklı yolu var roma rakamlarıyla listeleme, harfler ile listeleme, latin rakamlarıyla listeleme ve noktalar ile listeleme. Biz şimdilik noktalar şeklinde yani madde madde listeleme ile ilgileniyoruz. Liste oluştururken bundan önce öğrendiklerimizden farklı olarak birden fazla etiket kullanacağız. `` ve `` etiketleri liste oluşturmamızda bize yardımcı olacak.

`` etiketi ile öncelikle listemizi başlatacağız ve daha sonrasında her bir liste elemanı için `` etiketini kullanacağız. Bazı büyükşehirleri içeren bir liste oluşturalım ve örnek üzerinden ilerleyelim.

```
<ul>
  <li>İstanbul</li>
  <li>Ankara</li>
  <li>İzmir</li>
  <li>Balıkesir</li>
  <li>Kocaeli</li>
</ul>
```

Gördüğümüz gibi her bir maddeyi `` etiketlerine içerik olarak vererek elemanlar oluşturduk. Ve bu elemanları da `` etiketine içerik olarak verdik. Burada listeyi oluşturan etiketimiz *ul* etiketi, *li* etiketi ise liste elemanlarını belirtiyor. Şimdi bu kodun tarayıcıda vereceği çıktıya bakalım.

- İstanbul
- Ankara
- İzmir
- Balıkesir
- Kocaeli

CSS'e Giriş

CSS (Cascading Style Sheets) için genel tabirle HTML elemanlarını stillendirmemize olanak sağlayan bir dil diyebiliriz. Örneğin HTML'de *div* etiketini görmüştük. Bu etiketlerle oluşturduğumuz kutucukların boyutlarının ne olacağını, herhangi bir taraftan içeriye girdilenip girdilenmeyeceği gibi özellikleri CSS ile belirliyoruz.

Sözdizimi

CSS stil dosyalarını oluştururken bilmemiz gereken bir şey var. Bu dosyalarda aslında HTML dosyamızdaki bir şeyi (şeyi diyorum çünkü tek bir eleman da olabilir bir sınıfa dahil eleman grubu da olabilir, birazdan daha detaylı göreceğiz) şekil bakımından düzenliyoruz. Dolayısıyla neyi düzenlediğimizi bildirmemiz gerekiyor. Tam bunu yaptığımız kısma *seçici* (*selector*) diyoruz. Daha sonrasında ise süslü parantezler içine o seçtiğimiz şeyin sahip olmasını istediğimiz özellikleri yazıyoruz. Özellikleri yazarken **özellik: değer;** formatında yazıyoruz. Aslında en sonra noktalı virgülü koymasak da olur. Ancak bu durum sadece tek bir özellik belirteceksek geçerli. Birden fazla özellik tanımlarken noktalı virgüller ile ayırıyoruz.

Bahsettiklerimizi görsel üstünde daha kolay anlayabiliriz.



Seçici

Tam olarak neyin stilini düzenleyeceğimizi belirtmek için seçicilerden faydalandığımızdan bahsetmiştim. Görseldeki örnekte görebileceğiniz gibi seçici kısmına *h1* yazılmış, biz bunun aynı zamanda bir HTML etiketi olduğunu biliyoruz. Eğer CSS yazarken seçici kısmında bir HTML etiketinin adını kullanırsak orada belirttiğimiz stil özellikleri HTML dosyamızdaki o etiketlerin tamamına uygulanacak demektir. Yani biz görselde *h1* seçicisi ile yazdığımız *color* özelliğinde rengi kırmızı yapmıştık. Bu demek oluyor ki kullandığımız tüm *h1* etiketleri kırmızı olacak.

Ancak bazı durumlarda tüm etiketleri değil de sadece bir ya da birkaç tanesinin stilini değiştirmek isteyebiliriz. Bu durumlarda da sınıflar (*classlar*) yardımımıza koşuyor. Dökümanın HTML kısmında HTML etiketlerinin nitelikler alabildiğinden bahsetmiştik. Bu niteliklerden biri de *class* niteliği. Biz CSS kodlarımızı yazarken seçici olarak *class* niteliğinin değerini kontrol edip ona göre seçim yapabiliyoruz. Ancak bu sefer seçicinin başına . (nokta) koymamız gerekiyor. Örneğin *class* niteliğinin değeri *deneme* olan etiketleri seçmek istiyorsak **.deneme** şeklinde yazmamız gerekiyor. Hadi buna bir örnek yapalım.

```
<p class="deneme">Ben kırmızı bir yazı olacağım.</p>
```

Gördüğümüz gibi *class* niteliğinin değeri *deneme* olan bir *p* etiketi oluşturduk. Şimdi de bu etiketi şekillendireceğimiz CSS kodlarımızı yazalım.

```
.deneme{  
  color: red;  
}
```

Şimdi tarayıcımızda nasıl görünüyor ona bakalım;

Ben kırmızı bir yazı olacağım.

