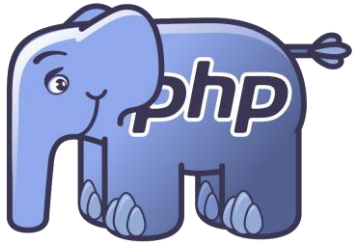
HTML / CSS – DUYARLI(RESPONSİVE) TASARIM

- 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 `<br>` *veya* `<br/>` 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.*
    - `<a href=”images.html” > FOTOĞRAFLAR </a>` *yazıya tıkayınca* ***images.html*** *sayfası açılacak:*
    - `<a href=” https://www.google.com” > Google </a>` *yazıya tıkayınca ilgili web adresi açılacak.*
- **Resimler:** `<img src=“images/basketball.jpeg” alt=“basketbol”>`

HTML Etiketleri

LİSTELER:

- **Sırasız Listeler**: Sırasız bir liste maddelerden oluşur. `<ul> <li>…</li><li>…</li></ul>`
- **Sıralı Listeler:** Maddeler rakamlar, roma rakamları veya harfler ile listelenir. `<ol> <li>…</li><li>…</li></ol>`

```
<h3> Alışveriş Listesi: </h3>

<ul>
    <li>Kahve</li>
    <li>Süt</li>
    <li>Çikolata</li>
</ul>
```

Alışveriş Listesi:

- Kahve
- Süt
- Çikolata

```
<h3> Alışveriş Listesi: </h3>
 <ol>
      <li>Kahve</li>
      <li>Süt</li>
      <li>Çikolata</li>
 </ol>
```

Alışveriş Listesi:

1. Kahve
2. Süt
3. Çikolata

```
<ol type="A">
     <li>Sucuk</li>
     <li>Menemen</li>
     <li>Bal</li>
 </ol>
```

Alışveriş Listesi:

A. Sucuk
B. Menemen
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

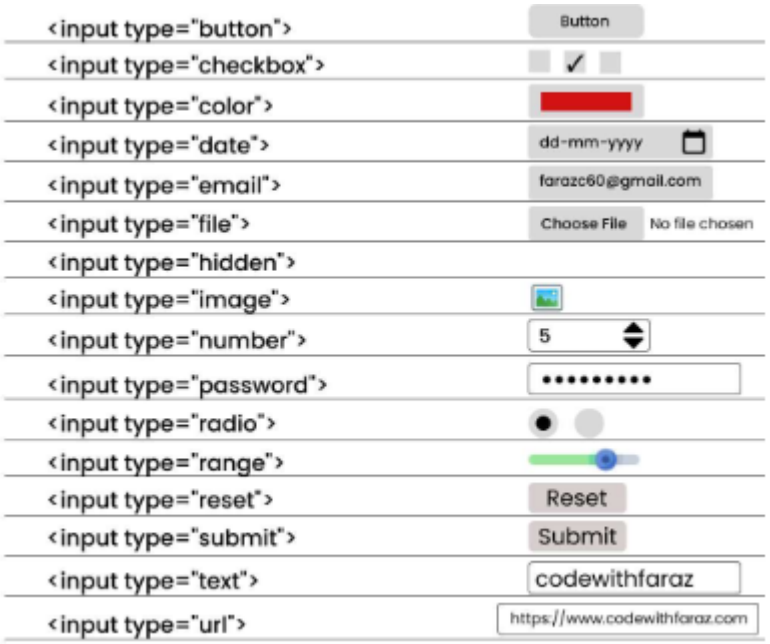
HTML Etiketleri

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 içerisine varsayılan olarak yazı yazmamızı sağlar*

Bazı input ve textarea parametreleri:

