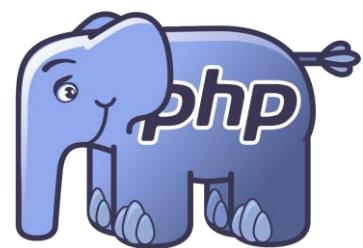


HTML / CSS – DUYARLI(RESPONSIVE) TASARIM

- HTML ve CSS Temelleri
- Duyarlı (Responsive) Tasarım



HTML - Hyper Text Markup Language

(Yüksek Metin İşaretleme Dili)



- HTML, web sayfalarını oluşturmak için kullanılan standart metin işaretleme dilidir.
- Web sitelerinin şekillendirilmesinde, içerik düzenleme ve yapılandırma kullanılır.
- HTML dosyasının uzantısı ***.htm** veya ***.html** olmak zorundadır (farklı amaçlar ile framework veya uygulamaya özel dosya tipi kullanılmadıkça).



HTML Sayfası Temel Yapısı

```
<!DOCTYPE Html>
<html>
    <head>
        <title> Benim Sayfam </title>
        <meta charset="UTF-8">
    </head>
    <body>
        <h1>Sayfama Hoş geldiniz!</h1>
    </body>
</html>
```

HTML Etiketleri

- **Başlık etiketleri:** `<h1>` ve `<h6>` dâhil olmak üzere aradaki tüm rakamlar kullanılarak tanımlanabilir
- **Paragraf etiketi:** `<p> Bu bir paragraf </p>`
- **Satır etiketi :** Satır atlama etiketi `
` veya `
` etiketidir, bu etiketin bitiş etiketi yoktur.
- **Yorum etiketi:** `<!-- Bu bir açıklama -->`
- **HTML Links – Hyperlinks:** Html sayfalarında bir başka sayfaya, içeriğe ya da url adresine bağlantı oluşturabiliriz.
 - **`href`:** Hangi bağlantıya gideceğini burada belirtiriz.
 - **`target`:** Sayfanın nerede açılacağını burada belirtiriz. Örneğin `target="_blank"` Bağlantıyı yeni bir pencerede açar.
 - ` FOTOĞRAFLAR ` yazıya tıkayınca `images.html` sayfası açılacak:
 - ` Google ` yazıya tıkayınca ilgili web adresi açılacak.
- **Resimler:** ``

HTML Etiketleri

LİSTELER:

- **Sırasız Listeler:** Sırasız bir liste maddelerden oluşur. ``
`......`
- **Sıralı Listeler:** Maddeler rakamlar, roma rakamları veya harfler ile listelenir. ``
`......`

`<h3> Alışveriş Listesi: </h3>`

`` Alışveriş Listesi:
`Kahve` • Kahve
`Süt` • Süt
`Çikolata` • Çikolata
``

`<h3> Alışveriş Listesi: </h3>`

`` Alışveriş Listesi:
`Kahve` 1. Kahve
`Süt` 2. Süt
`Çikolata` 3. Çikolata
``

`<ol type="A">`

Alışveriş Listesi:
`Sucuk` A. Sucuk
`Menemen` B. Menemen
`Bal` C. Bal
``

HTML Etiketleri

TABLOLAR:

- **<table>** Tablonun başlangıç ve bitiş etiketidir.
- **<tr>** Tablo satırı ekleme etiketidir.
- **<td>** Tablo veri hücresi ekleme etiketidir. Sütun ekleme olarak da düşünebiliriz.
- **<th>** Tablo başlığı ekleme etiketidir.
- **<thead>** Tablo başlığını belirgin hale getirmek için kullanılır. Görünüşe etkisi yoktur.
- **<tbody>** Tablo gövdesini belirgin hale getirmek için kullanılır. Görünüşe etkisi yoktur.

```
<table border="2">
  <thead>
    <tr>
      <th>İsim</th>
      <th>Ödev</th>
      <th>Proje</th>
      <th>Vize</th>
      <th>Final</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Volkan</td>
      <td> 50 </td>
      <td> 90 </td>
      <td> 80 </td>
      <td> 100 </td>
    </tr>
  </tbody>
</table>
```

İsim	Ödev	Proje	Vize	Final
Volkan	50	90	80	100

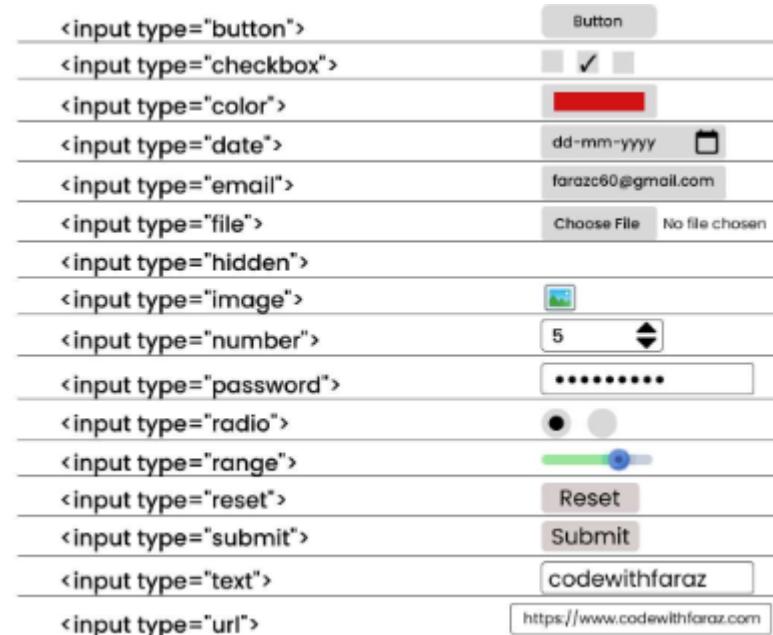
HTML Etiketleri

FORMLAR:

- **input**: Kullanıcının bilgi girişi yapması gereken yerlerde kullanılır.
- **type**: Formun türünü belirler.
- **text**: Metin girişi yapılması içindir.
- **name**: Değişken değerini belirler. Her input değeri için farklı isim vermemiz zorunludur.
- **value**: input elemanı içerisinde varsayılan olarak yazı yazmamızı sağlar

Bazı input ve textarea parametreleri:

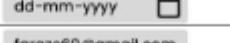
- **autofocus**: İlgili metin alanını seçili hale getirir ve böylece görsel olarak öne çıkmasını sağlar.
- **disabled**: Veri girişini pasif hale getirir.
- **readonly**: Sadece okunabilir metin gözükür. Bu alanı fare ile seçemeyiz.
- **required**: Doldurulması zorunlu alanlarda bu parametreyi kullanırız.