Gördüğünüz gibi kırmızı bir yazı elde ettik. Burada dikkat edilmesi gereken bir şey var. Seçici olarak bir HTML etiketinin adını kullanırsak sayfadaki o etiketlerin tamamına etki edeceğiniz söylemiştik. Ancak seçici olarak sınıf (class) kullanırsak bu demek oluyor ki HTML içinde hangi etiket olduğu farketmeksizin o class değerine sahip tüm etiketler üzerinde çalışacak.

HTML ile CSS

HTML ile sayfalarımızı oluşturduğumuzu ve CSS ile de sayfalarımızı biçimlendirdiğimizi/güzelleştirdiğimizi öğrendik. Peki hangi CSS dosyası hangi HTML dosyasına etki edeceğini nasıl bilebiliyoruz? Biz belirtiyoruz. HTML'e giriş kısmında HTML şablonundan bahsetmiştik. Orada `<head>` etiketi ile bazı şeyleri belirtiyorduk, örneğin *title*. Bu etiketi içine `<link>` etiketi ekliyoruz ve *rel* niteliğine *stylesheet* değerini veriyoruz. Daha sonra *a* etiketinden de hatırladığımız *href* niteliğine de CSS dosyamızın dizinini değer olarak veriyoruz.

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

Son olarak dökümanımız bu şekilde görünüyor.

Başlayalım

Dökümanın bu bölümüne geldiysek artık kişisel web sayfamızın temellerini atabilecek kadar bilgiye sahibiz demektir. Hadi o zaman sayfamızı oluşturmaya başlayalım! Öncelikle dilediğiniz bir yerde (önerim masaüstü) *websitem* adında bir klasör oluşturun. Ve daha sonra bu klasörün içine uzantısı *.html*, adı *index* olan bir dosya oluşturalım. Yani *index.html* dosyası. Bir de yanına *style.css* dosyası oluşturalım. Daha önceden hazırladığımız, kendimize ait bir fotoğrafı da yine bu klasör içine kopyalayalım ve adını *fotografim* olarak değiştirelim. Türkçe karakter kullanmadığımıza dikkat edin. Klasörümüzün son görünümü aşağıdaki gibi olmalı.

```
--- websitem\  
    |-- index.html  
    |-- style.css  
    |-- ben.jpg (ve ya .png)
```

Şimdi bir sonraki bölüme geçip kodlarımızı yazmaya başlayabiliriz.

HTML'i Hazırlayalım

İlk yapacağımız şey HTML sayfamızı hazırlamak olacak. Şekillendirme kısmına daha sonra geçeceğiz. Şimdi daha önce indirip kurduğumuz ya da zaten kullandığımız kod editörü ile *index.html* dosyasını açalım. İçine HTML şablonumuzu yazacağız. Bu noktada kopyala yapıştır yapmanızı önermesem de kabul edilebilir. Ancak buradan sonraki aşamalarda kopyala/yapıştır yazmak yerine dökümandan bakarak kendinizin yazması daha sağlıklı olacaktır. Hatta bazı noktaları değiştirmekten/fazlasını eklemekten çekinmeyin. Eğer bu dökümanı bir etkinlikte uyguluyorsanız zaten sizinle ilgilenen deneyimli birisi var demektir. Olası bir hata durumunda ona danışabileceğinizi aklınızdan çıkarmayın. Eğer dökümanı evde uyguluyorsanız olası bir durumda Twitter üzerinden [@bugraisguzar](#) (bu dökümanı hazırlayan *Kodlama Gönüllüleri* gönüllüsü) ile iletişime geçerek sorularınızı sorabilirsiniz. Ya da ben@bisguzar.com adresine mail atabilirsiniz.

```
<html>
<head>
  <title>Ben Buğra İşgüzar</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>

</body>
</html>
```

Başlangıç için *index.html* sayfamızı bu şekilde düzenleyelim. Ben *title* yani başlık kısmına kendi adımı yazmayı tercih ettim. Siz de kendi adınızla değiştirebilir, *benim sitem*, *ilk sitem* gibi istediğiniz herhangi bir başlığı koyabilirsiniz. Yaratıcılığınıza kalmış!

Burada dikkat etmemiz gereken kısım `<link rel="stylesheet" href="style.css">` satırı. bu satırda CSS dosyamızı çağırdığımızı daha önce öğrenmiştik. HTML ile CSS dosyalarımız aynı dizinde bulunduğu için *href* niteliğine direk dosyanın adını değer olarak verebiliriz. Şimdi sayfamızın içeriğini oluşturmaya başlayabiliriz. Bu aşamadan sonra yazdığımız kodların tamamını *body* etiketinin içeriği olarak yazacağız.