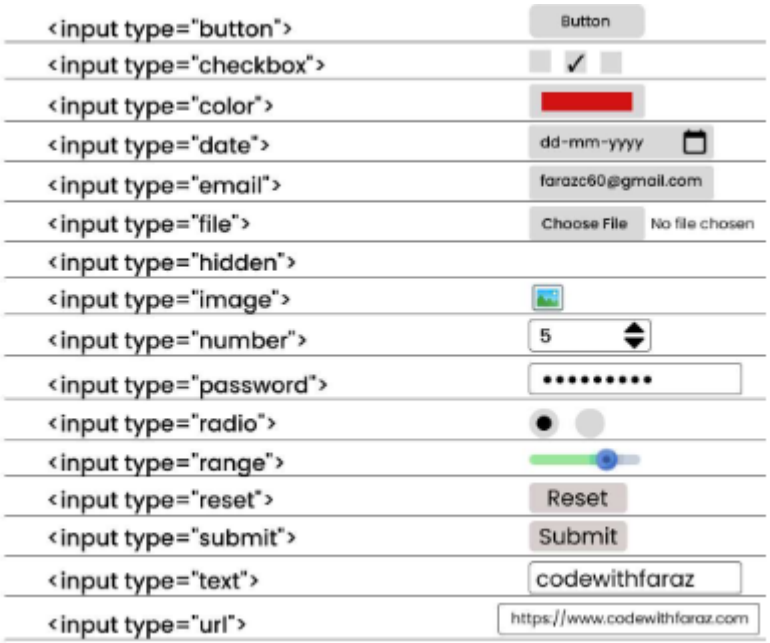
- ***autofocus:*** *İlgili metin alanını seçili hale getirir ve böylece görsel olarak öne çıkmasını sağlar.*
- ***disabled:*** *Veri girişini pasif hale getirir.*
- ***readonly:*** *Sadece okunabilir metin gözükür. Bu alanı fare ile seçemeyiz.*
- ***required:*** *Doldurulması zorunlu alanlarda bu parametreyi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



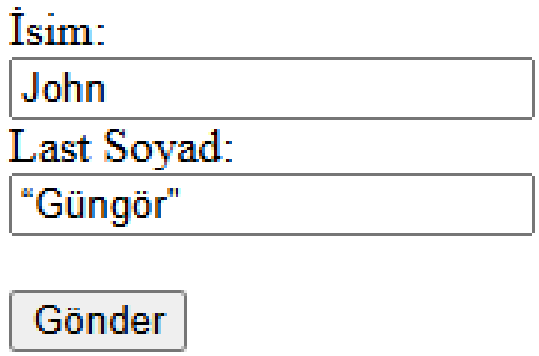
HTML Etiketleri

input type parametresine ait diğer özellikler:

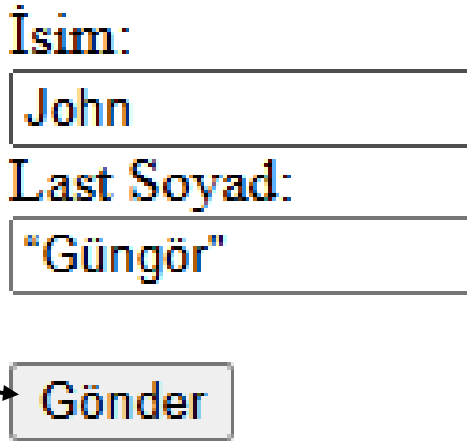
- ***password:*** *Şifreli alanlarda bu özelliği kullanırız. Karakterler yıldız şeklinde gözükür.*
- ***reset:*** *Formdaki girilmiş tüm verileri temizler.*
- ***file:*** *Dosya yükleme butonudur. Bu butona tıklayınca bir pencere açılı yüklenecek dosya buradan*
- *seçilecektir.*
- ***hidden:*** *Gizlemek istediğimiz kısımlar için bu özelliği kullanırız.*
- ***image:*** *Gönder butonu yerine resim koymak için bu özelliği kullanırız.*
- ***url:*** *Web adresi girişi için kullanılırız.*
- ***tel:*** *Telefon numarası girişi için kullanılırız.*
- ***number:*** *Sayı seçimleri için kullanırız. Aşağı yukarı oklar gözükür.*
- ***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 özelliği kullanırız.*



