



II. BÖLÜM

WEB PROGRAMLAMA



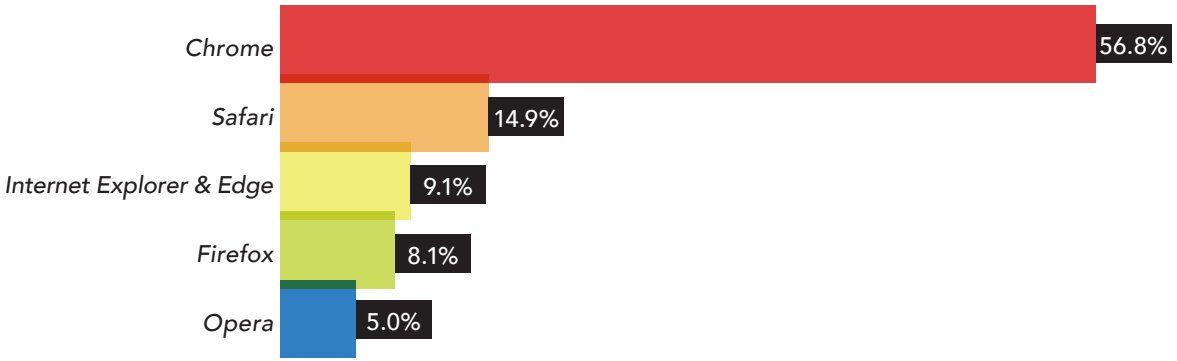
1. İNTERNET VE WEB SERVİSLERİ

1. İnternet ve Web Servisleri

Arkadaşlar biliyorum ki bilgisayarda oyun oynamayı çok seviyorsunuz. Özellikle çevrimiçi oyun oynamak daha heyecan verici sanırım. Akıllı cihazların da hayatımıza hızla girmesiyle bazılarımız için gittiğimiz her yerde oyun oynamak vazgeçilmez bir tutku oldu. Bununla birlikte bilgisayar ya da mobil cihazımızı açtığımızda İnternet'i bilgi aramak, alışveriş yapmak, arkadaşlarımızla sohbet etmek, arkadaşlarımızın paylaşımlarını beğenmek, paylaşımlara yorum yazmak ve daha birçok farklı amaçla kullanıyoruz değil mi? Peki hiç düşündünüz mü? İnternet nasıl çalışır, nasıl geliştirilir? Acaba siz de bir İnternet sitesi yapabilir misiniz? Gelin ilk olarak İnternet'in nasıl çalıştığına yanıt arayalım.

Web Tarayıcıları

İnternet'e bağlanmak için hepimiz birer web tarayıcısı (browser) kullanıyoruz. Fakat kullanmış olduğumuz web tarayıcılar farklılık gösterebiliyor. Ağustos 2017 verilerine göre dünya genelinde en çok kullanılan tarayıcılar Chrome, Safari, İnternet Explorer, Firefox, Opera olarak sıralanmaktadır¹ (Bkz. Şekil 1).



Şekil 1: Web tarayıcılarının dünya genelinde yüzde olarak kullanım grafiği

Bir web tarayıcısı bizim görüntülemek istediğimiz siteyi ya da kaynağı, sitenin yüklü olduğu bilgisayardan ya da sunucudan çağırarak (istek yaparak) ekrana yansıtır. Dolayısı ile web tarayıcısının bulunduğu bilgisayar istemci, görüntülenmek istenen sitenin yüklü olduğu bilgisayar ise sunucu olarak adlandırılır.

Şimdi bir örnek üzerinde bu durumu senaryolaştıralım:

- Chrome web tarayıcısını açtınız.
- Tarayıcınızın adres satırına <http://www.google.com> yazdınız ve site açıldı.
- Bu durumda sizin bilgisayarınız istemci, google web sitesi kaynak oldu.

Peki sunucuya nasıl eriştik ve sunucu nerede?

- Sunucuya <http://www.google.com.tr> ismi ile erişim sağladık.
- Web tarayıcımıza bu ismi girdiğimizde ismin bağlı olduğu adresteki bilgisayarlar ise istek yaptığımız sunuculardır. Bu sunucuların İnternet ortamında yerini gösteren adresleri IP adresleridir. Google'nin Türkiye web sitesinin ipsi 216 58 209 195'tir. Akılda daha kalıcı olması için "alan

¹ <https://www.w3counter.com/globalstats.php?year=2017&month=8>

adı sunucuları” <http://www.google.com.tr> adresi ile 216 58 209 195 ipsini eşleştirilir. Dolayısı ile daha kolay bir erişim sağlamış oluruz.

Gündelik yaşamımızda web tarayıcılarını hangi amaç için kullanırsak kullanalım web tarayıcıları aslında arka planda aynı işlemleri yapar. Yani bizlerin kaynak olarak isimlendirdiğimiz resim, metin, animasyonları görüntüleme ve bu kaynaklar üzerinde ekleme, silme, düzenleme yapmamıza olanak sağlar. Bunun yanı sıra site tarafından izin verilmişse görüntülediğimiz kaynakları kendi bilgisayarımıza indirmemiz için aracı görevi üstlenir. Son olarak siz İnternet’te gezinirken ziyaret etmiş olduğunuz sitelerin ve bu siteleri gezerken girmiş olduğunuz verilerin bilgilerini sonradan size hatırlatmak üzere tutar.

Web Teknolojileri

World Wide Web (www), Tim Berners-Lee tarafından 1989 yılında icat edilmiştir. O günden bugüne İnternet ve ilgili teknolojilerde pek çok gelişme ve değişim meydana gelmiştir. Örneğin geçmişte tarayıcılarda sadece metin görüntüleyebiliyor ve başka herhangi bir işlem gerçekleştiremiyorduk. İşte web teknolojisinin bu ilk versiyonu **Web 1.0** olarak adlandırılmaktadır. Daha sonra web teknolojileri; tarayıcıda görüntülediğimiz site üzerinde yeni kaynaklar ekleme, kaynaklar üzerinde değişiklik yapma gibi olanaklar sağlamaya başladı. Bu İnternet teknolojisi ise **Web 2.0** olarak adlandırılmaktadır. Web 2.0 ile çalışan birçok örnek verebileceğinizi düşünüyorum. Biraz ipucu ister misiniz? “Paylaşım” size ne ifade ediyor? Facebook, Instagram, Youtube, Wikipedia ... Bunlara benzer daha birçok web 2.0 ile hazırlanmış Blog, Sosyal Ağ, Viki, Elektronik Portfolyo gibi site örnekleri verebiliriz. Hadi bu türden bir site ortamında neler yapabiliyoruz sıralamaya çalışalım (Bkz. Şekil 2).



Şekil 2: Web 2.0 ile hazırlanmış bir sitede neler yapabiliriz?

Sanki bir şeyi atladık gibi değil mi arkadaşlar? İnternet’te bizim yapabildiklerimizin yanında bir de İnternet’in bizim için bizden habersiz olarak yapabildikleri var. Örneğin Google’da arama yaparken bir kelime yazıyoruz Google tamamlıyor. Aynı şekilde bir alışveriş sitesinden futbol ayakkabısı ya da elbise beğeniyorsunuz. Bir sonraki web tarayıcınızı açtığınızda benzer ürünlerin ekranınızda reklamının ya-

pıldığını görebiliyorsunuz. İşte bu web teknolojisi de akıllı web ya da anlamsal ağ yani Web 3.0 olarak adlandırılmaktadır. Gelin şimdi web 1.0, web 2.0 ve web 3.0 teknolojilerini karşılaştıralım (Bkz. Tablo 1).

Web 1.0	Web 2.0	Web 3.0
Web sayfası sadece okunabilir.	Web sayfası hem okunabilir hem de yazılabilir.	Web sayfası yazılabilir okunabilir ve programlanabilir.
Etkileşim tek yönlüdür. Yani bilgisayar gösterir, kullanıcı okur.	Etkileşim çok yönlüdür. Yani web sayfasındaki tüm kullanıcılar birbirleri ile etkileşim kurabilir.	Web sayfasındaki kullanıcılar ile bilgisayarlar ve bilgisayarlar arasında etkileşim vardır.
Kullanıcı ara yüzleri daha sade ve genellikle metin tabanlı tasarım vardır.	Zenginleştirilmiş, etkileşimli ve dokunmatik arayüz tasarımı vardır.	Yaşamla bütünleşik duyu- larla şekillenebilen arayüz- ler ve kullanıcıya göre de-ği- şen arayüz tasarımı vardır.

Tablo 1: Web 1.0, Web 2.0 ve Web.3.0 Karşılaştırma

2. İŞARETLEME DİLİNE GİRİŞ (HTML)

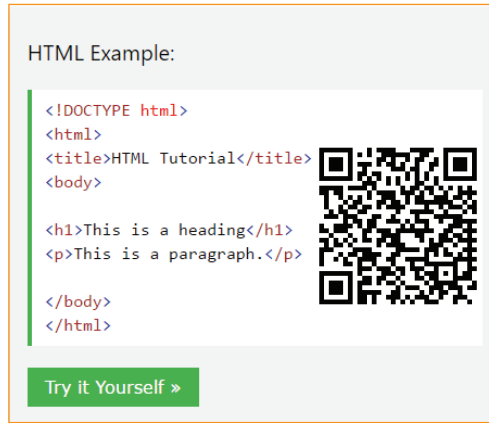
2. İşaretleme Diline Giriş (HTML)

İnternet'in nasıl çalıştığı ve web teknolojilerinin neler olduğu hakkında bilgi sahibi olduk. Bu ünite ile örnek bir web sayfasının temellerini oluşturmaya başlıyoruz.

Ön Hazırlık

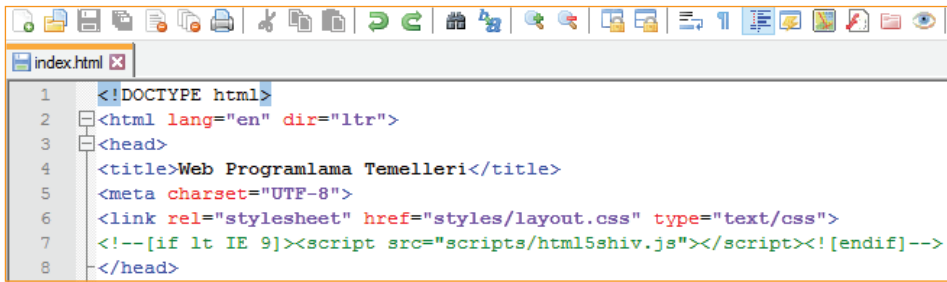
Kodlama yaparken yararlanabileceğimiz birçok editör var. Biz bu bölümde iki editörü tanıtacağız. Bu editörler dışında öğretmeninizin önerdiği editörleri de kullanabilirsiniz.

I. <https://www.w3schools.com/> Adresindeki Web Sitesi: Öncelikle siteye bağlandıktan sonra HTML kodların yazılabileceği ekrana bağlanmak için yeşil butona tıklıyoruz. Ya da QR Kodunu mobil aygıtınızdan okutabilirsiniz (Bkz. Şekil 4).



Şekil 3: www.w3schools.com editörü

II. NotPad++ Editörü: Çeşitli dilleri destekleyen ücretsiz kaynak kodu düzenleyicisidir. <https://notepad-plus-plus.org/> adresinden kişisel bilgisayarınıza indirip kolayca kurulum yapabilirsiniz.



Şekil 4: Notpad ++ editörünün genel görünümü

Editörler hakkında bilgi sahibi olduk. Şimdi çalışma dizinimizi hazırlayalım. Öncelikle, masaüstüne ismimize ait bir dizin oluşturalım. Yapacağımız bütün uygulamaları dosya uzantısı .html olacak şekilde (Ör: "dosyaismi".html) bu dizin içerisine kaydedeceğiz.

HTML

HTML, Hiper Metin İşaretleme ya da Biçimlendirme Dili (Hyper Text Markup Language) olarak Türkçe'ye çevrilmiştir. Web sitesi oluşturmak için kullanılan standart dildir. İlerleyen bölümlerde başka dillerle de tanışacağız. Şimdi, HTML kodlarının nasıl çalıştığını kavramaya çalışalım.

```
!DOCTYPE html> <!-- Kodlamada kullanılacak HTML standartını belirtir. Bu örnek için HTML5 kullanılacaktır. -->
<html> <!-- Kodun başlangıcını belirtir.-->
  <body> <!-- Web sayfasındaki metinler, sesler, resimler, animasyonların kodlandığı bölümün başlangıcı-->
    <h1> h1 ve p kodlarının işlevi </h1>
    <p> h1 kodu bir paragrafın başlığını yazarken kullanılabilir.</p>
    <p> p kodu ise; yeni bir paragrafa başlarken kullanılabilir. </p>
  </body> <!-- Web sayfasındaki metinler, sesler, resimler, animasyonların kodlandığı bölümün bitişi -->
</html> <!-- Kodun bitişini gösterir. -->
```

Resimdeki kodu istediğiniz bir editörü kullanarak yazınız.

Dikkat!

- Kodların yazılırken küçük büyük işaretleri <> arasına <html>, <p>, <body> gibi kod yazıldığını fark ettiniz mi?
- Herhangi bir koda başlarken < > işaretlerini, kodu bitirirken </ > işaretlerini kullandığımızı fark ettiniz mi?

Kodları çalıştırınız ve ekran çıktısını inceleyiniz.

Dikkat!

- Sadece siyah renkteki yazıların çıktı olarak sunulduğunu fark ettiniz mi?
- <!-- --> işaretleri arasındaki yazılı metinlerin sayfada görüntülenmediğini fark ettiniz mi?

Resim Ekleme

Şimdi birlikte bir resmin web sayfasına nasıl eklendiğine bakalım. Web sayfamıza kodunu kullanarak resim ekleyebiliriz. Öncelikle İnternet'ten kendi ilgisimize göre bir resim bulalım. Konumuz kodlama olduğu için Google'da görsellerden "web programlama" anahtar kelimesi ile arama yapıyoruz. İlgimizi çeken herhangi bir resme sağ tıklayarak resim adresini kopyalıyoruz.

```
<!DOCTYPE html>
<html>
  <body>
    <h2>Resim Ekleme</h2>
    
    <!--
      img kodu açıklamalar
      -----
      src: webden kopyaladığımız resmin adresi
      alt: imleç resmin üzerine geldiğinde görüntülenecek yazı
      style: resmin boyutları
    -->
    <p> Geleceğin kodlayıcıları olarak hepimiz çok çalışmak zorundayız. </p>
  </body>
</html>
```


Dikkat!

- Yeşil renkteki yazıların web sayfasında görüntülenmediğini fark ettiniz mi?
- < > işaretleri arasına birden fazla kod yazılabildiğini fark ettiniz mi?

```

```

Kodu çalıştırdığınızda resim ve yazının alt alta olduğunu göreceksiniz. Peki bunları yan yana nasıl getirebiliriz? Bunun birçok yolu var. Fakat başlangıç aşamasında olduğumuz için sadece resmi sağa ya da sola yaslamamızı sağlayan float komutunu göreceğiz.

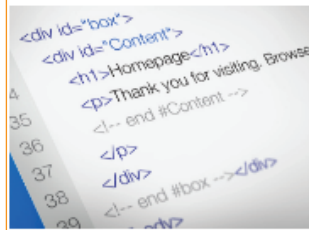
- Resim eklerken resmin genişliği (width) ve yüksekliğini (height) belirtmek için style kodunu kullanmıştık.
- Resmi sağa ya da sola yaslamak için de yine bu kodun içerisine float:left; kodunu ekliyoruz ve kodu test ediyoruz.

Dikkat!

- style kodu içerisinde birden fazla kod yazarken kodlar arasına ";" işareti eklendiğini fark ettiniz mi?
- style="width:200px;height:150px;float:left;"

- Eğer kodu çalıştırdıysanız resim ile yazının bitişik durduğunu fark etmiş olmalısınız. Resim ve yazı arasında boşluk bırakmak istediğimizde ise margin:50px; komutunu style kodu içerisine ekliyoruz.
- Fakat bu kez resmin üstünde-sağında-solunda ve altında boşluklar oluştu. Biz sadece yazı ile boşluk olsun istiyorduk. Bunun için margin kodunda bir değişiklik yapmamız gerekiyor.
- margin:50px kodunu margin-right:50px olarak değiştirdiğimizde istediğimiz sonuca ulaşacağız.

Resim Ekleme



Geleceğin kodlayıcıları olarak hepimiz çok çalışmak zorundayız.

Kodlayıcı Görevi

- margin-top, margin-left, margin-right, margin-bottom komutlarının ne işe yaradığını keşfetmeye çalışınız.

Başka Bir Sayfaya Bağlantı (Link) Verme

Öncelikle bu işlem için gerekli olan kodumuzu tanıyalım. HTML ile sayfalar arası bağlantı (link) vermemizi <a> kodu sağlar.

Bu iki kod arasına, üzerine tıkladığımızda bizi farklı bir sayfaya yönlendirecek metni ya da resmi ekliyoruz.

<a>Kodlama Öğreniyorum Sayfası

Daha sonra href ile yönlendirilecek sayfayı yazıyoruz.

```
<a href="https://www.w3schools.com">Kodlama Öğreniyorum Sayfası</a>
```

Yönlendirilen sayfanın farklı bir sekmede açılmasını istiyorsak target="_blank" kodunu kullanıyoruz.

```
<a href="https://www.w3schools.com" target="_blank">Kodlama Öğreniyorum Sayfası</a>
```

Kodlayıcı Görevi 1

- Şimdi siz de komutlarının ne işe yarayabileceğini keşfediniz.

Kodlayıcı Görevi 2

- Herhangi bir resmin üzerine tıklanığında okulunuzun web sitesine yönlendirme sağlayan kodu yazınız.

HTML5

Tim Beurners Lee Web'i icat ettiğinden bu güne kadar hem İnternet kullanıcılarının hem de kodlayıcıların ihtiyaçları değişmiştir. Buna yönelik olarak Web'i programlarken kullanmış olduğumuz diller de kendilerini güncellemektedir. Şuan en güncel html sürümü HTML5'tir. Eski HTML sürümlerinde siteyi tasarlamış olduğunuz kodların bazıları farklı web tarayıcıları tarafından desteklenmiyordu. HTML5, tüm modern tarayıcılarda desteklenir. Aynı zamanda tek bir doğru kodlama şekli yoktur. Aşağıdaki kodların hepsi en son güncel tarayıcılar tarafından çalıştırılabilir. Böylece kodlayıcının işi daha kolaylaşmış olur.

```
<input type="text" value="John" disabled>
```

```
<input type="text" value=John>
```

```
<input type="text" value="John Doe">
```

```
<input type="text" value='John Doe'>
```

- HTML5'te, var olan etiketler (kodlar) dışında etiketler yaratabiliriz (Örneğin isminizi bir etiket olarak kodlayabilirsiniz.). Bu özellik, kodlayıcılara kendilerine özgü kodlama yöntemleri geliştirmelerine olanak sağlar.

```
<!DOCTYPE html>
<html>
<head>
  <script>document.createElement("yenesne")</script> <!--ie9 ve öncesi için gerekli-->
  <style>
    yenesne {
      display: block;
      background-color: #dddddd;
      padding: 50px;
      font-size: 30px;
    }
  </style>
</head>
<body>
  <h1>Bilinmeyen Nesnenin Öğretilmesi?</h1>
  <yenesne>Daha önce tagı olmayan nesnemi yarattım</yenesne>
</body>
</html>
```

- HTML5, bir web sayfasının farklı bölümlerini tanımlamak için yeni anlamsal etiketler sunar. Anlamsal etiket; <video>, <form> gibi içeriği hakkında fikir sahibi olduğumuz elementlerdir. Yani ne tür bir kaynağı etiketlediği açıktır. Anlamsal olmayan elementler ise <div>, <p> gibi içeriği hakkında fikir sahibi olamadığımız elementlerdir. Yani içerisine birçok farklı türden kaynak alabilir. HTML5 ile <header>, <nav>, <section>, <article>, <footer>, <figure>, <mark>, <details>, <progress>, <dialog> gibi yeni anlamsal elementler sunulmaktadır. Şimdi bu elementleri tanıyalım.

<code><header> </header></code> Bir belge veya bölüm için bir başlık belirtir.	<code><section> </section></code> Genellikle başlıklı tematik bir içerik gruplandırmasıdır. Bir anasayfa giriş, içerik ve iletişim bilgileri gibi bölümlere ayrılabilir.	<code><article> </article></code> Forum yazısı, blog yazısı, gazete makalesi gibi. Bağımsız, kendine yeten içeriği belirtir.	<code><footer> </footer></code> Bir belge veya bölüm için altbilgiyi belirtir.
<code><footer> </footer></code> Bir belge veya bölüm için altbilgiyi belirtir.	<code><nav> </nav></code> Gezinti bağlantılarını belirtir.	<code><aside> </aside></code> İçeriği (yan bar gibi) bir kenara koyar.	<code><figure> </figure></code> <code><figcaption> </figcaption></code> Bir resim ve resim yazısı, bir <figure> ögesinde gruplandırılabilir.
<code><mark> </mark></code> İşaretili/vurgulanmış metinleri tanımlar.	<code><details> </details></code> Kullanıcının görüntüleyebileceği veya gizleyebileceği ek ayrıntıları tanımlar.	<code><progress> </progress></code> Görevin ilerlemesini temsil eder.	<code><dialog> </dialog></code> Bir iletişim kutusu veya pencere tanımlar.

1. UYGULAMA

Notpad++ editöründe aşağıdaki basamakları takip ederek kodlama yapalım.

- İlk olarak genel sayfa yapısını oluşturuyoruz.

```

1  <!DOCTYPE html> <!-- Sayfanın HTML5 standartında kodlanacağını belirtir -->
2  <html> <!-- HTML etiketini açıyoruz. -->
3  <body> <!-- Sayfanın içeriğini bu etiket içerisine kodluyoruz. -->
4
5  <section> <!-- HTML5 elementlerini tanıtmak için bir bölüm açıyoruz. -->
6
7  <header> <!-- Bölümümüzün başlık etiketini açıyoruz. -->
14
15 <article> <!-- Bölüm içerisindeki birinci makale içeriği -->
21
22 <article> <!-- Bölüm içerisindeki ikinci makale içeriği -->
28 </section> <!-- HTML5 elementlerini tanıttığımız bölümü kapatıyoruz.-->
29
30 <footer> <!-- Sayfanın altında görünecek altbilgi -->
31 <header>
34 <nav>
40 </footer>
41
42 </body> <!-- body etiketini kapatıyoruz. -->
43 </html> <!-- HTML etiketini açıyoruz. -->

```

Yukardaki kodu incelediğimizde sayfamız bir bölüm (<section>) ve altbilgiden (<footer>) oluşuyor. Bölüm içerisinde bir başlık etiketi (<header>), iki makale içeriği etiketi (<article>) var. Altbilgi içerisinde ise yine bir başlık ve gezinim etiketi (<nav>) var.

2) Şimdi bölüm içeriğini kodluyoruz.

```
5 <section> <!-- HTML5 elementlerini tanıtmak için bir bölüm açıyoruz. -->
6
7 <header> <!-- Bölümümüzün başlık etiketini açıyoruz. -->
8   <figure> <!-- Bölümümüzün başlığı içerisinde bir resim elementi görünsün istiyoruz. -->
9     
11     <figcaption>Fig.1 - HTML5</figcaption> <!-- Resmin açıklaması -->
12   </figure> <!-- Resim etiketini kapatıyoruz. -->
13 </header> <!-- Bölümümüzün başlık etiketini kapatıyoruz. -->
14
15 <article> <!-- Bölüm içerisindeki birinci makale içeriği -->
16   <header> <!-- Birinci makale başlığı -->
17     <h1>Anlamsal Elementler</h1> <!-- h1,h2,h3 Metnin büyüklüğünü değiştirir. -->
18   </header> <!-- Birinci makale başlığını kapatıyoruz. -->
19   <p>form, table, article, audio, video vb.</p> <!-- p bir paragrafa başlarken kullanılır. -->
20 </article> <!-- Birinci makale içeriğini kapatıyoruz -->
21
22 <article> <!-- Bölüm içerisindeki ikinci makale içeriği -->
23   <header> <!-- İkinci makale başlığı -->
24     <h2>Anlamsal Olmayan Elementler</h2> <!-- h1,h2,h3 Metnin büyüklüğünü değiştirir. -->
25   </header> <!-- İkinci makale başlığını kapatıyoruz. -->
26   <p>div, p, span vb. </p> <!-- p bir paragrafa başlarken kullanılır. -->
27 </article> <!-- Birinci makale içeriğini kapatıyoruz -->
28
29 </section> <!-- HTML5 elementlerini tanıttığımız bölümü kapatıyoruz.-->
```

➔ Bölümün başlık etiketi içerisine bir resim eklemek için ilk olarak <figure> etiketini açtık. Daha sonra, öğrenmiş olduğumuz etiketini kullanarak bir resim ekledik. Resmin başlığı ya da açıklaması için <figcaption> etiketini kullandık. Son olarak bölümün başlık etiketini kapattık.

➔ Bölüm içerisine bir içerik, açıklama metni, bilgi eklemek istediğimizde <article> etiketini kullandık. Daha sonra, yine <header> etiketi içerisine <h1> etiketini kullanarak içeriğin ya da makalenin başlığını ekledik. Son olarak makale metnini <p> etiketi içerisine ekledik ve <article> etiketini kapattık.

Kodlayıcı Görevi

- <h1> <h2> <h3> <p> etiketlerinin ne işe yaradıklarını test ediniz.
- Kodunuzu bu etiketler olmadan tekrar çalıştırınız. Ne tür farklılıklar olduğunu gözlemleyiniz.

3) Şimdi de sayfanın alt bilgisini kodluyoruz.

```
31 <footer> <!-- Sayfanın altında görünecek altbilgi -->
32   <header>
33     <h2>Footer Bağlantıları</h2>
34   </header>
35   <nav> <!-- Sayfanın altında görünecek bağlantılar (linkler) -->
36     <a href="/html/">HTML</a> |
37     <a href="/css/">CSS</a> |
38     <a href="/js/">JavaScript</a> |
39     <a href="/jquery/">jQuery</a>
40   </nav> <!-- Gezinim etiketini kapatıyoruz. -->
41 </footer> <!-- Altbilgiyi kapatıyoruz-->
```

Altbilgi (<footer>)etiketi içerisinde başlık ve gezinim bağlantıları yer alıyor. Gezinim bağlantılarını <nav> etiketi içerisine yazdık. Bağlantıları daha önce öğrenmiş olduğumuz <a> etiketi ile verdik. Şimdi sıra sizde kodlamanız sonucunda aşağıdaki gibi bir çıktı elde etmiş olmalısınız.



2. UYGULAMA

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <!-- <mark> ile etiketlenen yazı sarı ile vurgulanır. -->
6 <p>Bu metinde bir vurgulanmış <mark>yazı</mark> var.</p>
7
8 <br/>
9
10 <!-- <details> ile açılır kapanır içerikler oluşturabiliriz. -->
11 <details>
12 <summary> <!-- Üzerine tıklandığında açılır metin -->
13   Denizer YILDIRIM
14 </summary>
15 <p>Ankara Üniversitesi, Uzaktan Eğitim Merkezi</p>
16 <p>Öğretim Tasarımı & Web Programlama</p>
17 </details>
18
19 <br/><br/>
20
21 <!-- Herhangi bir görevin ilerlemesini gösteren etiket
22   value: görevin hangi aşamada olduğunu belirten değer
23   max: görevin tamamlanma aşamasını gösteren değer
24 -->
25 <progress value="22" max="100"></progress>
26
27 <br/><br/>
28
29 <!--Kullanıcıya bildirim vermek için kullanılan etiket -->
30 <dialog open> Dialog etiketi ile kullanıcıya uyarı verebiliyoruz. </dialog>
31
32 </body>
33 </html>
```

Kodlayıcı Görevi

- <dialog open> etiketi içerisindeki open kodunun ne işe yaradığını keşfediniz.

ÇOKLU ORTAM İÇERİK EKLEME

Sitenize HTML5 ile eklemiş olduğunuz mp4 video ve mp3 ses biçimleri şu anki mevcut tarayıcılar tarafından çalıştırabilir. Kodlamaya geçmeden önce İnternet'ten sitenize eklemek istediğiniz bir video ve sesi kendi isminizle oluşturmuş olduğunuz dizine indiriniz.

1. UYGULAMA

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <header> <p>Video Ekleme</p> </header>
6 <video id="video1" width="420" controls>
7 <!-- Video etiketini açıyoruz -->
8 <source src="mov_bbb.mp4" type="video/mp4">
9 <!-- Video kaynağını ve türünü belirtiyoruz. Siz indirmiş olduğunuz videonun ismini yazınız.-->
10 Tarayıcınız mp4 videoyu desteklemiyor.
11 <!-- Eğer tarayıcı eski ise videonun (mp4) çalıştırılmayacağı uyarısı veriyoruz. -->
12 </video>
13 <!-- Video etiketini kapatıyoruz -->
14
15 <br/><br/>
16 <!-- Video ve ses arasına iki satır boşluk bırakıyoruz. -->
17
18 <header> <p>Ses Ekleme</p> </header>
19 <audio controls>
20 <!-- Ses etiketini açıyoruz -->
21 <source src="horse.mp3" type="audio/mpeg">
22 <!-- Ses kaynağını ve türünü belirtiyoruz. Siz indirmiş olduğunuz ses dosyasının ismini yazınız.-->
23 Tarayıcınız mp3 sesi desteklemiyor.
24 <!-- Eğer tarayıcı eski ise sesin (mp3) çalıştırılmayacağı uyarısı veriyoruz. -->
25 </audio>
26 <!-- Ses etiketini kapatıyoruz. -->
```

Kodlayıcı Görevi

- Video ya da ses etiketleri içerisinde çeşitli kodlar yazılabilir. Bu kodların işlevleri aşağıdaki tabloda verilmiştir. Bu kodları test edelim.

Etiket	Özellik	Değer	Açıklama
Video, Audio	autoplay	autoplay	Sayfa yüklendiğinde oynatılmaya başlayacağını belirtir.
Video, Audio	controls	controls	Kontrol butonlarının görüntüleneceğini belirtir.
Video	height	pixels	Yüksekliği ayarlar.
Video, Audio	loop	loop	Her defasında tekrar başlayacağını belirtir.
Video, Audio	muted	muted	Videonun sesinin kapatılmasını belirtir.
Video	poster	URL	Kullanıcı oynat düğmesine basana kadar gösterilecek bir resmi belirtir.
Video, Audio	src	URL	Video dosyasının URL'sini belirtir.
Video	width	pixels	Video oynatıcısının genişliğini ayarlar.