HTML Etiketleri

input type parametresine ait diğer özellikler:

- **password:** Şifreli alanlarda bu özelliği kullanırız. Karakterler yıldız şeklinde gözükmür.
- **reset:** Formdaki girilmiş tüm verileri temizler.
- **file:** Dosya yükleme butonudur. Bu butona tıklayınca bir pencere açılacak yüklenecek dosya buradan seçilecektir.
- **hidden:** Gizlemek istediğimiz kısımlar için bu özelliğe kullanırız.
- **image:** Gönder butonu yerine resim koymak için bu özelliğe kullanırız.
- **url:** Web adresi girişi için kullanılırız.
- **tel:** Telefon numarası girişi için kullanılırız.
- **number:** Sayı seçimleri için kullanırız. Aşağı yukarı oklar gözükmür.
- **range:** Sürgü ile minimum ve maximum değerler seçmek için kullanırı.
- **color:** Renk paletinden renk seçeriz.
- **search:** Arama motoru için bu metin kutusunu kullanırız.
- **email:** Mail adresi girişi için kullanılırız.
- **date:** Gün, ay ve yıl seçmemizi sağlar.
- **month:** Ay seçmemizi sağlar.
- **week:** Hafta seçmemizi sağlar.
- **time:** Saat ve dakika seçmemizi sağlar.
- **datetime-local:** Tek bir seferde gün, ay, yıl, saat ve dakika seçmemizi sağlar.
- **button:** Buton oluşturmak için bu özelliğe kullanırız.

<input type="button">	
<input type="checkbox">	<input checked="" type="checkbox"/> 
<input type="color">	
<input type="date">	dd-mm-yyyy 
<input type="email">	farazc60@gmail.com
<input type="file">	Choose File No file chosen
<input type="hidden">	
<input type="image">	
<input type="number">	5 
<input type="password">	*****
<input type="radio">	<input checked="" type="radio"/> 
<input type="range">	
<input type="reset">	Reset
<input type="submit">	Submit
<input type="text">	codewithfaraz
<input type="url">	https://www.codewithfaraz.com



HTML Etiketleri

İsim:

John

Last Soyad:

"Güngör"

Gönder

```
<form action="/action_page.php">
  <label for="ad">İsim:</label><br>
  <input type="text" id="ad" name="ad" value="John"><br>
  <label for="soyad">Last Soyad:</label><br>
  <input type="text" id="soyad" name="soyad" value="Güngör"><br><br>
  <input type="submit" value="Gönder">
</form>
```

İsim:

John

Last Soyad:

"Güngör"

Gönder

action: Formun gönderileceği adresin yazılıdığı parametredir. Eğer yukarıdaki kutuya bir şeyle yazıp gönder butonuna tıklaysanız, girdığınız bilgileri "kullanici.php" adlı veri dosyasına göndermiş olursunuz.

method: Formun gönderilme yöntemidir. "**GET**" metodu kullanıldığında istekler adres satırında görüntülenir. Bir de "**POST**" metodu vardır. Simdilik bunlara değinmeyeceğiz.

target= "_blank": Form yeni bir pencerede açılır.

value: Butonun üstünde ne yazacağını belirler.



HTML Etiketleri

- **VIDEO EKLEME:**

```
<video src="dosya.mp4" type="audio/mp4" width="320px"  
height="240px" poster="foto.jpg" loop autoplay muted  
controls> </video>
```

- **SES EKLEME:**

```
<audio src="bethoven.mp3" type="audio/mpeg" loop autoplay  
muted controls> </audio>
```



Link ve Kaynaklar

- <https://htmlcheatsheet.com/>
- Git eğitimi:
<https://www.youtube.com/watch?v=rWGZoT7fePg>
- CSS
 - <https://css-tricks.com/>
 - <https://flexboxfroggy.com/#tr>
- <https://codepen.io/>

PHP

Personal Home Page *Hypertext Preprocessor*

MBP 192 Internet Programlama

Bahar 25-26

iEU Bilgisayar Prog.





PHP Nedir?

- HTML içine gömülebilen bir betik dilidir
- PHP Web Tabanlı ve Nesne yönelimli bir programlama dilidir.
- PHP büyük işletim sistemlerinde; **Linux**, birçok **Unix** türevi (HP-UX, Solaris, OpenBSD vb.), **Microsoft Windows**, **Mac OS**, çok çeşitli platformlarda çalışabilir.
- “Apache, IIS, nginx” gibi HTTP sunucularının büyük kısmını destekler.



Neden saat değişti?

- index.html

```
<h1>Merhaba, bu HTML</h1>
<p>Saat: ????? </p>
```

Merhaba, bu HTML

Saat: ?????

- index.php

```
<h1>Merhaba, bu PHP</h1>
<p>Saat:
<?php echo date('H:i:s'); ?>
</p>
```

Merhaba, bu PHP

Saat: 21:24:30

Bilgisayarlarda PHP bulunmuyorsa – Online Compiler



- HTML ile düzgün çalışan:

https://www.w3schools.com/php/phptryit.asp?filename=tryphp_compiler

- Alternatif:

- <https://www.programiz.com/php/online-compiler/>
- <https://onecompiler.com/php>



Sistemde PHP kontrolü

- php -v : işletim sisteminde yüklü PHP versiyonu

```
C:\Users\RiverIsland>php -v
PHP 8.2.12 (cli) (built: Oct 24 2023 21:15:15) (ZTS Visual C++ 2019 x64)
Copyright (c) The PHP Group
Zend Engine v4.2.12, Copyright (c) Zend Technologies
```



PHP Syntax ve Temel Kurallar

- **Kural 1:** PHP kodları `<?php`
`?>` arasında yazılır

```
<?php  
    // PHP kodu buraya  
?>
```

- **Kural 2:** Her satır **noktalı virgül**
(;) ile **biter**

```
<?php  
    echo "Merhaba"; // ✓ Doğru  
    echo "Dünya"   // ✗ Yanlış - noktalı virgül yok  
?>
```

- **Kural 3:** PHP küçük/büyük harf duyarlıdır.

- `index.php` ✓
- `index.html` ✗ (PHP çalışmaz)

- **Kural 5:** Kod yazımında yorum satırları kullanılır.
Okunabilir kod!

```
<?php  
    echo "Merhaba Dünya"; // Bu bir açıklama satırıdır.  
    # Bu da başka bir tek satırlı yorum örneğidir.  
    /* Bu bölümdeki kodlar,  
       berber randevu sistemindeki  
       veritabanı bağlantısını sağlar.  
    */  
    echo "Bağlantı başarılı.";  
?>
```



echo = Ekrana yazdır

Ekrana

- “Merhaba Dünya!”
- $5 + 3$ toplamını
- H1 başlığı olarak
“HTML de
yazdırabilir”

yazdıralım...

```
<?php  
echo "Merhaba Dünya";  
echo 5 + 3;  
echo "<h1>HTML de  
yazdırabilir</h1>";  
?>
```



HTML ile PHP içi içe çalışması

HTML içine PHP Gömme

```
<!DOCTYPE html>
<html>
<head>
    <title>PHP</title>
</head>
<body>
    <h1>Merhaba MBP 192 SINIFI</h1>
    <p>
        <?php echo "Emrah Yüksel"; ?>
    </p>
    <p>
        Lorem ipsum...
        <?php echo " Bu kod HTML etiketi
arاسına eklendi "; ?>
        Excepteur sint...
    </p>
</body>
</html>
```

