

Php Form İşlemleri

Form İşlemleri

Form İşlemleri

- Bütün programlama dillerinde kullanıcının girmiş olduğu bilgiler üzerinde işlem yapılır. Php'de kullanıcı tarafından bilgileri alabilmek için formlar kullanılır. Örnek olarak ziyaretçi defteri, kayıt formu ,anket vs.. diğer işlemlerde formlar kullanılır.
- Form elemanları sayfa içerisine HTML yardımıyla yerleştirilir ama içerikleri HTML yardımıyla okunamaz. Bu form elemanları üzerinde javascript yardımıyla çeşitli kontroller sağlanabilir ama bilgiyi sunucuya gönderemez. Formdan alınan bilgiler sunucuya gönderilecekse PHP gibi dinamik bir dil kullanımı zorunludur.

Form Elemanlari

- form
- input
- textarea
- select

<form> etiketi

Formlar yoluyla ziyaretçiden bilgi alabilir, aldığımız bilgi doğrultusunda birçok işlemi gerçekleştirebiliriz. Bir form <form> </form> etiketleri ile yazılır.

Form etiketi yardımcı elemanları

Girilecek Kod	Görevi
action = "dosya ismi"	Formdan girilecek bilgilerin değerlendirileceği dosyanın tam yolu.
method = "yöntem"	Formdan girilecek bilgilerin değerlendirici dosyaya gönderilme yöntemi

<input> etiketi

- Genel amaçlı bir form etiketidir. Sonlandırıcı etiketi yoktur. Girdi türü **type** ögesinde belirtilerek farklı girdilerin alınmasını sağlar.

<input> etiketi

type = "eleman türü"

name = "isim"

value = "değer"

size = "değer"

maxlength = "değer"

checked = "değer"

src = "dosya ismi"

alt = "metin"

align = "left|center|right"

readonly

disabled

Eleman türü belirtilir. text, password, checkbox, radio, submit, reset, file, hidden, image, button değerlerinden biri kullanılır. Etiket bu değerlerle nasıl kullanıldığını aşağıda örnekleriyle açıklamaya çalıştım.

Girilen verinin hangi değişken ismi altında değerlendirileceğini belirtmek için kullanılır. Mutlaka belirtilmelidir.

"radio" ve "checkbox" dışında bu parametrenin kullanımı isteğe bağlıdır. Belirtilecek "değer" eleman türüne göre değişir.

"text" ve "password" elemanlarında karakter sayısını, diğerlerinde ise piksel cinsinden genişliği belirtmek için kullanılır.

"text" ve "password" elemanlarında girilebilecek en çok karakter sayısını belirtmek için kullanılır. Öntanımlı değeri sınırsızdır.

Sadece "radio" ve "checkbox" elemanlarında seçili elemanı "on" olarak belirtmek için kullanılır. Diğerlerinde yoksayılr.

"image" elemanında resim dosyasını belirtmek için kullanılır.

Fare eleman üzerindeyken balon içinde görünecek açıklama metni için kullanılır.

Elemanın form üzerinde nasıl konumlanacağını belirtmek için kullanılır.

Veri girme amacıyla kullanılmayacak elemanlar içindir.

Veri girişi engellenecek elemanlar içindir.

<input> etiketi

type parametresi ile aşağıdaki elemanlar tanımlanabilir.

- Text
- Password
- Checkbox
- Radio
- Submit
- Reset
- Button

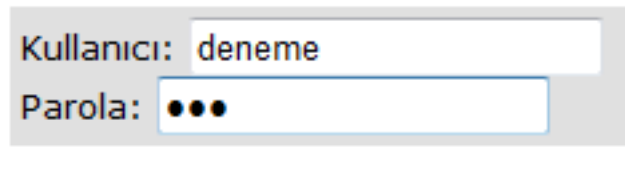
text

- Forma tek satırlık yazı yazılabilecek alan eklemek için kullanılır.
- Adı: `<input type="text" name="firstname">
`
Soyadı: `<input type="text" name="lastname">`



password

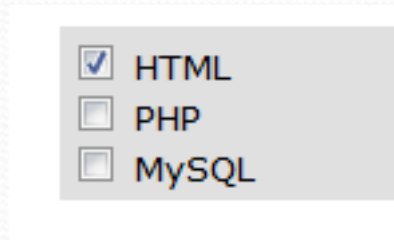
- Forma parola yazılabilecek alan eklemek için kullanılır. "text" elemanından farklı olarak bu alana girilen her karakter * ile gösterilir.
- Kullanıcı: `<input type="text" name="username">
`
Parola: `<input type="password" name="password">`



Kullanıcı: deneme
Parola: ●●●

checkbox

- Forma onay kutuları eklemek için kullanılır.
- `<input type="checkbox" name="kutu1" checked="on">` HTML
`<input type="checkbox" name="kutu2">` PHP
`<input type="checkbox" name="kutu3">` MySQL