<source> kodu içerisinde video ya da sesin nerede olduğunu belirtirken dikkat edilmesi gereken önemli bir nokta var. Aşağıdaki kod, kaynağın olduğu yeri belirtir.

```
src="mov_bbb.mp4"
```

Bu durumda kodlamanın yapıldığı dosyanın kayıt edildiği dizin aynı olmalıdır. Dizinler farklı ise aşağıdaki şekilde kodlama yapılmalıdır.

```
src="dizinismi/mov_bbb.mp4"
```

Peki, youtube gibi bir sosyal ağ sitesinden sitemize video eklemek istendiğinde ne yapılabilir? İşte bu durumda HTML5'teki <video> etiketi yerine başka kodlara ihtiyaç duyulur. Çünkü sosyal video paylaşım sitelerinin çalışma mantığı mp4 video biçimlerinden farklıdır. Şimdi ilk olarak herhangi bir sosyal video paylaşım sitesinden ilgi çekici bir video bulunuz ve daha sonra videonun altındaki paylaş (share) butonuna basarak karşınıza çıkan linki kopyalayınız. Bu link videonun kaynağı olacaktır. Şimdi üç farklı yöntemle bu videoyu site sayfasına eklenecektir.

2. UYGULAMA

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <!--Yöntem 1-->
6 <iframe width="420" height="345" src="https://www.youtube.com/embed/K6S40-VtZBI">
7 </iframe>
8
9 <!--Yöntem 2-->
10 <object width="420" height="315" data="https://www.youtube.com/embed/K6S40-VtZBI">
11 </object>
12
13 <!--Yöntem 3-->
14 <embed width="420" height="315" src="https://www.youtube.com/embed/K6S40-VtZBI">
15
16
17 </body>
18 </html>
```

Bu uygulama ile youtube'daki bir videoyu üç farklı yöntemle sayfaya eklenmiştir. Peki HTML5 ile <video> içerisinde controls, loop kodları eklenebiliyordu. Youtube'dan video eklenen üç yöntemde bunu nasıl yapılabilir?

```
src="https://www.youtube.com/embed/K6S40-VtZBI?autoplay=0&loop=1&controls=1">
```

Autoplay=0 : Sayfa yüklendiğinde videoyu otomatik başlatma.

Loop=1 : Video bittiğinde tekrar başlat.

Controls=1 : Video kontrol butonlarını görünür yap.

PROJE ŞABLONUNUN OLUŞTURULMASI

Öğrenilenleri tekrarlamak ve yeni kodlar öğrenmek amacıyla örnek bir site şablonu üzerinde çalışılacaktır. Şablonun ana sayfası Şekil 5'teki gibidir.



Şekil 5: Örnek bir anasayfa görünümü

İlk olarak şimdiye kadar öğrenilenlerden hareketle, yazılabilecek kodlar, Resim 5'teki gibi bir görüntü elde etmek için yeterli olmayacaktır. Bunun için HTML5 kodları ile sitenin ana çerçeveleri oluşturulmaya çalışılacaktır. Daha sonraki ünite de ise CSS kodları kavrandığında site resimdeki gibi bir görüntüye kavuşturulabilecektir.

Site Başlığı Ve Dil Kodlaması

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Web Programlama Temelleri</title>
5     <!-- Tarayıcı sekmesinde web sayfasının isminin görüntülenmesini sağlar-->
6     <meta charset="UTF-8">
7     <!-- Web sayfasının içerisindeki Türkçe karakterlerin görüntülenmesini sağlar -->
8   </head>
9
10  <body>
11    <!-- banner -->
12    <div>
13      <!-- İçerik -->
14    </div>
15    <div>
16      <!-- Footer -->
17    </div>
18  </body>
19 </html>
```

Yeni bir kodla karşılaştık. <head> etiketi arasına yazılan kodlar, sayfanın genel ayarlarının yapıldığı bölümdür. Bu etiket içerisinde,

- <title> ile tarayıcı sekmesinde sayfanın başlığını görüntüleyebiliriz.
- <meta> ile hangi dil içeriğinin kullanılacağını belirtebiliriz (UTF-8 Türkçe karakterlerin de desteklenmesini sağlar).
- <style> ve <script> ile farklı dosyaları sayfaya bağlayabiliriz. Bu konuya ilerleyen ünitelerde değinilecektir.

Site Banner'ı ve Gezinim Linklerinin Oluşturulması

```

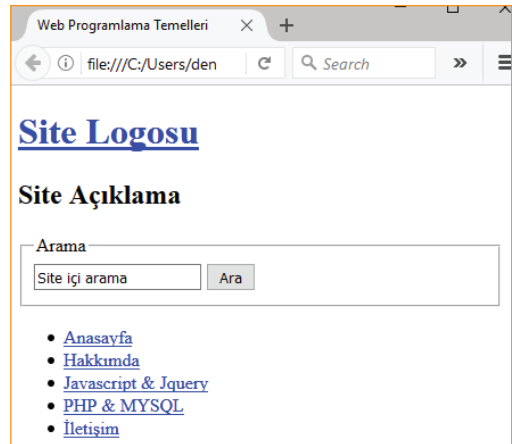
11 <!-- banner -->
12 <div>
13 <header>
14
15 <div id="hgroup">
16 <h1><a href="#">Site Logosu</a></h1>
17 <h2>Site Açıklama</h2>
18 </div>
19
20 <!-- Site içi arama için bir metin girişi-->
21 <form action="#" method="post">
22 <fieldset>
23 <legend>Arama</legend>
24 <input type="text" value="Site içi arama">
25 <input type="submit" id="sf_submit" value="Ara">
26 </fieldset>
27 </form>
28
29 <nav>
30 <ul>
31 <li><a href="#">Anasayfa</a></li>
32 <!-- Şuan tasarlamaya çalıştığımız sitenin anasayfa bağlantısı -->
33 <li><a href="#">Hakkımızda</a></li>
34 <!-- Siteyi kodlayanla ilgili kişisel bilgilerin yer alacağı hakkımızda sayfası -->
35 <li><a href="#">Javascript & Jquery</a></li>
36 <!-- İlerleyen bölümlerde Javascript ya da Jquery ile yapacağımız kodlamaların yer alacağı site sayfası -->
37 <li><a href="#">PHP & MYSQL</a></li>
38 <!-- İlerleyen bölümlerde PHP & MYSQL ile yapacağımız kodlamaların yer alacağı site sayfası -->
39 <li><a href="#">İletişim</a></li>
40 <!-- İletişim sayfası -->
41 </ul>
42 </nav>
43
44 </header>
45 </div>

```

Daha önceden div elementinin anlamsal olmayan yani içeriği her türden element alabilen bir element olduğu bilinmektedir. Burada **id = "hgroup"** olan bir div elementi oluşturulmuştur. Bu bölüm ilerleyen aşamalarda sitenin logo ve açıklamasının bulunduğu yer olacak.

Yeni Kodlar

- **Form etiketi:** Kullanıcıdan veri girişi almamızı sağlayan, tüm elementleri bir arada toplayan çerçevedir.
- **Fieldset:** Birbiri ile ilişkili elementleri bir arada toplayan bir çerçevedir.
- **ul ve li:** Madde işaretli metinler oluşturmamızı sağlayan elementlerdir. Birlikte kullanılan; madde işareti konulacak metin gruplarını, ise her bir maddeyi içerir.



Kodlayıcı Görevi

- Kullanıcının veri girişi yapmasına olanak sağlayan kaç tür <input> nesnesi vardır?
- İpucu: <input type="?">
- Madde imlerinin görünümünü nasıl değiştirebiliriz?
- İpucu: <ul type="?">

Sitenin Slayt Resmi ve Ana İçerik Bölümünün Oluşturulması

a) Sitenin Slayt Resminin Oluşturulması

```
47 <!-- İçerik -->
48 <div>
49 <!-- Sitenin Slayt Gösterisi ve Ana İçeriğini id' si container olan bir div elementi içerisinde topluyoruz. -->
50 <div id="container">
51
52 <!-- Sitenin slayt gösterisini id' si slider olan div elementi içerisinde topluyoruz.-->
53 <section id="slider">
63
64 <!-- Sitenin Ana İçeriğini id' si intro olan bir div elementi içerisinde topluyoruz. -->
65 <div id="intro">
90
91 </div>
92 </div>
```

```
<!-- Sitenin slayt gösterisini id' si slider olan div elementi içerisinde topluyoruz.-->
<section id="slider">
<figure>
<figcaption>
<h2>Slogan</h2>
<p> Bilgisayarlar öğrenebilen varlıklardır. Fakat öğretirken algoritmalara gereksinim duyar. </p>
<p> Şimdi söz geleceğin kodlayıcılarında: "Algoritma bizim işimiz :)"</p>
<footer><a href="#">Devamını Oku &raquo;</a></footer>
</figcaption>
</figure>
</section>
```



Slogan

Bilgisayarlar öğrenebilen varlıklardır. Fakat öğretirken algoritmalara gereksinim duyar.

Şimdi söz geleceğin kodlayıcılarında: "Algoritma bizim işimiz :)"

[Devamını Oku »](#)

b) Sitenin Ana İçerik Bölümünün Oluşturulması

```

47 <!-- İçerik -->
48 <div>
49 <!-- Sitenin Slayt Gösterisi ve Ana İçeriğini id' si container olan bir div elementi içerisinde topluyoruz. -->
50 <div id="container">
51
52 <!-- Sitenin slayt gösterisini id' si slider olan div elementi içerisinde topluyoruz.-->
53 <section id="slider">
63
64 <!-- Sitenin Ana İçeriğini id' si intro olan bir div elementi içerisinde topluyoruz. -->
65 <div id="intro">
90
91 </div>
92 </div>

```

```

<!-- Sitenin Ana İçeriğini id' si intro olan bir div elementi içerisinde topluyoruz. -->
<div id="intro">
<section >
<!-- article 1 -->
<article>
<h2> Biz Kodlayıcılar, </h2>
<p> Web programlama bölümünü bitirdiğimizde örnek bir site tasarlamış olacağız. <br/>
Siteyi tasarlarken hem kodların ne işe yaradığını öğrenecek hem de kodları örnek tasarımımız için uygulayacağız. <br/>
Programlama yapmaktan keyif aldıkça kendi kendinize daha ilgi çekici, beğeni oranı yüksek siteler tasarlayabileceksiniz. </p>
</article>
<!-- article 2 -->
<article>
<figure>
<ul>
<li><a href="a1.html"></a></li>
<!-- Madde işaretli, a1.html sayfasına bağlantılı genişliği:130px, yüksekliği:130px bir resim ekliyoruz. -->
<li><a href="a2.html"></a></li>
<!-- Madde işaretli, a2.html sayfasına bağlantılı genişliği:130px, yüksekliği:130px bir resim ekliyoruz. -->
<li><a href="a3.html"></a></li>
<!-- Madde işaretli, a3.html sayfasına bağlantılı genişliği:130px, yüksekliği:130px bir resim ekliyoruz. -->
</ul>
<figcaption><a href="#">Resim Galerisi <img alt="Resim Galerisi" /></a></figcaption> <!-- Buraya dikkat yeni bir kod! -->
</figure>
</article>
</section>
</div>

```

Biz Kodlayıcılar,

Web programlama bölümünü bitirdiğimizde örnek bir site tasarlamış olacağız.

Siteyi tasarlarken hem kodların ne işe yaradığını öğrenecek hem de kodları örnek tasarımımız için uygulayacağız.

Programlama yapmaktan keyif aldıkça kendi kendinize daha ilgi çekici, beğeni oranı yüksek siteler tasarlayabileceksiniz.



[Resim Galerisi »](#)

Yeni Kod!

Resim Galerisi » şeklinde bir görüntü nasıl elde edilir? Bunun için kodlara dikkat etmek gerekir.

```
<figcaption><a href="#">Resim Galerisi &raquo;</a></figcaption>
```

Görüldüğü üzere » simgesi » kodu ile elde edilir. Bu türden kodlar varlık isimleri olarak adlandırılır. Benzer şekilde birçok simgenin varlık ismi vardır. Yandaki QR kodlu bağlantıda² simgelerin karşılıklarına gelen kodlar bulunabilir.



Sitenin Alt Bilgi Bölümünün Oluşturulması

```
<!-- Footer -->
<div>
  <footer id="footer">
    <section>
      <h2>Hızlı Bağlantılar</h2>
      <nav>
        <ul>
          <li><a href="index.html">Anasayfa</a></li>
          <li><a href="hakkimda.html">Hakkımda</a></li>
          <li><a href="jskodlar.html">Javascript & JQuery</a></li>
          <li><a href="phpmysqlkodlar.php">PHP & MYSQL</a></li>
          <li><a href="iletisim.php">İletişim</a></li>
        </ul>
      </nav>
    </section>
    <p>Copyright &copy; 2017 - <a href="hakkimda.html">Siteyi Kodlayan kişi İsmi</a></p>
  </footer>
</div>
```

Hızlı Bağlantılar

- [Anasayfa](#)
- [Hakkımda](#)
- [Javascript & JQuery](#)
- [PHP & MYSQL](#)
- [İletişim](#)

Copyright © 2017 - [Siteyi Kodlayan kişi İsmi](#)

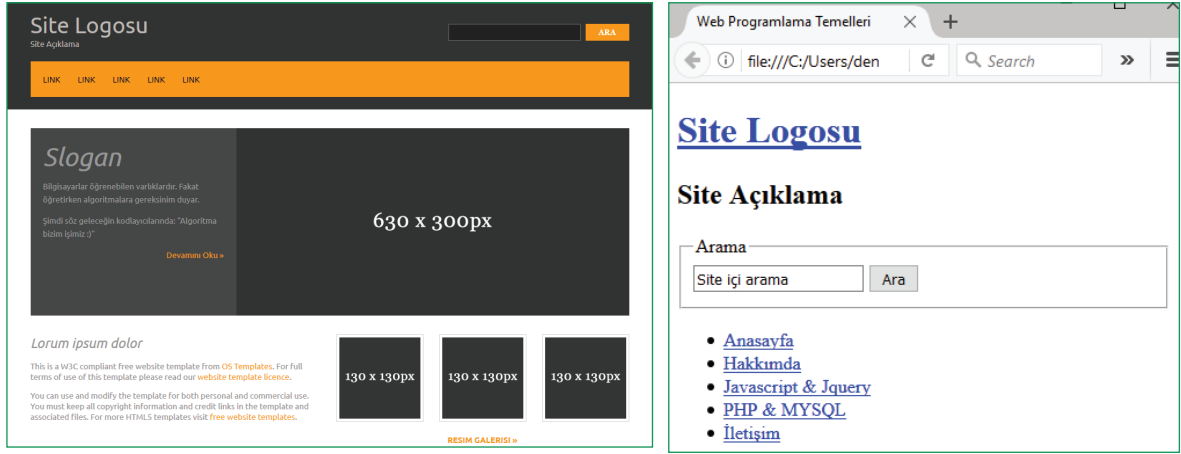
Bu ünite sonunda HTML5 ile site şablonunun ana çerçeveleri oluşturuldu. Böylece üniteadaki kazanımlara yönelik kodlama örneklerini uygulama fırsatı oldu. Şimdi yapılan kodlamaları, index.html şeklinde kendi isminizle oluşturduğunuz dizinin (klasörün) içerisine kaydediniz.

² https://www.w3schools.com/charsets/ref_html_entities_4.asp

3. STİL SAYFALARINA GİRİŞ

3. STİL SAYFALARINA GİRİŞ

Proje olarak geliştirilen web sitemizde şu ana kadar sadece HTML kodları kullanıldı. Fakat tasarlamaya çalışılan web sayfası ile şuana kadar yapılanlar karşılaştırıldığında yapılacak çok işlemin olduğu görülür.



Bu bölümde site görsel olarak yukarıdaki resme benzetilmeye çalışılacaktır. Bunun için kullanılacak kodlama Cascading Style Sheets (CSS)' dir. CSS, bir HTML belgesinin stilini yani HTML öğelerinin nasıl görüntüleneceğini açıklar. Şu an en yeni standardı CSS3'tür. Her yeni standart kendinden önceki tüm standartları destekler. Şimdi şekildeki gibi bir görüntü elde etmek için ilk CSS kodlaması yapılacaktır. CSS kodlamanın üç yöntemi vardır.

CSS'İN YAZILACAĞI YERE GÖRE KODLAMA YÖNTEMLERİ

1. HTML Etiketleri İçinde CSS Kodlama

```
<!DOCTYPE html>
<html>
<body style="background-color: grey;">
<!-- Sayfanın arkaplan rengini gri yapar -->

<h1 style="color: white; text-align: center;">İlk CSS Örneğimiz</h1>
<!-- h1 ile etiketlenmiş başlığımızın yazı rengini beyaz ve ortalanmış yapar. -->

<p style="font-family: verdana; font-size: 20px; text-align: justify;">Bu bir paragraf yazısıdır.
Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır.
Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır.
Bu bir paragraf yazısıdır.</p>
<!-- Paragrafı, yazı tipi verdana, yazı büyüklüğü 20px ve iki yana yaslı olacak şekilde biçimlendirir. -->
</body>
</html>
```

Dikkat!

- Bu yöntemle yapılan biçimlendirme sadece kodu yazılan elemente uygulanır. Şimdi sizde </body> kapatma etiketinden önce bir p etiketi ekleyiniz ve içeriğini doldurunuz. Sonra test ediniz.

İlk CSS Örneğimiz

Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır.

2. Bir Stil Bloğu (<style></style>) İçerisinde CSS Kodlama

```
<!DOCTYPE html>
<html>

<head>
<style>    /*Head bölümü içerisinde style etiketi açıyoruz.*/

body {      /*Hangi elementi biçimlendirmek istiyorsak elementten sonra bir parantez açıyoruz. */
  background-color: grey;    /* body elementinin arka plan rengini gri yapıyoruz. */
}
/* Elementin parantezini kapatıyoruz. */

h1 {
  color: white;    /* h1 elementinin yazı rengini beyaz yapıyoruz. */
  text-align: center; /* h1 elementi içerisindeki yazıları ortalanmış yapıyoruz. */
}

p {
  font-family: verdana; /* p elementinin yazı tipini verdana yapıyoruz. */
  font-size: 20px;    /* p elementinin yazı büyüklüğünü 20px yapıyoruz. */
}
</style>
</head>
<body>

<h1>İlk CSS Örneğimiz</h1>
<p>Bu bir paragraf yazısıdır.
Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır.
Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır.
Bu bir paragraf yazısıdır.</p>

</body>
</html>
```

Dikkat!

- Bu yöntemle yapılan biçimlendirme, sayfadaki kodu yazılan tüm elementlere uygulanır. Örneğin h1 etiketini ele alalım. Bu yöntemi kullandığınızda sayfadaki tüm h1 elementleri aynı biçimde olacaktır. Şimdi siz de </body> kapatma etiketinden önce bir h1 etiketi ekleyiniz ve içeriğini doldurunuz. Sonra test ediniz.

3. Harici Bir Stil Dosyası İçinde CSS Kodlama

Bu yöntemde sadece stil kodları *.css (ör: temel.css) şeklinde kaydedilir. Daha sonra web sayfasının head bölümü içerisine css uzantılı kaydedilmiş dosya, bağlantı verilir.

```
<!DOCTYPE html>
<html>

<head>
<link rel="stylesheet" href="styles/temel.css" type="text/css">
</head>
<body>

<h1>İlk CSS Örneğimiz</h1>
<p>Bu bir paragraf yazısıdır.
Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır.
Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır.
Bu bir paragraf yazısıdır.</p>

</body>
</html>
```

Dikkat!

- Bu yöntemle yapmış olduğunuz biçimlendirme 2. yöntemle aynı işlemi yapar. Yani biçim, kodu yazdığınız tüm elementlere uygulanır.
- `<link rel="stylesheet" href="styles/temel.css" type="text/css">` kodundaki href içeriğinde temel.css dosyasından önce styles dizini olduğunuz fark ettiniz mi?

ELEMENTİN ETİKETİNE GÖRE CSS KODLAMA

Sayfamızda biçimlendirme yaparken elementin etiketine (ör: div, body, input, article, header, footer) göre kodlama yapılır. Bu kodlama biçimini CSS kodlamanın 2. Yönteminde gördük.

```
body { /*Hangi elementi biçimlendirmek istiyorsak elementten sonra bir parantez açıyoruz. */
background-color: grey; /* body elementinin arka plan rengini gri yapıyoruz. */
}
/* Elementin parantezini kapatıyoruz. */

h1 {
color: white; /* h1 elementinin yazı rengini beyaz yapıyoruz. */
text-align: center; /* h1 elementi içerisindeki yazıları ortalanmış yapıyoruz. */
}

p {
font-family: verdana; /* p elementinin yazı tipini verdana yapıyoruz. */
font-size: 20px; /* p elementinin yazı büyüklüğünü 20px yapıyoruz. */
}
```

Bu kodlama yöntemi tek bir css kodu ile sayfanızdaki benzer etiketlerin aynı şekilde görüntülenmesine olanak sağlar. Yukarıdaki örneğimizi ele alacak olursak, sayfamızdaki h1 etiketli tüm elementler için benzer biçimlendirme uygulanmıştır. Bazı h1 etiketli elementlerin farklı biçimlendirmeye sahip

olması isteniyorsa ilk olarak “HTML etiketleri içinde CSS kodlama” yöntemi kullanılır. Fakat bu yöntem çok kullanışlı değildir.

```
<h1 style="color:green; background-color:grey;">Farklı Biçimlendirmeye sahip olmasını istediğimiz h1 etiketi</h1>
```

Kullanılacak diğer bir yöntem ise, “Elementin Özelliklerine Göre CSS Kodlama” dır.

ELEMENTİN ÖZELLİKLERİNE GÖRE CSS KODLAMA

HTML elementlerinin özelliklerine göre css kodlama yöntemini açıklamadan önce, “Elementin Özellikleri” bilinmelidir. Aşağıdaki HTML kodlarını incelendiğinde etiket isimleri dışında kırmızı etikete özgü özellikler (kırmızı renkli) olduğu görülecektir. Bu özelliklerden id ve class tüm etiketlerdeki ortak özelliktir.

```
<h1 id="" class="" style="" > ... </h1>
<img id="" class="" alt="" src="" title="" />
<a id="" class="" href="" target="" > ... </a>
```

Elementlerin sahip olduğu özelliklere göre css kodlamada genellikle id ve class özellikleri kullanılır. Örneğin class özelliği “renkli” ya da id özelliği “renkli” olan herhangi bir elementin biçimi değiştirilebilir. Bunun için id’ye göre kodlama yaparken “#” simgesi, class’ a göre kodlama yaparken “.” simgesi ile başlanır.

```
.renkli { color:orange; }
#renkli { color:green; }
```

Herhangi bir kodlama yönteminin kullanılabilmesi için önce, biçimlendirme yapmak istenilen etikete id ya da class özellikleri girilir. Fakat bazen buna gereksinim olmadan da kodlama yapılabilir. Örneğin <a> etiketi ile bir bağlantı oluşturulduğunda:

```
<a href="http://okulunuz.com" target="_blank" > ... </a>
```

Bu etiketi href özelliği http://okulunuz.com olacak şekilde bir seçim yaparak biçimlendirilebilir.

```
a[href="http://okulunuz.com"]
{
    ...
}
```

```

<!DOCTYPE html>
<html>

<head>
<style>
/* Bir önceki örnekte kodları altalta yazmıştık. Aşağıda görüldüğü gibi kodları yanyana da yazabiliriz*/
body { background-color: grey;}
p { font-family: verdana;font-size: 20px;color:white; }
.renkli { color:orange; }
#renkli { color:green; }

</style>
</head>
<body>

<h1>İkinci CSS Örneğimiz</h1>
<p>Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. </p>
<!-- Class özelliği renkli olan paragraf-->
<p class="renkli"> Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. </p>
<!-- id özelliği renkli olan paragraf-->
<p id="renkli">Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır.</p>
<!-- Herhangi bir özellik atanmamış standart paragraf-->
<p>Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. </p>
<!-- Class özelliği renkli olan paragraf-->
<p id="renkli"> Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. </p>
<!-- id özelliği renkli olan paragraf-->
<p class="renkli"> Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. </p>
</body>
</html>

```

İkinci CSS Örneğimiz

Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır.
Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır.
Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır.
Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır.
Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır.
Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır. Bu bir paragraf yazısıdır.

İç İç Geçmiş CSS Kodlama

Son olarak, iç içe geçmiş css kodlama örneklerini inceleyelim. Örnek:

- Sadece **header** elementi içerisindeki **class özelliği “renkli“** olan elementleri nasıl biçimlendirilir?
- Sadece **section** elementi içerisindeki **article** elementi içerisinde olan h1 elementini nasıl biçimlendirilir?
- Sadece **footer** elementi içerisindeki **google sitesine link verilmiş a** elementleri nasıl biçimlendirilir?

```

<!DOCTYPE html>
<html>

<head>
<style>
body {padding:25px; background-color: grey;font-family:arial;}

header .renkli {color:orange;}
/* header elementi içerisindeki class özelliği "renkli" olan elementlerin seçimi */

section article h1 {color:blue;}
/* section elementi içerisindeki article elementi içerisinde olan h1 elementinin seçimi */

footer [href*=google] {color:orange; text-decoration:none;}
/* footer elementi içerisindeki google sitesine link verilmiş a elementinin seçimi */
</style>
</head>
<body>

<h1 style="color:white;">Üçüncü CSS Örneğimiz</h1>
<p>-----</p>
<header>
  <h1>Parağraf Başlığı </h1>
  <p class="renkli"> Bu bir parağraf yazısıdır. Bu bir parağraf yazısıdır. </p>
</header>
<p>-----</p>
<section>
  <article>
    <h1>Parağraf Başlığı </h1>
    <p class="renkli"> Bu bir parağraf yazısıdır. Bu bir parağraf yazısıdır. </p>
  </article>
</section>
<p>-----</p>
<footer>
  <a href="http://ankara.edu.tr">Ankara üniversitesi</a>
  <a href="http://google.com.tr">Google Web Sitesi</a>
</footer>

<p>-----</p>
<p class="renkli"> Bu bir parağraf yazısıdır. Bu bir parağraf yazısıdır. </p>

</body>
</html>

```

Üçüncü CSS Örneğimiz

Parağraf Başlığı

Bu bir parağraf yazısıdır. Bu bir parağraf yazısıdır.

Parağraf Başlığı

Bu bir parağraf yazısıdır. Bu bir parağraf yazısıdır.

[Ankara Üniversitesi](#) [Google Web Sitesi](#)

Bu bir parağraf yazısıdır. Bu bir parağraf yazısıdır.

Dikkat!

- Class özelliği “renkli” olan iki p elementinin birbirinden farklı biçime sahip olduğunu farkettiler mi?
- İki h1 elementinin birbirinden farklı biçime sahip olduğunu farkettiler mi?
- Aynı footer elementi içerisinde olup farklı biçime sahip iki a elementini farkettiler mi?

PROJE SİTESİNİN CSS İLE GÖRSELLEŞTİRİLMESİ

Varsayılan Ayarları

Bir web sitesinin CSS ile görselleştirilirken ilk olarak yapılması gereken varsayılan olarak kullanılacak genel CSS ayarlarını oluşturmaktadır.

```
<style>
/*-----Genel Ayarlar-----*/
* {
  /* Web sayfasındaki tüm etiketlerin seçimi */
  margin:0; /* Elementlerin kendinden önce ve sonraki elementlerle olan mesafesi 0 pixel (px) olsun.*/
  padding:0; /* Elementlerin kendinden önce ve sonraki elementlerle olan mesafesi 0 px olsun.*/
  text-decoration:none; /* Elementlerin içerisinde kalan metinlerin altı çizili olmasın.*/
}
body{
  /*body elementinin seçimi*/
  font-size:13px; /* Elementin içerisindeki metinlerin yazı büyüklüğü 13 px olsun.*/
  font-family: Ubuntu; /* Elementin içerisindeki metinlerin yazı tipi Ubuntu olsun.*/
  color:#919191; /* Elementin içerisindeki metinler #919191 koduna karşılık gelen yazı renginde olsun. */
  background-color:#232323; /* Elementin arka plan rengi #232323 koduna karşılık gelen renkte olsun*/
}

header, container, footer { /* header, container, footer elementlerinin seçimi*/
  width:960px; /* Elementin Genişliği 960 px olsun.*/
  margin:auto; /* Elementin sağında ve solunda kalan boşluklar eşit oranda (ortalı) olsun. */
}

h1, h2, h3, h4, h5, h6{
  font-size:22px;
  font-weight:normal;
  font-style:italic;
  line-height:normal;
}

nav ul{
  list-style:none; /* madde işareti imleri (yuvarlak, kare, sayı) gösterilmesin. */
}

form, fieldset, legend{
  border:none;
}

input, textarea, select{
  font-size:12px;
  font-family:Georgia,"Times New Roman",Times,serif;
}

a {
  color:#ffffff; /* Linkler #ffffff koduna karşılık gelen yazı renginde olsun.*/
}
</style>
```

Site Şablonundaki Temel Çerçevelerin Biçimlendirilmesi

Şablonumuzda, body etiketi içerisinde **banner**, **content** ve **footer** olmak üzere üç ana çerçeve oluşturulmuştur. Şimdi bu bölümlerin her biri içerisindeki etiketler için şablona uygun şekilde biçimlendirme yapılacaktır.