HTML Etiketleri




```
<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>
```



action: *Formun gönderileceği adresin yazıldığı parametredir. Eğer yukarıdaki kutuya bir şeyler yazıp gönder butonuna tıklarsanız, girdiğiniz bilgileri "kullanici.php" adlı veri dosyasına göndermiş olursunuz.*
method: *Formun gönderilme yöntemidir.* ***“GET”*** *metodu kullanıldığında istekler adres satırında görüntülenir. Bir de* ***“POST”*** *metodu vardır. Şimdilik bunlara değinmeyeceğiz.*
target="_blank": *Form yeni bir pencerede açılır.*
value: *Butonun üstünde ne yazacağını belirler.*

HTML Etiketleri

- **VİDEO 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>
```

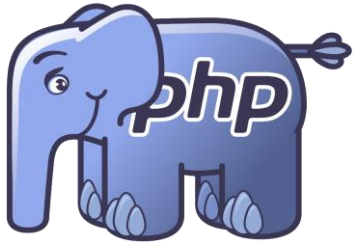
PHP

~~*Personal Home Page*~~
Hypertext Preprocessor

MBP 192 Internet Programlama

Bahar 25-26

İEU Bilgisayar Prog.



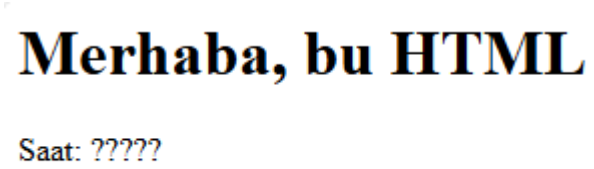
PHP Nedir?

- 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?

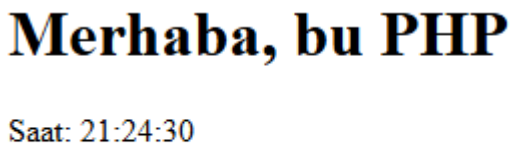
- index.html

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



- index.php

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



Kod: h2-test_page >> test.html, test.php

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 ..... ?>*** arasına yazılır

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

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

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

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

- **Kural 4:** PHP dosyaları ****.php*** uzantılı olmalı
  - `index.php` ✅
  - `index.html` ❌ (PHP çalışmaz)

- **Kural 5:** Kod yazımında yorum satırları kullanılır.

  Okunabilir kod!

```php
<?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

- “Merhab Dünya!”
- 5 + 3 toplamını
- H1 başlığı olarak “`HTML de yazdırabilir`”

