



## Özet

- **PHP Diline Giriş**
- PHP web sayfaları geliştirmek için kullanılan sunucu taraflı dillerden bir tanesidir.
- **PHP DESTEKLİ WEB SUNUCUSU KURULUMU**
- PHP dilini çalıştırmak için hem web sunucusu hem de PHP yorumlayıcısının bu sunucu üzerinde kurulması gerekir.
- XAMPP paketi hem sunucuyu hem de PHP yorumlayıcısını birlikte kurmanızı sağlar. Ücretsiz paketi Apache Friends web sitesinden (<https://www.apachefriends.org/tr/index.html>) indirip kurabilirsiniz.
- Kurulum sonrasında herhangi bir metin editörü ile yazacağınız PHP uzantılı dosyaları sunucudaki ana web dizinine (örneğin c:\xampp\htdocs\ kaydedin. Talıştırmak için tarayıcınızda <http://localhost/dosyaAdi.php> şeklinde yazıp Enter tuşuna basınız.
- PHP'ye özgü ayarlamaları yapmak için C:\xampp\php klasöründeki php.ini dosyasını açıp düzenleyebilirsiniz. Değişikliklerin geçerli olması için sunucunun (örneğin Apache) durdurulup tekrar başlatılması gerekir.
- **PHP DİLİNİN YAPISI**
- PHP komutları HTML sayfalarında <?php ile ?> arasında yazılır. Bu PHP kod bloğuna istediğiniz kadar PHP komutu yazabilirsiniz.
- Açıklama satırı eklemek için satır başına // sembollerini kolayabilir ya da /\* ile başlayıp açıklama sonuna \*/ koyabilirsiniz.
- **Değişkenler:** PHP'de değişkenler tanımlanmadan ilk değer atanarak kullanılabilir. Atanan değerın türüne göre değişkenin türü belirlenir. Farklı türde bir değer atanırsa değişken kabul eder ve o türe dönüşür.
- Değişken isimleri \$ sembolü ile başlar, sonrasında A-Z, a-z ve ya rakamlar yer alabilir. Bunlara ek olarak sadece alt tire karakteri kullanılabilir. Rakamlar en başta yer alamaz.
- Değişkenlerin tanımlandıkları yere ve/veya static kelimesine göre yerel, genel, statik ve parametre kapsamında olabilirler.
- Define kelimesi ile değeri değiştirilemeyen sabitler tanımlanabilir.
- Matematik işlemleri için +, -, \*, / ve % sembolleri kullanılırken, hızlı atama için +=, -=, \*=, /= ve %= ile . = operatörleri kullanılır.
- **Diziler:** Birden çok değeri saklayabilen değişkenlerdir. Elemanları farklı türde olabilir. Elemanlara ulaşmak için kullanılan indis değeri sıfırdan başlar. Eğer istenirse indis yerine her elemana bir de anahtar eklenebilir.
- Dizileri sıralama veya ters çevirmek için sort, rsort ve array\_reverse fonksiyonları kullanılabilir. Diziye sonradan yeni eleman eklemek veya elemanını çıkarmak mümkündür.
- **KONTROL YAPILARI**
- Programın akışını belirli koşullara göre değiştirmek ya da belirli komutların tekrar çalışmasını sağlamak için kontrol yapıları olarak tanımlanan if, switch, for, foreach, while ve do-while komutları kullanılır.
- **If komutu:** Bir koşula göre bazı komutların çalışmasını veya çalışmamasını sağlayan yapıdır. Tek başına, if-else ya da if-elseif şeklinde 3 farklı şekilde kullanılabilir.
- Koşulların doğru ya da yanlış değerini vermesi için karşılaştırma (>, <, == vb. ) ve mantık operatörleri (&&, ||) kullanılır.