Banner Çerçevesinin Biçimlendirilmesi

“**banner**” çerçevesinin html kodlarını açarak en üstteki div elementinin class özelliği “çerçeve1”, sitenin logosunun ve açıklamasının bulunacağı div etiketinin id özelliği “logogrup” olarak değiştirilir.

```

<!-- Banner -->
<div class="cercevel">
  <header>
    <div id="logogrup">
      <h1><a href="index.html">Site Logosu</a></h1>
      <h2>Site Açıklama</h2>
    </div>
    <form>
      <fieldset>
        <legend>Arama</legend>
        <input type="text" value="Site içi arama"
          onFocus="this.value=(this.value=='Site içi arama')? '' : this.value ;">
        <input type="submit" id="ara" value="Ara">
      </fieldset>
    </form>
    <nav>
      <ul>
        <li><a href="#">Anasayfa</a></li>
        <li><a href="#">Hakkımda</a></li>
        <li><a href="#">Javascript & JQuery</a></li>
        <li><a href="#">PHP & MYSQL</a></li>
        <li><a href="#">İletişim</a></li>
      </ul>
    </nav>
  </header>
</div>

```

Şimdi en dıştaki elementten başlayarak en içteki elemente doğru banner çerçevesinin CSS ayarları yapılabilir.

```

/*Çerçeve*/
.cercevel {
  width:100%;
  text-align:left;
  color:#C0BAB6;
  background-color:#333333;
}

/* Renk kodlarının hangi renge karşılık geldiğini
http://htmlcolorcodes.com/ adresinden bulabilirsiniz. */

/*---Header---*/
header{ padding:20px; width:960px; }

/*---Header içindeki logogrup idli element---*/
header #logogrup { float:left; margin:0 0 20px 0;} /* margin: üst sağ alt sol */
header #logogrup h1{ font-size:36px;}
header #logogrup h2{ font-size:13px;}

```

```

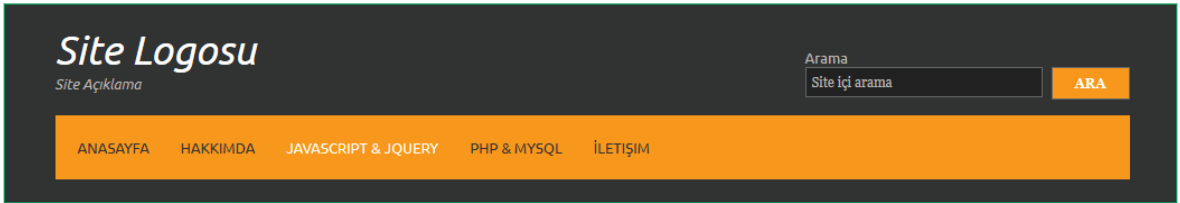
/*---Header içindeki form elementi---*/
header form{
  width:290px;
  float:right; /*sağa hizala*/
  margin-top:20px;
  padding:0;
}
header form input{
  float:left; /*sola hizala*/
  width:200px;
  margin:0;
  padding:5px;
  color:#C0BAB6;
  background-color:#232323;
  border:1px solid #666666;
}
header form #ara{
  float:right;
  width:70px;
  font-size:12px;
  font-weight:bold;
  text-transform:uppercase; /*yazıları büyük harfe dönüştürme*/
  color:#FFFFFF;
  background-color:#FF9900;
}

```

```

/*---Header içindeki nav elementi---*/
header nav {
  width:100%;
  margin:0;
  padding:20px 0;
  color:#C0BAB6;
  background-color:#FF9900;
  clear:both; /* Elementin solunda veya sağında herhangi bir başka element bulunamaz. */
}
header nav ul{padding:0 20px;}
header nav li{
  display:inline; /* Maddeleri alt alta değil de yanyana sıralar*/
  margin-right:25px;
  text-transform:uppercase;
}
header nav li a{color:#333333;} /*linklerin genel rengi*/
header nav li a:hover{color:#ffffff;} /*linklerin üzerine gelindiğinde rengi*/

```



Şekil 6: Header çerçevesinin görünümü

Content Çerçevesinin Biçimlendirilmesi

“content” çerçevesinin html kodlarını açarak etiketlere verilen id ve class özellikleri aşağıdaki gibi değiştirilir.

```

<!-- Content -->
<div class="cerceve2">
  <div id="container" class="clear">
    <!-- Slider -->
    <section id="slider" class="clear">
      <figure>
        <figcaption>
          <h2>Slogan</h2>
          <p> Bilgisayarlar öğrenebilen varlıklardır.
          Fakat öğrenirken algoritmalara gereksinim duyar. </p>
          <p> Şimdi söz geleceğın kodlayıcılarında:
          "Algoritma bizim işimiz :)"</p>
          <footer class="devam"><a href="#">Devamını Oku &raquo;</a></footer>
        </figcaption>
      </figure>
    </section>
    <!-- Acıklama -->
    <div id="intro">
      <section class="clear">
        <!-- article 1 -->
        <article class="acıklama">
          <h2>Ad Soyad</h2>
          <p>Açıklama açıklama açıklama açıklama açıklama açıklama
          açıklama açıklama açıklama açıklama açıklama açıklama
          açıklama açıklama açıklama açıklama açıklama... </a>.</p>
          <br/>
          <p>Açıklama açıklama açıklama açıklama açıklama açıklama
          açıklama açıklama açıklama açıklama açıklama açıklama
          açıklama açıklama açıklama açıklama açıklama... </a>.</p>
        </article>
        <!-- article 2 -->
        <article class="acıklama ek">
          <figure>
            <ul class="clear">
              <li>
                <a href="#">
                  
                </a>
              </li>
              <li>
                <a href="#">
                  
                </a>
              </li>
              <li class="last">
                <a href="#">
                  
                </a>
              </li>
            </ul>
            <figcaption><a href="#">Resim Galerisi &raquo;</a></figcaption>
          </figure>
        </article>
      </section>
    </div>
  </div>
</div>

```

Şimdi en dıştaki elementten başlayarak en içteki elemente doğru content çerçevesinin CSS ayarları yapılır.

```

<style>
/* -----Genel----- */
.cerceve2 { width:100%; margin:0; padding:0; text-align:left; color:#979797; background-color:#FFFFFF; }
.clear:after{ /* class özelliği clear olan elementten sonra */
  content:"."; /* içeriği . olan bir element yaratalım ve */
  display:block; /* bu elementten sonraki elementin bir alt satırda görüntülenmesini sağlayalım. */
}

/* -----Content----- */
#container{width:960px;
  margin: 0 auto;
  /* üst dış boşluğu 0px sağında ve solunda kalan boşluklar eşit oranda (ortalı) olsun. */
  padding:30px 0;
  /* üst iç boşluğu 30px diğer iç boşlukları 0px olsun.*/
}
#container section{
  display:block;
  /* elementin genişliğini bir üstündeki elementin genişliğine (width=100%) eşitleyelim. */
  margin-bottom:30px;
  padding:0;
}
#container .last{margin:0;}
#container .more{text-align:right;}

/* -----Slider----- */
#container #slider{
  width:100%; ,
  /* elementin genişliğini bir üstündeki elementin genişliğine (width=100%) eşitleyelim. */
}

#container #slider figure img{float:right; width:630px; height:300px;}

#container #slider figure figcaption{
  float:left; width:290px; height:260px;
  padding:20px; color:#989898; background-color:#464646;
  line-height:20px; /* satır yüksekliğini belirtir. */
}

#container #slider figure figcaption a{color:#FF9900; background-color:#464646;}
#container #slider figure h2{font-size:42px;}

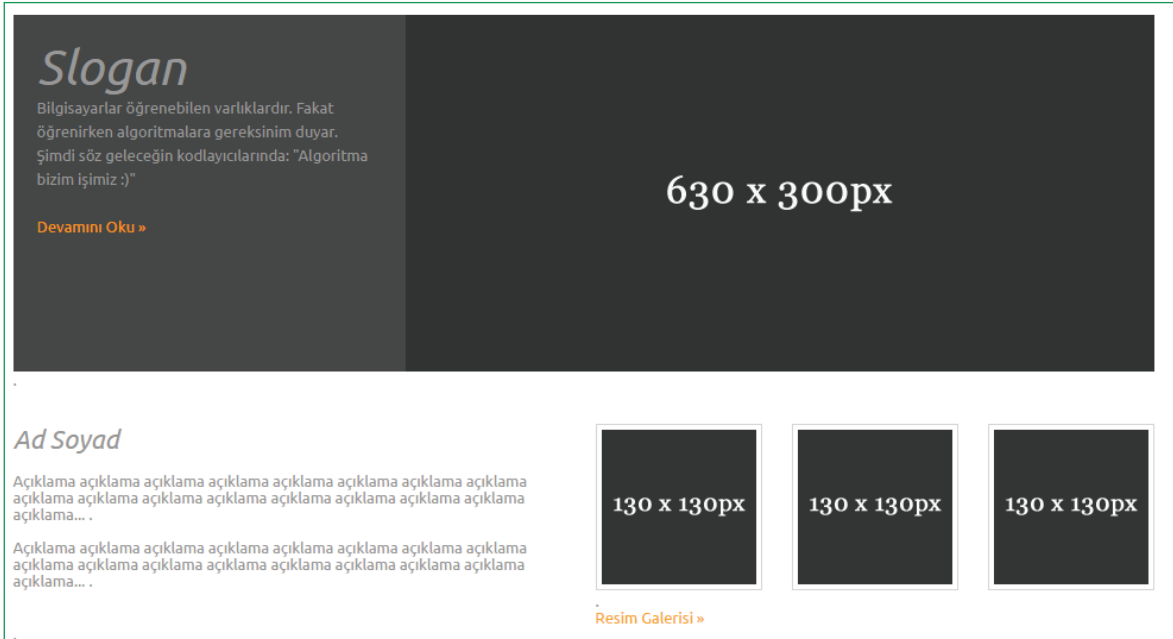
/* -----Intro----- */

#container #intro{margin-bottom:60px; padding-bottom:20px; }
#container #intro section .aciklama{width:470px; float:left; margin:0 20px 0 0;}
#container #intro section .ek {margin-right:0px; }
#container #intro section article h2 {margin-bottom:15px;}
#container #intro section article figure ul {list-style:none;}
#container #intro section article figure ul li {float:left; margin:0 25px 0 0;}
#container #intro section article figure ul li.last {margin-right:0;}
#container #intro section article figure ul li img { width:130px; height:130px; padding:4px;
  border-width:1px; /* resmin kenarları 1px genişliğinde olsun */
  border-style:solid; /* resmin kenarları solid (çizgi) stilinde olsun */
  border-color:#D6D6D6; /* resmin kenarları #D6D6D6 koduna karşılık gelen renkte olsun */
}

#container #intro section article figure figcaption a {color:orange;}

</style>

```

Şekil 7: Content çerçevesinin görünümü

Footer Çerçevesinin Biçimlendirilmesi

Son olarak “footer” çerçevesinin html kodları açılır ve en üstteki div elementinin class özelliği “cerceve3” değiştirilir. Diğer etiketler anlamsal elementler olduğu için özel bir id ya da class belirtmeye gerek yoktur.

```

<!-- Footer -->
<div class="cerceve3">
  <footer id="footer">
    <nav>
      <ul>
        <li><a href="#">Anasayfa</a></li>
        <li><a href="#">Hakkımda</a></li>
        <li><a href="#">Javascript & JQuery</a></li>
        <li><a href="#">PHP & MYSQL</a></li>
        <li><a href="#">İletişim</a></li>
      </ul>
    </nav>
    <p style="float:right">Copyright &copy; 2017 -
      <a href="#"> Siteyi Kodlayan kişi İsmi</a>
    </p>
  </footer>
</div>

```

Şimdi en dıştaki elementten başlayarak en içteki elemente doğru footer çerçevesinin CSS ayarları yapılır.

```
<style>
.cerçeve3 { width:100%; margin:0; padding:0; text-align:left; color:#919191; background-color:#232323;}
footer {padding:20px 0;}
footer nav {
  display:inline;
  /* block kendinden sonraki elementi satır başına atıyordu.
  inline-block ise, kendinden sonraki elementi yan yana koyar.*/
}
footer nav li {padding:4px; border-right:1px solid white;
display:inline;
/* menüü oluşturan madde imlerinin alt alta değil yanyana gelmesini sağlar.*/
}
footer nav li:last-child{ /* menünün en son elementini seçer */
border:none;
}
footer a{color:white;}
footer a:hover{color:orange;}
</style>
```

Site Logosu

Site Açıklama

Arama

ARA

[ANASAYFA](#) [HAKKIMDA](#) [JAVASCRIPT & JQUERY](#) [PHP & MYSQL](#) [İLETİŞİM](#)

Slogan

Bilgisayarlar öğrenebilen varlıklardır. Fakat öğrenirken algoritmalara gereksinim duyar. Şimdi söz geleceğin kodlayıcılarında: "Algoritma bizim işimiz :)"

[Devamını Oku »](#)

630 x 300px

Ad Soyad

Açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama...

Açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama açıklama...

130 x 130px

130 x 130px

130 x 130px

[Resim Galerisi »](#)

Anasayfa | Hakkımızda | Javascript & Jquery | PHP & MYSQL | İletişim

Copyright © 2017 - Siteyi Kodlayan kişi İsmi

Şekil 8: Sitenin genel görünümü

4. ETKİLEŞİM



4. ETKİLEŞİM

Şimdiye kadar HTML kodları ile oluşan bir site şablonu, CSS kodları ile görsel hâle geldi. Şimdi siteyi ziyaret eden kullanıcıların site içerisinde neler yapabileceği ile ilgili beyin fırtınası yapılacaktır. İçerik okunabilir, linkleri kullanarak sayfalar arası geçiş yapılabilir. Eklenmesi istenilen başka bir madde var mıdır?

Örneğin;

- Her hangi bir HTML elementi içerisinde yazılı olan bir metni değiştirebilir mi?
- Ya da bir div elementinin arka plan rengini değiştirebilir mi?

İşte, kullanıcıların bu türden işlemleri gerçekleştirebilmesi için HTML ve CSS kodları yeterli olmayabilir. Bunun için farklı kod yapılarına gereksinim duyulur. Bu noktada Javascript:

- HTML içeriğini değiştirebilir.
- HTML niteliklerini değiştirebilir.
- CSS stilini değiştirebilir.
- Bu işlemleri mantıksal sınımalara göre de yapabilir. Örneğin siteyi ziyaret eden kullanıcı harf yerine metin girişi yaptıysa, uyarı gösterebilir.

Özetle, HTML web sayfalarının içeriğini tanımlamak, CSS web sayfalarının biçimini belirlemek, Javascript ise web sayfalarının davranışını programlamak için kullanılır. Peki Javascript kodları nereye eklenir. Daha önce görüldüğü üzere, farklı CSS kodlama yöntemleri gibi Javascript kodlama yöntemleri de vardır.

Javascript Kodlama Yöntemleri

İşlevsel Olmayan Kullanımı

```
<!DOCTYPE html>
<html>
  <body>
    <h2>Hızlı ama işlevsel olmayan kullanım</h2>
    <p id="demo">Parağraf</p>
    <button
      onclick='document.getElementById("demo").innerHTML = "Butona tıklayınca parağraf değişti.";'>
      Değiştir
    </button>
  </body>
</html>
```

Hızlı ama işlevsel olmayan kullanım

Butona tıklayınca parağraf değişti.

Değiştir

Head Elementi Arasındaki Kullanımı

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function degistir() {
        document.getElementById("demo").innerHTML = "Butona tıklayınca parağraf deđiřti.";
      }
    </script>
  </head>
  <body>
    <h2>head elementi arasındaki kullanımı </h2>
    <p id="demo">Parađraf</p>
    <button type="button" onclick="degistir()">Deđiřtir</button>
  </body>
</html>
```

head elementi arasındaki kullanımı

Butona tıklayınca parađraf deđiřti.

Body Elementi Arasındaki Kullanımı

```
<!DOCTYPE html>
<html>
  <body>
    <script>
      function degistir() {
        document.getElementById("demo").innerHTML = "Butona tıklayınca parađraf deđiřti.";
      }
    </script>
    <h2>body elementi arasındaki kullanımı </h2>
    <p id="demo">Parađraf</p>
    <button type="button" onclick="degistir()">Deđiřtir</button>
  </body>
</html>
```

body elementi arasındaki kullanımı

Butona tıklayınca parađraf deđiřti.

Dış Bir Dosyaya Bağlantı Verilerek Kullanımı

```
<!DOCTYPE html>
<html>
  <body>
    <script src="myScript.js"></script>

    <!--Dosyanın içeriği:

      function degistir() {
        document.getElementById("demo").innerHTML = "Butona tıklayınca parağraf değişti.";
      }
    -->

    <!-- ya da
      <script src="https://www.w3schools.com/js/myScript.js"></script>
      <script src="js/myScript.js"></script>
      şeklinde de olabilirdi.
    -->

    <h2>Dış bir dosyaya bağlantı verilmesi </h2>
    <p id="demo">Parağraf</p>
    <button type="button" onclick="degistir()">Değiştir</button>
  </body>
</html>
```

Dış bir dosyaya bağlantı verilmesi

Butona tıklayınca parağraf değişti.

Değiştir

Şimdi bu yöntemlerden bazılarını kullanarak senaryo temelinde kodlama yapılacaktır.

HTML İÇERİĞİ DEĞİŞTİRME

Javascriptle HTML içeriğinin değiştirilebileceğinden bahsedilmiştir. Web sayfamızda bir buton aracılığıyla güncel tarih ve saatin gösterilmesi bu türden uygulamalara bir örnek olarak verilebilir.

```
<!DOCTYPE html>
<html>
<body style="font-family:Ubuntu;">

<h2>Güncel Tarih ve Saat Uygulaması</h2>

<button onclick="document.getElementById('demo').innerHTML = Date()">Göster</button>
<!--
  Kod Açıklaması
  onclick="document.getElementById('demo').innerHTML = Date()"
  onclick: kullanıcı butona tıkladığında
  document: sayfa içerisindeki
  getElementById('demo'): id'si demo olan elementin
  innerHTML: içeriğini
  Date(): güncel tarih ve saat ile değiştir.
-->

<p id="demo" style="background-color:grey;color:orange;"></p>
</body>
</html>
```

Aşağıdaki ekran görüntüsünde “Göster” butonuna her tıkladığında butonun altındaki saat de değişmektedir.



HTML NİTELİKLERİNİ DEĞİŞTİRME

Hatırlanıldığı üzere, Html sayfaları etiketlerden oluşmaktadır. Örneğin div, p, a, button, input, vb. sayfa ile kullanıcının etkileşimi sonucunda, herhangi bir etiketin özelliğini Javascript ile değiştirmek mümkündür. Örneğin kullanıcının herhangi bir butona tıklaması ile div olan bir etiketi butona ya da input olan bir etiketi butona dönüştürmek mümkündür.

```
<!DOCTYPE html>
<html>
<body style="font-family:'Verdana'; font-size:14px;">

<h2>Elementin Niteliğini Değiştirme</h2>

<input type="text" id="degistirilecekElement" value="Bu bir veri giriş elementi">

<button onclick="Degistir();">Değiştir</button>

<script>
  function Degistir() {

    document. // sayfa içerisindeki
    getElementsByTagName('INPUT')['degistirilecekElement'].
    // id'si "degistirilecekElement" olan input elementinin
    setAttribute('type', 'button');
    // type niteliğini button olarak değiştir.

    document. //sayfa içerisindeki
    getElementsByTagName('INPUT')['degistirilecekElement'].
    //id'si "degistirilecekElement" olan input elementinin
    setAttribute('value', 'Bu bir buton elementi');
    //value niteliğini "Bu bir buton elementi" olarak değiştir.
  }
</script>

</body>
</html>
```

Kullanıcı Butona Tıklamadan Önce

Elementin Niteliğini Değiştirme

Bu bir veri giriş elementi

Değiştir

Kullanıcı Butona Tıkladıktan Sonra

Elementin Niteliğini Değiştirme

Bu bir buton elementi

Değiştir

CSS DEĞİŞTİRME

HTML sayfaları biçimlendirilirken CSS kullanılmıştır. Bu biçimlendirme özelliklerini de kullanıcıların eylemlerine göre (Örneğin bir butona tıkladığında ya da bir veri girişi yaptığında) Javascriptle değiştirmek mümkündür.

```
<!DOCTYPE html>
<html>
<body style="font-family:Verdana; font-size:14px;">

<h2>Elementleri Gizleme ya da Görünür Yapma</h2>

<p id="p1" style="background-color:grey;color:orange;">
Bu bir parağraf. Bu bir parağraf. Bu bir parağraf.
</p>

<input type="button" value="Gizle"
onclick="document.getElementById('p1').style.display='none'">
<!--
  Elementin Gizlenmesi
  butona tıkladığında, id'si p1 olan elementin stilleri içerisinde
  display özelliğine none yap.
  Not: ...style.visibility='hidden' da kullanılabilir.
-->

<input type="button" value="Göster"
onclick="document.getElementById('p1').style.display='block'">
<!--
  Elementin Görünür Yapılması
  butona tıkladığında, id'si p1 olan elementin stilleri içerisinde
  display özelliğine block yap.
  Not: ...style.visibility='visible' da kullanılabilir.
-->
</body>
</html>
```

Elementleri Gizleme ya da Görünür Yapma

Gizle Göster

Elementleri Gizleme ya da Görünür Yapma

Bu bir parağraf. Bu bir parağraf. Bu bir parağraf.

Gizle Göster

Şimdiye kadar Javascript kullanarak basit etkileşimli kodlamalar yapıldı. Artık hedefi daha yükseklere taşımanın sırası gelmedi mi sizce? Hatırlarsanız, javascript' de mantıksal sınımalara göre işlem yapılabileceğinden bahsedilmişti. Örneğin

- Eğer ... olduysa, yap.
- Eğer ... ise ... değiştir.
- Eğer ... ise ... göster.

Bu şekilde sayısız senaryolar oluşturulabilir. Peki bu sınıama işlemini javascript ile nasıl yapabiliriz?

MANTIKSAL SINAMA

Bu konu başlığı altında javascript' de mantıksal sınıama yapılmasına olanak sağlayan kontrol yapıları görülecektir.

if-else kontrol yapısı

```
<script>
  if (durum1) {
    // durum1 doğru ise, yapılacak kodlama
  } else if (durum2) {
    // durum2 doğru ise, yapılacak kodlama
  } else {
    // durum1 ve durum2 dışındaki tüm diğer durumlarda yapılacak kodlama
  }
</script>
```

Şimdi, bir senaryo üzerinden if-else kontrol yapısı kullanılacaktır. Örneğin siteye giriş zamanına göre kullanıcıya “Günaydın”, “İyi günler” ya da “İyi akşamlar” mesajı veren bir kodlama yapılacaktır.

```
<!DOCTYPE html>
<html>
<body style="font-family:Verdana;font-size:14px;">
  <p style="font-weight:bold;">Saate Göre Mesaj Verme</p>

  <p id="demo"></p>

  <script>
    var mesaj, saat; // var kodu ile değişken tanımlıyoruz.

    saat = new Date().getHours();
    // Date()      : yeni bir date nesnesi yaratır
    // .getHours() : ve date nesnesinin sadece hours yani saatini alır.

    if (saat < 10) {           // eğer saat 10 dan küçükse
      mesaj = "Günaydın";     // mesaj değişkenini "Günaydın" olarak eşitle
    } else if (saat < 20) {   // eğer saat 20 den küçükse
      mesaj = "İyi Günler";   // mesaj değişkenini "İyi Günler" olarak eşitle
    } else {                  // eğer saat bu iki durumun dışında ise yani 20 ve 20 den büyükse
      mesaj = "İyi akşamlar"; // mesaj değişkenini "İyi Akşamlar" olarak eşitle
    }

    // Şimdi sıra bu mesaj değişkeninin ekranda gösterilmesine geldi.
    // Üç farklı yöntemle kullanıcıya mesaj verilebilir.
    document.getElementById('demo').innerHTML= mesaj;
    // Daha önce görmüş olduğumuz HTML içeriği değiştirme
    window.alert(mesaj);
    // Bir uyarı penceresi içerisinde mesajı verme
    document.write(mesaj);
    // Sayfaya javascriptle yazdırma
  </script>
</body>
</html>
```

SWİTCH KONTROL YAPISI

```
<script>
  switch(degisken) {
    case x:
      // deęişkenin x olma durumunda yapılacak kodlama
      break;
    case y:
      // deęişkenin y olma durumunda yapılacak kodlama
      break;
    default:
      // deęişkenin x ve y olmama durumunda yapılacak kodlama
  }
</script>
```

Şimdi bir senaryo üzerinden switch kontrol yapısı kullanılacaktır. Örneğin siteye hangi günde giriş yapıldığının mesajını veren bir kodlama yapılacaktır.

```
<html>
<body style="font-family:Verdana;font-size:14px;">

<p style="font-weight:bold;"> Hangi gün olduğunu bulan kontrol yapısı</p>
<p style="font-weight:bold;padding:30px;background-color:orange;" id="demo"></p>

<script>
  var gun, mesaj;           // var kodu ile gun ve mesaj deęişkenleri oluşturuldu.
  gun = new Date().getDay(); // getDay() ile o andaki tarihin gün numarasını gun deęişkenine eşitliyoruz.
  switch (gun) {           //

    case 0:                 // gün deęişkeni 0 ise;
      mesaj = "Pazar";     // mesaj deęişkenini "Pazar" a eşitle
      break;               // dięer durumları kontrol etmene gerek kalmadı.

    case 1:                 // gün deęişkeni 1 ise;
      mesaj = "Pazartesi"; // mesaj deęişkenini "Pazartesi" ye eşitle
      break;               // dięer durumları kontrol etmene gerek kalmadı.

    case 2:                 // gün deęişkeni 2 ise;
      mesaj = "Salı";      // mesaj deęişkenini "Salı" ya eşitle
      break;               // dięer durumları kontrol etmene gerek kalmadı.

    case 3:                 // gün deęişkeni 3 ise;
      mesaj = "Çarşamba"; // mesaj deęişkenini "Çarşamba" ya eşitle
      break;               // dięer durumları kontrol etmene gerek kalmadı.

    case 4:                 // gün deęişkeni 4 ise;
      mesaj = "Perşembe"; // mesaj deęişkenini "Perşembe" ye eşitle
      break;               // dięer durumları kontrol etmene gerek kalmadı.

    case 5:                 // gün deęişkeni 5 ise;
      mesaj = "Cuma";      // mesaj deęişkenini "Cuma" ya eşitle
      break;               // dięer durumları kontrol etmene gerek kalmadı.

    case 6:                 // gün deęişkeni 6 ise;
      mesaj = "Cumartesi"; // mesaj deęişkenini "Cumartesi" ye eşitle
      // başka durum olmadığı için break kullanmadık.
  }
  document.getElementById("demo").innerHTML = "Bugün: " + mesaj;
</script>

</body>
</html>
```

Kodlayıcı Görevi

- Kontrol yapılarında kullanılan kodlar test edildiğinde sayfa açılır açılmaz javascript kodlarının çalıştığı fark edilecektir. Daha önceki örneklerde kullanıcının sayfa ile etkileşimi vardı yani kullanıcı bir butona tıkladığında (ya da imleç herhangi bir nesnenin üzerine getirildiğinde) javascript kodları çalışıyordu.
- Şimdi siz kodlayıcılardan sayfaya bir etkileşim ekleyerek kontrol yapılarının çalıştırılması beklenmektedir.

Tekrarlı (Döngü) Yapılar

Diğer programlama dillerinde olduğu gibi, Javascript' de de aynı kod her seferinde farklı bir değer için çalıştırılmak isteniyorsa tekrarlı yapılar (Döngüler) kullanılır. Örneğin birden 100'e kadar sayı yazdırmak istenirse 100 satır kodlama yapmak yerine bunu 3 satır kodlama ile yazdırabilmek mümkündür. Şimdi, senaryolar temelinde tekrarlı yapılar kullanılacaktır.

For Döngüsü

```
for (ifade 1; ifade 2; ifade 3) {
  // Çalıştırılacak kod bloğu
}

// ifade1: Kod bloğu başlamadan önce döngünün başlama değerini belirtir.
// ifade2: Kod bloğunun çalışma koşulunu tanımlar.
// ifade3: Kod bloğu çalıştırdıktan sonra her defasında çalıştırılacak kodu tanımlar.
```

1-100 arasındaki sayıların 3'e ya da 5'e bölünenlerini yazdıran javascript kodlaması

```
<!DOCTYPE html>
<html>
<style>
  body {font-family:Verdana;font-size:14px;}
  #demo {padding:15px; margin:10px; background-color:#ffc000;}
</style>
<body>
  <h2>For Döngüsü Örneği</h2>
  <p id="demo"></p>

  <script>
    // var ile bir i değişkeni tanımlıyoruz.
    var i;

    // i=1 den i<=100 (100 dahil) olana kadar her seferinde for içerisindeki kod bloğunu çalıştır.
    // (i++) Kod bloğu her seferinde çalıştırıldığında ise i değişkenini son değerinden 1 artır.
    for(i=1; i<=100; i++)
    {
      // i%3: i değerinin 3 ile bölümünden kalanı verir.
      // i%5: i değerinin 5 ile bölümünden kalanı verir.
      // || (veya işareti) : koşullardan herhangi birinin sağlanması durumunda kod bloğu çalıştırılır.

      if(i%3==0 || i%5==0){
        // (+) : bir önceki değer üzerine yeni değer eklenmesini sağlar.
        // (i + ", ") : i değerinden sonra ", " string değerini ekler.
        document.getElementById('demo').innerHTML += i + ", ";
      }
    }

  </script>
</body>
</html>
```

For Döngüsü Örneği

3, 5, 6, 9, 10, 12, 15, 18, 20, 21, 24, 25, 27, 30, 33, 35, 36, 39, 40, 42, 45, 48, 50, 51, 54, 55, 57, 60, 63, 65, 66, 69, 70, 72, 75, 78, 80, 81, 84, 85, 87, 90, 93, 95, 96, 99, 100,

While Döngüsü

```
while (koşul) {  
    // Çalıştırılacak kod bloğu  
}  
// koşul: belirtilen koşul doğru olduğu sürece kod bloğu çalıştırılır.
```

1 den n' e kadar olan sayıların toplamı

```
<!DOCTYPE html>  
<html>  
<style>  
    body    {font-family:Verdana;font-size:14px;}  
    #demo1, #demo2 {padding:15px; margin:10px; background-color:#f2f2f2;}  
</style>  
<body>  
    <h2>While Döngüsü Örneği</h2>  
    <p id="demo1"></p>  
    <p id="demo2"></p>  
  
    <script>  
        // var ile kullanacağımız değişkenleri tanımlıyoruz.  
        var i, n, tektoplam, cifttoplam;  
  
        // değişkenlerin başlangıç değerini atıyoruz.  
        i=1;    tektoplam=0; cifttoplam=0;  
  
        // kullanıcıdan bir veri girmesini istiyoruz.  
        n= prompt("Bir sayı giriniz.");  
  
        // i <= n (ya da i<n+1) : 1, n+1 den küçük olana kadar  
        // her seferinde kod bloğunu çalıştır.  
        while(i<=n)  
        {  
            // i nin ikiye bölümünden kalanın sıfır olup olmadığını kontrol et.  
            if (i%2==0){ // eğer sıfırsa (yani çift sayı)  
                // cifttoplam değişkenine i kadar toplam sayı ekle  
                cifttoplam += i;  
            } else { // eğer sıfır değilse (yani tek sayı)  
                // tektoplam değişkenine i kadar toplam sayı ekle  
                tektoplam += i;  
            }  
            i++; // i yi en son değerinden 1 artır.  
        }  
  
        document.getElementById('demo1').innerHTML+=  
        "1 den "+ n +" e kadar olan <mark>çift</mark> sayıların toplamı: <mark>"+ cifttoplam + "  
</mark>";  
  
        document.getElementById('demo2').innerHTML+="1 den "+ n +" e kadar olan <mark  
style=background-color:orange;>tek</mark> sayıların toplamı: <mark style=background-  
color:orange;>"+ tektoplam + "</mark>";  
  
    </script>  
</body>  
</html>
```

Bir sayı giriniz.

Tamam İptal

While Döngüsü Örneği

1 den 5 e kadar olan çift sayıların toplamı: 6

1 den 5 e kadar olan tek sayıların toplamı: 9

Dikkat!