yazdıralım…

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

Kod: h2-start >> start.php

HTML il PHP içi içe çalışması

HTML İçine 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
           arasına eklendi "; ?>
        Excepteur sint...
    </p>
</body>
</html>
```

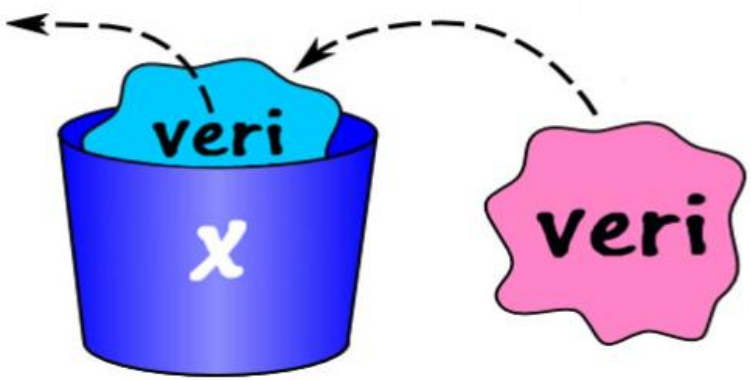
PHP İçinden HTML Çıktısı Verme - “ ” dikkat

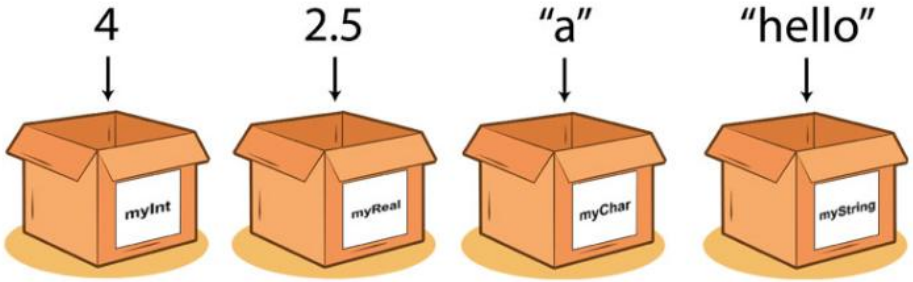
```
<!DOCTYPE html>
<html>
<head>
    <title>PHP</title>
</head>
<body>
    <?php
    echo "<h1>Merhaba MBP 192 </h1>";
    echo "<p> PHP Dersi </p>";
    echo "<p> Lorem ipsum...
        Bu kod HTML etiketi arasına
eklendi
        Excepteur sint... </p>";
    ?>
</body>
</html>
```

Kod: h2-start >> 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 çağırmaktır.
  - **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 anlaşılır.





Değişkenler ve Veri Tipleri

Değişken Tanımlama Kuralları

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

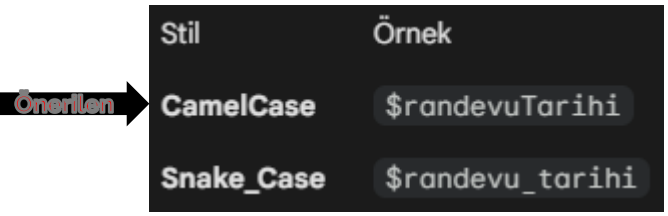
Değişken Adı	Durum	Neden?
`$randevuID`	**Uygun**	Harf ile başlar, standart kurallara uyar.
`$_musteri`	**Uygun**	Alt çizgi ile başlayabilir.
`$fiyat_2026`	**Uygun**	İçinde sayı barındırabilir.
`$2randevu`	**Uygun Değil**	Sayı ile başlayamaz.
`$toplam-tutar`	**Uygun Değil**	Tire (-) karakteri içeremez (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: h2-start >> değişkenler.php

Değişken Veri Tipleri ve Tanımlama

Değişken Yazım Standartları (CamelCase vs Snake_Case)



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.

- Kod: h2-start >> veri_tipleri.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 $toplam-tutar = 500; // PHP burada '$toplam' 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) değişirse kodunuz bozulabilir. // Önerilen: $ucret = 100;`

- Kod: h2-start >> 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 sınıftan bir **örnek** (**instance**) üretilmelidir.

```php
<?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 Tasarımı");

// 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 bölümlerinden erişilebilir olduğunu belirleyen kurallar b.ütünüdür.
- Değişkenlerin nerede tanımlandığı, onların ömrünü ve görünürlüğünü 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 `global` 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 çağrıldığında kaldığı yerden devam etmesini istiyorsanız static anahtar kelimesini kullanırsınız.

- Kod: h2-start >> degisken_kapsam.php

Örnek

Adınızı, soyadınızı, yaşınızı, 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
<?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: h2-start >> operatorler.php

Aritmetik Operatörler

Sayısal değerlerle standart matematiksel işlemler yapmak için kullanılırlar.