HTML
 PHP
 MySQL

radio

- Forma radyo düğmeleri eklemek için kullanılır.
- `<input type="radio" name="cinsiyet" value="erkek" checked="on">`
Erkek
`<input type="radio" name="cinsiyet" value="disi">` Dişi



Erkek
 Dişi

submit, reset, button

- Formda belirtilen bilgileri ilgili dosyaya yollamak için kullanılacak düğmeler yerleştirmek için kullanılır.
submit formu kabul eder ve yollar,
reset ise girilen bilgileri sıfırlar(siler).
button herhangi bir amaçla kullanmak içindir. Diğer ikisi gibi öntanımlı bir davranışı yoktur.

submit, reset, button

- Adınız: `<input type="text" name="firstname">
`
Soyadınız: `<input type="text" name="lastname">
`
Eposta adresiniz: `<input type="text" name="email">
`
`<input type="radio" name="cinsiyet" value="erkek" checked="on">`
Erkek

`<input type="radio" name="cinsiyet" value="disi">` Dişi

`<input type="submit" value="Send">` `<input type="reset"`
`value="Sıfırla">`

Adınız:

Soyadınız:

Eposta adresiniz:

Erkek
 Dişi

```
Düğmeye resim ekleme
<button name="reset" type="reset"
value="reset">
  Sıfırla
</button>
```

<textarea> etiketi

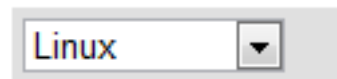
- İngilizce “metin alanı” anlamına gelen bu kelime forma yazı yazılabilecek alan eklemek için kullanılır. Yardımcı öğeler olarak **rows** ve **cols** kullanılır.

<select> etiketi

- Seçim listesi oluşturmakta kullanılır. <option> etiketi ile liste öğeleri belirtilir. <optgroup> etiketi ise bu liste elemanlarını konularına göre gruplamakta kullanılır.
- Bir select etiketi en azından bir option etiketi içermeli ve optgroup etiketleri iç içe olmamalıdır. Tek veya çok seçimli (**multiple** parametresi ile), sabit (**size** parametresi ile) veya aşağıya açılan listeler oluşturulabilir.

<select> etiketi

- ```
<select name="os_type">
<option value="win">Windows 9x</option>
<option value="winnt">Windows NT</option>
<option value="linux">Linux</option>
<option value="unix">UNIX</option>
<option value="os2">OS/2</option>
<option value="macos">MacOS</option>
</select>
```





# <select> etiketi

```
<select name="menu" multiple>
 <option selected="" value="none">Hiçbiri</option>
 <optgroup label="Çorbalar">
 <option value="c1">Tarhana</option>
 <option value="c2">Ezogelin</option>
 <option value="c3">İşkembe</option>
 </optgroup>
 <optgroup label="Yemekler">
 <option value="y1">İmambayıldı</option>
 <option value="y2">Sultanbeğendi</option>
 <option value="y3">Kadınbudu</option>
 </optgroup>
 <optgroup label="Tatlılar">
 <option value="t1">Kazandibi</option>
 <option value="t2">Tavukgöğsü</option>
 <option value="t3">Keşkül</option>
 </optgroup>
</select>
```



# Formdan Veri Gönderme

- Formdan veri göndermek için **method** parametresi kullanılır.
- Method parametresi iki değer alır **GET ve POST**
- `<form action="test.php" method="GET">`
- .....
- `</form>`
- `<form action="test.php" method="POST">`
- .....
- `</form>`

# Get metoduyla bilgi gönderme

- GET metodunda formlara girilen bilgiler adres çubuğunda gönderilirken görünürler. Formlara girilen bilgiler, sayfa adresinin sonuna şu şekilde eklenir.
- **`http://www.site.com/sayfa.php?form1=değer1&form2=değer2&...`**
- Burada **form1**, **form2**,... ile gösterilen ifadeler sayfada formlara verilen isimler yani name değerleridir. **değer1**, **değer2**,... ise kullanıcının bu formlara girdiği değerleri ifade etmektedir. Dikkat ederseniz tüm bunlar sayfa adresinden ? ile ayrılmaktadır, her form ve değerinin arasında & işareti bulunmaktadır.

# Get metoduyla bilgi gönderme

- Bu yöntemde;
- Gönderilen bilgiler adres çubuğunda görüldüğü için güvenlik düşüktür. Özellikle şifre girişlerinde kullanılmamalıdır.
- Adres çubuğundan gönderilebilecek veri miktarı sınırlı olduğunda fazla miktardaki veriler gönderilemez.