Buradan sonra öncelikle ne yapacağımızdan bahsedeceğiz ve daha sonra hazırladığımız kodu göreceğiz. Lütfen hazır koda bakmadan önce kendiniz yazmak istemiyorsanız bile aklınızda hangi etiketleri nasıl ve nerede kullanacağımızı hayal edin. Böylelikle bilgilerin kalıcılığı yüksek oranda artacaktır. Bu tamamen sizin öğrenme yöntemlerinize göre değişiklik gösterecektir ancak dökümana katılım gösterirseniz kalıcılık kesinlikle artacaktır.

Öncelikle sayfamızın bir başlık kısmı olacak o yüzden bir *header* etiketi oluşturalım. Bu etiketin içine de önce *h1* etiketi ile adımızı soyadımızı yazalım altına da fotoğrafımızı koyalım. Onun altına da bir menü listesi oluşturalım. Menüümüzü daha sonra işlevsel hale getireceğiz, şimdilik sadece şablonumuzu hazırlayalım. Menüümüzde bulunacak elemanlar;

- Özet
- Deneyim
- Eğitim
- Yetenekler
- İletişim

Hadi şimdi sayfamızın *header*ini yani başlık kısmını hazırlayalım.

```
<header>
  <h1>Buğra İŞGÜZAR</h1>
  
  <ul>
    <li>Özet</li>
    <li>Deneyim</li>
    <li>Eğitim</li>
    <li>Yetenekler</li>
    <li>İletişim</li>
  </ul>
```

</header>

Burada adınızı soyadınızı düzenlemeyi unutmayın. Şimdi tarayıcımızdaki çıktıya bakalım.

Buğra İŞGÜZAR



- Özet
- Deneyim
- Eğitim
- Yetenekler
- İletişim

Evet istediğimiz çıktıyı alıyoruz. Fotoğrafın çok büyük olması dikkat çekmiş olabilir, boyutlarını daha sonra CSS ile düzenleyeceğiz. Şimdi menüde oluşturduğumuz kısımların içeriklerini oluşturmaya başlayalım. Header kısmını bitirdiğimize göre sayfamızın içeriğini oluşturmaya başlıyoruz demektir. Ana içeriği de *main* etiketi içine yazacağız. Menüdeki her elemanı bir bölüm olarak ayıracağız ve bu yüzden *section* etiketi ile her bölümün sınırlarını belirleyeceğiz. Dolayısıyla *main* etiketi içinde

section etiketleri olacak. Öncelikle özet kısmını oluşturalım. Bir *section* etiketi oluşturalım ve onun içine de *h2* etiketi ile **Özet** başlığı atalım. Daha sonrasında da bir paragraf oluşturalım ve kendimiz hakkında kısa bir özet yazalım. En son etiketleri kapattığımızdan emin olalım.

```
<main>
  <section>
    <h2>Özet</h2>
    <p>Yazılıma meraklı bir sağlıkçıyım. Aynı zamanda gezmeyi/organizasyonlarda görev almayı seviyorum.</p>
  </section>
</main>
```

Bu kısımları kendimize göre düzenlemeyi, kendi özetimizi yazmayı unutmayalım. İstedığınız kadar uzatabilir, hatta belki birkaç resim ekleyebilirsiniz. Şimdi de deneyim kısmını oluşturalım.

```
<section>
  <h2>Deneyim</h2>
  <ul>
    <li>
      <b>Simaka Türkiye</b><br>
      <small>Web Developer</small>
      <br>
      Aralık 2017 - Haziran 2018<br>
    </li>

    <li>
      <b>bisguzarcom</b><br>
      <a href="http://www.bisguzar.com">bisguzar.com</a><br>
      <small>Yazar</small>
      <br>
      Ağustos 2017 - Halen<br>
    </li>
  </ul>
</section>
```

Bu bölümün de yine *main* etiketinin içinde olacağını unutmayalım. Kendinize göre düzenleyip yeni deneyimler ekleyebilir, fazlasını silebilirsiniz. Şimdi eğitim bilgilerimizi içerecek kısmı oluşturalım.

```
<section>
  <h2>Eğitim</h2>
  <ul>
    <li>
      <strong>Marmara Üniversitesi</strong><br>
      <a href="https://www.marmara.edu.tr/" itemprop="url">marmara.edu.tr</a><br>
      İlk ve Acil Yardım
      <br>
      2018 - Halen<br>
    </li>
  </ul>
</section>
```

Bu kısmı da kendi bilgilerinizle oluşturduğunuzu varsayarak devam edelim. Sırada yetenekler bölümümüz var.

```
<section>
  <h2>Yetenekler</h2>
  <ul>
    <li>
      Python/Flask
    </li>
    <li>
      Sqlite/MongoDB
    </li>
  </ul>
```

```

        HTML/CSS
    </li>
    <li>
        Cyclist
    </li>
</ul>
</section>