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

- Kod: h2-start >> operatorler.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
		?>

-=	Çıkartarak Atama	<?php
		$sayi=5;
		echo $sayi-=10;
		// İşlem Sonucu -5
		?>

*=	Çarparak Atama	<?php
		$sayi=5;
		echo $sayi*=10;
		// İşlem Sonucu 50
		?>

/=	Bölerek 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: h2-start >> operatorler.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
==	Eşittir	$a == $b
===	Denktir	$a === $b
!=	Eşit değildir	$a != $b
!==	Denk değildir	$a !== $b
<>	Eşit değildir	$a <> $b
<	Küçüktür	$a < $b
>	Büyüktür	$a > $b
<=	Küçük ve eşittir	$a <= $b
=>	Büyük ve eşittir	$a => $b

- Kod: h2-start >> operatorler.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: h2-start >> operatorler.php

String (Metin) Operatörleri

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

`.` : Birleştirme operatörü.

PHP

```php
$ad = "Emrah";
$soyad = "Yüksel";
echo $ad . " " . $soyad; // Çıktı: Emrah Yüksel
```

- Kod: h2-start >> 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.

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

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

- Kod: h2-start >> operatorler.php

Artırma ve Azaltma Operatörleri

```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
`$y = ++$x;`	1. `$x` artar	`$x` ve `$y` aynı olur.
	2. Yeni `$x`, `$y` 'ye atanır.	
`$y = $x++;`	1. Mevcut `$x`, `$y` 'ye atanır	`$y` eski değerde kalır, `$x` artar.
	2. `$x` artar.	

Hata Operatörü

- Mecbur dolmadıkça KULLANMAYIN!!!
- **Try-Catch blokları KULLANIN**
- @ işareti ile temsil edilir.
- Önüne geldiği ifadenin oluşturabileceği hata mesajlarını (warning, notice vb.) bastırmak ve kullanıcıya gösterilmesini engellemektir.

- **RİSKLER**
  - **Hata Ayıklamayı Zorlaştırır:** Gerçekten kritik bir hata oluştuğunda mesaj gizlendiği için sorunun kaynağını bulmak güçleşebilir.
  - **Performans:** @ operatörü, PHP'nin hata raporlama seviyesini geçici olarak değiştirdiği için yoğun kullanımda performansı olumsuz etkileyebilir.
  - **Kritik Hatalar:** Bu operatör, programın çalışmasını tamamen durduran "Fatal Error" (Ölümcül Hata) türündeki hataları engelleyemez.

```php
$dosya = @file('mevcut_olmayan_dosya.txt');
// Dosya yoksa bile ekrana hata mesajı basılmaz.
```

- Kod: h2-start >> operatorler.php

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 yazılarak 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.

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

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

$metinSayi = "42isik";
$sayi = (int)$metinSayi;
// Sonuç: 42 (Metnin başındaki sayıyı alır)
$metinSayi2 = "isik42";
$sayi2 = (int)$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 Sayi: " . $metinSayi . " → Sayı: " . $sayi; // Metin Sayi: 42isik → Sayı: 42
echo "<br>";
echo "Metin Sayi 2: " . $metinSayi2 . " → Sayı 2: " . $sayi2; // Metin Sayi 2: isik42 → Sayı 2: 0
echo "<br>";
```


- Kod: h2-start >> tur_donusumu.php

Manuel Tür Dönüşümü (Type Casting)

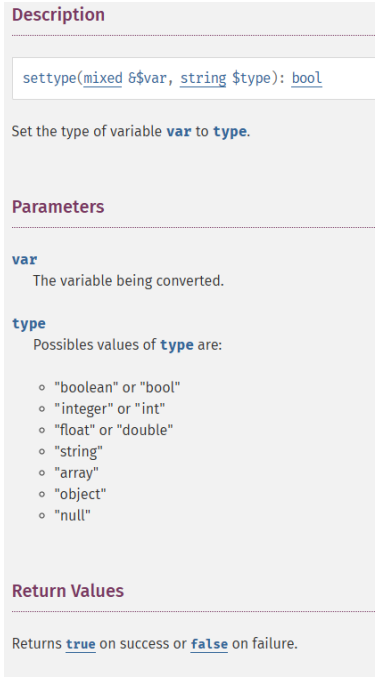
Dönüşüm Türü	Kullanım	Açıklama
(int), (integer)	`(int)$degisken`	Tam sayıya dönüştürür.
(bool), (boolean)	`(bool)$degisken`	Mantıksal değere (true/false) dönüştürür.
(float), (double)	`(float)$degisken`	Ondalıklı sayıya dönüştürür.
(string)	`(string)$degisken`	Metne (karakter dizisine) dönüştürür.
(array)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