```
// ----String Birleştirme İşlemi-----
document.getElementById('demo1').innerHTML
+=
"1 den "+ n +" e kadar olan <mark>çift</mark> sayıların toplamı: <mark>" + cifttoplam + "</mark>";
```

Javascript' de herhangi bir değişkenin değerini sadece metin, sadece sayı atanabileceği gibi HTML ya da CSS kodları olarak da atanabilir. Yukarıdaki kodlama bir string birleştirme işlemidir. Burada kullanıcının girmiş olduğu n sayısı ile toplam sayıları bir anlam ifade edebilecek şekilde birleştirilmiştir. Javascript, bu kodları web sayfası içerisine sadece yazmakla görevlidir. Bu kodun nasıl görüntüleneceği ise daha önceden de değinildiği üzere tarayıcının görevidir.

Do While Döngüsü

```
do {
  // Çalıştırılacak kod bloğu
} while (koşul)
// koşul: belirtilen koşul doğru olduğu sürece kod bloğu çalıştırılır.

// "do while" döngüsünün while döngüsünden farkı
// -----
// koşulun doğru olup olmadığını kontrol etmeden önce kod bloğunu bir kez çalıştırmasıdır.
```

Kodlayıcı Görevi

- do while döngüsünü kullanarak kullanıcının girmiş olduğu sayı kadar rastgele sayı üreten bir javascript kodlama yapınız.
- İpucu
 - Math.random()
 - Math.floor()

FONKSİYON YAZIMI

Fonksiyonlar belirli bir görevi gerçekleştirmek için tasarlanmış kod bloklarıdır. Herhangi bir eylem sonucunda çağrıldıklarında yürütülmeye başlarlar.

Fonksiyonlarla ilgili daha önceden siteyi ziyaret eden bir kullanıcının bir butona tıklamasıyla elementlerin niteliğini değiştirebilen bir kodlama yapılmıştı. Bu senaryo üzerindeki kodlama iki bölüme ayrılabilir.

I. Elementlerin niteliğini değiştiren kod bloğu

```
<script>
  function Degistir() {

    document. // sayfa içerisindeki
    getElementsByTagName('INPUT')['degistirilecekElement'].
    // id'si "degistirilecekElement" olan input elementinin
    setAttribute('type', 'button');
    // type niteliğini button olarak değiştir.

    document. //sayfa içerisindeki
    getElementsByTagName('INPUT')['degistirilecekElement'].
    //id'si "degistirilecekElement" olan input elementinin
    setAttribute('value', 'Bu bir buton elementi');
    //value niteliğini "Bu bir buton elementi" olarak değiştir.

  }
</script>
```

II. Kullanıcı bir butona tıkladığında elementlerin niteliğini değiştiren kod bloğunun çağrılması

```
<button onclick="Degistir();">Değiştir</button>
```

Şimdi fonksiyon kullanım yöntemlerini karşılaştırmalı olarak inceleyelim.

```
// Yöntem 1
// Basit Fonksiyon Kullanımı
function fonksiyonAdi1() {
  // fonksiyon çağrıldığında çalıştırılacak kod bloğu
}

// Yöntem 2
// Fonksiyonun kod bloğunun çalıştırılabilmesi için
// dışarıdan değişken gönderilmesi gereken
// fonksiyon kullanımı
function fonksiyonAdi2(a, b, ...) {
  // fonksiyon çağrıldığında çalıştırılacak kod bloğu
  // örneğin a ile b nin toplamını hesaplama
}

// Yöntem 3 (Bu yöntem yukarıdaki iki yöntem için de kullanılabilir)
// Fonksiyonun kod bloğunu çalıştırdıktan sonra
// bir değer döndüren fonksiyon kullanımı
function fonksiyonAdi3(...){
  // fonksiyon çağrıldığında çalıştırılacak kod bloğu
  return geriDondurulecekDeger;
}
```

Uygulama

Aşağıda kullanıcının doğum tarihine göre yaşını hesaplamayan bir kodlama örneği paylaşılmıştır.

- 1- Kullanıcının doğum tarihini girebileceği bir sayfa tasarımının yapılması

Fonksiyon Kullanım Örneği

- 2- Kullanıcının doğum tarihine göre yaşını hesaplayan fonksiyonun yazılması

```
// Yöntem 1 ile kullanıcının yaşını hesaplama
function Hesapla(){

    // id'si yıl olan bir input nesnesinden kullanıcının doğum tarihini alma
    var dogumtarihi = document.getElementById('yil').value;

    // güncel tarihi alma
    var gunceltarih = new Date().getFullYear();

    // yaşı hesaplama
    var yas = gunceltarih-dogumtarihi;

    //yaşı hesapladıktan sonra kullanıcıya gösteren fonksiyonu çağırma
    Goster(yas);
}
```

- 3- Kullanıcının yaşını gösteren fonksiyonun yazılması

```
// Yöntem 2 ile kullanıcının yaşını gösterme
function Goster(a){
    // id'si mesaj olan elementin içerisinde yaşı kullanıcıya gösterme
    document.getElementById('mesaj').innerHTML= "Yaşınız: "+ a;
}
```

4- Javascript Kodları ile HTML kodlarının birleştirilmesi

```
<!DOCTYPE html>
<html>
<style>
  body {font-family:Verdana;font-size:14px;}
  h2 {padding:5px;background-color:orange;}
  dogumtarihi {
    background-color:#f3f3f3;padding:30px;display:block;
  }
  mesaj {
    background-color:#f4f4f4;padding:30px;display:block;
    margin-top:30px;color: orange;visibility:hidden;
  }
</style>
<body>
  <h2>Fonksiyon Kullanım Örneği</h2>
  <div>
    <dogumtarihi>
      <input id=yil type=text value='Doğum Tarihini Giriniz'>
      <input type=submit value='Hesapla' onclick="Hesapla();">
    </dogumtarihi>
    <mesaj id="mesaj">

  </mesaj>
</div>

<script>
  // Yöntem 1 ile kullanıcının yaşını hesaplama
  function Hesapla(){

    // id'si yil olan bir input nesnesinden kullanıcının doğum tarihini alma
    var dogumtarihi = document.getElementById('yil').value;

    // güncel tarihi alma
    var gunceltarih = new Date().getFullYear();

    // yaşı hesaplama
    var yas = gunceltarih-dogumtarihi;

    //yaşı hesapladıktan sonra kullanıcıya gösteren fonksiyonu çağırma
    Goster(yas);
  }

  // Yöntem 2 ile kullanıcının yaşını gösterme
  function Goster(a){
    // id'si mesaj olan elementin içerisinde yaşı kullanıcıya gösterme
    document.getElementById('mesaj').innerHTML= "Yaşınız: "+ a;
    document.getElementById('mesaj').style.visibility='visible';
  }
</script>
</body>
</html>
```


Bu örnekte, Yöntem 1 ve Yöntem 2' ye göre fonksiyon kullanımları birleştirilmiştir. Şimdi, örnek Yöntem 3 ' e göre değiştirilecektir.

```
<script>
// Yöntem 1 ile Kullanıcının Yaşını Gösterme
function Goster(){

    // id'si yil olan elementin değerini alma.
    var dogumtarihi = document.getElementById('yil').value;

    // id'si mesaj olan elementin içerisinde yaşı kullanıcıya gösterme
    document.getElementById('mesaj').innerHTML= "Yaşınız: "+ Hesapla(dogumtarihi);
    document.getElementById('mesaj').style.visibility='visible';
}
// Yöntem 3 ile Kullanıcının yaşını hesaplayıp geri döndürme
function Hesapla(a){

    // güncel tarihi alma
    var gunceltarih = new Date().getFullYear();

    // fonksiyon içerisine gönderilen değeri kullanarak yaşı hesaplama
    var yas = gunceltarih-a;

    //yaşı hesapladıktan sonra, fonksiyonun çağrıldığı yere gönderme
    return yas;
}
</script>
```

Kodlayıcı Görevi

- Yukarıdaki örnekte verilen senaryoya göre Yöntem 1 ve Yöntem 3 kullanıldı. Peki sizce kullanıcı ile etkileşim sırasında ilk olarak hangi fonksiyonu çağırılmalıdır?

```
<dogumtarihi>
  <input id=yil type=text value='Doğum Tarihini Giriniz'>
  <input type=submit value='Hesapla' onclick=" ? ">
</dogumtarihi>
```

DİZİLER

Diziler, birden çok değeri tek bir değişkende depolamak için kullanılır. Önceki örneklerde, her bir değer için yeni değişken tanımlamaları yapıldı. Örneğin:

```
var oto1= "Volvo";
var oto2= "Opel";
var oto3= "Ford";
var takim1= "Çorumspor";
var takim2= "Muğlaspor";
var takim3= "Gençlerbirliği";
```

Bu türden bir değişken tanımlama yerine, benzer niteliklere sahip olan değişkenler bir dizi değişken-
de toplanabilir ve daha kolay yönetilebilir. Örneğin:

```
var otolar= ["Volvo", "Opel", "Ford"];  
var takimler= ["Çorumspor", "Muğlaspor", "Gençlerbirliği"];
```

ya da

```
var otolar = new Array("Volvo", "Opel", "Ford");  
var takimler= new Array("Çorumspor", "Muğlaspor", "Gençlerbirliği");
```

Yukarıdaki iki örnek de aynı şeyi yapmaktadır. Basitlik, okunabilirlik ve kodun çalıştırılma hızı için birincisinin kullanılması tavsiye edilmektedir. Peki, dizi içerisindeki bir öğeye (ör: Muğlaspor) nasıl ulaşılabilir?

Diziler, içerisindeki öğelere erişmek için sayıları kullanır. Bu sayılar 0' dan başlayarak dizinin iç-
risindeki öğe sayısından 1 eksik değer arasında olabilir. Örneğin takimler dizisi 3 öğe içermektedir.
Burada "Muğlaspor" değerini ekrana yazdırmak için;

```
document.getElementById("demo").innerHTML = takimler[1];
```

kodlaması kullanılabilir. Dizinin tüm öğelerini ekrana yazdırmak için ise,

```
document.getElementById('mesaj').innerHTML = takimler;
```

Uygulama

Şimdi bir senaryo temelinde yukarıdaki örnekler kodlanacaktır. Kullanıcının girmiş olduğu değere
göre dizinin elemanını ekranda gösteren bir kodlama aşağıdaki şekilde olabilir.

1- Kullanıcının veri giriş yapabilmesi için HTML kodları

```
<dizi>  
  <input id=eleman type=text value='Dizinin kaçınıcı elemanı'>  
  <input type=submit value='Göster' onclick="Goster(document.getElementById('eleman').value);">  
</dizi>
```

2- Kullanıcının girmiş olduğu değere göre dizinin öğesini gösterme

```
<script>  
  function Goster(a){  
    var otolar= ["Volvo", "Opel", "Ford"];  
    var takimler= ["Çorumspor", "Muğlaspor", "Gençlerbirliği"];  
    document.getElementById('mesaj').innerHTML = takimler[a];  
  }  
</script>
```

Ekran Çıktısı

Dizi Kullanım Örneği

Gençlerbirliği

Kodlayıcı Görevi

- Kullanıcı herhangi bir değer girmez ya da dizinin öğe sayısı aralığında olmayan (0' dan küçük ya da dizi uzunluğundan 1 eksik değerden büyük) bir değer girerse ne olur? Test ediniz?

Hatırlanacağı üzere, bu gibi durumlar için bir mantıksal sınama yapılması gerektiği daha önceden öğrenilmişti. Buna göre kullanıcı herhangi bir değer girmez ya da dizinin öğe sayısı aralığında olmayan bir değer girdiğinde dizinin tüm öğelerini ekrana yazan bir kodlama yapalım.

```

<script>
function Goster(a){
  var takimler= ["Çorumspor", "Muğlaspor", "Gençlerbirliği"];
  if (a==0 || a==1 || a==2)
  {
    document.getElementById('mesaj').innerHTML = takimler[a];
  } else {
    document.getElementById('mesaj').innerHTML = takimler;
  }
}
</script>

```

Ekran Çıktısı

Dizi Kullanım Örneği

Çorumspor,Muğlaspor,Gençlerbirliği

BASİT DİZİ İŞLEMLERİ

Önceki örneklerde sabit bir dizi tanımlayarak, dizi içerisindeki öğelere erişilmeye çalışıldı. Şimdi ise, diziyi öğe eklemek, silmek, değiştirmek gibi işlemler görülecektir.

1- Dizinin öğelerini değiştirme

En basit şekliyle dizinin herhangi bir öğesi aşağıdaki gibi değiştirilebilir.

```
<script>
  var meyveler = ["Muz", "Portakal", "Elma"];

  meyveler[0] = "Kiwi";
  meyveler[1] = "Kiraz";
  meyveler[2] = "Vişne";

</script>
```

2- Kullanıcının seçmiş olduğu değeri dizinin başına ya da sonuna ekleme ve dizinin başından ya da sonundan veri silme

HTML kodları

```
<basitdizi>
  <select id="oto">
    <option value="Anadol">Anadol</option>
    <option value="Mercedes">Mercedes</option>
    <option value="Saab">Saab</option>
    <option value="Audi">Audi</option>
    <option value="Tata">Tata</option>
    <option value="Subaru">Subaru</option>
  </select>
  <button id="Ekle" onclick="Ekle(document.getElementById('oto').value);">Ekle</button>
  <button id="Sil" onclick="Sil();">Sil</button>
</basitdizi>
```

Javascript Kodları

```
<script>
  var otolar = ["Volvo", "Opel", "Ford"];
  function diziGoster(){
    document.getElementById('demo').innerHTML= otolar;
  }
  function Ekle(a) {
    otolar.push(a); //gönderilen değeri dizinin sonuna ekler.
    otolar.unshift(a); //gönderilen değeri dizinin başına ekler.
    diziGoster();
  }
  function Sil(){
    otolar.pop(); //dizinin sonuncu öğesini siler.
    otolar.shift(); //dizinin birinci ([0]) öğesini siler.
    diziGoster();
  }
</script>
```

3- Dizinin Herhangi Bir Yerine (ör: 3. Ögesinden sonra) Yeni Bir Öge Ekleme ya da Herhangi Bir Ögesini Değiştirme

HTML kodları

```
<basitdizi>
  <select id="oto">
    <option value="Anadol">Anadol</option>
    <option value="Mercedes">Mercedes</option>
    <option value="Saab">Saab</option>
    <option value="Audi">Audi</option>
    <option value="Tata">Tata</option>
    <option value="Subaru">Subaru</option>
  </select>
  <input id="kacinci" value="Kaçıncıdan itibaren">
  <input id="degistirilecek" value="Değiştirilecek Öge">
  <button id="islem" onclick="diziIslem();">İşlem</button>
</basitdizi>
```

Javascript Kodları

```
<script>
var otolar = ["Volvo", "Opel", "Ford"];
function Ekle(a, b, c) {
  otolar.splice(a,b,c);
  // b=0 geleceği için bu bir ekleme işlemidir.
  document.getElementById('demo').innerHTML= otolar;
}
function Degistir(a, b, c){
  otolar.splice(a,b,c);
  // b 0 dan farklı bir değer geleceği için bu bir değiştirme işlemidir.
  document.getElementById('demo').innerHTML= otolar;
}
function diziIslem(){
  var yenioge = document.getElementById('oto').value;
  var kacinci = document.getElementById('kacinci').value;
  var degistirilecek = document.getElementById('degistirilecek').value;
  if (degistirilecek == 0) {
    Ekle(kacinci, degistirilecek, yenioge);
  } else {
    Degistir(kacinci, degistirilecek, yenioge);
  }
}
</script>
```

Ekleme

Dizinin herhangi bir yerine yeni bir öge ekleme ya da herhangi bir ögesini değiştirme

Volvo,Mercedes,Opel,Ford

Mercedes ▾ 1 0 İşlem

Değiştirme

Dizinin herhangi bir yerine yeni bir öge ekleme ya da herhangi bir ögesini değiştirme

Volvo,Mercedes,Ford

Mercedes ▾ 1 1 İşlem

4. Dizinin Öğelerini Sıralama

Programlama dilleri ile iki tür sıralama yapılabilir. İlk olarak harf, metin ya da sayıları alfabetik sıralayabilir (Ör: 0, 1, 10, 2, 20, a, ab, b, bd). İkinci olarak ise sayıları numerik olarak sıralayabilir (Ör: 0, 1, 2, 10, 20). Javascriptle de dizi içerisindeki öğeleri alfabetik ya da numerik şekilde sıralayabilirsiniz. Bu işlem için .sort() komutu kullanılır. sort(), varsayılan olarak değerleri metin olarak sıralar. "Ankara", "10" ve "?" öğelerini kıyaslandığında, rakamlar harflerden daha önce geldiği için 10 ilk sırada olmalıdır. "a" harfi "m" den önce geldiği için de dizinin sıralaması 10, Ankara, Muğla olur. Sayılar metin hâlinde sıralanırsa "25", "100"den büyüktür, çünkü "2", "1"den büyüktür.

HTML Kodları

```
<style>
  body {font-family:Verdana;font-size:14px;}
  h2 {padding:5px;background-color:orange;}
  h3, basitdizi {
    background-color:#f3f3f3;padding:30px;display:block;
  }
  .demo {
    background-color:#f4f4f4;padding:30px;display:block;
    margin-top:30px;color: orange;
  }
  button:hover{background-color:orange;
  }
</style>

<h2>JavaScript Dizi Sıralama</h2>

<button onclick="AlfabetikSiralama()">Alfabetik Sıralama</button>
<button onclick="NumerikSiralama()">Numerik Sıralama</button>

<p id="demo" class="demo"></p>
```

Javascrpt Kodları

```

<script>
  //Sadece sayılardan (int) oluşan dizi
  var sayiIcerenDizi = [40, 100, 1, 5, 25, 10];
  //Sadece metinden (string) oluşan dizi
  var metinIcerenDizi = ["ankara", "muğla", "istanbul"];
  //Hem sayı hem de metinden oluşan dizi
  var harfMetinVeSayiIcerenDizi = [06, 48, 34, "ankara", "muğla", "istanbul"];
  //Dizileri ekrana yazdıran fonksiyon
  function Goster() {
    document.getElementById("demo").innerHTML = sayiIcerenDizi + "<br/><br/>";
    document.getElementById("demo").innerHTML += metinIcerenDizi + "<br/><br/>";
    document.getElementById("demo").innerHTML += harfMetinVeSayiIcerenDizi;
  }

  //sayfa ilk yüklendiğinde dizilerin ilk halini göstermek için fonksiyonu çağırıyoruz.
  Goster();

  function AlfabetikSiralama() {
    //sort() dizi öğelerinin alfabetik sıralanmasını sağlar.
    sayiIcerenDizi.sort();
    metinIcerenDizi.sort();
    harfMetinVeSayiIcerenDizi.sort();
    //dizi öğelerinin sıralanmış hallerini göstermek için fonksiyonu çağırıyoruz.
    Goster();
  }

  function NumerikSiralama() {
    // dizi öğelerinin numerik sıralanması için
    // sort() kodunu farklı bir biçimde kullanıyoruz.

    sayiIcerenDizi.sort(function(a, b){return a - b});
    metinIcerenDizi.sort(function(a, b){return a - b});
    harfMetinVeSayiIcerenDizi.sort(function(a, b){return a - b});

    // sort(function(a, b){return a - b}) kodu
    // her dizi öğesini iikili olarak karşılaştırır.
    // eğer küçükten büyüğe sıralamak istiyorsak a-b
    // büyükten küçüğe sıralamak istiyorsak b-a döndürülür.

    Goster();
  }
</script>

```

Ekran Çıktıları

JavaScript Dizi Sıralama

Alfabetik Sıralama
Numerik Sıralama

1,10,100,25,40,5

ankara,istanbul,muğla

34,48,6,ankara,istanbul,muğla

JavaScript Dizi Sıralama

Alfabetik Sıralama
Numerik Sıralama

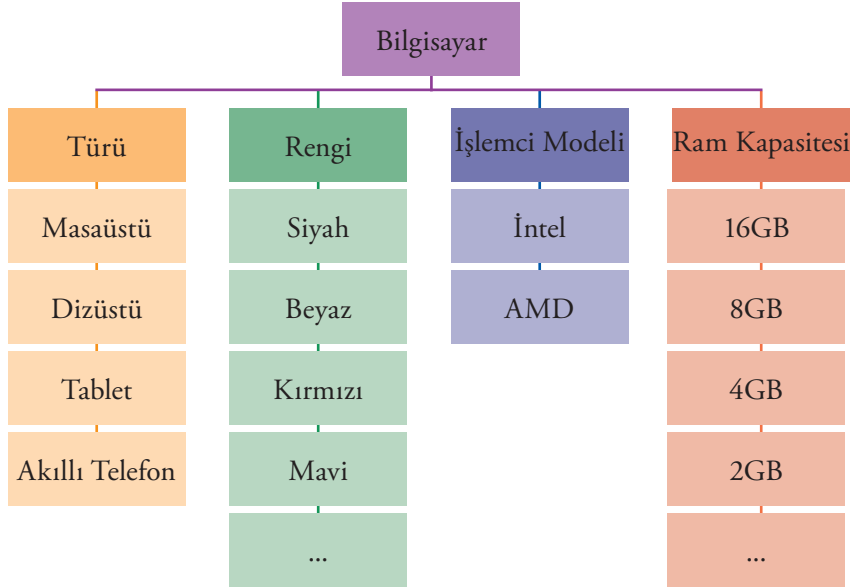
1,5,10,25,40,100

ankara,muğla,istanbul

6,34,48,ankara,muğla,istanbul

NESNELER

Gerçek yaşam içerisinde herhangi bir nesne düşününüz. Örneğin bir bilgisayarı ele alalım. Genel bir nesne ismini içeriyor değil mi? Bir diğer ifadeyle bir bilgisayar tür, renk, işlemci, ram gibi özelliklerin alt özelliklerinin birleşiminden doğan bir varlığı simgeliyor.



Şekil 9: Bilgisayar Nesnesinin Özellikleri

Benzer şekilde kişileri, arabaları nesneleştirmek mümkündür. Peki bu nesne yapısı kodlama yapmak ne işe yarar?

Dizileri kullanarak benzer niteliklere sahip olan değişkenler tek bir değişkende toplanabilir.

```
var bilgisayarlar = ["Asus", "Casper", "Dell", "IBM", "Lenova", "Vestel"]
```

Peki, şekil 9 'daki bilgisayar özellikleri bir değişkende nasıl tutulabilir? İşte bu durumda nesne temelli değişken tanımlanabilir. Javascript'te bu işlem aşağıdaki gibidir:

```
var bilgisayar = { turu:"Dizüstü", marka:"Asus", rengi:"siyah", islemci:"AMD", ram:"4GB" };
```

Dizi içerisindeki herhangi bir öğeye erişirken dizi sıra numarası (Ör: dizi [0]) kullanılırdı. Nesne içerisindeki bir öğeye erişirken de, "nesne" . "özellik" şeklinde erişilebilir.

```
var bilgisayar = { turu:"Dizüstü", marka:"Asus", rengi:"siyah", islemci:"AMD", ram:"4GB" };  
document.getElementById("demo").innerHTML = bilgisayar.turu + "|" + bilgisayar.marka;
```

Uygulama

Kullanıcının girmiş olduğu bilgisayar özelliklerini bir nesneye aktaran ve bu nesnelere bir dizi içerisinde tutup girilen bilgisayar özelliklerini bir tabloda gösteren sayfa kodlamasını yapınız.

1. Kullanıcının Girmiş Olduğu Bilgisayar Özelliklerini Bir Nesneye Aktarma

HTML Kodu

```

<h2>Dizi ve Nesne Kullanımı</h2>
<div>
  <nesne>
    Türü:
    <select id="tur">
      <option value="Dizüstü">Dizüstü</option>
      <option value="Masaüstü">Masaüstü</option>
      <option value="Tablet">Tablet</option>
      <option value="Akıllı Telefon">Akıllı Telefon</option>
    </select>
    <br/><br/>
    Markası:
    <select id="marka">
      <option value="Asus">Asus</option>
      <option value="Casper">Casper</option>
      <option value="Dell">Dell</option>
      <option value="IBM">IBM</option>
      <option value="Lenova">Lenova</option>
      <option value="Vestel">Vestel</option>
    </select>
    <br/><br/>
    Rengi:
    <input id="renk" type="text">
    <br/><br/>
    İşlemci Modeli:
    <select id="islemci">
      <option value="Intel">Anadol</option>
      <option value="AMD">Mercedes</option>
    </select>
    <br/><br/>
    RAM Kapasitesi:
    <input id="RAM" type="text">
    <br/><br/>
    <button id="Ekle" onclick="Ekle();">Ekle</button>
    <button id="Sil" onclick="Sil();">Sil</button>
  </nesne>
</div>
<div id="tablo"><!-- Nesneleri burada tablo şeklinde göstereceğiz. --></div>

```

Javascript Kodu

```
function Ekle() {
    // Özellikleri atanmamış bir nesne tanımladık
    var bilgisayar = { tur:" ", marka:" ", islemci:" ", ram:" ", fiyat:" "};

    // Daha sonra kullanıcının girmiş olduğu verileri atıyoruz.
    bilgisayar.tur = document.getElementById('tur').value;
    bilgisayar.marka = document.getElementById('marka').value;
    bilgisayar.islemci = document.getElementById('islemci').value;
    bilgisayar.ram = document.getElementById('ram').value;
    bilgisayar.fiyat = document.getElementById('fiyat').value;
}
```

2. Oluşturulan Nesnelere Bir Dizi İçerisinde Tutma

```
var dizi = []; //içi boş bir dizi tanımladık.
function Ekle() {
    // Özellikleri atanmamış bir nesne tanımladık
    var bilgisayar = { tur:" ", marka:" ", islemci:" ", ram:" ", fiyat:" "};

    // Daha sonra kullanıcının girmiş olduğu verileri atıyoruz.
    bilgisayar.tur = document.getElementById('tur').value;
    bilgisayar.marka = document.getElementById('marka').value;
    bilgisayar.islemci = document.getElementById('islemci').value;
    bilgisayar.ram = document.getElementById('ram').value;
    bilgisayar.fiyat = document.getElementById('fiyat').value;

    // Kullanıcının girmiş olduğu verileri tutan nesneyi diziyeye ekliyoruz.
    dizi.push(bilgisayar);
}
```

3. Dizi İçerisindeki Nesnelere Bir Tabloda Gösterme

Yeni CSS Kodları

```
/*tablonun tek (1,3,5,7) satırlarının arkaplan rengini ayarladık*/
tr:nth-child(odd) {background: #CCC;}

/*tablonun çift (2,4,6,8) satırlarının arkaplan rengini ayarladık*/
tr:nth-child(even) {background: #bff442;}

/*tablonun ilk satırının (başlık) arkaplan rengini ayarladık*/
tr:first-child {background: #ffa500;}

/*tablonun satırları içerisindeki hücrelerin iç boşluğunu ayarladık*/
td {padding:5px; }
```

Javascript Kodu

```

var dizi = []; // içi boş bir dizi tanımladık.
// tablonun HTML kodlarını tutacağımız bir değişken tanımladık.
var tabloKodu;

function diziGoster(){
    // ilk olarak bir tablo etiketi açtık.
    tabloKodu = "<table>";

    // tablonun ilk satırını (başlıklar) oluşturduk.
    tabloKodu += "<tr>";
    tabloKodu += "<td>Türü</td>";
    tabloKodu += "<td>Markası</td>";
    tabloKodu += "<td>İşlemcisi</td>";
    tabloKodu += "<td>Belleği</td>";
    tabloKodu += "<td>Fiyatı</td>";
    tabloKodu += "</tr>";

    // dizinin içerisinde ne kadar nesne ögesi varsa,
    for (var i = 0; i < dizi.length; i++)
    {
        // o kadar yeni satır ekliyoruz.
        tabloKodu += "<tr>";

        // her yeni satırda
        // dizi içerisindeki nesnenin özellikleri
        // ayrı ayrı sütunlarda gösterilecek.
        tabloKodu += "<td>"+dizi[i].tur+"</td>";
        tabloKodu += "<td>"+dizi[i].marka+"</td>";
        tabloKodu += "<td>"+dizi[i].islemci+"</td>";
        tabloKodu += "<td>"+dizi[i].ram+"</td>";
        tabloKodu += "<td>"+dizi[i].fiyat+"</td>";

        // açtığımız yeni satır etiketini kapatıyoruz.
        tabloKodu += "</tr>";
    }

    // kodlamanın başında açtığımız tablo etiketini kapattık.
    tabloKodu += "</table>";

    // son olarak, oluşturmuş olduğumuz HTML' yi kullanıcıya göstermek için
    // id'si tablo olan elementin içerisine tabloKodu değişkenini ekledik.
    document.getElementById('tablo').innerHTML = tabloKodu;
}

```

```

function Ekle() {
  // Özellikleri atanmamış bir nesne tanımladık
  var bilgisayar = { tur:" ", marka:" ", işlemci:" ", ram:" ", fiyat:" "};

  // Daha sonra kullanıcının girmiş olduğu verileri atıyoruz.
  bilgisayar.tur = document.getElementById('tur').value;
  bilgisayar.marka = document.getElementById('marka').value;
  bilgisayar.islemci = document.getElementById('islemci').value;
  bilgisayar.ram = document.getElementById('ram').value;
  bilgisayar.fiyat = document.getElementById('fiyat').value;

  // Kullanıcının girmiş olduğu verileri tutan nesneyi diziye ekliyoruz.
  dizi.push(bilgisayar);