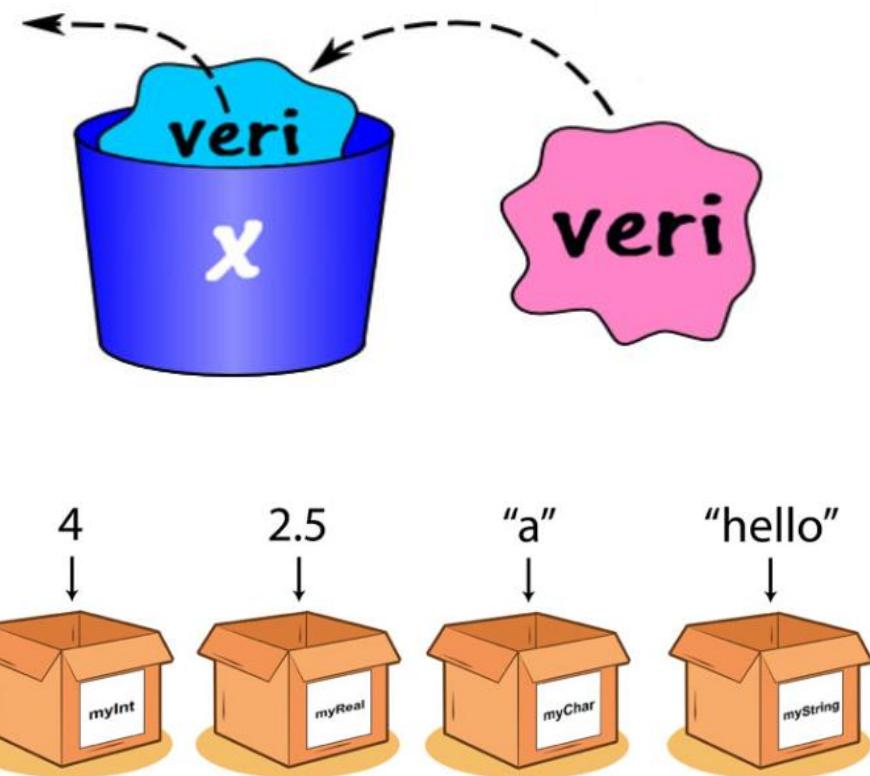
PHP İçinden HTML Çıktısı Verme - " " dikkat

```
<!DOCTYPE html>
<html>
<head>
    <title>PHP</title>
</head>
<body>
    <?php
        echo "<h1>Merhaba MBP 192 </h1>";
        echo "<p> PHP Dersi </p>";
        echo "<p> Lorem ipsum...
            Bu kod HTML etiketi arasına
eklendi
            Excepteur sint... </p>";
    ?>
</body>
</html>
```

Kod: php-ders-sunum-kodlari >> 2-html_ic_ice.php, 3-
html_ic_ice.php

Değişkenler

- Temek kullanım amacı, bir veriyi program boyunca tekrar tekrar yazmak yerine, ona bir isim atayıp o isimle çağrılmaktır.
 - Veri Yönetimi:** Değişkenin değerini bir kez değiştirdiğinizde, o değişkeni kullandığınız her yerde değer güncellenmiş olur
 - Okunabilirlik:** Karmaşık veriler yerine anlamlı isimler (\$fiyat, \$musteri_adi) kullanarak kodun ne yaptığı kolayca anlaşılabilir.



Değişkenler ve Veri Tipleri

Değişken Tanımlama Kuralları

- Tüm değişkenler **\$** işaretini ile başlar sonra değişken ismi yazılır.
- Değişken adları **bir harf** veya **alt çizgi (_) ile başlamalıdır.**
- Sayı ile başlayamazlar** ancak içinde sayı barındırabilirler.
- Büyük-küçük harfe duyarlıdırlar
 - (\$ad ile \$AD farklı değişkenlerdir).

Değişken Adı	Durum	Neden?
\$randevuID	Uygun	Harf ile başlar, standart kurallara uyar.
\$_musteri	Uygun	Alt çizgi ile başlayabilir.
\$fiyat_2026	Uygun	İçinde sayı barındırabilir.
\$2randevu	Uygun Değil	Sayı ile başlayamaz.
\$toplam-tutar	Uygun Değil	Tire (-) karakteri içeremez (alt çizgi kullanılmalıdır).
\$ad soyad	Uygun Değil	Arada boşluk bırakılamaz.
\$değişken	Riskli	Türkçe karakter kullanımı teknik olarak çalışsa da önerilmez.

```
syntax error, unexpected integer "8", expecting variable or "{" or "$"
View Problem (Alt+F8) No quick fixes available
<?php $8 = 9; ?>
<?php echo $8; ?>
```

- Kod:** php-ders-sunum-kodlari >> değişkenler.php

Değişken Tanımlama Örnekleri



Doğru

- **\$randevuID:** Harf ile başlar, standartlara uygundur.
 - `$randevuID = 101; echo "Randevu Numarası: " . $randevuID;`
- **\$_musteri:** Alt çizgi ile başlama
 - `$_musteri = "Volkan"; echo "Müşteri Adı: " . $_musteri;`
- **\$fiyat_2026:** İçinde sayı barındırabilir
 - `$fiyat_2026 = 250.00; echo "Gelecek Yıl Fiyatı: " . $fiyat_2026;`

Hatalı

- **\$2randevu:** Sayı ile başlayamaz
 - `// HATALI KULLANIM $2randevu = "14:30"; // PHP bu satırda 'Parse error: syntax error, unexpected '2'' hatası verir.`
- **\$toplam-tutar:** PHP, tire işaretini (-) çıkarma işlemi olarak algıla
 - `// HATALI KULLANIM $toplam-tutar = 500; // PHP burada '$toplaml' değişkeninden 'tutar'ı çıkarmaya çalıştığını sanır ve hata verir.`
- **\$ad soyad:** Arada boşluk bırakılamaz.
 - `// HATALI KULLANIM $ad soyad = "Mustafa Volkan"; // PHP iki farklı kelime gördüğü için ne yapacağını bilemez.`
- **\$değişken:** Türkçe karakter kullanımı.(önerilmez)
 - `// RİSKLİ KULLANIM $ücret = 100; // Teknik olarak PHP 7+ sürümlerinde çalışır ancak dosya kodlaması (UTF-8) değiştirse kodunuz bozulabilir. // Önerilen: $ucret = 100;`

\$degisken_adi = degisken_degeri



Değişken Veri Tipleri ve Tanımlama

Değişken Yazım Standartları
(CamelCase vs Snake_Case)

Stil	Örnek
CamelCase	\$randevuTarihi
Snake_Case	\$randevu_tarihi

var_dump(), bir değişkenin yapısı hakkında hem **değerini** hem de **türünü** (veri tipini) gösteren fonksiyondur

- **String (Metin):** Tırnak içerisinde yazılan her türlü metinsel veridir.
 - `$isim = "Volkan";`
- **Integer (Tam Sayı):** Ondalık kısmı olmayan pozitif veya negatif sayılardır.
 - `$yil = 2026;`
- **Float / Double (Ondalıklı Sayı):** Virgüllü (noktalı) sayılardır.
 - `$fiyat = 150.50;`
- **Boolean (Mantıksal):** Genellikle karar yapılarında kullanılır. Sadece iki değer alabilir:
 - `True` (doğru) veya `False` (yanlış).
- **Array (Dizi):** Tek bir değişken içinde birden fazla değer tutmamızı sağlar.
 - `$hizmetler = array("Saç Kesimi", "Sakal Tıraşı");`
- **Object (Nesne):** Dökümanlarda belirtilen "nesne yönelimli" yapının temelidir. Sınıflardan (class) türetilen verilerdir.
- **NULL:** Değişkenin hiçbir değer taşımadığını ifade eder.