```

Bu bölümleri oluştururken her birini *section* etiketi içinde yazacağımızdan bahsetmiştik. Her section yani bölümün de içinde ne olduğunu belirten başlıklar yazdık farkettiğiniz gibi. Daha sonra da o bölümün içeriğini. Şimdi son olarak iletişim bölümümüzü de oluşturup sayfamızın HTML kısmına son verelim.

```

<section>
    <h2>İletişim</h2>
    <ul>
        <li>
            <a href="mailto:ben@bisguzar.com">E-mail</a>
        </li>
        <li>
            <a href="www.twitter.com/bugraisguzar">@bugraisguzar</a>
        </li>
    </ul>
</section>

```

Burada dikkatinizi çeken bir kısım olmalı. e-mail yazısına bağlantı verirken *a* etiketine *href* değeri olarak **mailto:ben@bisguzar.com** adresini verdik. Ancak bunun bir web adresi olmadığını görebiliyoruz. Bu da *a* etiketinin özelliklerinden biri. *mailto* yapısı ile kullandığımızda bu bağlantıya tıklayan kişinin varsayılan e-mail programı (windowsda genellikle outlook) açılacak ve *ben@bisguzar.com* adresine mail göndermek için pencere çıkacak.

HTML sayfamızın son halini kontrol edelim.

```

<header>
    <h1>Buğra İŞGÜZAR</h1>
    
    <ul>
        <li>Özet</li>
        <li>Deneyim</li>
        <li>Eğitim</li>
        <li>Yetenekler</li>
        <li>İletişim</li>
    </ul>
</header>
<main>
    <section>
        <h2>Özet</h2>
        <p>Yazılıma meraklı bir sağlıkçıyım. Aynı zamanda gezmeyi/organizasyonlarda görev almayı seviyorum.</p>
    </section>
    <section>
        <h2>Deneyim</h2>
        <ul>
            <li>
                <b>Simaka Türkiye</b><br>
                Web Developer
                <br>
                Aralık 2017 - Haziran 2018<br>
            </li>
            <li>
                <b>bisguzarcom</b><br>
                <a href="www.bisguzar.com">bisguzar.com</a><br>
                Yazar
                <br>
                Ağustos 2017 - Halen<br>
            </li>
        </ul>
    </section>

```

```
</section>
<section>
  <h2>Eğitim</h2>
  <ul>
    <li>
      <strong>Marmara Üniversitesi</strong><br>
      <a href="https://www.marmara.edu.tr/" itemprop="url">marmara.edu.tr</a><br>
      İlk ve Acil Yardım
      <br>
      2018 - Halen<br>
    </li>
  </ul>
</section>
<section>
  <h2>Yetenekler</h2>
  <ul>
    <li>
      Python/Flask
    </li>
    <li>
      Sqlite/MongoDB
    </li>
    <li>
      HTML/CSS
    </li>
    <li>
      Cyclist
    </li>
  </ul>
</section>
<section>
  <h2>İletişim</h2>
  <ul>
    <li>
      <a href="mailto:ben@bisguzar.com">E-mail</a>
    </li>
    <li>
      <a href="www.twitter.com/bugraisguzar">@bugraisguzar</a>
    </li>
  </ul>
</section>
</main>
```

Yazdıklarımızı *index.html* dosyasına kayıt edelim ve bu dosyayı bir tarayıcı ile açıp kontrol edelim. Evet HTML ile işimizi *şimdilik* bitirdik, bir sonraki bölüme geçelim ve sayfamızı CSS ile biraz güzelleştirelim.

Görünüşü Güzelleştirelim

Şu andan itibaren daha çok hazırladığımız sayfamızın görünüşünü günümüz tasarım anlayışına göre şekillendireceğiz. Bu sayfayı tek başlık altında hazırladığımız için anlam karmaşası yaşayabilirsiniz. Bu gibi durumlarda mentörünüzden destek istemekten çekinmeyin. Dolayısıyla yazdığımız her ufak kod sayfanın görünüşünde farklılık yaratacak. Bu farklılıkları canlı şekilde takip edebilmek için *index.html* sayfasını tarayıcı ile açalım ve açık kalsın. Bu bölümün ilerleyen noktalarında tarayıcıya geri dönüp sayfayı yenileyeceğiz ve yaptığımız değişiklikleri gözlemleyebileceğiz. Yazdığımız CSS kodlarının karşılığını somut olarak anında gördüğümüz zaman kalıcılığı da artacaktır. Bu bölümde yapacağımız tüm işlemleri *style.css* dosyamız üzerinde yapacağız. Dolayısıyla dosyamızı kod editörümüz ile açalım.