  // Her yeni bir nesne eklendiğinde, dizi içerisindeki nesnelere
  // ekrana yazdıracak fonksiyonu çağırıyoruz.
  diziGoster();
}

function Sil() {
  dizi.pop();
  diziGoster();
}

```

Ekran Çıktısı

Dizi ve Nesne Kullanımı

Türü:

Markası:

İşlemci Modeli:

RAM Kapasitesi:

Fiyatı (KDV' siz):

Türü	Markası	İşlemcisi	Belleği	Fiyatı
Masaüstü	Casper	Intel	6GB	1500 TL
Tablet	Vestel	Intel	2GB	600 TL
Akıllı Telefon	Lenova	AMD	2GB	750 TL
Dizüstü	Asus	AMD	8GB	2500 TL

5. VERİ TABANI YÖNETİMİ

5. VERİ TABANI YÖNETİMİ

Veri tabanı yönetimine geçmeden önce, bir önceki ünite de Javascript ile kullanıcının girmiş olduğu bilgisayar özelliklerini ekranda gösteren uygulamayı biraz tartışalım.

Türü	Markası	İşlemcisi	Belleği	Fiyatı
Masaüstü	Casper	Intel	6GB	1500 TL
Tablet	Vestel	Intel	2GB	600 TL
Akıllı Telefon	Lenova	AMD	2GB	750 TL
Dizüstü	Asus	AMD	8GB	2500 TL

Kullanıcı sayfaya bağlandığında, yukarıdaki gibi bilgisayarlar özelliklerini girmiş olsun. Sonrasında tarayıcısını açıp kapattığında bu girilen bilgilere ulaşılabilir mi?

Şu ana kadar yazılan kodlar buna izin vermez. Peki girilen bilgilere sonradan ulaşabilmek için ne yapılabilir? Bunun web programlamada temel olarak iki yöntemi var:

- Birincisi istemci tarafında cookie (çerez) yönetimi
- İkincisi ise sunucu tarafında veri tabanı yönetimi

Bu bölümde, kullanıcı tarafından girilen verilerin tutulması için sunucu tarafında veritabanı yönetim sistemlerinin kullanımı üzerinde durulacak. Eğer Microsoft Excel ve Microsoft Access'in kullanımı konusunda bilgi sahibiyse bu ünite kazanımlarını daha kolay kavrayıp uygulayabilirsiniz. Bununla birlikte, daha önce bu programları hiç kullanmadıysanız bile kitaptaki uygulama örnekleri ile veri tabanı yönetiminin nasıl yapıldığı ile ilgili bilgi düzeyinizi artırabilirsiniz.

VERİ TABANI

Veri tabanı; kolayca erişilebilecek, yönetilebilecek ve güncellenebilecek şekilde düzenlenmiş bir bilgi topluluğudur. Veriler, satırlar, sütunlar ve tablolar hâlinde organize edilir ve ilgili bilgileri bulmayı kolaylaştırmak için kayıt hâlinde eklenir. Örneğin okullarınızda kullanılan e-Okul Web Sitesi sizinle ilgili tüm bilgileri bir veri tabanında tutmaktadır. Bunun yanı sıra hastanelerdeki hasta, doktor, tedavi, tetkik bilgileri, bankalardaki; müşteri, mevduat, kredi bilgilerinin hepsi bir veri tabanında tutulmaktadır.

Veri tabanı sistemleri, veri tekrarlarını ortadan kaldırarak veri bütünlüğünün sağlanmasına ve verilerin bir düzen içinde tutulmasına olanak sağlar. Kodlayıcıların kullanabileceği birçok veri tabanı sistemi vardır. Bunlar yandaki şekilde gösterilmektedir. Bu ünite de, ücretsiz bir veri tabanı yönetim sistemi olan MariaDB yönetimi ele alınacaktır.

MariaDB, MySQL'in kaynak kodundan türemiş, ücretsiz olarak kullanılabilen ilişkisel veri tabanı sistemidir. İlişkisel veri tabanı ise veri tabanı içerisinde birçok tablo kullanıldığı ve tablolar arasında ilişkiler kurulduğu bir yapıdır.

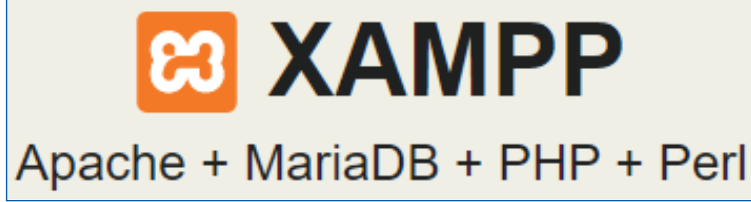
Tablolar arasındaki bir ilişki bir tabloya, başka bir tablodaki kaydı bağlamayı sağlar. Bu şekilde veriler daha az yer kaplar ve işlemleri kolaylaştırır.

Veritabanı Sistemleri

- MySQL
- MariaDB
- Microsoft SQL
- Microsoft Access
- PostgreSQL
- Oracle
- Sybase
- Berkeley

XAMPP KURULUMU

XAMPP ücretsizdir, yüklenmesi kolaydır. İçerisindeki PhpMyAdmin paneli MariaDB veri tabanının yönetimini sağlar.



PhpMyAdmin, MySQL veri tabanı ve türevi olan MariaDB yönetimi için en popüler uygulamalardan biridir. PHP ile yazılmış ücretsiz bir araçtır. Yandaki QR Kodu ya da dipnottaki adresi kullanarak kullanmış olduğunuz işletim sistemine uygun olan yazılımı indirebilirsiniz.



XAMPP kurulumunun nasıl yapılacağı ile ilgili internette, “XAMPP Kurulumu” anahtar kelimeleri ile bir arama yaptığınızda, birçok görsel anlatımlı metin ya da video bulabilirsiniz. Bunun dışında, yandaki QR kodunu kullanarak ya da dipnottaki web adresini ziyaret ederek yazar tarafından seçilmiş video anlatımını kullanabilirsiniz.



Bilgisayarınıza XAMPP kurduğunuzda, kurulum yapmış olduğunuz bilgisayarı basit bir web sunucusuna dönüştürmüş olursunuz. Daha önceden bilgisayarınız bir web sitesine bağlanırken ve istemci olarak çalışırken artık sunucu olarak da çalışabilir. Daha açık bir ifadeyle kendi bilgisayarınızda oluşturmuş olduğunuz bir web sitesinin kaynaklarına yine kendi bilgisayarınızın tarayıcısını kullanarak erişebilirsiniz. Bir sonraki ünite de bu konu daha detaylı incelenecektir.

MARIADB VERİ TABANI YÖNETİMİ

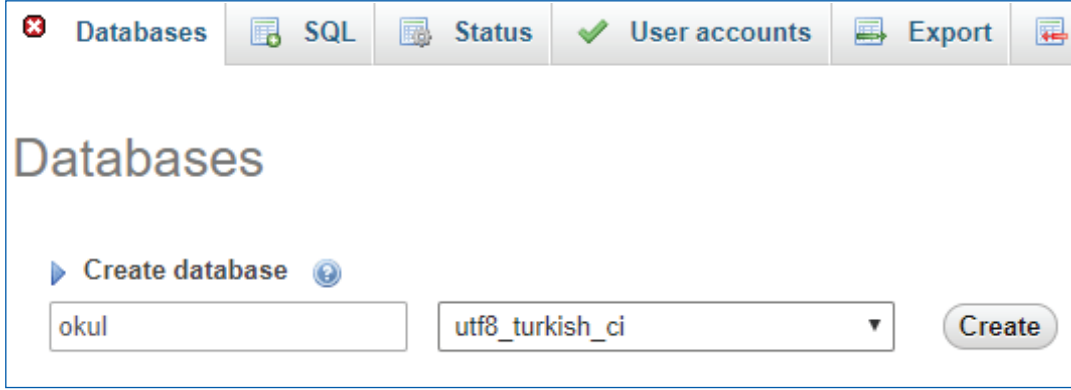
Bilgisayarınızda herhangi bir tarayıcı açılıp ve <http://localhost/phpmyadmin/> adresine bağlanılır. Burada localhost, web sitesi kaynaklarının tarayıcının açık olduğu bilgisayarda olduğu belirtilmektedir. Eğer XAMPP kurulu olmayan bir bilgisayarda çalışıyorsanız dipnottaki adresini kullanarak, phpmyadmin sitesinin kullanıcıların testine açtığı demo veri tabanı yönetimini kullanabilirsiniz.

Veri tabanı Oluşturma

The screenshot shows the phpMyAdmin demo server interface. The browser address bar displays <https://demo.phpmyadmin.net/master-config/index.php>. The page title is "phpMyAdmin Demo Server". A warning message states: "You are using the demo server. You can do anything here, but please do not change root, debian-sys-maint and pma users. More information is available at demo.phpmyadmin.net." The interface includes a navigation menu on the left with options like "New", "information_schema", "menagerie", "mysql", "opengis", "performance_schema", "phpmyadmin", "sakila", and "world". The main content area shows "General settings" with "Server connection collation" set to "utf8mb4_unicode_ci" and "Appearance settings" with "Language" set to "English" and "Theme" set to "pmahomme".

phpMyAdmin yönetim panelini kullanarak yeni bir veri tabanı oluşturabilir ya da oluşturmuş olduğunuz veri tabanları içerisinde yönetmek istediklerinizi seçebilirsiniz.

Yeni bir veri tabanı oluştururken, şekildeki gibi veri tabanı ismi (okul) ve veri tabanının dil desteği seçilir (utf8_turkish_ci). Daha sonra create (oluştur) butonuna tıklanır. Böylece Türkçe karakterleri (ı, ö, ğ vb.) destekleyen okul isimli bir veri tabanı oluşturulmuş olur.



The screenshot shows the 'Databases' section of phpMyAdmin. At the top, there are navigation tabs: 'Databases' (with a red 'x' icon), 'SQL', 'Status', 'User accounts' (with a green checkmark), and 'Export'. Below the tabs, the word 'Databases' is displayed in a large font. Underneath, there is a 'Create database' button with a play icon and a help icon. Below this, there is a text input field containing 'okul' and a dropdown menu showing 'utf8_turkish_ci'. To the right of these fields is a 'Create' button.

Oluşturulan veri tabanına daha sonradan erişilmek istendiğinde tekrar Databases sekmesine tıklanarak ulaşılabilir. Eğer veri tabanını silmek isterseniz veri tabanının en solundaki kutucuğu seçerek Drop (Çıkar) butonuna tıklayabilirsiniz.

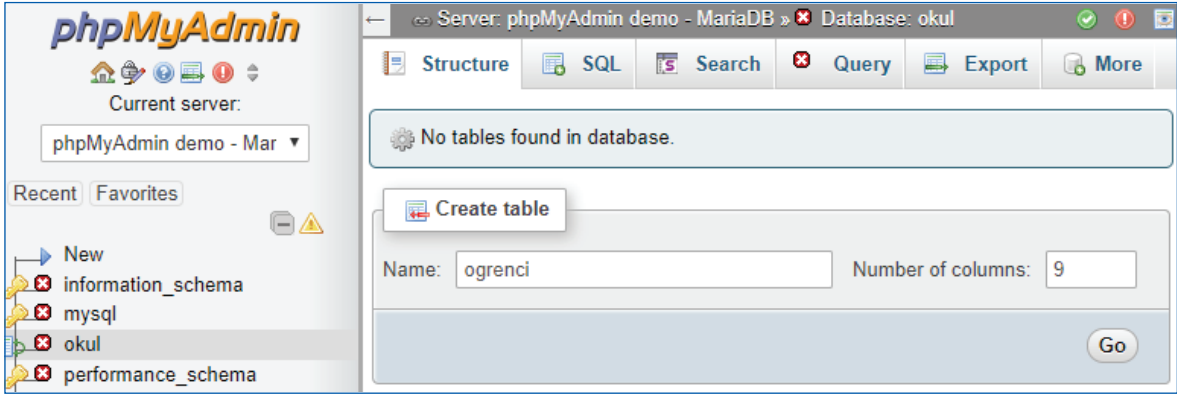
Database	Collation	Master replication	Action
<input type="checkbox"/> information_schema	utf8_general_ci	Replicated	Check privileges
<input type="checkbox"/> mysql	latin1_swedish_ci	Replicated	Check privileges
<input checked="" type="checkbox"/> okul	utf8_turkish_ci	Replicated	Check privileges
<input type="checkbox"/> performance_schema	utf8_general_ci	Replicated	Check privileges
<input type="checkbox"/> phpmyadmin	utf8_bin	Replicated	Check privileges
Total: 5		latin1_swedish_ci	

Check all With selected: Drop

Tablo Oluşturma

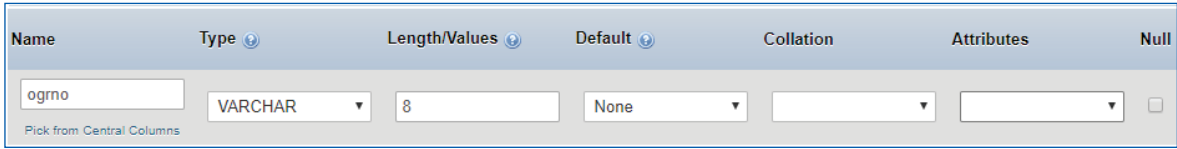
Bir veri tabanı tablolardan, tablolar ise satır ve sütunlardan oluşur. Şimdi oluşturduğunuz veri tabanına tablo ekleyiniz. Bunun için öncelikle tablo içeriğimiz üzerinde tartışalım. Bir okulu ele aldığımızda, okul içerisinde; öğretmen, öğrenci, alt yapı, mali işler vb. alt varlıklar vardır. İlişkisel bir veri tabanı, bu varlıklarla ilgili bilgileri ayrı ayrı tablolarda tutar. Örneğin öğrenci varlığı ile ilgili bilgiler neler olabilir? Öğrenci Numarası, Adı, Soyadı, Yaşı, Doğum Tarihi, Adresi, Dersi, 1. Yazılı Notu, 2. Yazılı Notu, 3. Yazılı Notu, Ortalama, Dersten Geçti ya da Kaldı Durumu vb... Bu örnekler daha da çoğaltılabilir. Şimdilik yukarıdaki bilgileri içeren 12 sütundan oluşan bir öğrenci tablosu oluşturunuz

- Databases sekmesine tıkla → okul veri tabanına tıkla
- Structure sekmesi altında tablo oluşturacağınız ekranla karşılaştınız. Şekildeki gibi tablo ismi (ogrenci) ve tablonun sütun sayısı (12) yazılarak daha sonra Go butonuna tıklanır.



Not: “utf8_turkish_ci” dil desteği veri tabanımızın içeriğinin Türkçe karakterleri desteklemesini sağlar fakat veri tabanı ismi ve altındaki tablo isimleri Türkçe karakter içermeyecek şekilde yazılır.

Tabloyu oluşturduktan sonra tablo sütunlarının özelliklerini ayarlayacağınız ekranla karşılaşacaksınız. Şu an bizim için önemli olan 7 özellik üzerinde duracağız.



Name (Sütun İsmi)

Öğrenci varlığı ile ilgili, Öğrenci Numarası, Adı, Soyadı, Dersi, 1. Yazılı Notu, 2. Yazılı Notu, 3. Yazılı Notu, Ortalama Dersten Geçti ya da kaldı durumu vb bilgiler tutulabileceğinden bahsedilmmişti. Sütun ismi verilirken bir sonraki ünite de sütunlara kolay erişilebilmesi için aşağıdaki maddelere göre kodlama yapılması önerilir.

- ✓ Türkçe karakter içermemelidir.
- ✓ İki kelime arasında boşluk kullanılmamalı bunun yerine kelimelerin ilk harfleri büyük harfle yazılmalıdır (Ör: OgrNo).
- ✓ Uzun sütun isimleri anlamlı şekilde kısaltılmalıdır (Ör: YazNot1, YazNot2).

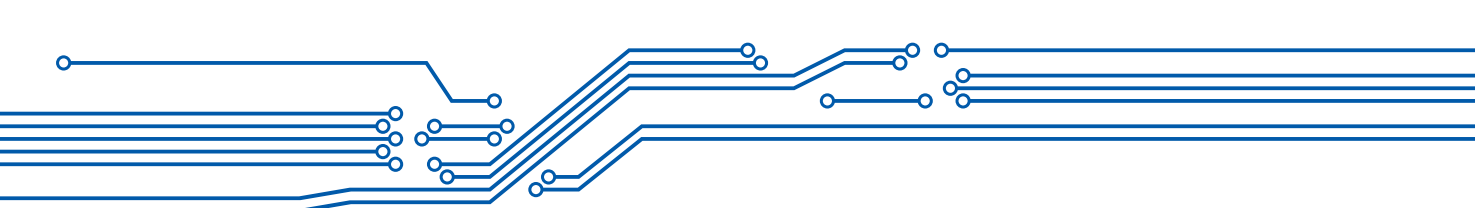
Type (Veri Türü)

Sütunlar farklı türde veri içerebilir. Örneğin bazı tür bilgiler sadece numerik değer (10250) içerirken bazı tür bilgiler ise sadece metin (Gazi Mustafa Kemal Atatürk) içerebilir. Bununla birlikte tarih (12.12.2017) de bir veri türüdür. Bu yüzden sütunlarda ne tür bilgi tutulacağını önceden tanımlanması gerekir. MySQL ya da MariaDB’de bir sütun için tanımlanabilecek birçok veri türü vardır. Şu an Numerik, String ve Tarih-Saat türleri üzerinde durulacaktır.

1. Numerik Türler

Bu türler tam sayısal veri türlerini ve ondalık sayı veri türlerini içerir.

a. BOOLEAN: Doğru ya da yanlış bir durumu belirten, sadece 0 ya da 1 numerik değerlerini alabilen bir veri türüdür. Sıfır değeri yanlış kabul edilir. 1 ise doğru sayılır. Örneğin

- 
- i. Öğrenciye tablet dağıtıldı (1) ya da dağıtılmadı (0).
 - ii. Öğrencinin evinde bilgisayar var (1) ya da yok (0).
 - iii. Öğrenci dersi geçti (1) ya da geçmedi (0).

b. TINYINT: Çok küçük tamsayı değerlerini ifade eder. Eğer negatif değerler kullanılacaksa -128 ile 127 arasındaki değerleri, sadece pozitif değerler kullanılacaksa 0 - 255 arasındaki değerleri kapsar. Örneğin yaş için, not için kullanılabilir.

c. SMALLINT: Eğer negatif değerler kullanılacaksa -32768 ile 32767 arasındaki değerleri, sadece pozitif değerler kullanılacaksa 0 - 65535 arasındaki değerleri kapsar. Örneğin yaş için, not için kullanılabilir.

d. INT: Eğer negatif değerler kullanılacaksa -2147483648 ile 2147483647 arasındaki değerleri, sadece pozitif değerler kullanılacaksa 0 - 4294967295 arasındaki değerleri kapsar. Eğer ihtiyaç duyulan veri bu aralık dışında ise BIGINT kullanılabilir.

e. DECIMAL: Ondalık sayıların istenen formatta biçimlendirmesini sağlayan veri türüdür. Örneğin DECIMAL (10, 2) şeklindeki bir veri tanımlaması toplam 10 numerik değer ve 2 ondalık basamak içeren bir veri türünü işaret eder. Örneğin parasal işlemlerde kullanılabilir. En yüksek 1 000 000 00 TL iş hacmine sahip olan bir şirketi düşünelim. Buna göre DECIMAL (9,2) veri tanımlaması uygundur. DECIMAL için en yüksek basamak sayısı 65'tir. Desteklenen en yüksek ondalık sayıların sayısı (D) 30'dur. Dolayısı ile en yüksek DECIMAL (65, 30) tanımlaması yapılabilir. Bu tanımlamanın yeterli olmadığı durumlarda, INT' da olduğu gibi daha geniş aralıklı veri türleri kullanılabilir. Bunlar FLOAT ve DOUBLE' dir. Bu veri türlerinin tanımlanması da benzer şekildedir FLOAT(x, y) ya da DOUBLE (x,y).

2. String Türler

String; içerisinde harf, rakam, simge barındırabilir. Kitabın Javascript bölümünde, dizi sıralama işleminde numerik değerler içerisinde kullanılan rakamlarla, string değişken içerisinde kullanılan rakamlar arasındaki farklılıktan bahsedilmişti. Bununla birlikte string ve numerik verilerin bir ayrımını da ha ortaya koymamız gerekiyor. Örneğin 150 (string) ve 200 (int) verisini içeren iki veri birbiri ile toplanamaz. Çünkü string sayısal bir değişkeni ifade etmez. 150, 1, 5 ve 0'ın yan yana gelmesinden oluşan bir metni ifade eder. Bu doğrultuda, veri tabanı yönetimi için de veri türlerini tanımlarken bu durumlar göz önünde bulundurulmalıdır. Veri tabanı içerisinde string bir veri tanımlamanın yolları:

a. VARCHAR (X): En yüksek 255 karakter içerisinde barındırabilen bir string veri tanımlar. X yerine gireceğimiz rakamla bu 255 karakter düşürülebilir. Örneğin VARCHAR(1) sadece içerisinde 1 karakter barındırabilir. Bu karakter harf, rakam ya da bir simge olabilir.

b. TEXT (x): En yüksek 65,535 karakter içerisinde barındırabilen bir string veri tanımlar.

3. Tarih ve Saat Veri Türleri

a. DATETIME: Tarih ve Saati birlikte tutan veri türüdür. Desteklenen aralık '1000-01-01 00: 00: 00.000000' ile '9999-12-31 23: 59: 59.999999' arasındadır. MySQL ve türevi MariaDB, 'YYYY-AA-GG HH: MM: SS [.fraction]' formatında DATETIME değerlerini görüntüler.

b. DATE: Sadece tarihi tutan veri türüdür. Desteklenen aralık '1000-01-01' ile '9999-12-31' arasındadır. MySQL ve türevi MariaDB, 'YYYY-AA-GG' biçiminde TARİH değerlerini görüntüler. Bu format değiştirilebilir.

c. TIME: Sadece saati tutan veri türüdür. Aralığı '-838: 59: 59.000000' ile '838: 59: 59.000000' arasındadır. MySQL, TIME değerlerini 'HH: MM: SS [.fraction]' formatında görüntüler ancak dizelerin veya sayıların kullanılmasıyla TIME sütunlarına değer atmasına izin verir.

d. YEAR: Dört haneli bir biçimde bir yılı tutan veri türüdür. MySQL ve türevi MariaDB yıl değerlerini YYYY biçiminde görüntüler. Değerler 1901 - 2155 ve 0000 olarak görüntülenir.

Lenght/Values

- ✓ Tanımlanan numerik veri türleri için toplam digit veya ondalık değerlerin belirtilmesi
- ✓ String veri türleri için toplam karakter uzunluğunun belirtilmesi

Name	Type	Length/Values
ogrnno	VARCHAR	8
yas	TINYINT	2
ortalama	DECIMAL	4,2

Default

Veri tanımlaması yapıldıktan sonra bazı sütunlara herhangi bir değer girilmemesi durumunda varsayılan olarak bu sütunlarda hangi değerın görüntüleneceğinin ayarlanmasını sağlar. None, As defined, NULL ve CURRENT_TIMESTAMP seçenekleri vardır. Eğer varsayılan olarak

- ✓ Kendinizin tanımlandığı bir değerın görüntülenmesini istiyorsanız, "As defined",
- ✓ Daha önceden hiçbir değer girilip ya da silinmediğini göstermek için "NULL",
- ✓ Sadece tarih zaman türleri için eğer bir değer girilmedi ise o anın güncel tarih ve saatinin otomatik girilmesini istiyorsanız CURRENT_TIMESTAMP seçeneklerini kullanabiliriz.

Name	Type	Length/Values	Default
ogrnno	VARCHAR	8	None
yas	TINYINT	2	As defined: 6
ortalama	DECIMAL	4,2	NULL
tarihsaat	DATETIME		CURRENT_TIME

Collation

Veri tabanını oluştururken dil desteği “utf8-turkish-ci” olarak belirtilmişti. Fakat sütunlara ayrı ayrı dil desteği atamanız da mümkündür. Örneğin adres tutan bir string veri kümesi için dil desteğini “utf8-unicode-ci” seçebilirsiniz. Böylece birden fazla dil desteği sağlamış olursunuz.

Name	Type	Length/Values	Default	Collation
adres	TEXT	255	NULL	utf8_unicode_ci

Attributes

Veri tanımlaması yaparken integer(int) negatif değer alınabilir. Sadece pozitif değerli veri girişi yapılması isteniyorsa Attributes özelliği “UNSIGNED” olarak değiştirilmelidir.

Name	Type	Length/Values	Default	Collation	Attributes
ortalama	DECIMAL	4,2	NULL		UNSIGNED

NULL

Eğer sütuna veri girişi zorunlu tutulmak istenilmiyorsa null seçeneğine tik atılması gerekir. Böylece ilk veri girişinde sütun doldurulmadan boş çözümlenebilir.

Name	Type	Length/Values	Default	Collation	Attributes	Null
ortalama	DECIMAL	4,2	NULL		UNSIGNED	<input checked="" type="checkbox"/>

Bir veri tabanı içerisinde tablo oluştururken önemli olan 7 özellik ve veri türleri görüldü. Okul veri tabanı içerisinde öğrenci varlığı ile ilgili oluşturulacak tablo şekildeki gibidir.

Name	Type	Length/Values	Default	Collation	Attributes	Null
OgrNo	VARCHAR	8	None	utf8_turkish_ci		<input type="checkbox"/>
Pick from Central Columns						
Adi	VARCHAR	50	NULL	utf8_unicode_ci		<input checked="" type="checkbox"/>
Pick from Central Columns						
Soyadi	VARCHAR	50	NULL	utf8mb4_unicode		<input checked="" type="checkbox"/>
Pick from Central Columns						
Yasi	TINYINT	2	As defined: 6		UNSIGNED	<input type="checkbox"/>
Pick from Central Columns						
DogumTarihi	DATE		NULL			<input checked="" type="checkbox"/>
Pick from Central Columns						
Adresi	TEXT		NULL	utf8_unicode_ci		<input checked="" type="checkbox"/>
Pick from Central Columns						
Dersi	VARCHAR	50	NULL	utf8_turkish_ci		<input checked="" type="checkbox"/>
Pick from Central Columns						
Yaz1	TINYINT	3	NULL		UNSIGNED	<input checked="" type="checkbox"/>
Pick from Central Columns						
Yaz2	TINYINT	3	NULL		UNSIGNED	<input checked="" type="checkbox"/>
Pick from Central Columns						
Yaz3	TINYINT	3	NULL		UNSIGNED	<input checked="" type="checkbox"/>
Pick from Central Columns						
Ortalama	DECIMAL	5,2	As defined: 0.00			<input type="checkbox"/>
Pick from Central Columns						
Durum	TINYINT	1	NULL			<input checked="" type="checkbox"/>
Pick from Central Columns						

Tablonun sütunlarını resimdeki gibi ayarladıktan sonra Save (Kaydet) butonu tabloya kaydedilir. Eğer tabloya yeni sütun eklemek istenirse öğrenci tablosu seçiliyken Structure sekmesine tıklanır. Karşınıza daha önce oluşturmuş olduğunuz sütunlar gelir. Bu ekranda Add bölümünü kullanarak istediğiniz sayıda ve var olan sütunlar arasında istediğiniz bir yere yeni sütun ya da sütunlar ekleyebilirsiniz.

#	Name	Type	Collation	Attributes	Null	Default
<input type="checkbox"/>	1	OgrNo	varchar(8)		No	None
<input type="checkbox"/>	2	Adi	varchar(50)		Yes	NULL
<input type="checkbox"/>	3	Soyadi	varchar(50)		Yes	NULL
<input type="checkbox"/>	4	Yasi	tinyint(2)	UNSIGNED	No	6
<input type="checkbox"/>	5	DogumTarihi	date		Yes	NULL
<input type="checkbox"/>	6	Adresi	text		Yes	NULL
<input type="checkbox"/>	7	Dersi	varchar(50)		No	None
<input type="checkbox"/>	8	Yaz1	tinyint(3)	UNSIGNED	Yes	NULL
<input type="checkbox"/>	9	Yaz2	tinyint(3)	UNSIGNED	Yes	NULL
<input type="checkbox"/>	10	Yaz3	tinyint(3)	UNSIGNED	Yes	NULL
<input type="checkbox"/>	11	Ortalama	decimal(5,2)		No	0.00
<input type="checkbox"/>	12	Durum	tinyint(1)		Yes	NULL

Check all With selected:

column(s)

Bir öğrencinin birden çok dersi olabilir. Fakat aynı öğrencinin veri tabanı içerisinde aynı iki dersi içerecek şekilde kaydı olmamalıdır. Çünkü veri tabanı düzenli, tekrar etmeyecek şekilde verilerin tutulmasını sağlamak için kullanılır. Dolayısı ile bu tablo üzerinde aynı öğrenci ve aynı ders olacak şekilde tekrar eden bir kayıt istenmez. Bu durumda birincil anahtar (primary) eklememiz gerekir.

Birincil Anahtar (Primary)

Tablo içerisinde tekrar etmesini istemediğiniz kayıtlar için kullanılabilir. Örneğin OgrNo sütununa birincil anahtar eklediğinizde, tüm tablo içerisinde öğrenci numarası tekrar edemez. Fakat tabloda, OgrNo ve Dersi sütunlarının aynı olması durumunda kayıt girişlerinin yapılamaması istenir. Bu durumda, her iki sütunun seçilerek tablonun altındaki Primary butonuna tıklanması gerekmektedir.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input checked="" type="checkbox"/>	1	OgrNo				No	None	Change
<input type="checkbox"/>	2	Adi			Yes	NULL		Change
<input type="checkbox"/>	3	Soyadi			Yes	NULL		Change
<input type="checkbox"/>	4	Yasi		UNSIGNED	No	6		Change
<input type="checkbox"/>	5	DogumTarihi			Yes	NULL		Change
<input type="checkbox"/>	6	Adresi			Yes	NULL		Change
<input checked="" type="checkbox"/>	7	Dersi			No	None		Change
<input type="checkbox"/>	8	Yaz1		UNSIGNED	Yes	NULL		Change
<input type="checkbox"/>	9	Yaz2		UNSIGNED	Yes	NULL		Change
<input type="checkbox"/>	10	Yaz3		UNSIGNED	Yes	NULL		Change
<input type="checkbox"/>	11	Ortalama			No	0.00		Change
<input type="checkbox"/>	12	Durum			Yes	NULL		Change

Check all *With selected:* Browse Change Drop Primary

Böylece şekildeki gibi aynı OgrNo' lu öğrencinin farklı derslerle ilgili kaydı tutulurken bir sonraki kayıta eğer yine 01012323 OgrNo'lu ve matematik dersini içeren bir kayıt girilmesine izin verilmeyecektir.

OgrNo	Adi	Soyadi	Yasi	DogumTarihi	Adresi	Dersi	Yaz1	Yaz2	Yaz3	Ortalama	Durum
01012323	Mete	Yıldırım	6	2011-01-23	Park Eymir Toki	Matematik	90	80	100	90.00	1
01012323	Mete	Yıldırım	6	2011-01-23	Park Eymir Toki	Türkçe	85	95	95	91.66	1

Kodlayıcı Görevi

- Structure Sekmesi altındaki diğer butonların ne işe yaradığını test ediniz.
- Okul veri tabanı içerisinde, öğretmen varlığı ile ilgili bir tablo oluşturunuz.

Kayıt İşlemleri

Oluşturmuş olduğunuz tabloların içerisindeki kayıtlarla ilgili Seçme, Ekleme, Güncelleme ve Silme işlemleri yapılabilir. Bu işlemler yapılırken iki yöntem kullanılabilir:

Arayüz (phpMyAdmin) Kullanımı

Kayıt Ekleme

Kayıt girmek istediğiniz tabloyu seçtikten sonra, Insert Sekmesine tıklayınız. Daha sonra alanlara değerleri yazınız ve Go butonuna basınız.

Column	Type	Function	Null	Value
OgrNo	varchar(8)			01012424
Adi	varchar(50)		<input type="checkbox"/>	Selvi
Soyadi	varchar(50)		<input type="checkbox"/>	Güzel
Yasi	tinyint(2) unsigned			15
DogumTarihi	date		<input checked="" type="checkbox"/>	
Adresi	text		<input checked="" type="checkbox"/>	
Dersi	varchar(50)			Matematik
Yaz1	tinyint(3) unsigned		<input type="checkbox"/>	50
Yaz2	tinyint(3) unsigned		<input type="checkbox"/>	60
Yaz3	tinyint(3) unsigned		<input type="checkbox"/>	90
Ortalama	decimal(5,2)			66.66
Durum	tinyint(1)		<input type="checkbox"/>	

Go

Kodlayıcı Görevi

- Oluşturmuş olduğunuz Okul veri tabanı içerisindeki, ogrenci ve ogretmen tabloları içerisine en az 10' ar kayıt giriniz.

Kayıt Seçme

Herhangi bir tabloyu (ör: ogrenci) seçtiğinizde, tablonun içerisinde var olan kayıtlar “Number of rows” (Kayıt Sayısı) da belirtilen sayı kadar (Ör: 25 kayıt) sayfalar hâlinde getirilecektir. Eğer isterseniz “Filter rows” (kayıtları filtreleme) den istenen metin, tarih ya da sayıyı içeren kayıtları getirebilirsiniz. Örneğin şekilde Selvi metnini içeren tüm kayıtlar getirildi. Burada dikkat edilmesi gereken bir nokta, filtreleme bölümüne girmiş olduğunuz anahtar kelimenin tüm sütunlarda aranmasıdır. Anahtar kelime “100” olarak girilmiş olsaydı, yazılı notlarından herhangi biri 100 olan kayıtların hepsini getirilebilecekti.

Show all | Number of rows: 25 | Filter rows: Selvi

Sort by key: None

+ Options

	OgrNo	Adi	Soyadi	Yasi	DogumTarihi	Adresi	Dersi	Yaz1	Yaz2	Yaz3	Ortalama	Durum
<input type="checkbox"/>	01012424	Selvi	Güzel	15	2002-04-21	Güçlükaya mah. Fizik		75	85	100	86.66	1
<input type="checkbox"/>	01012424	Selvi	Güzel	15	2002-04-21	Güçlükaya Mah. Kimya		97	100	87	94.66	NULL
<input type="checkbox"/>	01012424	Selvi	Güzel	15	NULL	NULL Matematik		50	60	90	66.66	0

Kayıt Güncelleme

Herhangi bir kaydı güncellemek istediğinizde, kaydın solundaki Edit butonuna tıkladıktan sonra kayıt eklemede yaptığınız gibi yeni değer girebilir ve güncelleme işlemi yapabilirsiniz.

		OgrNo	Adi	Soyadi	Yasi	DogumTarihi	Adresi
<input type="checkbox"/>	Edit Copy Delete	01012424	Selvi	Güzel	15	2002-04-21	Güçlükaya Mah.
<input checked="" type="checkbox"/>	Edit Copy Delete	01012424	Selvi	Güzel	15	NULL	NULL

Check all With selected: Edit Copy Delete Export

Kayıt Silme

Herhangi bir kaydı silmek istediğinizde, kaydın solundaki Delete butonuna tıkladıktan sonra karşınıza gelen ekrandaki uyarıyı onaylayarak silme işlemi yapabilirsiniz.

		OgrNo	Adi	Soyadi
<input type="checkbox"/>	Edit Copy Delete	01012323	Mete	Yıldırım
<input type="checkbox"/>	Edit Copy Delete	01012323	Mete	Yıldırım
<input type="checkbox"/>	Edit Copy Delete	01012424	Selvi	Güzel
<input type="checkbox"/>	Edit Copy Delete	01012424	Selvi	Güzel
<input checked="" type="checkbox"/>	Edit Copy Delete	01012424	Selvi	Güzel

Confirm

Do you really want to execute "DELETE FROM `ogrenci` WHERE `ogrenci`.`OgrNo` = '01012424' AND `ogrenci`.`Dersi` = 'Matematik'?"

OK Cancel

SQL Kodları

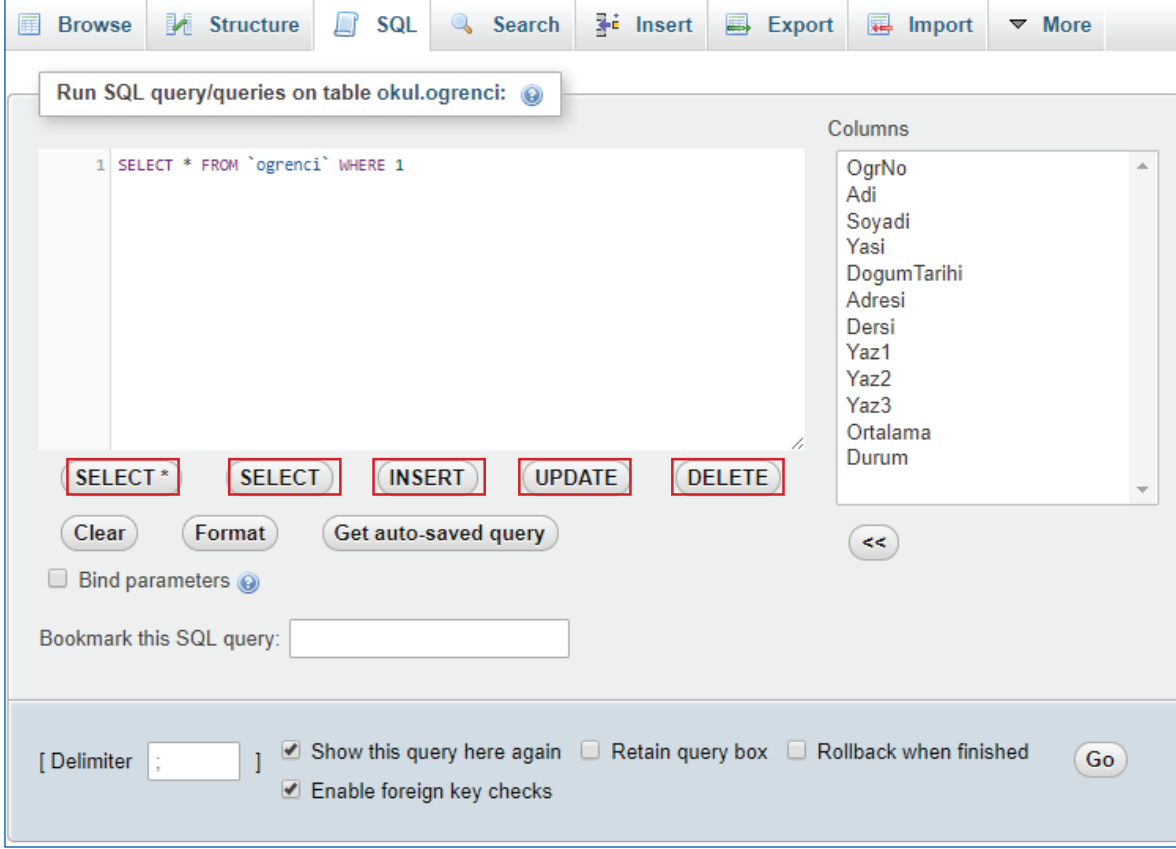
phpMyAdmin kullanıcılara kod yazımı olmadan kolay bir veri yönetimi sunar. Peki, arka planda çalışan kod nasıldır?

Veri tabanı yönetimi aslında SQL kodları ile yapılır. phpMyAdmin panelini kullanarak yapmış olduğumuz veri ekleme, seçme, güncelleme ve silme işlemleri için sql kodlarının nasıl çalıştığını kavrayıp uygulamamız gerekir? Çünkü, bir sonraki bölümde bu kodlara ihtiyaç duyacağız.

Web programlama yaparken web sitesi kaynakları ile veri tabanı yönetimi birbirinden ayrılır. Web sitesi programlarken bizim kaynak dosyalarımız ve veri tabanı arasında ilişki kuracak kodlara ihtiyaç vardır. Örneğin web siteleri sizden iletişim için mail adresinizi, isminizi girmenizi ister ve daha sonra bunu veri tabanlarına yazar. Böylece sizle ilgili iletişim bilgilerini tutmuş olur. Gerektiğinde size bu veri tabanındaki bilgileri kullanarak mail atar ya da bir sosyal ağda yazmış, silmiş yada güncellemiş olduğunuz yorumlarda da aynı tür işlemler yapılır. İşte web tabanlı programlamada bu tür işlemler (veri ekleme, veri getirme, veri güncelleme, veri silme) için phpMyAdmin panelinin arka planda çalıştırdığı SQL kodları kullanılır.

Phpmyadmin panelinde bu kodları çalıştırarak nasıl yazıldıklarını kavramaya çalışılacak. Bir sonraki ünite, bu kodları yazarak web sayfası üzerinden veri ekleme, seçme, güncelleme ve silme işlemlerini yapılabilecek.

Öğrenci tablosu seçili iken SQL sekmesine tıklayınız. Karşınıza bir tablo üzerinde yapabileceğiniz, veri ekleme, seçme, güncelleme ve silme işlemlerini sql kodları ile yapabileceğiniz ekran geliyor. Burada SELECT, INSERT, UPDATE ve DELETE' e ayrı ayrı tıklayınız ve ekranda değişen kodları inceleyiniz.



The screenshot shows a SQL query editor interface. At the top, there is a menu bar with options: Browse, Structure, SQL, Search, Insert, Export, Import, and More. Below the menu bar, there is a title bar that says "Run SQL query/queries on table okul.ogrenci:". The main area is divided into two parts. On the left, there is a text area containing the SQL query: "1 SELECT * FROM `ogrenci` WHERE 1". On the right, there is a list of columns: OgrNo, Adi, Soyadi, Yasi, DogumTarihi, Adresi, Dersi, Yaz1, Yaz2, Yaz3, Ortalama, and Durum. Below the text area, there are several buttons: "SELECT *", "SELECT", "INSERT", "UPDATE", and "DELETE". There are also buttons for "Clear", "Format", and "Get auto-saved query". A checkbox labeled "Bind parameters" is checked. Below that, there is a text input field for "Bookmark this SQL query:". At the bottom, there is a footer area with a "Delimiter" field set to ";", a "Go" button, and several checkboxes: "Show this query here again" (checked), "Retain query box" (unchecked), "Rollback when finished" (unchecked), and "Enable foreign key checks" (checked).

SELECT

Veri seçimi için kullanılan SQL komutudur.