# Get methoduyla bilgi alma

- `<form>...</form>` etiketinin **action** özelliğinde belirtilen sayfada **GET** methoduyla yani URL'ye eklenerek gönderilen değişkeleri almak için `$_GET` dizisi kullanılır.
- `$_GET` dizisinde indis olarak değişken adları yada form adları kullanılır.
- Örneğin;  
**`http://www.site.com/index.php?konu=salam&mesaj=Merhaba`** şeklinde GET ile gönderilen **konu** ve **mesaj** bilgilerini şu şekilde alırız.
- `<?php`
- `$gelen_konu=$_GET["konu"];`
- `$gelen_mesaj=$_GET["mesaj"];`
- `?>`

# Örnek

- GET metodu ile gönderilen ad ve soyad bilgilerini alıp ekrana yazdıralım.

# Örnek Devam

```
<!--index.php-->
```

```
<html>
```

```
<body>
```

```
 <form name="form1" method="get" action="yaz.php">
```

```
 Adı: <input type="text" name="adi">

```

```
 Soyadı: <input type="text" name="soyadi">

```

```
 <input type="submit" name="gönder" value="Gönder">
```

```
 </form>
```

```
</body>
```

```
</html>
```

# Örnek Devam

```
<!--yaz.php-->
```

```
<html>
```

```
<body>
```

```
<?php
```

```
echo "Adınız: ",$_GET["adi"],"
";
```

```
echo "Soyadınız: ",$_GET["soyadi"],"
";
```

```
?>
```

```
</body>
```

```
</html>
```



# Örneğin Açıklaması

- Burada iki sayfamız vardır.
- İlk sayfamız olan **index.php** sayfasında formlarımız var. Bu formlara girilen bilgilerin **yaz.php** sayfasına gönderileceği **action="yaz.php"** ile belirtilmiştir. **method="get"** ifadesiyle bilgilerin GET metodu ile gönderileceği belirtilmiştir. Unutmayınız form yapılarında çoğu zaman POST metodu kullanılır. GET metodu pek kullanılmaz. Burada sadece GET metodu ile bilgi göndermeye ve almaya örnek vermekteyiz.
- Bilindiği gibi GET metodu daha çok `<a>...</a>` etiketi ile oluşturulan linkler ile bilgi göndermede kullanılır.
- İkinci sayfamız olan **yaz.php** sayfasında formlardan gelen bilgiler alınıyor. Daha sonra sayfaya yazdırılması için **echo** fonksiyonuna veriliyor. Bu bilgileri değişkenlere atayarak da işlem yapabilirsiniz.
- **Not:** **index.php** ile **yaz.php** sayfalarının aynı dizinde olduğu kabul edilerek **action** kısmında **yaz.php** sayfasının yolu belirtilmemiştir. Aksi halde dosyanın tam yolu yada göreceli yolu yazılmalıdır.

## Örnek2

- Rastgele üretilen iki sayıyı **yaz.php** sayfasına bir link ile (GET metodu ile) gönderip ekrana yazdıralım.

# Örnek2 (Devam)

```
<!--index.php-->
```

```
<html>
```

```
<body>
```

```
<?php
```

```
$s1=rand(1,100);
```

```
$s2=rand(1,100);
```

```
echo "Bulunan sayılar:
";
```

```
echo "Sayı 1=$s1
Sayı 2=$s2
";
```

```
echo "Bu linke tıklayarak
sayıları gönderebilirsiniz";
```

```
?>
```

```
</body>
```

```
</html>
```

# Örnek2 Devam

```
<!--yaz.php-->
```

```
<html>
```

```
<body>
```

```
<?php
```

```
echo "Sayı 1: ",$_GET["sayi1"],"
";
```

```
echo "Sayı 2: ",$_GET["sayi2"],"
";
```

```
?>
```

```
</body>
```

```
</html>
```

# Post methoduyla bilgi gönderme

- Bu methodda formlara girilen bilgiler gönderilirken görünmemektedir. Bundan dolayı daha güvenilirdir. Ayrıca büyük miktardaki veriler de bu yöntemle gönderilmektedir. Örneğin dosya yüklemede kullanılan ve fazla miktarda verilerin bulunduğu formlarda POST metodu kullanılmaktadır.