Öncelikle sayfamızın genel görüntüsünü düzenleyerek başlayalım. Web siteleri de bir kompozisyonndan oluşur. Ve bu kompozisyonu dönemin trendleri belirler. Modern herhangi bir web sitesine girdiğiniz zaman ana içeriğin sayfanın orta taraflarına doğru konumlandırıldığını görebilirsiniz. Biz de sayfamızın içeriğini ortalayalım ve okunabilirliği artıralım. Bunun için seçicisi *body* etiketi olan bir css kodu hazırlayalım.

```
body {  
  
}
```

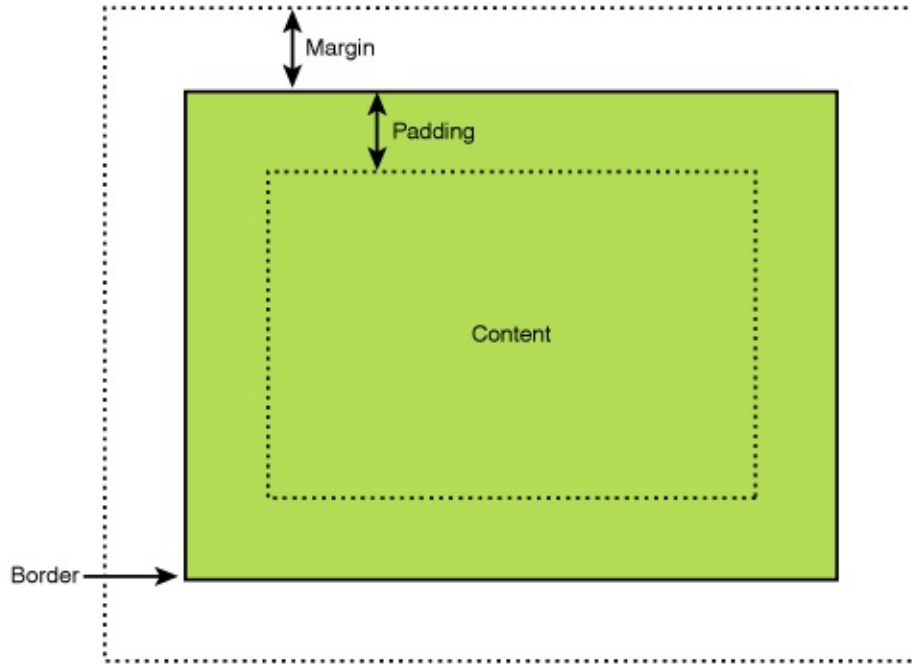
İlk olarak kullanacağımız özelliğimiz *margin* özelliği. Margin özelliği ilgili elemanın dış kısımlarından ne kadar öteleneceğini belirlemek için kullanılıyor. Biz şimdi değeri *0 auto* olan bir *margin* özelliği ekleyelim bu kodumuza.

```
body {  
  margin: 0 auto;  
  max-width: 50em;  
}
```

Dosyayı kaydettikten sonra tarayıcıma geçip *index.html* dosyasını yenilersek üst-alt ve yanlardan biraz boşluk oluştuğunu ve içeriğimizin ortalandığını görebiliriz. Burada *margin* özelliğine değer olarak aslında iki farklı değer verdik. Bu değerlerden ilki olan *0* değeri elemanın üst ve alttan ne kadar öteleneceğini ifade ediyor. 2. değer olan *auto* ise sol ve sağ dıştan ne kadar öteleneceğini belirtiyor. Bunların yanında bir de *max-width* özelliğini *50em* olarak ayarladık gördüğünüz gibi. Bunun sebebi ise genişliği belli olmayan bölme/div (body etiketini de bir bölme/div olarak düşünebilirsiniz) aslında genişlik olarak ekranın alabildiği kadar geniştir. Dolayısıyla biz genişliğin tamamını kaplayan bir elemanı yanlardan öteleyemeyiz çünkü ötelenebilecek boş bir değer yok. Değer olarak verdiğimiz *50em* ise bir ölçü birimi. CSS stilleme dilinde *em* ölçü birimi aslında mevcut yazı fontunun kaç katı olacağını belirtiyor. Örneğin **2em = 2*yazı fontu**.

Örneklendirelim bu konuyu. Ekran genişliğimizin 100em olduğunu düşünelim. Bunun 50em'lik kısmını bizim body etiketimiz kullanacak. geriye 50em kalacak. Bu etiket de sağ ve soldan otomatik olarak öteleneceği için soldan 25em sağdan da 25em ötelenecek ve ortalananak demektir.

Tarayıcıda canlı olarak da görebildiğimiz gibi içeriğimiz yanlardan ortalandı ancak üst ve alttan hâlâ sınırlara dayalı. Bu sefer üst ve alttan öteleme için margin etiketini değil de *padding* etiketini kullanalım. Margin etiketinin o elemanın dıştan ötelenmesini sağladığından bahsetmiştik. *padding* etiketi ise elemanın içten ötelenmesini sağlıyor.