```
SELECT sutun1, sutun2, ...  
FROM tablo_adi  
WHERE koşul  
ORDER BY sutun1, sutun2, ... ASC|DESC;
```

Bir tablo içerisinde veri seçimi nasıl olabilir biraz üzerinde düşününüz.

I. Tablodaki tüm sütunları içerecek şekilde tüm kayıtları seçebilirsiniz.

```
SELECT * FROM tablo_adi
```

II. Tablodaki bazı sütunları içerecek şekilde tüm kayıtları seçebilirsiniz.

```
SELECT sutun1, sutun2, sutun3 FROM tablo_adi
```

III. Tablodaki tüm sütunları içerecek şekilde bazı kayıtları seçebilirsiniz.

```
SELECT * FROM tablo_adi WHERE sutun1 = 'filtre'
SELECT * FROM tablo_adi WHERE sutun1 = 'filtre' AND sutun2 = 'filtre'
SELECT * FROM tablo_adi WHERE sutun1 = 'filtre' OR sutun2 = 'filtre'
```

IV. Tablodaki bazı sütunları içerecek şekilde bazı kayıtları seçebilirsiniz.

```
SELECT sutun1, sutun2, sutun3 FROM tablo_adi WHERE sutun1 = 'filtre'
```

V. Seçilen kayıtları istenen bir sütuna göre sıralayabiliriz.

```
SELECT * FROM tablo_adi ORDER BY sutun1 ASC           /* A dan Z ye */
SELECT * FROM tablo_adi ORDER BY sutun1 DESC        /* Z den A ya */
SELECT sutun1, sutun2, sutun3 FROM tablo_adi ORDER BY sutun1 DESC
SELECT * FROM tablo_adi WHERE sutun1 = 'filtre' ORDER BY sutun1 ASC
SELECT * FROM tablo_adi WHERE sutun1 = 'filtre' OR sutun2 = 'filtre' ORDER BY sutun1 ASC, sutun2 DESC
```

INSERT

Veri ekleme için kullanılan SQL komutudur.

```
INSERT INTO tablo_adi (sutun1, sutun2, sutun3, ...)
VALUES (sutun1_deger, sutun2_deger, sutun3_deger, ...);
```

Bir tablo içerisinden veri ekleme nasıl olabilir.

I. Tüm sütunlara değerlerinin girilerek kayıt eklenmesi

```
INSERT INTO tablo_adi VALUES ('sutun1', 'sutun2', 'sutun3', ...)
```

*/*Tüm sütunların değerleri girilir.*/**

```
INSERT INTO ogrenci VALUES ('01012525','Sezgin','Ardıç','10','2007-05-20','Kalaba Mah.','Müzik','50','75','100','75.00','1')
```

II. Sadece istenen sütunlara değerlerinin girilerek kayıt eklenmesi

```
INSERT INTO tablo_adi ('sutun1', 'sutun2', 'sutun3' ...)
```

*/*Null izni olmayan sütunlarla birlikte kayıt eklenmek istenen sütunlar yazılır.*/**

```
VALUES ('sutun1_deger', 'sutun2_deger', 'sutun3_deger' ...)
```

*/*İsmi yazılan sütunların değerleri girilir.*/**

```
INSERT INTO ogrenci (OgrNo, Dersi) VALUES ('01012626','Beden Eğitimi')
```

UPDATE

Veri güncelleme için kullanılan SQL komutudur.

```
UPDATE tablo_adi
SET sutun1 = sutun1_yenideger, sutun2 = sutun2_yenideger, ...
WHERE kosul;
```

```
UPDATE ogrenci SET Adi='Deniz Mete' WHERE OgrNo='01012323';
```

/ Öğrenci numarası 01012323 olan tüm kayıtlardaki öğrenci ismini Deniz Mete olarak değiştirir */*

```
UPDATE ogrenci SET Durum='0' WHERE OgrNo='01012323' AND Ortalama<'70.00';
```

/ Öğrenci numarası 01012323 ve ortalama puanı 70.00 altında olan tüm kayıtlardaki öğrencinin geçti kaldı durumunu 0 (kaldı) olarak değiştirir. */*

```
UPDATE ogrenci SET Adres='Gölbasi Mah. Oğuzlar ÇORUM';
```

/ Koşulsuz bir update kodu (Where olmadan) yazarsanız, tablonuz içerisindeki bütün kayıtların, belirtmiş olduğunuz sütun yada sütunların (Ör: Adres) değeri aynı olur. Update kodunu bu şekilde kullanırken çok dikkatli olmalısınız!*/*

DELETE

Veri silme için kullanılan SQL komutudur.

```
DELETE * FROM tablo_adi
WHERE kosul;
```

```
DELETE * FROM ogrenci WHERE OgrNo='01012323';
```

/ Öğrenci numarası 01012323 olan tüm kayıtları tablodan siler*/*

```
DELETE * FROM ogrenci WHERE OgrNo='01012323' OR OgrNo='01012424';
```

/ Öğrenci numarası 01012323 veya 01012424 olan tüm kayıtları tablodan siler*/*

```
DELETE * FROM ogrenci WHERE OgrNo='01012323' AND Ders='Matematik';
```

/ Öğrenci numarası 01012323 ve dersi Matematik olan kaydı tablodan siler*

Tek kayıt olacağını nasıl bilebiliyoruz? İpucu: Primary (Birincil Anahtar)/*

```
DELETE * FROM ogrenci
```

/ Koşulsuz bir delete kodu (Where olmadan) yazarsanız, tablonuz içerisindeki bütün kayıtları silersiniz. Delete kodunu bu şekilde kullanırken çok dikkatli olmalısınız!*/*

Kodlayıcı Görevi

- En az 10 kayıt girişi yapmış olduğunuz öğretmen tablosu üzerinde insert, select, update ve delete sql cümleciklerini kullanarak, veri ekleme, veri seçme, veri güncelleme ve veri silme işlemlerini yapınız.

6. ETKİLEŞİM VE VERİ YÖNETİMİ

6. ETKİLEŞİM VE VERİ YÖNETİMİ

Etkileşim denildiğinde, web programlama ile ilgili Javascript kodlama üzerinde durulmuştu. Veri yönetimi denildiğinde ise sunucu üzerinde çalışan (sunucu tabanlı) araçlar içerisinde Mysql' in türevi olan MariaDB yönetimi ve sql kodlama üzerinde durmuştuk.

Kodlayıcı Görevi

- Javascript ile veri yönetimi yapılabilir mi? Daha açık bir ifadeyle Javascript kullanarak Mysql ya da MariaDB içerisindeki veriler üzerinde ekleme, seçme, güncelleme ve silme işlemleri yapılabilir mi? Neden?

Web programlama yaparken web sitesi kaynakları ile veri tabanı yönetiminin birbirinden ayrıldığını hatırlayınız. Web sitemiz şu ana kadar öğrendiğiniz kodlar üzerinden hareketle HTML, CSS ve Javascript kodlarının bir birleşiminden meydana geliyordu. Bunun dışında veri yönetimi için phpMyAdmin paneli ile tanıtılmıştı. Fakat bu panel site kodlarından ayrılıyordu. Bu panel ile site kodlarını konuşurmak içinse sql kodlarının kullanılma gereksiniminden bahsedilmişti.

HTML, CSS ve Javascript kodları veri yönetimi yapmak yani sql kodlarını çalıştırmak için yeterli değildir. Bunun için sunucu tabanlı çalışan bir başka kod yapısına ihtiyaç vardır. Bu ihtiyacı PHP, ASP, CSharp vb. programlama dilleri karşılayabilir. Biz bu ünite de PHP üzerinde durulacak.

PHP (HYPERTEXT PREPROCESSOR)

PHP dinamik ve etkileşimli web sayfaları yapmak için sunucu tabanlı bir kodlama dilidir. "Hypertext Preprocessor" kelimelerinin kısaltılmasıdır. Yaygın olarak kullanılan, açık kaynak kodlama dilidir.

PHP dosyaları

- Metin, HTML, CSS, JavaScript ve PHP kodu içerebilir.
- PHP kodu sunucuda yürütülür ve sonuç tarayıcıya düz HTML olarak döndürülür.
- PHP dosyalarının uzantısı ".php" dir.

PHP,

- Dinamik sayfa içeriği üretebilir.
- Sunucudaki dosyaları oluşturabilir, açabilir, okuyabilir, yazabilir, silebilir ve kapatabilir.
- Form verilerini toplayabilir.
- Çerezleri gönderebilir ve alabilir.
- Veri tabanınızdaki verileri ekleyebilir, silebilir, verileri değiştirebilir.
- Veri şifreleyebilir.

PHP kodlarını tarayıcılar HTML, CSS ya da Javascriptte olduğu gibi tek başına çalıştıramaz. Bu yüzden PHP kodlarını yorumlayabilecek bir ara birime ihtiyaç vardır. Hatırlarsanız veri yönetimi için bir ara birim kurulumu (XAMPP) yapılmıştı. Bu ara birim, içerisinde hem veri yönetimi yapabileceğimiz phpMyAdmin panelini hem de PHP kodlarını çalıştırabileceğimiz Apache sunucusunu içerisinde barındırmaktadır. Eğer XAMPP kurulumunu bilgisayara yaptıysanız yeni bir kurulum yapmanıza gereksinim yoktur.

Php kodlama yaparken site kaynak dosyalarınızı, XAMPP kurulumunu yapmış olduğunuz dizin içerisinde (Varsayılan olarak C:\xampp\htdocs) saklamanız gerekir. Böylece Apache tarafından php kodlarınız yorumlanabilir. Basit bir örnek uygulama üzerinde testini yapalım. Notpad++ üzerinde aşağıdaki kodu yazın ve dosya uzantısı .php olacak şekilde kaydediniz.

```

<!DOCTYPE html>
<html>
<style>
    body    {font-family:Verdana;font-size:14px;}
    .baslik {padding:5px; background-color:green;}
</style>
<body>
<p class="baslik"> Örnek 1 <p>

<?php // Bir PHP kodu <?php ile başlar.

    // php kodları büyük küçük harfle yazılabilir.
    ECHO "Adınız ... <br>";
    echo "Soyadınız ... <br>";
    Echo "Düşünceleriniz ...<br>";
    // echo sayfaya yazdırma işlemi için kullanılabilir.

?> <!-- Bir PHP kodu ile biter. -->

</body>
</html>

```

Daha sonra bu dosyayı C:\xampp\htdocs dizinine kopyalayınız. Aynı şekilde htdocs altında yeni bir klasör (Ör: phpsite) oluşturup bu klasör içerisine de kaydedebilirsiniz. Şimdi sıra tarayıcıdan bu dizine erişmeye geldi. <http://localhost/test.php> ya da <http://localhost/phpsite/test.php> adreslerini kullanarak ilk php örneğimizin çıktısına erişilebilir.

Örnek 1

Adınız ...
Soyadınız ...
Düşünceleriniz ...

Php kodlarını nasıl çalıştırabileceğinizi gördük şimdi php üzerinde daha detaylı duracağız.

DEĞİŞKENLER

PHP'de bir değişken tanımlanırken, \$ işaretiyle başlanır ve sonrasında değişken adı yazılır.

- Değişken adı ise, kısa (x ve y gibi) veya daha açıklayıcı (yaş, arabaadi, toplam) olabilir.
- Değişken adı bir harf veya altçizgi karakteriyle başlar. Bir sayı ile başlayamaz.
- Değişken adı yalnızca alfasayısal karakterler (A-z, 0-9) ve altçizgi (_) içerebilir.
- Değişken adları büyük küçük harflere duyarlıdır. (\$yas ve \$YAS iki farklı değişkendir)

Şimdi bu ölçütlere göre tanımlanmış bazı değişkenleri bir uygulama üzerinde inceleyiniz.

```
<html>
<style>
  body {font-family:Verdana;font-size:14px;}
  .baslik {padding:5px; background-color:green;}
  vurgu { font-weight:bold;}
</style>
<body>
<p class="baslik"> Örnek 2 <p>
<?php

//php' de deęişkenin türünü (int, string, decimal vs.) belirtmeye gerek yoktur.
$ad = "Deniz Mete";
$soyad = "YILDIRIM";
$yas= 2;
$agirlik= 12.20;
$boy1 = 80.50;

echo "$ad $soyad' ın $yas yaşındaki fiziksel özellikleri <br/>";
echo "-----<br/>";
echo "Kilosu: <vurgu> $agirlik </vurgu> <br/> Boyu: <vurgu> $boy1 </vurgu> ";

?>
</body>
</html>
```

Örnek 2

Deniz Mete YILDIRIM' ın 2 yaşındaki fiziksel özellikleri

Kilosu: **12.2**
Boyu: **80.5**

Bu örnekte herhangi bir etkileşim olmadığını fark etmiş olmalısınız. Web tarayıcısını açtığınızda doğrudan yukarıdaki ekranla karşılaştınız. Etkileşimli bir örnek tasarlayınız. Örneğin site kullanıcılarından kendi doğum tarihini girmesini isteyiniz ve buna göre kullanıcının yaşını hesaplayınız. Bu ve benzer örnekler için kodlamaya geçmeden önce PHP Form Kullanımı ile ilgili bilgi sahibi olmanız gerekiyor.

FORM KULLANIMI

Daha önceden PHP nin sunucu tabanlı çalıştığını dile getirmiştik. Bunun anlamı, kullanıcı herhangi bir eyleme geçtiğinde diğer bir ifadeyle sayfa ile etkileşim kurduğunda (örneğin veri girdi ve butona bastı) her seferinde öncelikle sunucuya bizim bilgisayarımızdan bir istek gönderilir daha sonra kodlar sunucuda yorumlanır, HTML'ye dönüştürülür ve son olarak bizim bilgisayarımıza geri gönderilir. Bilgisayarımızdan sunucuya veri gönderilirken daha önceden de görmüş olduğumuz form elementleri kullanılır. Form elementlerine girilen değerler ise POST ya da GET yöntemleri ile taşınır. İlk olarak POST yöntemini ele alınız.

POST YÖNTEMİ

```

<!-- test.php sayfasındayız. -->
<p class="baslik"> Örnek 3 <p>

<form action="test.php" method="post">

<!-- action:      formdaki butona tıklandıktan sonra,
                  hangi sayfaya yönlendirmek istiyorsak
                  o sayfanın dosya ismi ve uzantısını yazıyoruz.
                  Bu örneğimizde, aynı sayfaya (test.php) yönlendirme
                  yapacağız.
                  Başka bir sayfaya da yönlendirme yapabiliriz.

                  method:   verilerin hangi yöntemle taşınacak (POST ya da GET)
-->
İsim: <input type="text" name="ad"><br>
E-posta: <input type="text" name="eposta"><br>
<input type="submit">
</form>

```

```

<!-- Butona tıklandıktan sonraki işlemler:
Veriler POST yöntemi ile sunucuya aktarılır.
Şimdi bu verileri kullanabilmek için gerekli php kodunu yazalım.
-->
<?php
//ilk olarak istediğimiz veriler gönderilmiş mi kontrol ediyoruz.

// isset: veri var mı?
// $_POST[".."]: post edilen verileri ulaşma
if ( isset($_POST["ad"]) && isset($_POST["eposta"])){
    // name' i ad ve eposta olan post edilmiş veri var olduğunda
    yapılacaklar...

    echo "Sistem <vurgu>" . $_POST["ad"] . "</vurgu> adlı kişiye
    <vurgu>" . $_POST["eposta"] . "</vurgu> eposta adresini kullanarak
    mail atacak." ;
    // php de . işareti kullanılarak string ifadeler ve değişkenler
    birleştirilebilir.
} else {
    // name' i ad ve eposta olan post edilmiş veri var olmadığında
    yapılacaklar...
    echo "<p style=background-color:grey;margin:10px;padding:5px;>Sayfa
    POST edilmedi. Bir diğer ifadeyle methodu post olan bir form ögesi
    içerisindeki butona tıklanmadı.</p>";
}
?>

```

Sayfa İlk Açıldığında

Örnek 3

İsim:

E-posta:

Sayfa POST edilmedi. Bir diğer ifadeyle methodu post olan bir form ögesi içerisindeki butona tıklanmadı.

Veri Girişi Yapıp Göndere Basıldığında

Örnek 3

İsim:

E-posta:

Sistem **Mutlu** adlı kişiye **mutlu@test.com** eposta adresini kullanarak mail atacak.

GET YÖNTEMİ

Bu yöntemle çalışırken POST yöntemi ile yazmış olduğunuz kodlar üzerinde aşağıdaki değişiklikleri yapmanız yeterli olacaktır.

```
<form action="test.php" method="get">
```

```
<?php
//ilk olarak istediğimiz veriler gönderilmiş mi kontrol ediyoruz.

// isset: veri var mı?
// $_GET[".."]: get ile taşınan verilere ulaşma
if ( isset($_GET["ad"]) & isset($_GET["eposta"])){
    // name' i ad ve eposta olan get ile taşınan veri var olduğunda
    yapılacaklar...

    echo "<p style=background-color:grey;margin:10px;padding:5px;>
Sistem <vurgu>" . $_GET["ad"] . "</vurgu> adlı kişiye <vurgu>" .
$_GET["eposta"] . "</vurgu> eposta adresini kullanarak mail
atacak.</p>" ;
    // php de . işareti kullanılarak string ifadeler ve değişkenler
birleştirilebilir.
} else {
    // name' i ad ve eposta olan post edilmiş veri var olmadığında
    yapılacaklar...
    echo "<p style=background-color:grey;margin:10px;padding:5px;>Sayfa
POST edilmedi. Bir diğer ifadeyle methodu post olan bir form ögesi
içerisindeki butona tıklanmadı.</p>";
}
?>
```

← → ↻ ↗ ⓘ localhost/phpsite/test.php?ad=Mutlu&eposta=mutlu%40test.com

Örnek 3

İsim:

E-posta:

Sistem **Mutlu** adlı kişiye **mutlu@test.com** eposta adresini kullanarak mail atacak.

Dikkat!

- Taşınan verilerin ismi ve değeri ile birlikte tarayıcının URL satırında görüntülediği dikkatinizi çekti mi?

Kodlayıcı Görevi

- Kullanıcının bir sayfada, ismini, soy ismini, şifresini, e-postasını, cinsiyetini ve mesajını girmesini sağlayınız. Daha sonra bu verileri başka bir sayfada (ör: test2.php) gösteriniz.

İsim:
Soyisim:
E-mail:
Cinsiyet:
 Erkek
 Kadın
Mesaj:
Formu gönder

Uygulama

Şimdi kullanıcının dosya göndermesini sağlayacak bir uygulama yapınız. Bu uygulama için dosya adları dosyayukle.php ve dosyaal.php olan iki sayfa için kodlama yapınız.

Senaryo

- ✓ Kullanıcı dosyayukle.php sayfasını ziyaret eder.
- ✓ Form elementlerini kullanarak herhangi bir dosya seçer ve gönder butonuna basar.
- ✓ dosyaal.php sayfasında post edilen bir dosya olup olmadığı, dosyanın istenen özelliklere uyup uymadığı kontrol edilir.
- ✓ Eğer post edilen bir dosya yoksa ya da istenen özelliklere uymuyorsa GET yöntemi kullanılarak dosyayukle.php sayfasına bir uyarı gönderilir.
- ✓ Eğer post edilen bir dosya varsa ve istenen özelliklere uyuyorsa, post edilen dosya site dizini altında dosyalar klasörüne kaydedilir. Son olarak dosyaal.php sayfasında kullanıcıya dosyanın başarılı şekilde gönderildiği ile ilgili uyarı mesajı verilir.

HTML Kodları (dosyayukle.php)