Değişken Tanımlama ve Veri Tipleri



- \$hizmet_adi = "Saç ve Sakal Kesimi"; (String)
- \$sure_dakika = 45; (Integer)
- \$ucret = 350.50; (Float)
- \$kampanya_var_mi = true; (Boolean)
- \$ekipmanlar = ["Makas", "Ustura", "Tarak"];

```
$hizmet_adi = "Saç ve Sakal Kesimi"; // (String)
echo "<h3>$hizmet_adi = \"Saç ve Sakal Kesimi\"; </h3>";
echo "Hizmet Adı: " . $hizmet_adi . "<br>";
var_dump($hizmet_adi);

$sure_dakika = 45; // (Integer)
echo '<h3>$sure_dakika = 45; </h3>';
echo "Hizmet Süresi: " . $sure_dakika . "<br>";
var_dump($sure_dakika);

$ucret = 350.50; // (Float)
echo '<h3>$ucret = 350.50; </h3>';
echo "Hizmet Ücreti: " . $ucret . "<br>";
var_dump($ucret);

$kampanya_var_mi = true; // (Boolean)
echo '<h3>$kampanya_var_mi = true; </h3>';
echo "Kampanya Var mı?: " . ($kampanya_var_mi ? "Evet" : "Hayır") . "<br>";
var_dump($kampanya_var_mi);

$ekipmanlar = ["Makas", "Ustura", "Tarak"]; // (Array)
echo '<h3>$ekipmanlar = ["Makas", "Ustura", "Tarak"]; </h3>';
var_dump($ekipmanlar);
```

- **Kod:** php-ders-sunum-kodlari >> veri_tipleri.php

- **Kod:** php-ders-sunum-kodlari >> veri_tipleri.php



Object (Nesne) veri tipi

- PHP'deki **Object (Nesne)** veri tipi, dökümanlarınızda vurgulanan "Nesne Yönelimli" (Object-Oriented) programlama yaklaşımının temelidir.
- Bir nesne, sadece bir değer değil; o değere ait **özellikleri (properties)** ve bu özellikler üzerinde işlem yapabilen **işlevleri (methods)** bir arada tutan kompleks bir yapıdır.
- Nesne veri tipini kullanabilmek için önce bir şablon olan **Sınıf (Class)** tanımlanmalı, ardından bu sınıfın bir **örnek (instance)** üretilmelidir.

```
<?php
// 1. Sınıf (Şablon) Tanımlama
class Berber {
    public $isim;
    public $uzmanlik;

    // Nesne oluşturulduğunda çalışan fonksiyon
    public function __construct($ad, $alan) {
        $this->isim = $ad;
        $this->uzmanlik = $alan;
    }

    public function selamVer() {
        return "Merhaba, ben " . $this->isim . ". Uzmanlık
alanım: " . $this->uzmanlik;
    }
}

// 2. Nesne (Object) Oluşturma
$berberNesnesi = new Berber("Volkan", "Saç Tasarımcısı");

// 3. Yapısını İnceleme
echo "<h2>Object Veri Tipi Yapısı:</h2>";
var_dump($berberNesnesi);

echo "<br><br>";
// Nesne içindeki metoda erişim
echo $berberNesnesi->selamVer();
?>
```

Sınıf (Class): Nesnenin nasıl olacağını belirleyen genel plandır

Nesne (Object): Bu plandan üretilmiş, hafızada yer kaplayan somut veridir.

var_dump Çıktısı: Bu komutu kullandığınızda, PHP size değişkenin tipinin object olduğunu, hangi sınıfa ait olduğunu ve içindeki verileri detaylıca gösterir.



Değişken Kapsamı (Scope)

- **Değişken kapsamı (variable scope)**, bir değişkenin kodun hangi bölmelerinden erişilebilir olduğunu belirleyen kurallar bütünüdür.
- Değişkenlerin nerede tanımlandığı, onların ömrünü ve görünürüğünü doğrudan etkiler.
- Kapsam Tipleri:
 - Yerel Kapsam (Local Scope)
 - Global Kapsam (Global Scope)
 - Statik Kapsam (Static Scope)

Kapsam Türü	Tanımlandığı Yer	Erişilebilirlik
Yerel (Local)	Fonksiyon içi	Sadece o fonksiyon içinden.
Global	Fonksiyon dışı	Fonksiyonlar hariç her yer (fonksiyon içinde <code>global</code> ile).
Statik (Static)	Fonksiyon içi	Sadece o fonksiyon içinden ama değeri sıfırlanmaz.

• Yerel Kapsam (Local Scope)

- Bir fonksiyon içerisinde tanımlanan değişkenler "yerel" değişkendir. Bu değişkenlere sadece tanımlandıkları fonksiyonun içinden erişilebilir. Fonksiyon bittiğinde bu değişkenler bellekten silinir.

• Global Kapsam (Global Scope)

- Herhangi bir fonksiyonun dışında tanımlanan değişkenler "global" değişkendir. Bu değişkenlere fonksiyonların dışındaki her yerden erişilebilir. Ancak, bir fonksiyonun içinden global bir değişkene erişmek için global anahtar kelimesi kullanılmalıdır.
- **\$GLOBALS Superglobal** PHP, tüm global değişkenleri `$GLOBALS['degisken_adi']` isimli bir dizide saklar. Fonksiyon içinde global yazmak yerine bu diziyi de kullanabilirsiniz.

• Statik Kapsam (Static Scope)

- Normalde bir fonksiyonun çalışması bittiğinde içindeki yerel değişkenler yok edilir. Ancak bir değişkenin değerini kaybetmemesini, fonksiyon her çağrılığında kaldığı yerden devam etmesini istiyorsanız static anahtar kelimesini kullanırsınız.

- **Kod:** `php-ders-sunum-kodlari >> degisken_kapsam.php`



Örnek

Adınızı, soyadınızı, yaşıınızı, bölümünüzü farklı değişkenlere metin(string) olarak atayın.

Mezuniyet durumunuzu "mezun" isimli bir değişkene mantıksal(Boolean") olarak atayın.