Bu görselde yeşil alan HTML elemanını temsil ediyor. Yani bizim senaryomuzda `body` etiketini. Şimdi de `body` seçicimize yazdığımız özelliklere bir de `padding` özelliğini ekleyelim ve değerini de `4em 1em` yapalım. Bunlar da sırasıyla üst-alt ve sol-sağ içe ötelemelerini temsil ediyor.

```
body {
  margin: 0 auto;
  max-width: 50em;
  padding: 4em 1em;
}
```

Şu anda kodumuzun son hali yukarıdaki gibi. Ve tarayıcıya geçip sayfayı yenilediğimizde istediğimiz her şeyin olduğunu görebiliyoruz. Biraz da yazıları okunabilir kılalım. Çünkü sayfamızın genel yapısı artık göze daha fazla hitap ediyor ancak yazılar günümüz standartlarına pek de uymuyor. Bunun için aşağıdaki kodları da yine aynı seçicimizin özellikleri arasına ekleyelim.

```
font-family: "Helvetica", "Arial", sans-serif;
line-height: 1.5;
color: #555;
```

Bu kodları sırasıyla açıklayalım. İlk satırda aslında yazılarımızın fontunu değiştirdik. `body` etiketi sayfamızı oluştururken kullandığımız diğer tüm HTML etiketlerini kapsadığı için bu etikette yapacağımız değişiklikler diğer tüm etiketlerde de geçerli olacak. Özetle HTML ve CSS'de bir hiyerarşiden bahsedebiliriz. Bu da demek oluyor ki fontu *Helvetica* olarak değiştirmemiz tüm sayfada fontun değişmesi anlamına geliyor.

İkinci satırda da satırların boyutunu ayarladık. `1.5` değeri vererek normal satır yüksekliğini normalinin `1.5` katı olarak ayarlamış olduk. Son satırda da yazı rengini griye yakın bir renk olan `#555` rengi ile değiştirdik. Bir de arka plan rengini düzenleyerek bu kısımdaki düzenlemelerimizi sonlandıralım. `background-color` özelliğine de `#f2f2f2` değerini verelim.

```
body {
  margin: 0 auto;
  max-width: 50em;
  padding: 4em 1em;

  font-family: "Helvetica", "Arial", sans-serif;
  line-height: 1.5;
  color: #555;
}
```

```
background-color: #f2f2f2;
}
```

Kodumuzun son hali bu şekilde olacak. Eğer siz kodunuzu kendiniz yazdıysanız kıyaslayabilirsiniz. Tekrar tarayıcımız üzerinden kontrol edelim ve sayfamızın ilk haliyle şimdiki hali arasındaki belirgin farkı gözetleyelim.

Daha öncesinde hatırlıyorsanız sayfamıza görsel eklediğimizde görselimiz abartı boyutlardaydı. Tabii bu benim görselim boyutlarından dolayıydı, sizin eklediğiniz fotoğrafınız daha küçük ve ya büyük olabilir. Ancak biz yine de resimin kaplayabileceği yüksekliği sınırlarsak standart bir görünüm elde edebiliriz. *max-width* özelliğinin elemanın genişliğini sınırlandırdığından bahsetmiştik. Elemanın yüksekliğini sınırlandırmak için de *max-height* özelliğini kullanacağız. Görselimizi *img* etiketi ile eklediğimiz için seçicisi *img* etiketi olan yeni bir css kodu hazırlayalım ve *max-height* özelliğine *150px* değerini verelim.

```
img {
  max-height: 200px;
}
```

Bu şekilde görünüyormuş olmalı. Bu sefer ölçü birimi olarak **px** yani pikselleri kullandık. Bu kodlarımızı da yine *style.css* dosyası içine yazmaya devam ediyoruz. Dolayısıyla *style.css* dosyamızın son hali aşağıdaki gibi bir hal almış olmalı.

```
body {
  margin: 0 auto;
  max-width: 50em;
  padding: 4em 1em;

  font-family: "Helvetica", "Arial", sans-serif;
  line-height: 1.5;
  color: #555;

  background-color: #f2f2f2;
}

img {
  max-height: 200px;
}
```

Sırada ufak küçük dokunuşları yapmak kaldı. *body* etiketinin özelliklerini düzenlerken *color* özelliğine griye yakın bir ton vermiştik. Dolayısıyla sayfamızdaki tüm yazılar griye yakın bir tonda yazıyor. Ancak biz başlıkların daha koyu hatta siyah olmasını istiyoruz ki öne çıkabilsin ve genel okunabilirlik artsın. Bunun için başlıklara özel bir renk tanımlarsak genel renk tanımlamasını ezebiliriz/çığneyebiliriz (yani genelgeçer olanı yok sayıp özel alan oluşturabiliriz) ve istediğimizi elde edebiliriz. Ancak hatırlarsanız biz HTML kodlarımızı hazırlarken başlıkları *h1* ve *h2* etiketleri ile oluşturmuştuk. Dolayısıyla bu sefer seçicimiz iki farklı etiketi seçecek. Biz bu etiketleri , (*virgül*) ile ayırarak birden fazla etiketi seçmeyi sağlayabiliyoruz. Öncelikle seçicimizi hazırlayalım.

```
h1, h2{
}
```