- Kod: h2-start >> tur_donusumu.php

Özel Dönüşüm Fonksiyonları

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

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



Description

```
settype(mixed &$var, string $type): bool
```

Set the type of variable **var** to **type**.

Parameters

var
The variable being converted.

type
Possibles values of **type** are:

- "boolean" or "bool"
- "integer" or "int"
- "float" or "double"
- "string"
- "array"
- "object"
- "null"

Return Values

Returns **true** on success or **false** on failure.

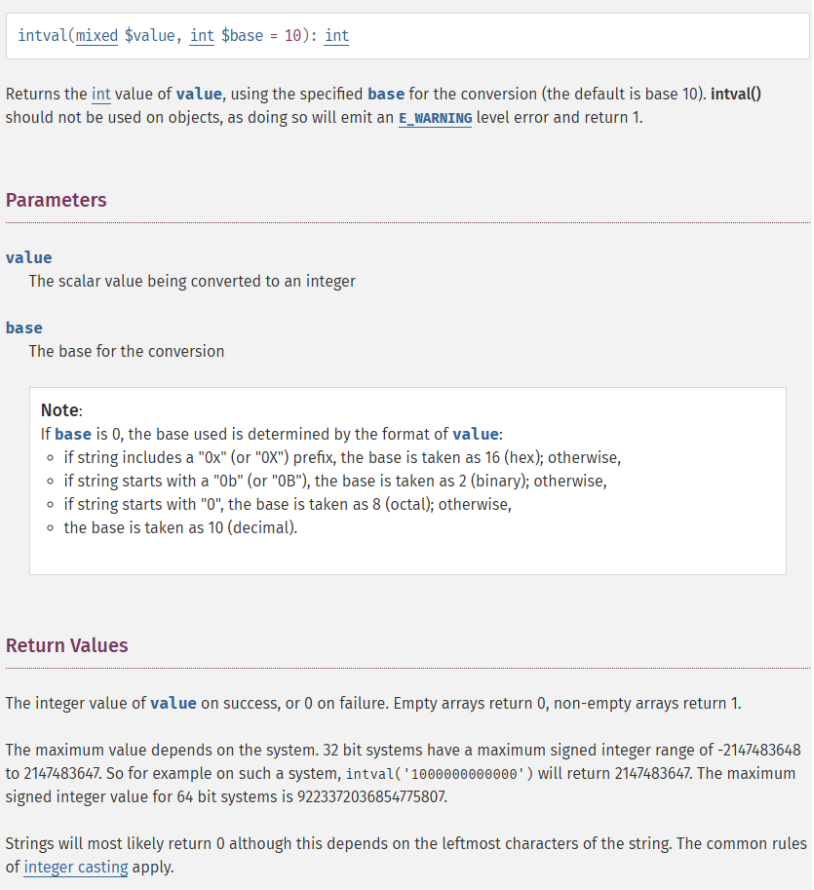
https://www.php.net/manual/en/function.settype.php

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.

h2-start >> tur_donusumu.php

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('1000000000000')` 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.

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

h2-start>>ozel_tur_donusum_fonk.php

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 işareti.

Link ve Kaynaklar

- https://htmlcheatsheet.com/
- Git eğitimi: https://www.youtube.com/watch?v=rWG70T7fePg
- CSS
  - https://css-tricks.com/
  - https://flexboxfroggy.com/#tr
- https://codepen.io/