```
<!--  
    action: butona tıkladıktan sonra dosyaal.php  
    dosyasına yönlendirilen  
  
    method: veri taşıma yöntemi post olan  
    bir form oluşturuyoruz.  
-->  
<form action="dosyaal.php" method="post">  
    <!--Dosya yükleme için type özelliği file olan bir  
    input nesnesi oluşturuyoruz.-->  
    <input type="file" name="dosya" />  
    <input type="submit" value="Gönder" />  
</form>
```

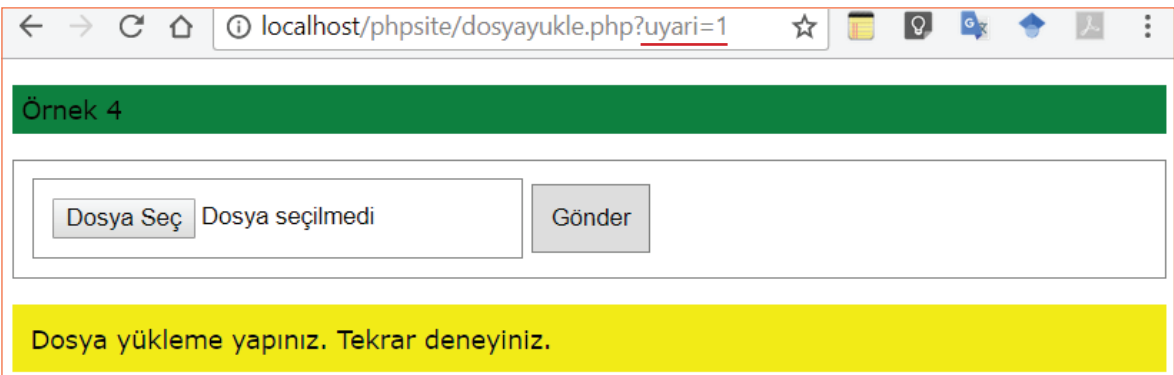
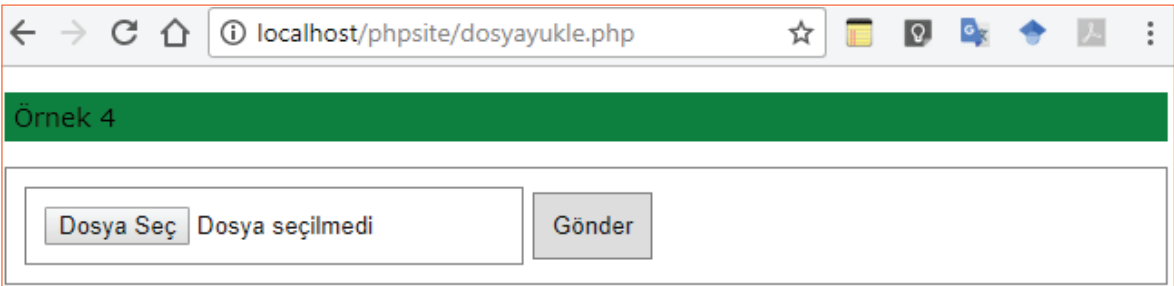
CSS Kodları (dosyayukle.php)

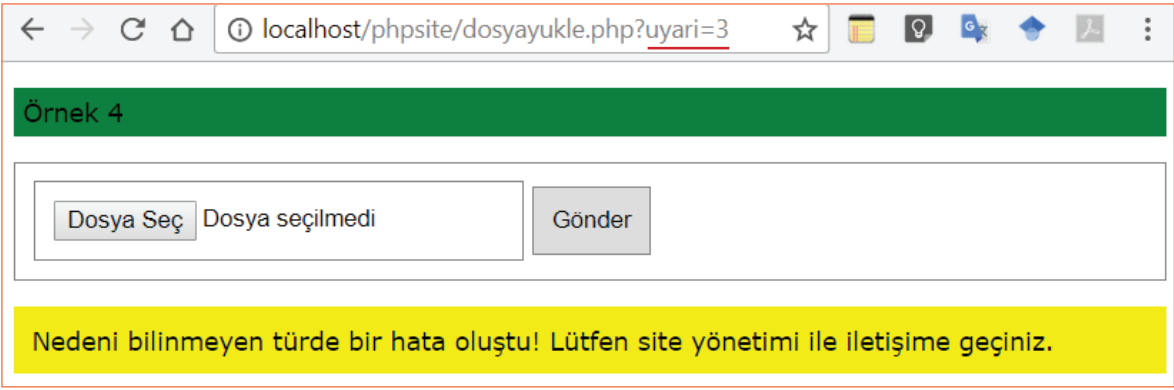
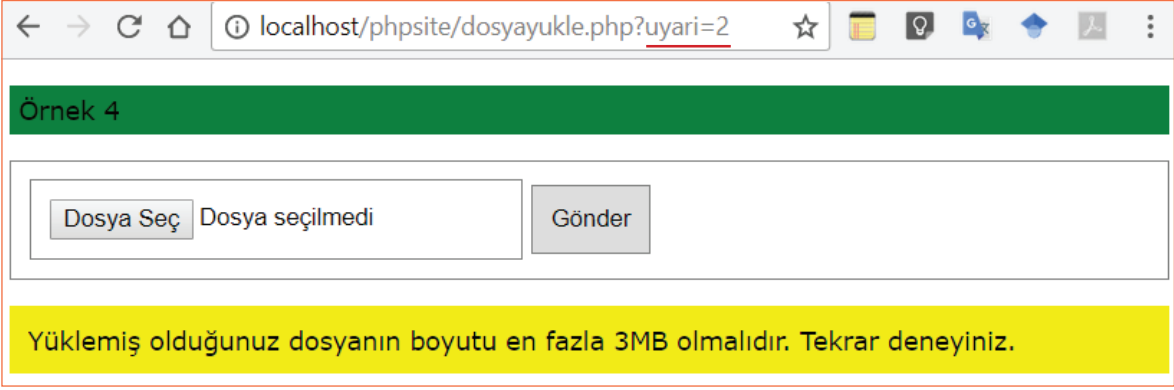
```
<style>
  body    {font-family:Verdana;font-size:14px;}
  .baslik {padding:5px; background-color:green;}
  form { border:1px solid grey; padding:10px;}
  input { border:1px solid grey; padding:10px;}
  .uyari {padding:10px; background-color:yellow;}
</style>
```

PHP Kodları (dosyayukle.php)

```
<?php
  if (isset($_GET["uyari"])){
    // Eğer get yöntemi ile gönderilmiş bir uyarı varsa ve değeri 1 ise;
    //Uyarı numarasına göre (1 veya 2) hata mesajını ekrana yazdır.
    if ($_GET["uyari"] == 1) {
      echo "<p class=uyari> Dosya yükleme yapınız.
      Tekrar deneyiniz. </p>";
    } else if ($_GET["uyari"] == 2){
      echo "<p class=uyari> Yüklemiş olduğunuz dosyanın boyutu
      en fazla 3MB olmalıdır. Tekrar deneyiniz. </p>";
    } else {
      echo "<p class=uyari> Nedeni bilinmeyen türde bir hata oluştu!
      Lütfen site yönetimi ile iletişime geçiniz. </p>";
    }
  }
?>
```

Ekran Görüntüsü (dosyayukle.php)





CSS Kodları (dosyaal.php)

```
<style>
  body      {font-family:Verdana;font-size:14px;}
  .baslik   {padding:5px; background-color:green;}
  .uyari    {padding:10px; background-color:yellow;}
  a { padding:10px; border-radius:4px; background-color:orange;
    color: white; text-decoration:none;}
  a:hover  { padding:10px; border-radius:4px; background-color:grey;
    color: white; text-decoration:none;}
</style>
```

PHP Kodları (dosyaal.php)

Senaryoya göre dosyanın gönderilip gönderilmediği, istenen özelliklere (ör: 3MB dan küçük olması) uygun olup olmadığı kontrol edilecekti. Bu doğrultuda, genel kod algoritması aşağıdaki şekilde oluşturulabilir.

```

<?php
// isset(): ? ile belirtilen nesne var mı yok mu diye kontrol etmek
// $_FILES["?"]: ? ile belirtilen adda post edilen dosyayı çekmek
if(isset($_FILES["dosya"])) {
    // eğer adı "dosya" olan bir dosya post edilmişse;
    // dosyanın özelliklerini kontrol et
    // -----

    $boyut = $_FILES["dosya"]["size"];
    //dosyanın boyutunu $boyut değişkenine eşitle.
    if($boyut > (1024*1024*3)){
        //eğer dosya boyutu 3MB dan büyükse;
        // dosyayukle.php sayfasına git ve uyarı 2'yi get yöntemi ile gönder.
    } else {
        // eğer dosya boyutu 3MB dan küçükse;
        // dosyayı sunucuya kaydet ve dosyaal.php sayfasında uyarı göster.
        // .....
    }
} else {
    // eğer adı "dosya" olan bir dosya post edilmemişse;
    // dosyayukle.php sayfasına git ve uyarı 1 i get yöntemi ile gönder.
}
?>

```

Bu algorithmada, dosyayukle.php sayfasına geri nasıl yönlendirme sağlanacağı ve dosyanın sunucuya kaydedilmesi ile ilgili işlemler gösterilmemiştir. Şimdi bu iki kodlama üzerinde duralım.

Php ile Sayfa Yönlendirme Kodu

```
header ("Location:yonlendirileceksayfa.php");
```

Get Yöntemi ile Sayfa Yönlendirme

```
header ("Location:yonlendirileceksayfa.php?degiskenismi=deger");
```

Dosyanın Sunucuya Kaydedilmesi

```

$dosya = $_FILES['dosya']['tmp_name'];
// ($_FILES['dosya']['tmp_name']) :
// Yüklenen dosyanın sunucuda saklandığı sıradaki geçici dosyayı
// $dosya değişkenine aktarma

copy($dosya, 'dosyalar/' . $_FILES['dosya']['name']);
// $dosya 'yı
// site ana dizininde oluşturmuş olduğunuz dosyalar klasörü içerisinde
// $_FILES['dosya']['name'] ismine göre kaydetme.

// Kullanıcıya dosyanın yükleme durumu ile ilgili uyarı verme
echo "<p class=baslik>Örnek 4<p>";
echo "<p style=uyari>Dosyanız başarılı şekilde sunucuya yüklendi!<p>";
echo "<a href=dosyayukle.php>Yeni Dosya Yükle</a>";

```

Şimdi bu kodları algorithmadaki yerlerine uygun şekilde yazalım. Daha sonra sayfamızı test edelim.

Örnek 4

Dosyanız başarılı şekilde sunucuya yüklendi!

[Yeni Dosya Yükle](#)

Kodlayıcı Görevi

- Tarayıcıyı ilk açtığınızda dosyayukle.php sayfasını değil de, dosyaal.php sayfasına bağlanmak istediğimizde ne olur? Neden?
- Post edilen dosyanın boyutu ile birlikte başka hangi özellikleri kontrol edilebilir? Araştırınız ve kodlamasını yaparak özelliklere uymayan bir dosya için yeni bir uyarı mesajı oluşturunuz.

PHP İLE VERİ YÖNETİMİ

PHP ile veri tabanına veri ekleyebileceğimizden, veri tabanından veri silebileceğimizden ve veri tabanındaki verileri değiştirebileceğimizden bahsedilmişti. Bir uygulama üzerinden php ile veri yönetiminin nasıl yapılabileceği üzerinde durunuz.

Uygulama

- Kullanıcı uyeol.php sayfasını ziyaret eder. Karşısına çıkan formu doldurur ve veri tabanına yazdırılması için gönderir.
- Site yöneticisi olan bir kişi, yönetim panelinden giriş yapar ve bu verileri görüntüler. Yanlış veri girişi yapmış olan kullanıcıların bilgilerini siler ya da günceller.

Bu senaryo için

- Veri tabanında kullanıcı isimli bir tabloya,
- Kullanıcılar bilgilerini girebileceği bir uyeol.php sayfasına,
- Yöneticilerin giriş yapabileceği bir giriş.php sayfasına,
- Yöneticilerin giriş yaptıktan sonra verileri görebileceği, güncelleyebileceği ya da silebileceği bir yönetim.php sayfasına ihtiyaç var.

1- Veri tabanı Tablosunun Hazırlanması

- Öncelikle “site” isimli bir veri tabanı oluşturunuz.
- Daha sonra, kullanıcı ile ilgili “kullanıcı adı”, “şifresi”, “adı soyadı”, “e-postası”, “öz geçmişi”, “rolü” bilgilerini tutacak bir tablo tasarımı yapınız.

#	Name	Type	Collation	Attributes	Null	Default
1	kullaniciAdi	varchar(8)			No	None
2	sifre	varchar(8)			No	None
3	adiSoyadi	varchar(25)			No	None
4	ePosta	varchar(25)			No	None
5	ozgecmisi	text			No	None
6	rolu	int(1)			No	0

“rolü” sütunu kullanıcının sistemdeki yetkilendirmesi ile ilgili bilgi tutan alandır. Bu örneğimizde, eğer rolü 1 ise yönetici 0 ise yetkisiz kullanıcıdır. Dolayısı ile yeni kayıt yapan bir kullanıcının rolü 0 olacaktır. Bu yüzden bu alanın varsayılan değerini 0 olarak göstermelisiniz.

Dikkat!

- Tablonuzu oluşturduktan sonra yönetici rolünde bir kullanıcı kaydı girişi yapmayı unutmayınız!

2-Kullanıcıların Bilgilerini Girebileceği uyeol.php Sayfasının Hazırlanması

Veri tabanında kullanıcı ile ilgili bilgi tutabileceğimiz tablomuzu oluşturduk. Şimdi senaryomuzdaki sırayı takip ederek, resimdeki gibi bir uyeol.php sayfası oluşturalım. Öncelikle sayfanın tasarımını daha sonra ise, veritabanı bağlantısı ve veri ekleme işlemlerini yapacağız.

The image shows a web form titled 'uyeol.php' with an orange background. It contains five input fields: 'Kullanıcı Adı', 'Şifreniz', 'Adı-Soyadı', 'E-Posta', and 'Özgeçmiş'. A 'Kayıt Ol' button is positioned at the bottom center of the form.

CSS Kodu

```
<style>
  body    { font-family:Verdana;font-size:14px;}
  p       {padding:5px;background-color:grey;color:white; border-radius:5px;}
  form    { border:1px solid grey; padding:10px;
           background-color:orange; border-radius:5px;}
  input   { border-radius:5px; border-style:solid;border-color:white;}
  textarea { border-radius:5px; border-style:solid;border-color:white;}
</style>
```


Kullanılan Form Elementleri

```
<form action="uyeol.php" method="post">
<table width="400" border="0">
  <tr>
    <td width="115">Kullanıcı Adı</td>
    <td width="269"><input name="kadi" type="text" /></td>
  </tr>
  <tr>
    <td>Şifreniz</td>
    <td><input name="sifre" type="password" /></td>
  </tr>
  <tr>
    <td>Adı-Soyadı</td>
    <td><input name="adsoyad" type="text" /></td>
  </tr>
  <tr>
    <td>E-Posta</td>
    <td><input name="eposta" type="text" /></td>
  </tr>
  <tr>
    <td>Özgeçmiş</td>
    <td><textarea rows="4" cols="25"></textarea></td>
  </tr>
  <tr>
    <td></td>
    <td><input type="submit" value="Kayıt Ol" /></td>
  </tr>
</table>
</form>
```

Php Kodu

Daha önceden de bahsetmiş olduğum gibi, kodlamış olduğumuz web sayfası ile veri tabanı birbirinden farklı platformlardır. Biz kodlayıcı olarak önce bu platformları birbiri ile konuşturmalıyız. Bunu yaparken php kodlarını kullanacağız. İşlem adımlarımız şekildeki gibidir.



- mysql_connect



- mysql_select_db



- mysql_query



- mysql_close

Veritabanına Bağlan

```
<?php

$veritabanisunucusu = "localhost";
// veritabanı sunucumuz kendi bilgisayarımızda olduğu için
// sunucumuzun adı localhost ya da ip adresi 127.0.0.1 dir.
// her ikisini de kullanabiliriz.

$vt_kullanicisi = "root";
$vt_kullanicisi_sifre = "1234";
// phpmyadmin kurulumundan sonra güvenlik için veritabanı sunucunuzda
// bir yönetici kullanıcı adı ve şifresi tanımlamanız gerekiyor.
// Php den veritabanına bağlantı kurarken de bu kullanıcı adı ve şifresi ile
// sunucuya bağlanabilirsiniz.

$baglanti = mysql_connect($veritabanisunucusu, $vt_kullanicisi, $vt_kullanicisi_sifre);
// Veritabanımıza mysql_connect kodu ile bağlanıyoruz.
// Kodla beraber, veri tabanı sunucusunun adı ya da ip adresi
// veritabanı kullanıcısı ve veritabanı kullanıcısının şifresi
// belirtilmelidir.
// Kod çalıştığıında bağlantı gerçekleşirse true,
// gerçekleşmezse false değeri döndürür.
// döndürülen değer $baglanti değişkenine aktarılır.

// Son olarak bağlantı gerçekleşti mi gerçekleşmedi mi kontrol edelim.
if ($baglanti==true) {
    echo "Bağlantı Başarılı!";
} else { die("Bağlantı Başarısız! Hata: " . mysql_connect_error());}

?>
```

Eğer bağlantı başarılı şekilde gerçekleştirildi ise, veri tabanı seçme işlemine geçilir. Senaryomuz gereği “site” isminde veri tabanı oluşturmuştuk.

Veritabanı Seç

```
if ($baglanti==true) {

    // Açıklayıcı Yöntem
    if (mysql_select_db("site") == true) {

        // Veritabanı seçimini mysql_select_db kodu ile yapıyoruz.
        // eğer kod çalıştırıldığında herhangi bir sorunla karşılaşılmazsa;
        // true değer döndürülür.

        echo "Veritabanı seçimi başarılı!";

    } else {
        die("Veritabanı Bulunamadı. Hata: " . mysql_connect_error());
        // Girmiş olduğunuz isimli bir veritabanı olmayabilir.
        // Bu gibi bir durumda, mysql_select_db sonucu false döner.
        // die kodu ile hata gösterilebilir.
    }

    // Daha Kısa Yöntem
    mysql_select_db("site") or die("Veritabanı Bulunamadı. Hata: " . mysql_connect_error());

} else { die("Bağlantı Başarısız! Hata: " . mysql_connect_error());}
```

Eğer veri tabanı seçme işleminde de herhangi bir sorunla karşılaşılma ise, artık sorgumuzu gönderebiliriz. Senaryomuz gereği kayıt ekleme işlemini yapacağız. Bu işlem için, kullanıcı tablosuna ekleyeceğimiz kayıtların değerlerini form elementlerinden gelen bilgilerle dolduracağız. Öncelikle bir veri ekleme sql cümleciğinin nasıl olduğunu hatırlayalım.

```
INSERT INTO tablo_adi (sutun1, sutun2,sutun3, ...)
VALUES ('sutun1_deger', 'sutun2_deger', 'sutun3_deger', ...);
```

Php ile bir sql cümleciğinin veritabanında çalıştırılmasını sağlayan kod ise `mysql_query`' dir.

Sorgu Gönder

```
// --Form verilerini çekme--
$kadi = "".$_POST["kadi"]."";
$sifre = "".$_POST["sifre"]."";
$adsoyad = "".$_POST["adsoyad"]."";
$eposta = "".$_POST["eposta"]."";
$ozgecmis = "".$_POST["ozgecmis"]."";
// Form verilerinin öncesinde ve sonrasında ' işareti eklediğimiz dikkatinizi çekmiştir.
// Veritabanına sql sorgusu göndereceğimiz için değerleri sorgu biçimine uygun hale getirdik.

$sqldegerler = $kadi ." , ". $sifre ." , ". $adsoyad ." , ". $eposta ." , ". $ozgecmis;
// Daha sonra tüm değerleri aralarına virgül koyarak birleştirdik.

$insert_sql= "insert into kullanıcı (kullanıcıAdi,sifre,adiSoyadi,ePosta,ozgecmisi)
values ($sqldegerler)";
// Değerleri atanmış formata uygun bir insert sql cümlesi oluşturduk.

$eklemeislemi = mysql_query($insert_sql);
// Son olarak oluşturmuş olduğumuz insert sql cümleciğinin
// mysql_query ile veritabanında çalıştırılmasını sağladık.
// Eğer kayıt başarılı şekilde eklendi ise true,
// tersi ise false değeri döndürülür.

if ($eklemeislemi==true){
    echo "Kayıt başarılı şekilde eklendi.<br/>";
}else {
    die("Kayıt Eklenemedi. Hata: ". mysql_connect_error());
}
// Kayıt ekleme işleminin gerçekleşip gerçekleşmediğini kontrol ediyoruz.
```

Bağlantıyı Kapat

Kayıt ekleme işlemimiz de bittiğine göre artık veri tabanı bağlantımızı kapatalım.

```
mysql_close($baglanti);
// vt sunucusu (MySQL/mariaDB) ile bağlantımızı koparttık.
```

uyeol.php sayfanızı ilk açtığınızda şekildeki gibi hatalar göreceksiniz. Bu hataları sayfayı ilk açtığımızda post edilen bir form elementi olmadığı için alıyoruz. Hatırlarsanız form elementlerinin verilerini çekerken sayfanın post edilme durumunu kontrol ediyorduk. Şimdi php kodunuzun başladığı yere sayfada post edilen bir form elementi olup olmadığını kontrol eden bir kontrol yapısı ekleyiniz. Sadece bir tane form elementini (Ör: "kadi" isimli form elementi) kontrol etmeniz yeterli olacaktır.

Notice: Undefined index: kadi in **C:\xampp\htdocs\phpsite\uyeol.php** on line **91**
Notice: Undefined index: sifre in **C:\xampp\htdocs\phpsite\uyeol.php** on line **92**
Notice: Undefined index: adsoyad in **C:\xampp\htdocs\phpsite\uyeol.php** on line **93**
Notice: Undefined index: eposta in **C:\xampp\htdocs\phpsite\uyeol.php** on line **94**
Notice: Undefined index: ozgecmis in **C:\xampp\htdocs\phpsite\uyeol.php** on line **95**

```
if (isset($_POST["kadi"]))  
{  
    // Yapmış olduğumuz tüm veritabanı işlemleri  
}
```

Bunun dışında bir küçük problemle daha karşılaşmanız olasıdır. Kayıt ekleme işlemlerini yaptıktan sonra phpmyadmin panelinden kullanıcı tablonuza baktığınızda, Türkçe karakterlerin (ç,ğ,ı,ö,ş) gösteriminde hatalar olduğunu göreceksiniz. Bu türden bir hata olmaması için Php kodlarında mysql ile veri işlemleri yapmadan önce karakter kümesini UTF-8 olarak ayarlamalısınız. Bu işlemi veritabanı bağlantısını kurduktan sonra yapabilirsiniz.

```
if ($baglanti==true) {  
    echo "Veritabanı bağlantısı sağlandı!<br/>";  
  
    mysql_query("SET NAMES UTF8");  
    // Karakter setinin UTF- olarak ayarlanması  
  
    // Kodun devamı ...
```

Üye olma ekranının görünümü

uyeol.php

Kullanıcı Adı	<input type="text" value="dnzyil"/>
Şifreniz	<input type="password" value="....."/>
Adı-Soyadı	<input type="text" value="Denizer Yıldırım"/>
E-Posta	<input type="text" value="test@test.com"/>
Özgeçmiş	<input type="text" value="Doğum tarihim
Eğitim durumum
... görevlerde bulundum."/>
	<input type="button" value="Kayıt Ol"/>

Veritabanı bağlantısı sağlandı!
Veritabanı seçimi başarılı!
Kayıt başarılı şekilde eklendi.

Phpmyadmin panelindeki kullanıcı tablosunun görünümü

kullaniciAdi	sifre	adiSoyadi	ePosta	ozgecmisi	rolu
dnzyil	12345	Denizer Yıldırım	test@test.com	Doğum tarihim ... Eğitim durumum gö...	0

Kodlayıcı Görevi

- Şimdi aynı şekilde siz de 10 kayıt giriniz?
- Bu kayıtlardan bir tanesinin rolünü phpmyadmin panelini kullanarak 1 yapınız.

3. Yöneticilerin giriş yapabileceği giriş.php sayfasının hazırlanması

Bu sayfada yönetici olan bir kişinin kullanıcı adı ve şifresini girmesi istenecektir. Daha sonra, bu bilgiler veri tabanındaki bilgilerle karşılaştırılacaktır. Eğer bilgiler birbiri ile tutarlı ise, oturum başlatılıp yönetim.php sayfasına yönlendirme yapılacaktır.

giris.php

Kullanıcı Adı

Şifre

Form Elementlerinin HTML Kodu

```
<form name="giris" action="giris.php" method="post">
  <table cellpadding="8" cellspacing="0" >
    <tr>
      <td width="100">Kullanıcı Adı</td>
      <td><input type="text" name="kullaniciadi"></td>
    </tr>
    <tr>
      <td width="100">Şifre</td>
      <td><input type="password" name="sifre"></td>
    </tr>
    <tr>
      <td colspan="2" align="right">
        <input type="submit" value="Giriş">
      </td>
    </tr>
  </table>
</form>
```

PHP kodu

Veritabanına
Bağlan

Veritabanı
Seç

Sorgu
Gönder

Bağlantıyı
Kapat

Veri ekleme işleminde olduğu gibi Veritabanına Bağlanma, Veri tabanı Seçme ve bağlantı kapama işlemlerini aynı şekilde yapıyoruz.

```
// --Form verilerini çekme--
$kadi = "".$_POST["kadi"]."";
// Form verilerinin öncesinde ve sonrasında
// ' işaretini eklediğimiz dikkatinizi çekmiştir.
// Veritabanına sql sorgusu göndereceğimiz için
// değerleri sorgu biçimine uygun hale getirdik.

$sql = "select * from kullanıcı where kullanıcıAdi= $kadi";
// Kullanıcı adı, giriş.php sayfasında
// kadi isimli form elementine girilen değer olan bir kaydı
// getirecek sql cümlesi...

$sorgulamaislemi = mysql_query($sql);
// Son olarak oluşturmuş olduğumuz select cümleciğinin
// mysql_query ile veritabanında çalıştırılmasını sağladık.
// Eğer kayıt başarılı şekilde getirildi ise
// bir kaynak verisi döndürülür.
// Bu kaynak verisi boş kayıt olabileceği gibi,
// birden fazla kayıta içerebilir.
// Tersi durumda ise false döndürülür.
```

```
if ($sorgulamaislemi != false) {
    // sorgu sonrasında bir hata ile karşılaşılmamışsa,
    if (mysql_num_rows($sorgulamaislemi) > 0) {
    } else {
        // sorgu sonrasında bir hata ile karşılaşıldıysa,
        echo "Sql Cümlesinde Hata Var!";
    }
}
```

```
if (mysql_num_rows($sorgulamaislemi) > 0) {
    // Eğer kaynak verisindeki kayıt sayısı 0 dan büyükse;
    // Kullanıcının girmiş olduğu kadi veritabanında var demektir.
    // varsa şifresi doğrumu ona bakalım!!!

    $satisir = mysql_fetch_assoc($sorgulamaislemi);
    // ilk olarak kaynağımız içerisindeki verilere,
    // kolayca ulaşabilmemiz için
    // kaynağımızı mysql_fetch_assoc() ile bir diziye dönüştürüyoruz.

    if ($satisir['sifre'] == $_POST["sifre"]) {
    } else {
        // Kullanıcının girmiş olduğu kadi veritabanında yok demektir.
        echo "Kullanıcı Adı Bulunamadı!";
    }
}
```

```

if ($satis['sifre'] == $_POST["sifre"]) {
    // Veritabanındaki şifre ile kullanıcının girmiş olduğu şifre eşleşiyorsa;
    // kullanıcının rolü kontrol edilir.
    if ($satis["rolu"] == 1) {
    } else {
        // Kullanıcının girmiş olduğu şifre ile veritabanındaki şifre eşleşmiyorsa;
        echo "Şifrenizi Yanlış Girdiniz!";
    }
}

```

```

if ($satis["rolu"] == 1) {
    // kullanıcının rolü 1 (yönetici) ise;
    //oturum başlatılıp yönetim.php sayfasına yönlendirme sağlanabilir.
    session_start(); // oturumu başlat
    $_SESSION["adsoyad"] = $satis["adiSoyadi"];
    header("Location: yönetim.php");
}else{
    // kullanıcının rolü 0 (yönetici değil) ise;
    echo "Yönetici Rolünüz Atanmamıştır. Giriş Yapamazsınız.";
}

```

Sayfayı test ettiğinizde, eğer tüm kodlarınız içerisinde herhangi bir sorunla karşılaşırsa, giriş.php sayfasında size hata verilecek, karşılaşılmazsa yönetim.php sayfasına yönlendirileceksiniz.

4- Yöneticilerin giriş yaptıktan sonra verileri görebileceği, güncelleyebileceği ya da silebileceği yönetim.php sayfasının hazırlanması

Sayfamızın senaryomuzdaki gibi tüm işlemleri yapabilecek hale gelmesi için kodlama sırası aşağıdaki gibidir.

```

<!DOCTYPE html>
<?php ### 1. Oturum kontrolü

<html>
<style>
    body { font-family:Verdana;font-size:14px;}
    p {padding:5px;background-color:grey;color:white; border-radius:5px;}
    input { border-radius:5px; border: 1px solid #42bff4; padding: 5px;}
    table {table-layout:fixed;}
    td {padding:5px; border: 1px solid green;}
    tr:first-child td{font-weight:bold; background-color:#42bff4;}
</style>
<body>
<p>yonetim.php</p>

<?php ### 2. Veritabanı bağlantısı ve veritabanı seçimi
<?php ### 5. Veri güncelleme ve silme işlemleri
<?php ### 4. Veritabanındaki kullanıcı tablosundan tüm kayıtların getirilmesi
<?php ### 3. Veritabanı bağlantısının kapatılması

</body>
</html>

```

1. Oturum kontrolü

İlk olarak sayfayı sadece adı soyadı için oturum açılmış bir kullanıcı görebilir. Dolayısı ile oturum açılıp açılmadığı ile ilgili bir kontrolümüz olmalıdır.

```
<?php
    session_start();
    // session kullanımını başlatıyoruz.

    // giris.php sayfasında;
    // kullanıcı adı ve şifresi doğrulanan kişinin
    // adsoyad için bir session açmıştık.
    // Not: kullanıcı idsi için de açabilirdik.

    if (isset($_SESSION["adsoyad"]) == false)
    {
        // eğer daha önce adsoyad isimli bir oturum açılmamışsa
        // yani $_SESSION["adsoyad"]) false ise;
        // giriş sayfasına yönlendiriyoruz.

        header ("Location: giris.php");
    }
?>
```

2. Veri tabanı bağlantısı ve veri tabanı seçimi

```
<?php ### Veritabanı bağlantısı ve veritabanı seçimi

$veritabanisunucusu = "localhost";

$vt_kullanicisi = "root";
$vt_kullanicisi_sifre = "1234";

$baglanti = mysql_connect($veritabanisunucusu,$vt_kullanicisi,$vt_kullanicisi_sifre);

if ($baglanti==true) {

    mysql_query("SET NAMES UTF8");

    mysql_select_db("site")
    or die("Veritabanı Bulunamadı. Hata: ".mysql_error());
} else {die("Veritabanı bağlantısı kurulamadı! Hata:".mysql_error());}

?>
```

3. Veri tabanı bağlantısının kapatılması

```
<?php
    mysql_close($baglanti) or die("VT bağlantısı kapatılırken bir hata
    oluştu!. Hata:".mysql_error());
?>
```


4. Veri tabanındaki en güncel kayıtların sayfada gösterimi

Bu bölümde veri tabanındaki kullanıcı tablosundan tüm kayıtları çekerek ve bu kayıtları sayfada gösterebilecek bir kodlama yapacağız. Bunun için iki basamaklı bir kodlama yapacağız.

1. Veri tabanındaki kullanıcı tablosundan tüm kayıtların getirilmesi

```
<?php ### Veritabanındaki kullanıcı tablosundan tüm kayıtların getirilmesi

$sql = "select * from kullanıcı";
// Veritabanındaki kullanıcı tablosundaki tüm kayıtları getiren sql cümlesi...

$sorgulamaislemi = mysql_query($sql);
// select cümleciğinin mysql_query ile veritabanında çalıştırılmasını
// ve geriye bir veri kaynağı döndürdük.

if ($sorgulamaislemi != false) {
    // sorgu sonrasında bir hata ile karşılaşılmamışsa,
    // artık verileri tablo halinde yöneticiye gösterebiliriz.