Bu şekilde iki farklı etiketi seçecek ve yazdığımız özelliklerin her ikisi üzerinde de uygulanmasını sağlayacağız. Şimdi de *color* özelliğine *black* değerini verelim ve başlıklarda yazıların siyah olmasını sağlayalım.

```
h1, h2{
  color: black;
}
```

Şimdi tüm dosyayı kayıt edelim ve tarayıcı üstünden eserimizi kontrol edelim. Kısa bir CSS yazarak sayfamızı nasıl okunabilir hale getirdiğimizi gördük. Tabii ki CSS bu kadar kısa sürede kavranabilecek bir şey değil. Biz burada sadece ne olduğu ve nasıl/ne amaçlarla kullanıldığı konularında bilgi sahibi olmayı hedefliyorduk ve olduk da. *style.css* sayfamızın son halini kıyaslama yapmak isteyenler için aşağıya bırakalım. Renk özelliklerini, boşluk değerlerini kafanıza göre düzenleyip kendi kişisel dosyanızı oluşturmak daha zevkli olacaktır. Çekinmeyin!

```
body {  
    margin: 0 auto;  
    max-width: 50em;  
    padding: 4em 1em;  
  
    font-family: "Helvetica", "Arial", sans-serif;  
    line-height: 1.5;  
    color: #555;  
  
    background-color: #f2f2f2;  
}  
  
img {  
    max-height: 200px;  
}  
  
h1, h2{  
    color: black;  
}
```

Son Dokunuşlar

HTML kodlarımızı yazarken bir menü oluşturduğumuzu hatırlıyorsunuz. Bu menüyü artık işlevsel yapma zamanı geldi. *a* etiketinin sadece web sitelerine bağlantı vermediğinden bahsetmiştik. Bu etiketin bir özelliği de sayfa içinde bağlantı verebiliyor olması. Peki bu bağlantıyı nereye veriyor? Diğer HTML etiketlerine. Peki hangi etikete bağlantı vereceğimizi nasıl seçiyoruz? Etiketlerin **id** niteliğine eşsiz kimlik atayarak. Şimdi oluşturduğumuz *özet*, *deneyim*, *eğitim*, *yetenekler*, *iletişim* bölmelerine eşsiz id yani kimlikler atayalım. Sırasıyla bu bölümlere atayacağımız id değerlerini listeliyorum.

- bölme -> alacağı değer
- özet -> *ozet*
- deneyim -> *deneyim*
- eğitim -> *egitim*
- yetenekler -> *yetenekler*
- iletişim -> *iletisim*

Özetle o bölüme belirlediğimiz başlıktan Türkçe karakterleri düzenleyerek idyi tanımlayacağız. İlk bölümümüz olan *özet* kısmının section etiketinin id'sini tanımlayalım.

```
<section>
  <h2>Özet</h2>
  <p>Yazılıma meraklı bir sağlıkçıyım. Aynı zamanda gezmeyi/organizasyonlarda görev almayı seviyorum.</p>
</section>
```

HTML kodlarımızı yazarken oluşturduğumuz *özet* bölümü böyle görünüyor olmalıydı (içeriğin sizin bilgilerinize göre farklılık gösterebileceğini unutmayalım). Şimdi *section* etiketinin *id* niteliğine *ozet* değerini tanımlayalım.

Sonuç olarak kodlarımız artık şu şekilde görünüyor olmalı.

```
<section id="ozet">
  <h2>Özet</h2>
  <p>Yazılıma meraklı bir sağlıkçıyım. Aynı zamanda gezmeyi/organizasyonlarda görev almayı seviyorum.</p>
</section>
```

Bu işlemi diğer tüm *section* etiketleri için tekrarlayalım. Ancak her birine içeriğine göre farklı bir id tanımlamaya dikkat edelim. Bu yaptığımız işlemler her bir *section* etiketine bir kimlik vermiş olduk. Şimdi menümüzdeki elemanlara birer bağlantı ekleyerek bu kimliğin olduğu bölgeye yönlendirmesini sağlayacağız. Böylece aynı sayfa içinde bağlantılar oluşturmuş olacağız.

Menü kısımımızın HTML kodları nasıldı bir hatırlayalım.

```
<ul>
  <li>Özet</li>
  <li>Deneyim</li>
  <li>Eğitim</li>
  <li>Yetenekler</li>
  <li>İletişim</li>
</ul>
```

Şimdi buradaki her elemana ilgili alanına bağlantı ekleyeceğiz. Örneğin *Özet* elemanına id'si *ozet* olan bölgeye yönlendiren bir bağlantı ekleyeceğiz. Hepsini eklediğimizde şuna benzer görünüyor olmalı.

```
<ul>
  <li>
    <a href="#ozet">Özet</a>
  </li>
  <li>
```



```
        <a href="#deneyim">Deneyim</a>
    </li>
    <li>
        <a href="#egitim">Eğitim</a>
    </li>
    <li>
        <a href="#yetenekler">Yetenekler</a>
    </li>
    <li>
        <a href="#iletisim">İletişim</a>
    </li>
</ul>
```

Kayıt edip sayfamızı tarayıcı ile açarsak artık menümüzün de çalıştığını görebiliriz. Ve kişisel sayfamızı başarıyla hazırladık! Sıra sayfamızı internette yayınlamaya geldi, bir sonraki bölüm de bunun hakkında.