Ekrana

Merhaba, ben ...ad soyad...

Yaşım:

Bölümüm:

Mezunsanız "Mezunoldum!"

Değilseniz "Hala öğrenciyim." olarak yazdırın

```
<?php
$ad = "Ayşe";
$soyad = "Yılmaz";
$yas = 19;
$universite = "İzmir Ekonomi
Üniversitesi";
$mezun = false;

echo "Merhaba, ben $ad $soyad<br>";
echo "Yaşım: $yas<br>";
echo "Okulum: $universite<br>";
if ($mezun) {
    echo "Mezun oldum!";
}
else {
    echo "Hala öğrenciyim.";
}
?>
```

OPERATÖRLER

- PHP'de operatörler, değişkenler ve değerler üzerinde işlemler gerçekleştirmek için kullanılan özel sembollerdir.
 - Aritmetik Operatörler
 - Atama Operatörleri
 - Karşılaştırma Operatörleri
 - Mantıksal Operatörler
 - String (Metin) Operatörleri
 - Artırma ve Azaltma Operatörleri
 - Hata Operatörü
-
- **Kod:** `php-ders-sunum-kodlari >> operatorler.php`

Aritmetik Operatörler

Sayısal değerlerle standart matematiksel işlemler yapmak için kullanılırlar.

Operatör	İsim	Örnek	Sonuç
+	Toplama	<code>\$a + \$b</code>	Toplam
-	Çıkarma	<code>\$a - \$b</code>	Fark
*	Çarpma	<code>\$a * \$b</code>	Çarpım
/	Bölme	<code>\$a / \$b</code>	Bölüm
%	Modül	<code>\$a % \$b</code>	Bölümden kalan
**	Üs Alma	<code>\$a ** \$b</code>	a 'nın b . kuvveti

```
<!DOCTYPE html>
<html>
<body>
    <h2>Matematiksel İşlemler</h2>
    <?php
        $a = 10;
        $b = 3;

        echo "A sayısı: $a, B sayısı: $b <br><br>";

        echo "Toplama ($a + $b): " . ($a + $b) . "<br>";
        echo "Çıkarma ($a - $b): " . ($a - $b) . "<br>";
        echo "Çarpma ($a * $b): " . ($a * $b) . "<br>";
        echo "Bölme ($a / $b): " . ($a / $b) . "<br>";
        echo "Modül ($a % $b): " . ($a % $b) . "<br>";
        echo "Üs Alma ($a ** $b): " . ($a ** $b) . "<br>";

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

- **Kod:**

- `php-ders-sunum-kodlari >> operatorler.php`
- `php-ders-sunum-kodlari >> aritmetik_operatorleri.php`

Atama Operatörleri

- Değişkenlere değer atamak için kullanılırlar.
- Bileşik atama operatörleri kodu kısaltmanıza yardımcı olur.

Operatör	Açıklama	Örnek İşlem
=	Atama İşlemi	<?php \$adSoyad="Emrah Yüksel"; \$sayi=5; \$deger=TRUE; ?>
+=	Toplayarak Atama	<?php \$sayi=5; echo \$sayi+=10; // İşlem Sonucu 10 ?>
-=	Çıkararak Atama	<?php \$sayi=5; echo \$sayi-=10; // İşlem Sonucu -5 ?>
=	Çarparak Atama	<?php \$sayi=5; echo \$sayi=10; // İşlem Sonucu 50 ?>
/=	Bölgerek Atama	<?php \$sayi=5; echo \$sayi/=10; // İşlem Sonucu 0,5 ?>
%	Mod Alarak Atama	<?php \$sayi=5; echo \$sayi%=10; // İşlem Sonucu 5 ?>
.=	Birleştirerek Atama	<?php \$ad="Emrah"; echo \$ad.=" Yüksel"; // İşlem Sonucu Emrah Yüksel ?>

- Kod:

- php-ders-sunum-kodlari >> operatorler.php
- php-ders-sunum-kodlari >> atama_operatorleri.php

Karşılaştırma Operatörleri

İki değeri karşılaştırmak için kullanılır ve sonuç olarak **true** veya **false** (boolean) döndürürler.

`==` : Eşittir (Sadece değer kontrolü).

`===` : Özdeştir (Hem değer hem de **veri tipi** kontrolü).

`!=` veya `<>` : Eşit değildir.

`!==` : Özdeş değildir.

`>`, `<`, `>=`, `<=` : Büyüklük/Küçüklük kontrolleri.

`<>` (**Spaceship**): Sol taraf küçükse `-1`, eşitse `0`, büyükse `1` döndürür.

Operatör	Açıklama	Örnek İşlem
<code>==</code>	Eşittir	<code>\$a == \$b</code>
<code>===</code>	Denktir	<code>\$a === \$b</code>
<code>!=</code>	Eşit değildir	<code>\$a != \$b</code>
<code>!==</code>	Denk değildir	<code>\$a !== \$b</code>
<code><></code>	Eşit değildir	<code>\$a <> \$b</code>
<code><</code>	Küçüktür	<code>\$a < \$b</code>
<code>></code>	Büyüktür	<code>\$a > \$b</code>
<code><=</code>	Küçük ve eşittir	<code>\$a <= \$b</code>
<code>=></code>	Büyük ve eşittir	<code>\$a => \$b</code>

- Kod:

- [php-ders-sunum-kodlari >> operatorler.php](#)
- [php-ders-sunum-kodlari >> karsilastirma_operatorleri.php](#)

Mantıksal Operatörler

- Birden fazla koşulu birleştirmek için kullanılır.
- **ÖRNEK:** Berber randevu sisteminizde
 - «*Eğer kullanıcı giriş yapmışsa VE admin ise»*

&& (and): Ve (Her iki koşul da doğru olmalı).

|| (or): Veya (En az bir koşul doğru olmalı).

! (not): Değil (Koşulun tersini alır).

Operatör	Açıklama	Örnek İşlem
\$a and \$b	Ve	\$a ve \$b her ikisi de doğruysa sonuç doğrudur (TRUE).
\$a or \$b	Veya	\$a veya \$b doğruysa sonuç doğrudur.
\$a xor \$b	Ayrıcalıklı Veya	\$a veya \$b doğruysa sonuç doğru, her ikiside doğruysa sonuç yanlıştır (FALSE).
! \$a	Değil	\$a doğru değilse sonuç doğrudur.
\$a && \$b	Ve	\$a ve \$b her ikisi de doğruysa sonuç doğrudur.
\$a \$b	Veya	\$a veya \$b doğruysa sonuç doğrudur.

- Kod:

- [php-ders-sunum-kodlari >> operatorler.php](#)
- [php-ders-sunum-kodlari >> mantiksal_operatorler.php](#)

Artırma ve Azaltma Operatörleri

Değişken değerini bir artırmak veya azaltmak için kullanılır.

`++$x` : Önce artır, sonra işlem yap.

`$x++` : Önce işlem yap, sonra artır.

`--$x` / `$x--` : Azaltma işlemleri.