# Post metoduyla bilgi gönderme

- HTML dosyası:

```
<form action="yaz.php" method="post">
```

```
Adınız : <input type="text" name="adiniz">

```

```
Soyadınız : <input type="text" name="soyadiniz">

```

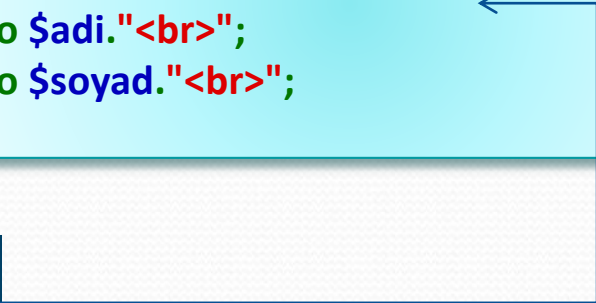
```
<input type="submit" value="Gönder">
```

```
</form/>
```

Yaz.php



```
<?php
$adi = $_POST[adiniz];
$soyad = $_POST[soyadiniz];
echo $adi."
";
echo $soyad."
";
?>
```



Formlardan POST metodu ile gönderilen bilgileri almak için \$\_POST dizisi kullanılır.

# Örnek

- Formlara girilen iki sayının toplamını topla.php sayfasında ekrana yazdıralım.
- (Örnek çalışmada **Metin Kutusu + Gönder Düğmesi** kullanınız.)

# Örnek Devam

```
<!--index.php-->
```

```
<html>
```

```
<body>
```

```
 <form name="form1" method="post" action="topla.php">
```

```
 Birinci Sayı: <input type="text" name="sayi1">

```

```
 İkinci Sayı: <input type="text" name="sayi2">

```

```
 <input type="submit" name="topla" value="topla">
```

```
 </form>
```

```
</body>
```

```
</html>
```



# Örnek Devam

```
<!--topla.php-->
```

```
<html>
```

```
<body>
```

```
<?php
```

```
$s1=$_POST["sayi1"];
```

```
$s2=$_POST["sayi2"];
```

```
$toplam=$s1+$s2;
```

```
echo "Sayıların toplamı: $s1+$s2=$toplam";
```

```
?>
```

```
</body>
```

```
</html>
```

# Örnek2

- Kullanıcının girdiği iki sayıyı yine kullanıcının açılır listeden seçtiği işleme göre **hesapla.php** sayfasında işleme alıp sonucu ekranda gösterelim.
- (Örnekte **Metin Kutusu + Açılır Liste + Gönder Düğmesi** kullanınız.)

# Örnek2 Devam

```
<!--index.php-->
```

```
<html>
```

```
<body>
```

```
<form name="form1" method="post" action="hesapla.php">
```

```
 Birinci Sayı: <input type="text" name="sayi1">

```

```
 İkinci Sayı: <input type="text" name="sayi2">

```

```
 İşleminizi seçiniz:
```

```
 <select name="islem">
```

```
 <option value="+">Topla</option>
```

```
 <option value="-">Çıkar</option>
```

```
 <option value="*">Çarp</option>
```

```
 <option value="/">Böl</option>
```

```
 </select>
```

```
 <input type="submit" name="hesapla" value="Hesapla">
```

```
</form>
```

```
</body>
```

```
</html>
```

# Örnek2 Devam

```
<!--hesapla.php-->
<html>
<body>
<?php
$s1=$_POST["sayi1"];
$s2=$_POST["sayi2"];
$islem=$_POST["islem"];

// Bu kısımda switch-case yapısı da kullanılabilirdi.
if($islem=="+")
 $sonuc=$s1+$s2;
elseif($islem=="-")
 $sonuc=$s1-$s2;
elseif($islem=="*")
 $sonuc=$s1*$s2;
else
 $sonuc=$s1/$s2;

echo "İşlemin Sonucu:
$s1$islem$s2=$sonuc";
?>
</body>
</html>
```

# Örnek3

- Kullanıcının girmiş olduğu kullanıcı adı ve şifreye göre **giris.php** sayfasında giriş izni veren yada vermeyen php kodunu yazalım.
- (Örnekte **Metin Kutusu + Şifre Kutusu + Gönder Düğmesi** kullanınız.)

# Örnek3 Devam

```
<!--index.php-->
```

```
<html>
```

```
<body>
```

```
<form name="form1" method="post" action="giris.php">
```

```
 Kullanıcı Adınız: <input type="text" name="k_adi">

```

```
 Şifreniz:<input type="password" name="sifre">

```

```
 <input type="submit" name="giris" value="Giriş">
```

```
</form>
```

```
</body>
```

```
</html>
```

# Örnek3 Devam

```
<!--giris.php-->
```

```
<html>
```

```
<body>
```

```
<?php
```

```
// İstenen kullanıcı adını ve şifreyi sabit değişkenler olarak tanımlayalım.
```

```
define("username","emyo");
```

```
define("password","au");
```

```
if($_POST["k_adi"]==username and $_POST["sifre"]==password)
```

```
 echo "Giriş izni verildi. Hoşgeldiniz.";
```

```
else{
```

```
 echo "Kullanıcı adını yada şifreyi yanlış girdiniz. Tekrar deneyiniz!
";
```

```
 echo "Geri";
```

```
}
```

```
?>
```

```
</body>
```

```
</html>
```