    ### Kayıtların sayfada gösterimi
}
?>
```

2. Kayıtların sayfada gösterimi

```
### Kayıtların sayfada gösterimi
### Tablonun üst bilgilerinin yazdırılması

echo "<table>";
// --- php ile table html kodlarının sayfaya yazdırılması ---

echo "<tr>";
// table etiketi altında <tr> etiketi

echo "<td>Kullanıcı Adı</td>";
echo "<td>Şifre</td>";
echo "<td>Adı Soyadı</td>";
echo "<td>e-posta</td>";
echo "<td>Özgeçmiş</td>";
echo "<td>Rolü</td>";
echo "<td>Düzenleme</td>";
echo "<td>Silme</td>";
// tr etiketinin altında td etiketleri
// ilk satır sütunların başlıklarını içermektedir.

echo "</tr>";
// ilk satırın kapatılması
```

```

### Kayıtların sayfada gösterimi
### Tablo içeriğinin veritabanından döndürülen veri kaynağı içerisindeki
### ($sorgulamaislemi) verilerle doldurulması
while($satis = mysql_fetch_assoc($sorgulamaislemi)) {
    // $sorgulamaislemi kaynağındaki veriler bitene kadar
    // her seferinde $satis değişkenine
    // bir kaydın bilgileri dizi olarak getirilir.
    // Kaynaktaki veriler bittiğinde while döngüsü sona erer.

    echo "<tr>";
    echo "<td>" . $satis["kullaniciAdi"]. "</td>";
    echo "<td>" . $satis["sifre"]. "</td>";
    echo "<td>" . $satis["adiSoyadi"]. "</td>";
    echo "<td>" . $satis["ePosta"]. "</td>";
    echo "<td>" . $satis["ozgecmisi"]. "</td>";
    echo "<td>" . $satis["rolu"]. "</td>";
    // while döngüsü sonlandırılana kadar,
    // tablonun ilk satırını oluştururken yapmış olduğumuz gibi
    // tabloya bir kayıt eklenir ve sütunların içeriği
    // $satis dizi elemanları ile doldurulur.

    echo '<td><a href="yonetim.php?eylem=duzenle&kadi='
    .$satis['kullaniciAdi'].'"> Düzenle </a></td>';
    // Düzenleme isimli sütun için bir link oluşturulur.
    // Bu linkte, eylem ve kadi bilgileri get yöntemi ile birlikte taşınır.

    echo '<td><a href="yonetim.php?eylem=sil&kadi='
    .$satis['kullaniciAdi'].'"> Sil </a></td>';
    // Silme isimli sütun için bir link oluşturulur.
    // Bu linkte, eylem ve kadi bilgileri get yöntemi ile birlikte taşınır.

    echo "</tr>";
    // Satır etiketi kapatılır.
}
echo "</table>";
// Tablo etiketi kapatılır

```

Ekran Görüntüsü

Kullanıcı Adı	Şifre	Adı Soyadı	e-posta	Özgeçmişi	Rolü	Düzenleme	Silme
dnzyil	123456	Denizer Yıldırım	test@test.com	Doğum tarihim Eğitim durumum görevlerde bulundum.	1	Düzenle	Sil
Myildiri	09876	Mutlu Yıldırım	happy@test.com	Ankara Üniversitesi Siyasal Bilimler ve kamu Yönetimi mezunuyum.	0	Düzenle	Sil
Sguzel	345987	Selvi Güzel Yıldırım	selvi@test.com	MEB Fen ve Teknoloji Öğretmeni...	0	Düzenle	Sil

5. Veri güncelleme ve silme işlemleri

Bir önceki aşamada tablo içerisine verilerle birlikte düzenle ve silme linkleri oluşturmuştuk. Kodu daha detaylı incelediğimizde get yöntemi ile eylem ve kadi değişkenlerinin taşındığını göreceğiz. Dolayısıyla ilk olarak sayfamızda get yöntemi ile taşınan bu isimlerde veriler olup olmadığını kontrol etmemiz gerekmektedir. Bu bölümdeki kodlamamızın genel algoritması aşağıda gösterilmektedir.

```

<?php ### 5. Veri güncelleme ve silme işlemleri

    if (isset($_GET["kadi"])&&isset($_GET["eylem"]))
    {
        if ($_GET["eylem"]=="duzenle"){
            ### 1.Veritabanındaki kullanıcı tablosundaki tüm kayıtları getiren sql cümlesi...
        }
    }else if ($_GET["eylem"]=="sil"){
        ### 3. Veri tabanından veri silme işlemi
    }
    ### 2. Veri Güncelleme İşlemi
?>

```

Veri Güncelleme Formunun Hazırlanması

İlk olarak bir kaydın sütun verilerinde güncelleştirme yapabilmemiz için veri eklemede olduğu gibi, form öğelerini kullanmamız gerekmektedir. Fakat burada önemli bir ayrıntıdan bahsetmemiz gerekiyor. Veri ekleme için boş form öğelerine değerler yazıyorduk. Şimdi ise değerleri girilmiş verimiz üzerinde değişiklik yapmak istiyoruz. Dolayısı ile, güncelleme yapmak istediğimiz kaydın sütun değerlerini veri tabanından çekerek, form elemanları içerisine yazmamız gerekmektedir.

```

### Veri Güncelleme İşlemleri
$sql = "select * from kullanıcı where kullanıcıAdi='".$_GET["kadi"].'";
// Veritabanındaki kullanıcı tablosundaki tüm kayıtları getiren sql cümlesi...

$sorgulamaislemi = mysql_query($sql);
// select cümleciğinin mysql_query ile veritabanında çalıştırılmasını sağladık.

if ($sorgulamaislemi != false) {
    // sorgu sonrasında bir hata ile karşılaşılmamışsa,
    // artık verileri form halinde yöneticiye gösterebiliriz.
    echo "<form action=yonetim.php method=post>";
    echo "<table>";

    // Tablo içeriğinin kullanıcı tablosundaki verilerle doldurulması
    $satisir = mysql_fetch_assoc($sorgulamaislemi);
    // $sorgulamaislemi kaynağında aynı kullanıcı adı ile birden fazla kayıt olamayacağı
    için while döngüsü açmadık.
    // kullanıcı tablomuzda birincil anahtar olarak kullanıcıAdi sütununu atamıştık.

    echo "<tr>";
    echo "<td>Kullanıcı Adı: <input name=kadi type=text value='".$satisir['kullanıcıAdi'].
    "'/></td>";
    echo "<td>Şifresi:<input name=sifre type=text value='".$satisir['sifre'].
    "'/></td>";
    echo "<td>Adı Soyadı: <input name=adsoyad type=text value='".$satisir['adıSoyadı'].
    "'/></td>";
    echo "<td>E-postası:<input name=eposta type=text value='".$satisir['ePosta'].
    "'/></td>";
    echo "<td>Özgeçmiş:<textarea name=ozgecmis>".$satisir['ozgecmisi'].</textarea></td>";
    echo "<td>Rolü:<input name=rol type=text value='".$satisir['rolu'].
    "'/></td>";
    // form öğeleri yazdırılarak, form değerleri ilgili dizi elemanlarına eşitlendi.

    echo '<td><input type="submit" value="Kaydet" /></td>';
    // son sütunda tipi submit olan bir kaydet butonu oluşturulur.

    echo "</tr>";
    // Satır etiketi kapatılır.

    echo "</table>";
    echo "</form>";
    // Tablo ve form etiketleri kapatılır

```

Ekran Görüntüsü

yonetim.php

Kullanıcı Adı: Myildiri	Şifresi: 09876	Adı Soyadı: Mutlu Yıldırım	E-postası: happy@test.com	Özgeçmişi: Ankara Üniversitesi Siyasal Bilimler ve kamu Yönetimi	Rolü: 0	Kaydet
-----------------------------------	--------------------------	--------------------------------------	-------------------------------------	--	-------------------	--------

Kullanıcı Adı	Şifre	Adı Soyadı	e-posta	Özgeçmişi	Rolü	Düzenleme	Silme
dnzyil	123456	Denizer Yıldırım	test@test.com	Doğum tarihim Eğitim durumum görevlerde bulundum.	1	Düzenle	Sil
Myildiri	09876	Mutlu Yıldırım	happy@test.com	Ankara Üniversitesi Siyasal Bilimler ve kamu Yönetimi mezunuyum.	0	Düzenle	Sil
Sguzel	345987	Selvi Güzel Yıldırım	selvi@test.com	MEB Fen ve Teknoloji Öğretmeni...	0	Düzenle	Sil

Veri Güncelleme İşlemi

Oluşturmuş olduğumuz form üzerinde yönetici değişiklik yaptıktan sonra kaydet butonuna bastığında sayfa tekrar yönetim.php sayfasına post edilecektir. Dolayısı ile herhangi bir form elementinin post edilip edilmediğinin kontrolünden sonra artık veri güncelleme için gerekli olan sorguyu oluşturabiliriz.

```
if (isset($_POST["kadi"]) == true)
{
    // kadi isminde bir form elementi post edilmişse anlıyoruz ki,
    // bizden güncelleştirme yapmamız isteniyor.

    $kadi = "".$_POST["kadi"]."";
    $sifre = "".$_POST["sifre"]."";
    $adsoyad = "".$_POST["adsoyad"]."";
    $eposta = "".$_POST["eposta"]."";
    $ozgecmis = "".$_POST["ozgecmis"]."";
    $rol = "".$_POST["rol"]."";
    // update sql cümlesi için gerekli olan değerleri
    // uygun biçimde yeni değişkenlere aktarıyoruz.

    $updatesql = "Update  kullanici set kullaniciAdi=$kadi, sifre=$sifre
, adiSoyadi=$adsoyad, ePosta=$eposta, ozgecmisi=$ozgecmis,rolu=$rol
where kullaniciAdi=$kadi";
    // daha sonra yeni değişkenleri update cümleciğinde uygun yere
    ekliyoruz.

    mysql_query($updatesql) or die ("Veri güncelleme işlemi başarısız
oldu. Hata: ". mysql_error());
    // Son olarak oluşturmuş olduğumuz update cümleciğini veritabanına
    çalıştırılması için gönderiyoruz.
}
```

Veri güncelleme işlemi yapıldıktan sonra veri tabanındaki verilerin güncel hâli ekranda gösterilecektir. Örneğin “Myildiri” kullanıcı isimli kaydın mail adresi ve rolünü değiştirelim.

Kullanıcı Adı	Şifre	Adı Soyadı	e-posta	Özgeçmişi	Rolü	Düzenleme	Silme
dnzyil	123456	Denizer Yıldırım	test@test.com	Doğum tarihim Eğitim durumum görevlerde bulundum.	1	Düzenle	Sil
Myildiri	09876	Mutlu Yıldırım	mutluyildirim@test.com	Ankara Üniversitesi Siyasal Bilimler ve kamu Yönetimi mezunuyum.	1	Düzenle	Sil
Sguzel	345987	Selvi Güzel Yıldırım	selvi@test.com	MEB Fen ve Teknoloji Öğretmeni...	0	Düzenle	Sil

Veri Silme İşlemi

```
### Veri Silme İşlemleri
$yadi= "' . $_GET["yadi"] . "'";
// delete sql cümlesi için gerekli olan değeri
// uygun biçimde yeni değişkenlere aktarıyoruz.

$silmesql ="Delete from kullanıcı where kullanıcıAdi='". $_GET["yadi"] .
"'";
// daha sonra yeni değişkenleri update cümleciğinde uygun yere ekliyoruz.

mysql_query($silmesql) or ("Veri silme işlemi başarısız oldu. Hata: ".
mysql_error());
// Son olarak oluşturmuş delete cümleciğini veritabanına çalıştırılması
için gönderiyoruz.
```

Veri silme işlemi yapıldıktan sonra veri tabanındaki verilerin güncel hali ekranda gösterilecektir. Örneğin üç kaydımız vardı. “Myildiri” kullanıcı isimli veriyi sildiğimizde ekranda sadece iki kayıt gösterilmelidir.

Ekran Görüntüsü

Kullanıcı Adı	Şifre	Adı Soyadı	e-posta	Özgeçmişi	Rolü	Düzenleme	Silme
dnzyil	123456	Denizer Yıldırım	test@test.com	Doğum tarihim Eğitim durumum görevlerde bulundum.	1	Düzenle	Sil
Sguzel	345987	Selvi Güzel Yıldırım	selvi@test.com	MEB Fen ve Teknoloji Öğretmeni...	0	Düzenle	Sil

7. WEB TABANLI PROJE GELİŐTİRME

7. WEB TABANLI PROJE GELİŞTİRME

Şimdiye kadar web temelli programlamada kullanılan HTML5, CSS3, Javascript, PHP ve Mysql hakkında bilgi sahibi olduk. Kitabın bu bölümüne geçmeden önce sizlerden bu dillerin temel yazım kurallarını, arasındaki farklılıkları kavramış olmanız ve temel ve orta seviyedeki örnekleri uygulayabiliyor olmanız beklenmektedir. Bu bölümde ise bu dillerin hepsini kullanarak bir proje geliştirmeniz beklenmektedir. Böylece, kitabın bu bölümüne kadar öğrenmiş olduğunuzu seçmiş olduğunuz bir proje temelinde daha detaylı ve bütünsel bir şekilde uygulama fırsatı bulacaksınız.

Bir web sitesi geliştirme projesinde, sürecin başından sonuna kadar size yol haritası olabilecek aşağıdaki gibi bir yöntem izlemeniz önerilir.

PLANLAMA

İlk olarak web sitesinin ne amaçla kullanılacağını açıkça ortaya koymanız gerekmektedir. Bir web sitesinin amaçlarının önceden belirlenmesi zamanınızı daha verimli kullanmanızı sağlayacaktır. İyi bir web sitesinden beklenen kullanıcının her şeye ulaşabilmesi değildir. Aksine kullanıcının amaçladığı eylemi kolay ve çabuk bir şekilde yapabilesidir.

Bununla birlikte, web sitesini kimlerin kullanacağı ve hangi bilgisayar ortamlarında (akıllı telefon, tablet, düşük çözünürlüklü kişisel bilgisayar ya da yüksek çözünürlüklü kişisel bilgisayar) kullanılacağı ile karar vermelisiniz. Böylece web sitesinin tasarımını, kullanıcının yaş, bilgisayar kullanma becerilerine uygun ya da hangi cihazların kullanımı daha ön plana çıkıyorsa ona uygun yapmanız mümkün olabilir.

Web tasarımcısının web sitesi içeriği hakkında karar vermesi gerekir. Web sitesinin metin ve resimleri dışında video, ses içerecek mi? İçerecekse bu video ve ses dosyaları nasıl temin edilecek? İnternette indirilecekse, telif hakları ile ilgili sorun olur mu? Bununla birlikte “Web sitesinde yetkilendirilmiş kullanıcılar olacak mı ya da yetkilendirilmiş kullanıcılar web sitesinde neler yapabilecek?” sorularına da cevap vermiş olmanız gerekmektedir. Örneğin hangi site içeriklerine yetkilendirilmemiş kullanıcılar erişebilecek? Böylece web sitenizin veri tabanının gereksinimleri ile ilgili de fikir sahibi olabilirsiniz.

TASARIM

İnternet ortamında daha önceden benzer siteleri araştırınız. Amacınıza göre hangi tür bir web sitesinin tasarımının daha uygun olacağına karar veriniz. Bu sitelerdeki gibi bir tasarım sizin için yeterli midir yoksa siz yeni bir tasarım ortaya koyabilir misiniz?

Siteniz menü yapısı nasıl olacak? Örneğin site menüsü tüm sayfalarda görünecek mi ya da her sayfa için ayrı bir menü mü olacak? Hangi menü yapısını kullandığınızda, web sitenizin amacına ve kullanıcılarına daha uygun olur?

En basit ve sade şekilde sitenin amacına uygun bir tasarımı kağıt üzerinde çizin. Sitenizde kaç sayfa olmasını bekliyorsanız bu sayfalar arasındaki bağlantıları da çizin. Eğer bu sayfalar arasında veri taşınma gereksinimi varsa kağıt üzerinde hangi yöntemi kullanacağınızı da belirtiniz.

Tüm ayrıntıları ile web sitenizin tasarımını ve sayfalar arasındaki bağlantıları çizdikten sonra başka bir arkadaşınızla, bilişim öğretmeni ve web sitesini kullanacak en az 3 kişi ile tasarım üzerinde konuşunuz. Böylece gözden kaçırmış olduğunuz ayrıntıları yakalama ve sorunlara çözüm bulma olanağını bulabilirsiniz.

Tasarımınızı kağıt üzerinde tamamladıktan sonra web sitenizin hangi bölümlerinde hangi programlama dillerini kullanacağınıza karar vermiş olmalısınız. Daha sonra “Web sitesinde gereksinim duyduğunuz kodlama ile ilgili daha önceden bir örnek yaptınız mı? Kodun nasıl yazılacağını hatırlıyor musunuz? Ek kaynaklara gereksinim duyuyor musunuz? Kimlerden yardım alabilirsiniz?” sorularına yanıt veriniz.

GELİŞTİRME

Tasarım sürecinden sonra, web sitenizi geliştirme sürecine geçebilirsiniz. Bu aşamada kağıt üzerindeki taslağınız doğrultusunda kitabın daha önceki bölümlerinde öğrenmiş olduğumuz kodlama becerilerini uygulayacaksınız. Bu süreç sizler için hem öğrendiklerinizi pekiştirme hem de yeni kodlama becerileri geliştirmenize olanak sağlayacaktır. Bu süreçte, öncelikle yedekli çalışmanız tavsiye edilir. Kodlama yaparken en basit ve kolay yollardan hedefe ulaşmayı deneyiniz. Çözüm bulamadığınız kodlama bölümlerinde, motivasyonunuzu düşürmeden hemen İnternette araştırmaya yöneliniz. İnternette neyi nasıl arayacağınız konusunda fikriniz yoksa bilişim öğretmeninden yardım isteyiniz.

TEST ETME

Web sitesi, tasarım ve geliştirme aşamaları içerisinde ve son hali ile test edilmelidir.

Web sitesinin içeriği ve işlevselliği ile ilgili test işlemleri

- Sayfa başlıklarını da içerecek şekilde metinleri yazım kurallarına uygun şekilde düzeltmek,
- Sayfalar arası yanlış ya da çalışmayan bağlantıları düzeltmek,
- Görüntülenmeyen ya da metinle tutarsız grafiklerin kaynağını (src) düzeltmek
- Sayfalar arası formlarla taşınan verileri kontrol etmek. Eğer veriler doğru şekilde taşınıyorsa, veritabanına bu verilerin doğru şekilde yazdırılıp yazdırılmadığını kontrol etmek.
- Diğer etkileşimli sayfa öğelerinin kullanıcı eylemine göre doğru şekilde çalışıp çalışmadığını kontrol etmek,
- Düşük hızlı bağlantıda yükleme hızını kontrol etmek için sayfaları test etmek.

Web sitesinin kullanılabilirliği ile ilgili test işlemleri

Kullanılabilirlik web sitesinin tasarımının ne kadar iyi olduğunun ölçüsüdür ve kullanılabilirliği yüksek olan bir site, kullanıcıların amaçlarını gerçekleştirmelerine olanak tanır.

Kullanılabilirlik testi için sitenin amacına uygun görevler belirlenir. Daha sonra kullanıcısı olacak kişilerden bu görevleri gerçekleştirmesi istenir. Her görevin gerçekleştirilip gerçekleştirilemediği ve gerçekleştirildi ise gerçekleştirilme süresi not edilir. Daha sonra test sonuçlarına göre web sitesinde düzeltmeler yapılır. Bu düzeltmeler, gerçekleştirilemeyen ya da uzun sürede gerçekleştirilebilen görevler için kullanıcı görüşleri alınarak sayfa tasarımındaki küçük değişiklikler renklerin belirginleştirilmesi, yazı puntolarının büyütülmesi, menü yapısının daha anlaşılır yapılması vb.) olabilir. Görevlerden hiçbiri gerçekleştirilemiyorsa amacına uygun bir site tasarımı yapılamamıştır. Bu durumda tasarımınızı yenilemeniz gerekli olabilir.

WEB TABANLI PROGRAMLAMA İÇİN PROJE ÖRNEKLERİ

Kitabın bu bölümünde sizlere üzerinde çalışabileceğiniz proje örnekleri tanıtılacaktır. Eminim ki siz kodlayıcılar da bu örnekler dışında birçok yaratıcı proje fikirleri ortaya atabilirsiniz. Bununla birlikte, projenin kodlanması için gerekli planlamaları yaparak öğrenmiş olduğumuz dilleri bu projeleri geliştirme amacıyla uygulayabilirsiniz.

Kişisel Gelişim Sitesi

Hatırlarsanız kitabın ilk iki bölümü sonunda bir site şablonu oluşturmuştuk. Bu şablon üzerinden devam

ederek kitap içerisinde uygulamış olduğumuz Javascript, Mysql ve Php örneklerini şablon içerisine yerleştirebilirsiniz. Bu site sizin Bilgisayar Bilimi dersindeki kişisel gelişim siteniz olabilir. Bununla birlikte, bu siteyi okulunuzdaki Bilgisayar Bilimi dersinin bir web sitesi haline getirebilirsiniz. Böylece sizden daha sonra bu deri alacak alt sınıflardaki arkadaşlarınız için yol gösterici bilgiler içeren bir site kodlamış olursunuz. Öte yandan, her yıl ortaya çıkan özgün projeleri bu siteden duyurarak projelerinizin geliştirilmesini sağlayabilirsiniz.

Fotoğraf Paylaşım Sitesi

Her yaz döneminde sınıf arkadaşlarınızın neler yaptıkları ile fikir sahibi olabileceğiniz resimlerin paylaşılacağı basit bir sosyal ağ sitesi geliştirebilirsiniz.

- Bu site basitçe üç sayfadan oluşabilir. Birincisi, site açıldığında paylaşılanların görülebileceği ana sayfa, ikincisi arkadaşlarınızın paylaşım yapmak için oturum açma sayfası ve sonuncusu arkadaşlarınızın oturum açtıktan sonra resim yükleme sayfası.
- Bu proje kapsamında paylaşım yapan kullanıcıyı, paylaşım tarihini ve paylaştığı resmin dosya ismini tutan bir veri tabanı oluşturabilirsiniz.
- Site ana sayfasını ilk ziyarette bu tablodaki verileri çekerek tasarımınızda (Ör: facebook sayfanızdaki gibi) uygun yerlere yazdırarak, resimlerin paylaşım yapan kişi ile birlikte alt alta görüntülenmesini sağlayabilirsiniz.
- Eğer çok fazla kullanıcı tarafından paylaşım yapılırsa ve dolayısı ile veri tabanında çok fazla kayıt varsa veri tabanından gelen verileri sayfalandırarak gösterebilirsiniz.
- Bu projeyi daha da ilerletmek isterseniz facebook ve youtube gibi siteleri inceleyerek sayfalandırma yerine sayfanın altına ilerledikçe veri gelmesini sağlayan Javascript kütüphaneleri kullanabilirsiniz.

Ödev Notlandırma Sistemi

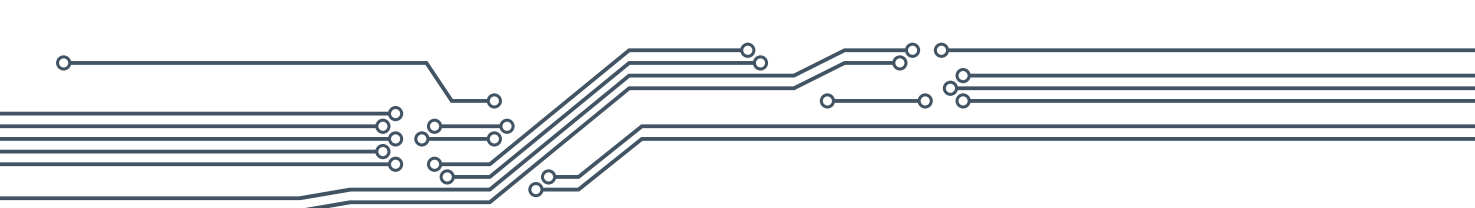
Bilişim öğretmeni için sizlere vermiş olduğu ödevleri kolayca takip ederek değerlendirebileceği bir web sitesi geliştirebilirsiniz. Bu siteyi basitçe senaryolaştıracak olursak;

- Siteye ilk erişildiğinde karşımıza bir oturum açma sayfası gelebilir.
- Oturum açan kullanıcının rolüne göre (öğrenci, öğretmen) kullanıcıyı öğrenci ya da öğretmen sayfasına yönlendirebiliriz.
- Öğrenci sayfasında kullanıcının haftalara göre gönderim yapabileceği bir form tasarımı yapabiliriz. Bu sayfada aynı zamanda daha önce gönderim yapılan ve öğretmen tarafından değerlendirilmiş ödevlerin notu ile birlikte listelenmesi sağlanabilir.
- Öğretmen sayfasında öğrenciler tarafından gönderim yapılmış ödevlerin listelenmesini sağlayabiliriz. Bununla birlikte, listelenen ödevlerden herhangi birine tıkladığında ödevle ilgili not ve açıklama girilmesini sağlayacak bir form tasarımı gerekir. Bu işlemi farklı sayfalarda yapabileceğiniz gibi aynı sayfa içerisinde de kodlamanız mümkündür.

Bu senaryoya göre veri tabanımızda kullanıcılar ve ödevlerle ilgili iki tablo tutulması yeterli olabilir. Fakat ödev tablosunu oluştururken öğrenci rolündeki bir kullanıcının birden fazla ödev gönderiminin olabileceğini göz önünde bulundurunuz ve birincil anahtar olacak sütunları buna göre belirtiniz.

Kodlama Öğretiyorum Sitesi

“En iyi öğrenme anlatma yoludur.” varsayımından hareket ederek iyi bir kodlayıcı, yazmış olduğu



kodu sade ve anlaşılır şekilde anlatabilendir. Böyle bir bakış açısı temelinde, siz de kod yazarken anlaşılabilir şekilde videolarınızı çekebilir ve bu videoları paylaşabileceğiniz basit bir blog sayfası tasarlayabilirsiniz. Böylece, akranlarınızla, aynı işi farklı kodlama yöntemleri ile yapabilen çözümleri paylaşabilir ve kodlama becerilerinizi geliştirebilirsiniz.

Şimdi basit bir şekilde bu sitenin senaryosunu yazalım. Siteye ilk giriş yapıldığında, karşımıza HTML5, CSS3, MYSQL Yönetimi ve PHP başlıklarını içeren bir menü yapısı geleceğini hayal edin. Bu menüler alt başlıklar da içerebilir. Başlıklardan herhangi birine tıkladığında, sizlerin videoya çekmiş oldukları kodlama örnekleri, tıklanan başlığa göre veri tabanından çağrılır.

Peki, bu şekilde bir senaryo için veri tabanımızda neler olmalı? Öncelikle temel kullanıcı bilgileri ile birlikte, sadece yetkilendirilen kişilerin video ekleyebilmesi için kullanıcı rolünü tutan bir tablo gereklidir. Bununla birlikte kullanıcıların eklemiş oldukları videonun yolu ve dosya ismini tutan videolar tablosu gereklidir. Video tablosunu oluştururken bir kullanıcının birden fazla video gönderiminin olabileceğini göz önünde bulundurunuz ve birincil anahtar olacak sütunları buna göre belirtiniz.

Soru Cevap Sitesi

ÖSYM sınavlarına yönelik okulunuz öğrencilerinin soru sorabilecekleri, yine okulunuz öğretmenleri tarafından bu soruların yanıtlanabileceği bir soru cevap sitesi tasarlayabilirsiniz. Öğrencilerin sitede soru sormalarını teşvik etmek için haftanın en değerli sorusunun seçimine yönelik bir kodlama yapabilirsiniz. Siteyi ziyaret eden kullanıcılar sorunun önemine göre bir değerlendirme notu verebilir. Bu notların ortalamasına göre haftalık önemli soruları listeleyebilir ve en önemli soruyu soran kullanıcının sitenin ana sayfasında görüntülenmesini sağlayabilirsiniz.

Anket Sitesi

Herhangi bir konuda okulunuz öğrencileri ya da velilerinin görüşlerini toplamak için bir anket sitesi tasarlayabilirsiniz. Bu site için okulunuz yöneticileri ya da öğretmenleri ile görüşerek önceden belirlenmiş anket sorularını internet ortamına geçirebilir, daha sonra web sitesinde hazırlamış olduğunuz anketi okulunuz öğrencileri ya da velilerine duyurabilirsiniz. Öğrencileri ve velilerden almış olduğunuz yanıtları bir veri tabanında tutabilir ve istenildiğinde bu verilere ilgili birimlerin erişebilmesini sağlayabilirsiniz. Bununla birlikte projenizi biraz daha geliştirmek isterseniz, yetkilendirilmiş kullanıcıların site içerisinde anket oluşturmasını sağlayarak sitenize daha dinamik bir işlev kazandırabilirsiniz.

Kıyaslama Sitesi

Popüler ürün ya da servisleri kıyasladığınız bir site tasarlayabilirsiniz. İnsanlara hangisinin daha iyi olduğunu gözlemlerinize ve araştırmalarınıza dayalı olarak bu web sitesinde gösterebilirsiniz.

Deney Sitesi

Fen laboratuvarındaki deney araçları ile yapılan deney videolarının paylaşıldığı bir web sitesi oluşturabilirsiniz. Bu web sitesini laboratuvarında bulunan araçları listeleyebilecek, kolayca güncelleştirilmesini sağlayabilecek ve istenildiğinde bu araçlar ile ilgili tanıtıcı bilgilere de ulaşılacak şekilde genişletebilirsiniz.

Ortak Wiki Alanı

Herhangi bir derse (Fizik, Kimya vb.) ilişkin terimlerin tanımlarının paylaşıldığı sınıflar arasında ortak kullanılabilen bir web sitesi oluşturabilirsiniz.

Kantin Sitesi

Okulunuzda işletilen kantinde satılan ürünlerin fiyatlarının takip edilebileceği, kantin işleticisi tarafından fiyat güncelleştirmelerinin yapılabileceği bir web sitesi tasarlayabilirsiniz.

Tasarruf Sitesi

Para tasarrufunun nasıl yapılması gerektiği hakkında bir site tasarlayabilirsiniz. Bunun için yaşlılar, üniversite öğrencileri, ev kadınları ile konuşarak onların deneyimlerini sitenizde paylaşabilirsiniz. Site-nize kullanıcılar üye olabilir ve kendi deneyimlerini paylaşabilir.

Özet

Yukarıdaki proje örnekleri, projenizi tasarlayıp geliştirmeden önce örnekler üzerinde tartışarak ufkunuzu genişletmek amacıyla paylaşılmıştır. Sadece bu site örnekleri ile sınırlı kalmayabilirsiniz. Bilgi çağı toplumundaki sorunlara çözüm olabilecek yeni proje fikirleri ortaya koyabilir ve bu projeleri web tabanlı programlama dilleri ile geliştirebilirsiniz.