```
$a = 5;  
echo ++$a; // Önce 6 yapar, sonra ekrana 6 yazar.
```

```
$a = 5;  
echo $a++; // Önce ekrana 5 yazar, işlem bitince arka planda $a'yı 6 yapar.  
echo $a; // Şimdi yazdırırsak sonuç 6 olur.
```

- Kod:
 - [php-ders-sunum-kodlari >> operatorler.php](#)
 - [php-ders-sunum-kodlari >> artirma-azaltma-operatoru.php](#)
 - [php-ders-sunum-kodlari >> artirma-azaltma-operatorleri.php](#)



Artırma ve Azaltma Operatörleri

KOD:

php-ders-sunum-kodlari >>
artirma_azaltma_operatorleri.php

```
<?php  
$sayi = 10;  
echo "Başlangıç: " . $sayi . "<br>"; // 10  
// Önce artırma akışı  
echo "Önce artır (++$sayi): " . ++$sayi . "<br>"; //  
Ekranda 11 görürüz.  
// Sonradan artırma akışı  
echo "Sonra artır ($sayi++): " . $sayi++ . "<br>"; //  
Ekranda hala 11 görürüz (çünkü işlemden sonra artacak).  
echo "Artırma sonrası son durum: " . $sayi . "<br>"; //  
Şimdi 12 olduğunu görürüz.  
// Azaltma akışı  
echo "Önce azalt (--$sayi): " . --$sayi; // Ekranda 11  
görürüz.  
?>
```

Örnek Kod	İşlem Akışı	Sonuçlar
<code>\$y = ++\$x;</code>	1. <code>\$x</code> artar 2. Yeni <code>\$x</code> , <code>\$y</code> 'ye atanır.	<code>\$x</code> ve <code>\$y</code> aynı olur.
<code>\$y = \$x++;</code>	1. Mevcut <code>\$x</code> , <code>\$y</code> 'ye atanır 2. <code>\$x</code> artar.	<code>\$y</code> eski değerde kalır, <code>\$x</code> artar.



String (Metin) Operatörleri

Metinleri birleştirmek için kullanılır. PHP'nin "**HTML içine gömülebilen**" yapısında çıktı üretirken sıkça kullanılır.

```
<?php
$mesaj = "Merhaba";
$mesaj .= " Dünya!"; // $mesaj = $mesaj . " Dünya!" işleminin kısa halidir.
echo $mesaj; // Çıktı: Merhaba Dünya!
?>

<?php
$ders_kodu = "MBP 192";
echo "<h1>" . $ders_kodu . " İnternet Programlama</h1>";
?>
```

- **Kod:**
 - [php-ders-sunum-kodlari >> operatorler.php](#)
 - [php-ders-sunum-kodlari >> metin_operatorleri.php](#)

SORU

- Aşağıdaki PHP kod bloğu çalıştırıldığında ekrana ne yazdırılır?

```
<?php
$sayi = 10;
$sayi += 5;
$sonuc = $sayi % 4;
echo "İşlem Sonucu: " . $sonuc;
?>
```

SORU

- Aşağıdaki PHP kod bloğu çalıştırıldığında ekrana ne yazdırılır? Beklenen çıktı nedir?

```
<?php
$puan = 50;
$ad = "Öğrenci";
echo $ad . " puanı: " . $puan++;
echo " | Yeni durum: " . $puan;
?>
```

SORU

- Bir randevu sisteminde, aşağıdaki koşulun true (doğru) dönmesi için hangi durumun gerçekleşmesi gereklidir?

```
<?php
if ($giris_yapildi == true && ($yetki == "admin" || $ucret <= 0)) {
    echo "Erişim başarılı.";
}
?>
```

- **A)** Kullanıcı giriş yapmamışsa ama admin ise.
- **B)** Kullanıcı giriş yapmışsa VE (ya admin ise YA DA ücret o ve altındaysa).
- **C)** Ücreti o olan herkes giriş yapabilir.
- **D)** Sadece adminler giriş yapabilir.

Tür dönüşümleri (Type casting)

Tür Dönüşümü Akışı ve
Dikkat Edilmesi Gerekenler

- Dönüşüm işlemleri yapılırken veri kaybı yaşanabileceği unutulmamalıdır
- Veritabanından gelen tüm veriler genellikle "string" olarak döner.
 - Bu verileri matematiksel işlemlerde kullanmadan önce **(int)** veya **(float)** ile dönüştürmek, uygulamanızın daha kararlı çalışmasını sağlar.

- **Otomatik Tür Dönüşümü (Type Juggling)**
 - işlem sırasında bir değişkenin türünü o anki bağlama göre kendisi belirler.
- **Manuel Tür Dönüşümü (Type Casting):**
 - Geliştiricinin, bir değişkenin türünü açıkça belirtmesi işlemidir.
 - Değişkenin önüne parantez içinde hedef tür yazarak gerçekleştirilir.
- **Özel Dönüşüm Fonksiyonları:**
 - Daha kontrollü dönüşümler yapmak için PHP'nin sunduğu hazır fonksiyonlardır

Otomatik Tür Dönüşümü (Type Juggling)

İşlem sırasında bir değişkenin türünü o anki bağlama göre kendisi belirler.

```
$sayi = 5;          // Integer
$metin = "10";     // String
$sonuc = $sayi + $metin; // Sonuç: 15 (Integer)
```

```
echo "<h3>Tam Sayıya Dönüşürme (int) veya (integer)</h3><br>";
$fiyat = 150.99;
$tamFiyat = (int)$fiyat;
// Sonuç: 150 (Veri kaybı yaşanır, .99 kısmı silinir)