## Özet (devamı)

- **Switch Komutu:** Bir değişkenin alabileceği değerlere göre programın akışını değiştirir. Aralık kontrolü yapamaz ancak açık değer kontrolü yapılabilir.
- **For Döngüsü:** Tekrar edilmesi istenen komutların tekrar sayısını, bir sayaç değişkeninin ilk ve son değerine göre ayarlandığı döngüdür. Dizilerin elemanlarına erişim için sıklıkla kullanılır. Sayaç istenilen bir değere başlatılabilir ve üst değer ulaşması için birer birer ya da farklı büyüklükte artırılabilir.
- **Foreach Döngüsü:** Bir dizinin tüm elemanlarına erişim sağlayana kadar komutların tekrar edilmesini sağlar. Eleman sayısı belirsiz dizilerde daha çok tercih edilir. Eğer dizi elemanları anahtar da içeriyorsa hem anahtara hem de değere erişim mümkündür.
- **While Döngüsü:** Tekrar edilecek komutların bir koşulun sonucuna göre tekrar edilmesini sağlar. Koşul döngü başında yer aldığı için koşul yanlış ise komutlar hiç çalıştırılmadan döngü terk edilebilir. Eğer sonuç sürekli doğru sonucunu üretirse, döngü sonsuz döngüye girebilir.
- **Do-While Döngüsü:** Tekrar durumunun koşula bağlı olduğu başka bir döngü yapısıdır. Ancak koşul, döngü komutlarının sonunda yer alır. Bu nedenler komutlar koşuldan bağımsız olarak en az bir kez çalıştırılır, sonraki tekrarlar ise koşula bağlıdır. Koşul sürekli doğru olursa yine sonsuz döngü riski vardır.

**DEĞERLENDİRME SORULARI**

- PHP'nin çalışması için aşağıdaki ikililerden hangisi gereklidir?
  - Tarayıcı ve Web Sunucusu
  - Web Sunucu ve Veri Tabanı
  - Veri Tabanı ve PHP Yorumlayıcısı
  - PHP yorumlayıcısı ve Web Sunucusu
  - Web Sunucusu ve Metin Editörü
- PHP komutlarının sonunda hangi karakter yer almalıdır?
  - .
  - ;
  - !
  - >
  - |
- Aşağıdakilerden hangisi geçersiz değişken adıdır?
  - \$2\_sayfaNo
  - \$sayfaNo2
  - \$sayfa\_2No
  - \$sayfaNoiki
  - \$SAYFA2No
- $\$a = 3 * 2 + 1$ ; işleminin sonucunda \$a değişkeninin değeri kaçtır?
  - 61
  - 2221
  - 4
  - 9
  - 7
- Diziler hakkında belirtilen ifadelerden hangisi yanlıştır?
  - Elemanları silinebilir
  - Yeni eleman eklenebilir
  - Tüm elemanlar aynı türde olmak zorundadır
  - Tüm anahtarlar farklı olmak zorundadır
  - Her değerın bir anahtarı vardır
- Aşağıdaki for döngülerinin hangisinde tekrar sayısı 10'dur?
  - for(\$a=2; \$a<=12; \$a++)
  - for(\$a=2; \$a<12; \$a++)
  - for(\$a=2; \$a<12; \$a+=2)
  - for(\$a=2; \$a>12; \$a++)
  - for(\$a=2; \$a==12; \$a++)

- $\$a=45$ ; if(\$a === "45") ifadesindeki koşulun sonucu nedir?
  - 45
  - "45"
  - "1"
  - Doğru
  - Yanlış
- Aşağıdaki komutların hangisinde sonsuz döngü riski vardır?
  - switch
  - if
  - while
  - if-elseif
  - Array\_reverse()
- Aşağıdaki değişken kapsamlarından hangisinde, fonksiyonda tanımlanmış değişken fonksiyon tekrar çağırıldığında önceki değerini korur?
  - Yerel
  - Genel
  - Global
  - Statik
  - Parametre
- Switch komutundaki default kelimesinin amacı nedir?
  - Önceki case'ler eşleşmez ise çalışacak komutları devreye sokmak
  - Çalışan en son case'e programın akışını geri göndermek
  - Kontrol edilen değişkenin değerini varsayılan değer ile değiştirmek
  - Kontrol edilen değişkenin değerini 1 artırmak
  - Açık değer yerine değer aralığı ile eşleşme kontrolü yapmak

**Cevap Anahtarı**

1.d, 2.b, 3.a, 4.e, 5.c., 6.b, 7.e, 8.c, 9.d, 10.a