Sayfamızı Yayınlayalım

Dökümana başlarken ihtiyacımız olacaklar kısmında bir *github.com* hesabına ihtiyacımız olacağını belirtmiştik. Eğer bir hesap açmadıysanız şimdi [www.github.com](https://github.com) adresine gidip bir hesap oluşturabilirsiniz. Hesap oluşturduktan sonra kendimize *github.com* üzerinde bir repository yani bir depo oluşturacağız. Dökünün *İnternet Nasıl Çalışır?* bölümünde web sitelerinin bir kaynaktan servis edildiğini söylemiştik. Burada bizim kaynağımız *github.com* olacak. Yani biz dosyalarımızı *github.com*'a göndereceğiz o da sayfamıza erişmek isteyen kullanıcılara servis edecek. Bunun için dosyalarımızı saklayabileceğimiz bir depoya ihtiyacımız var. Github'a giriş yaptıktan sonra <https://github.com/new> adresine gidelim ve buradan yeni bir depo oluşturalım. Depo adı kısmına (Repository Name) *github* kullanıcı adımızı yazıp sonunda *'github.io'* ekleyeceğiz. Benim kullanıcı adım *bisguzar* olduğu için depo adı kısmına ***bisguzar.github.io*** yazacağım.

Daha sonra *Create repository* butonuna tıklayarak depomuzu oluşturalım. Karşımıza depomuza nasıl dosya yükleyebileceğimiz ve depomuzu nasıl kontrol edeceğimizi anlatan kısa bir döküman çıktı.

Quick setup — if you've done this kind of thing before

Set up in Desktop

or

HTTPS

SSH

https://github.com/bisguzar/me.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```

echo "# me" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/bisguzar/me.git
git push -u origin master

```

...or push an existing repository from the command line

```

git remote add origin https://github.com/bisguzar/me.git
git push -u origin master


```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

Biz burada görselde de işaretlenen yere yani *uploading an existing file* bağlantısına tıklayacağız. Daha sonra *index.html*, *style.css* ve *ben.jpg/png* dosyamızı buraya sürükleyeceğiz.




Drag additional files here to add them to your repository

Or [choose your files](#)

index.html

style.css

ben.png



Commit changes

Dosyalar yüklendi.

Add an optional extended description...

☒ Commit directly to the `master` branch.
 ☐ Create a **new branch** for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes

Cancel

Dosyalarımızı sürükledikten sonra aşağıda listelenmiş şekilde görebiliyoruz görseldeki gibi. Daha sonra *Commit changes* alanındaki başlık kısmına yaptığımız değişikliği yazıyoruz. Biz dosya yüklediğimiz için *Dosyalra yüklendi.* yazabiliriz. Daha sonrasında da *Commit changes* butonuna tıklayarak işlemi tamamlayıp dosyalarımızı depomuza gönderiyoruz. Artık depomuzdaki o dökümanın gittiğini ve onun yerine dosyalarımızın geldiğini görebiliriz. Şimdi ise depo sayfamızın sağ üstündeki *Settings* sekmesine geçelim.

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

No description, website, or topics provided.

Edit

Manage topics

1 commit

1 branch

0 releases

1 contributor

Branch: master


New pull request

Create new file

Upload files

Find file

Clone or download


bisguzar Dosyalar yüklendi.

Latest commit a3db9f6 just now

ben.png

Dosyalar yüklendi.

just now

index.html

Dosyalar yüklendi.

just now

style.css

Dosyalar yüklendi.

just now

Help people interested in this repository understand your project by adding a README.

Add a README

Burada aşağıya doğru indiğimizde *GitHub Pages* alanını göreceğiz. Buradan *Source* seçeneğini *master branch* olarak seçelim ve *Save* butonuna basalım.

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Source

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more.](#)

master branch ▼

Save

Theme Chooser

Select a theme to publish your site with a Jekyll theme using the master branch. [Learn more.](#)

Choose a theme

Artık her şey hazır! Tarayıcımızdan https://github_kullanici_adimiz.github.io adresine gidersek kendi sayfamızı görebiliriz. Bağlantıyı kendi github kullanıcı adınıza göre düzenlemeyi unutmayın. Her şeyi halledip kendi kişisel sayfamızı oluşturduğumuz ve yayınladığımıza göre bunu paylaşmaya hazırız. Twitter üzerinden [@KodlamaG](#) ve [@bugraisguzar](#) adreslerini etiketleyerek bağlantınızı paylaşırsanız biz de gururlanabiliriz!