$metinSayi = "42isik";
$sayı = (int)$metinSayi;
// Sonuç: 42 (Metnin başındaki sayıyı alır)
$metinSayi2 = "isik42";
$sayı2 = (int)$metinSayi2;
// Sonuç: 0 (Metnin başında sayı olmadığı için sıfır olur)
echo "Fiyat: " . $fiyat . " → Tam Fiyat: " . $tamFiyat; // Fiyat: 150.99 → Tam Fiyat: 150
echo "<br>";
echo "Metin Sayı: " . $metinSayi . " → Sayı: " . $sayı; // Metin Sayı: 42isik → Sayı: 42
echo "<br>";
echo "Metin Sayı 2: " . $metinSayi2 . " → Sayı 2: " . $sayı2; // Metin Sayı 2: isik42 → Sayı 2: 0
echo "<br>";
```

- Kod: php-ders-sunum-kodlari >> tur_donusumu.php



Manuel Tür Dönüşümü (Type Casting)

Dönüşüm Türü	Kullanım	Açıklama
(int), (integer)	<code>(int)\$degisken</code>	Tam sayıya dönüştürür.
(bool), (boolean)	<code>(bool)\$degisken</code>	Mantıksal değere (true/false) dönüştürür.
(float), (double)	<code>(float)\$degisken</code>	Ondalıklı sayıya dönüştürür.
(string)	<code>(string)\$degisken</code>	Metne (karakter dizisine) dönüştürür.
(array)	<code>(array)\$degisken</code>	Diziye dönüştürür.
(object)	<code>(object)\$degisken</code>	Nesneye dönüştürür.

- Kod: php-ders-sunum-kodlari >> tur_donusumu.php

Özel Dönüşüm Fonksiyonları



Verileri daha kontrollü ve fonksiyonel bir şekilde dönüştürmek için özel fonksiyonlar kullanılır.

- **settype():**
 - değişkenin türünü kalıcı olarak değiştirir.

The screenshot shows the official PHP documentation for the `settype()` function. It includes sections for **Description**, **Parameters**, and **Return Values**. The **Description** section contains the function signature `settype(mixed $var, string $type): bool` and a brief description: "Set the type of variable `var` to `type`". The **Parameters** section details two parameters: `var` (the variable being converted) and `type` (possible values include boolean, integer, float, string, array, object, or null). The **Return Values** section states that it returns `true` on success or `false` on failure.

<https://www.php.net/manual/en/function.settype.php>

Değer Döndüren Fonksiyonlar (**intval**, **floatval**, **strval**)

- Bu fonksiyonlar değişkenin değerini alır, istenen tipe çevirir ve sonucu döndürür; ancak orijinal değişkene dokunmazlar
 - **intval(\$degisken):** Değişkenin tam sayı değerini döndürür.
 - **floatval(\$degisken):** Değişkenin ondalıklı sayı değerini döndürür.
 - **strval(\$degisken):** Değişkenin metin (string) değerini döndürür.



Özel Dönüşüm Fonksiyonları

`intval(mixed $value, int $base = 10): int`

Returns the `int` value of `value`, using the specified `base` for the conversion (the default is base 10). `intval()` should not be used on objects, as doing so will emit an `E_WARNING` level error and return 1.

Parameters

value
The scalar value being converted to an integer

base
The base for the conversion

Note:
If `base` is 0, the base used is determined by the format of `value`:
◦ if string includes a "0x" (or "0X") prefix, the base is taken as 16 (hex); otherwise,
◦ if string starts with a "0b" (or "0B"), the base is taken as 2 (binary); otherwise,
◦ if string starts with "0", the base is taken as 8 (octal); otherwise,
◦ the base is taken as 10 (decimal).

Return Values

The integer value of `value` on success, or 0 on failure. Empty arrays return 0, non-empty arrays return 1.

The maximum value depends on the system. 32 bit systems have a maximum signed integer range of -2147483648 to 2147483647. So for example on such a system, `intval('100000000000')` will return 2147483647. The maximum signed integer value for 64 bit systems is 9223372036854775807.

Strings will most likely return 0 although this depends on the leftmost characters of the string. The common rules of `integer casting` apply.

Değer Döndüren Fonksiyonlar (`intval`, `floatval`, `strval`)

- Bu fonksiyonlar değişkenin değerini alır, istenen tipe çevirir ve sonucu döndürür; ancak orijinal değişkene dokunmazlar
 - **`intval($degisken)`:**
Değişkenin tam sayı değerini döndürür.
 - **`floatval($degisken)`:**
Değişkenin ondalıklı sayı değerini döndürür.
 - **`strval($degisken)`:**
Değişkenin metin (string) değerini döndürür.

<https://www.php.net/manual/en/function.intval.php>

KOD: php-ders-sunum-kodlari >>ozel_tur_donusum_fonk.php⁴⁴



SORU??

Aşağıdaki PHP kod bloğu çalıştırıldığında tarayıcı ekranında (çıktıda) ne görünür? Lütfen nedenlerini açıklayarak belirtiniz.

```
<?php  
$gelen_ucret = "350"; // Formdan gelen veri (String)  
$hizmet_bedeli = 350; // Sistemdeki sabit değer (Integer)  
  
echo "1. Sonuç: ";  
var_dump($gelen_ucret == $hizmet_bedeli);  
  
echo "<br>2. Sonuç: ";  
var_dump($gelen_ucret === $hizmet_bedeli);  
?>
```

Ödev 1

- PHP fonksiyonlarında echo ve print nasıl çalışır, farkı nedir?
- Metin/Text veri tipinin "" (çift tırnak) veya '' (tek tırnak) içinde yazılmasını farkları nelerdir?
- Metin yazarken "" (çift tırnak) içinde çift tırnak nasıl kullanılır? Yok sayma (escape) kaçış işaretti.

Sabitler

Programın çalışması boyunca değeri değiştirilemeyen isimlerdir.

Sabit Tanımlama:

define() Fonksiyonu ile veya **const** ile tanımlanır.

- Dolar İşareti Yoktur:** Değişkenlerin aksine sabitler \$ ile başlamaz.
- Global Kapsam:** Sabitler tanımlandıkları andan itibaren tüm sayfa ve fonksiyonlar içerisinde otomatik olarak "global" kapsamdadır
- Değiştirilemezlik:** Program akışında bir sabite yeni bir değer atamaya çalışmak hata almanızı neden olur.
- Yazım Kuralı:** genellikle tamamı **BÜYÜK HARF** ile yazılır.

```
<!DOCTYPE html>
<html>
<body>
    <?php
        // define("SABIT_ADI", "değer");
        define("VERITABANI", "berber_otomasyon");
        define("KDV_ORANI", 0.20);

        const SITE_URL = "https://www.google.com";

        echo "Bağlanılan Veritabanı: " . VERITABANI . "<br>";
        echo "Uygulanan KDV: " . (KDV_ORANI * 100) . "% <br>";

        echo "<a href='" . SITE_URL . "'>Google'a Git</a>";
    ?>
</body>
</html>
```

Bağlanılan Veritabanı: berber_otomasyon
Uygulanan KDV: 20%
[Google'a Git](https://www.google.com)



SORU

Aşağıdaki kod blogunda yapılan hata nedir?

```
<?php
define("PI", 3.14);
$yari_cap = 5;
PI = 3; // Hatalı satır
echo PI * ($yari_cap ** 2);
?>
```

PHP'de sabitler bir kez tanımlandıktan sonra değerleri değiştirilemez. `PI = 3;` satırı bir sabit üzerinde yeniden atama yapmaya çalıştığı için **Parse Error** (Sözdizimi Hatası) verecektir.



Sihirli Sabitler (Magic Constants)

Yazıldıkları yere göre değerleri değişen özel sabitlerdir.

Normal sabitlerin aksine, PHP yorumlayıcısı tarafından çalışma anında otomatik olarak atanırlar.

```
<!DOCTYPE html>
<html>
<body>
    <h2>PHP Sihirli Sabitler Testi</h2>
    <?php
        echo "Bu kodun bulunduğu satır: " . __LINE__ . "<br>"; //
        echo "Dosya konumu: " . __FILE__ . "<br>"; //

        function testFonksiyonu() {
            echo "Şu an çalışan fonksiyon: " . __FUNCTION__; //
        }

        testFonksiyonu();
    ?>
</body>
</html>
```

Bu sabitler her zaman **iki alt çizgi (__) ile başlar ve biter**

KOD: php-ders-sunum-kodlari >> sihirli_sabitler.php

Değişken, Sabit ve Sihirli Sabit Farkı



- **Değişken (\$puan):** Çalışma anında değeri değişimdir, dolar işaretiyile tanımlanır.
- **Sabit (define):** Değerini asla değiştirmez, uygulama genelinde (global) erişilebilirdir.
- **Sihirli Sabit (__LINE__):** Değerini, kodun hangi dosyada veya hangi satırda olduğuna göre PHP tarafından otomatik değiştirilir.



SORU

Aşağıdaki kodun çıktısı nedir,
açıklayınız?

```
1  <!DOCTYPE html>
2  <html>
3  <body>
4      <h2>PHP Sihirli Sabitler</h2>
5      <?php
6          echo __LINE__;
7          echo "<br>";
8          echo __LINE__;
9      ?>
10     </body>
11     </html>
```



FORMLAR

Bir web sayfasında kullanıcılardan bilgi alabilmek için `<form>` etiketine ihtiyaç duyuyoruz.

method / action / name

```
<form method="POST" action="formlar_baslangic_POST.php">
    <label>1. Sayı:</label><br>
    <input type="number" name="sayi1" required><br>
    <label>2. Sayı:</label><br>
    <input type="number" name="sayi2" required><br><br>

    <input type="submit" value="Topla">
</form>
```



FORM: method / action / name

- **method** = “POST” / “GET”
 - Verinin sunucuya nasıl gönderileceğini belirler.
 - “**POST**” yöntemi, verileri URL'nin içinde değil, HTTP isteğiinin gövdesinde (arka planda) gizli bir şekilde gönderir
 - “**GET**” yönetimi, verileri URL'nin içinde iletir ve değerler görünür.
- **action**=“****.php”;
 - formun içindeki veriler “Gönder” (Submit) butonuna basıldığında **bu verilerin hangi adrese (URL) gönderileceğini** belirler.
 - Eğer bu niteliği boş bırakırsanız veya hiç yazmazsanız, form verileri **bulunduğunuz sayfanın kendisine** tekrar gönderilir.



FORM: method / action / name

- **name** <input name="sayı1">
 - **name değeri**, PHP tarafından veriyi yakalamak için kullandığımız "anahtar" ismidir.
 - Kullanıcı formu gönderdiğinde, tarayıcı bir veri paketi oluşturur. Bu paket şu mantıkla çalışır:
name_değeri = kullanıcının_yazdığı_değer
 - Eğer bir input elemanına name vermezseniz, o kutuya ne yazılırsa yazılısın PHP tarafına **hiç iletilemez**.

Method / Action / Name ilişkisi



- **action** verinin **nereye** gideceğini belirlerken,
- **method** (GET veya POST) verinin **nasıl** (açıkta mı yoksa gizli mi) gideceğini belirler.
- **name** gönderilecek verinin form yapısında nereden alınacağını belirtir.

Özellik	HTML (name)	PHP Karşılığı
Görevi	Veriye kimlik/İsim verir.	Kimliği kullanarak veriyi okur.
Yazım Şekli	<code><input name="eposta"></code>	<code>\$_POST['eposta']</code>
Eksik Olursa	Veri sunucuya gönderilmez.	<code>Undefined index</code> hatası alırsınız.



Alınan değerin Tür dönüşümü

- Formlardan gelen veriler varsayılan olarak **"string" (metin)** gelir.
- Tür dönüşümü ile kontrol edilmesi veya tür dönüşümü yapılması, çalışma açısından önemlidir.
- PHP, aritmetik işlem sırasında **Otomatik Tür Dönüşümü** yaparak bunları sayıya çevirir.

KOD: php-ders-sunum-kodlari >>formlar_baslangic_POST.php
php-ders-sunum-kodlari >> formlar_baslangic_GET.php

FORM Süreci

```
<form method="GET" action="formlar_baslangic_GET.php">
    <label>1. Sayı:</label><br>
    <input type="number" name="sayi1" required><br>
    <label>2. Sayı:</label><br>
    <input type="number" name="sayi2" required><br><br>

    <input type="submit" value="Topla">
</form>
<?php
// Formun gönderilip gönderilmediğini kontrol edelim
if (isset($_GET['sayi1']) && isset($_GET['sayi2'])) {

    // Formdan gelen verileri değişkenlere atayalım
    $s1 = $_GET['sayi1'];
    $s2 = $_GET['sayi2'];

    // Aritmetik toplama işlemi
    $toplam = $s1 + $s2;

    // Sonucu HTML içine birleştirerek yazdıralım
    echo "<h3>İşlem Sonucu</h3>";
    echo "<p>$s1 + $s2 = <b>$toplam</b></p>";
}
?>
```





FORM ÖRNEK

POST

```
<!DOCTYPE html>
<html>
    <head><title>Formlar ve POST Metodu</title></head>
    <body>
        <form method="" action="formlar_baslangic_POST.php">
            <label>1. Sayı:</label><br>
            <input type="number" name="sayi1" required><br>
            <label>2. Sayı:</label><br>
            <input type="number" name="sayi2" required><br><br>

            <input type="submit" value="Topla">
        </form>
        <?php
        // Formun gönderilip gönderilmemiğini kontrol edelim
        if (isset($_POST['sayi1']) && isset($_POST['sayi2'])) {
            // Formdan gelen verileri değişkenlere atayalım
            $s1 = $_POST['sayi1'];
            $s2 = $_POST['sayi2'];

            // Aritmetik toplama işlemi
            $toplam = $s1 + $s2;

            // Sonucu HTML içine birleştirerek yazdıralım
            echo "<h3>İşlem Sonucu</h3>";
            echo "<p>$s1 + $s2 = <b>$toplam</b></p>";
        }
    ?>

</body>
</html>
```

GET

```
<!DOCTYPE html>
<html>
    <head><title>Formlar ve GET Metodu</title></head>
    <body>
        <form method="GET" action="formlar_baslangic_GET.php">
            <label>1. Sayı:</label><br>
            <input type="number" name="sayi1" required><br>
            <label>2. Sayı:</label><br>
            <input type="number" name="sayi2" required><br><br>

            <input type="submit" value="Topla">
        </form>
        <?php
        // Formun gönderilip gönderilmemiğini kontrol edelim
        if (isset($_GET['sayi1']) && isset($_GET['sayi2'])) {
            // Formdan gelen verileri değişkenlere atayalım
            $s1 = $_GET['sayi1'];
            $s2 = $_GET['sayi2'];

            // Aritmetik toplama işlemi
            $toplam = $s1 + $s2;

            // Sonucu HTML içine birleştirerek yazdıralım
            echo "<h3>İşlem Sonucu</h3>";
            echo "<p>$s1 + $s2 = <b>$toplam</b></p>";
        }
    ?>

</body>
</html>
```