## Soru1--

## GET ve POST

Hem **GET** hem de **POST** bir dizi oluşturur **(örn. Dizi (key => value, key2 => value2, key3 => value3, ...).** Bu dizi, anahtarların form kontrollerinin isimleri olduğu ve değerlerin kullanıcıdan gelen giriş verileri olduğu anahtar / değer çiftlerini içerir.

Hem **GET** hem de **POST**, $ \_GET ve $ \_POST olarak değerlendirilir. Bunlar süper küreseldir, yani kapsamı ne olursa olsun her zaman erişilebilir olurlar ve onlara özel bir şey yapmadan herhangi bir işlev, sınıf veya dosyadan erişebilirsiniz.

$\_GET, geçerli komut dosyasına **URL** parametreleriyle iletilen bir değişken dizisidir.

$\_POST, geçerli komut dosyasına **HTTP POST** yöntemiyle iletilen bir değişken dizisidir.

## GET Yöntemi

**GET** yönteminde, veriler genellikle & (ve) işareti ile ayrılmış ad ve değer çiftleri dizeleri olan **URL** parametreleri olarak gönderilir . Genel olarak, **GET** verileri olan bir **URL** şöyle görünecektir:

http://www.example.com/action.php?**name**=css&***ders***=*22*

**URL**'deki kalın kısımlar **GET** parametreleridir ve italik kısımlar bu parametrelerin değeridir. Birden fazla parametre=value, ampersand (&) ile birleştirerek **URL**'ye gömülebilir . Webmaster sadece **GET** yöntemiyle basit metin verileri gönderebilir.

### GET Yöntemi Kullanmanın Avantaj ve Dezavantajları

* **GET** yöntemi, tarayıcınızın Konum kutusunda, sunucu günlüklerinde görünen uzun bir dize üretir.
* **GET** ayrıca gönderilecek bilgi miktarını da sınırlar. Sınırlama yaklaşık 2000 karakterdir. Ancak, değişkenler URL’de görüntülendiğinden, sayfayı favorilere eklemek mümkündür. Bu bazı durumlarda yararlı olabilir.
* Sunucuya gönderilecek parola veya diğer hassas bilgileriniz varsa asla **GET** yöntemini kullanmayın. Ancak bu veriler **MD5** şifreleme ile güvenli bir şekilde de gönderilebilir.
* **GET**, resimler veya cümle dosyaları(word,exe,pdf) gibi ikili verileri sunucuya göndermek için kullanılamaz.
* **GET** yöntemiyle gönderilen verilere **QUERY\_STRING** ortam değişkeni kullanılarak erişilebilir.
* PHP, **GET** yöntemini kullanarak gönderilen tüm bilgilere erişmek için $ \_GET ilişkilendirici dizi sağlar.

PHP, **URL** aracılığıyla gönderilen ya da bir HTML formu kullanılarak gönderilen tüm bilgilere erişmek için süper küresel $\_GET değişken sağlar.

## POST Yöntemi

**POST** yönteminde veriler, işleme komut dosyasıyla ayrı bir iletişimde, sunucuya bir paket olarak gönderilir. **POST** yöntemiyle gönderilen veriler **URL**'de görünmez.

### POST Metodu Kullanmanın Avantajları ve Dezavantajları

* **POST** yönteminin gönderilecek veri boyutunda herhangi bir kısıtlama yoktur.
* **POST** yöntemi, ASCII'nin, yanı sıra ikili veri(bir dosyayı karşıya yüklemek) göndermek için kullanılabilir.
* **GET**'den daha güvenlidir, çünkü kullanıcı tarafından girilen bilgiler asla **URL** sorgu dizesinde veya sunucu günlüklerinde görünmez.
* **POST** yöntemiyle gönderilen veriler **HTTP** üstbilgisinden geçer, bu nedenle güvenlik **HTTP** protokolüne bağlıdır. Güvenli **HTTP** kullanarak bilgilerinizin güvenli olduğundan emin olabilirsiniz.
* PHP, **POST** yöntemini kullanarak gönderilen tüm bilgilere erişmek için $ \_POST ilişkilendirici dizi sağlar.

PHP, $\_POST yöntemi ile gönderilen ya da bir HTML formu kullanılarak gönderilen tüm bilgilere erişmek için başka bir süper küresel değişken sağlar.

Soru2--

#### **Apache Kurulumu**

1. "webserver" ve "www" adlı iki dizin oluşturulur. Sunucu dosyaları "webserver" , site dosyaları ise "www" altında saklanır. Dizinleri, işletim sisteminin çökmesi, diskin formatlanması gibi sorunlardan kurtulmak için başka bir sürücüde oluşturunuz. Örnek D:/ olabilir.
2. Apache'nin 32 bit sürümü [https://www.apachelounge.com/d](https://www.apachelounge.com/download/VC11/) adresinden indirilir. Bu yazıda Apache 2.4.29 Win32 VC11 (httpd-2.4.29-win32-VC11.
3. İndirilen sıkıştırılmış dosya (.zip) içindeki "Apache24" klasörü "D:/webserver" dizinine kopyalanır.
4. "D:/webserver/Apache24/conf" içindeki "httpd.conf" dosyasını "Not Defteri" veya "Notepad++" ile açın. Aşağıdaki satırları belirtilen şekilde düzenleyin.  
     
   **ServerRoot "c:/Apache24"** sunucu ana dizininin belirtildiği bu satırı   
   **ServerRoot "d:/webserver/Apache24"** olarak değiştirin.   
     
   **#ServerName www.example.com:80** sunucu adının belirtildiği bu satırı  
   **ServerName localhost** olarak değiştirin.  
     
   **DocumentRoot "c:/Apache24/htdocs"** site dosyalarının bulunacağı dizinin belirtildiği bu satırı  
   **DocumentRoot "d:/www"** olarak değiştirin.  
     
   **<Directory "c:/Apache24/htdocs">** satırını  
   **<Directory "d:/www">** şeklinde değiştirerek site dosyalarımızın bulunacağı "d:/www" dizinine özel yapılandırmayı belirtin.   
     
   **ScriptAlias /cgi-bin/ "c:/Apache24/cgi-bin/"** satırını   
   **ScriptAlias /cgi-bin/ "d:/webserver/Apache24/cgi-** şeklinde değiştirerek sunucu cgi dosyalarının bulunduğu dizini belirtin.  
     
   **<Directory "c:/Apache24/cgi-bin">** satırını  
   **<Directory "d:/webserver/Apache24/cgi-** şeklinde değiştirerek sunucu cgi dosyalarının bulunduğu dizine özel yapılandırmayı belirtin.

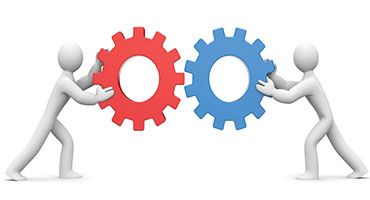
Soru3--

#### **PHP Kurulumu**



1. İndirilen Apache VC sürümüyle uyumlu 32 bit PHP dosyalarını [https://windows.php.net/downlo](https://windows.php.net/download/) adresinden indirin. Apache ve PHP'nin VC sürümlerinin aynı olması gerekir. Apache VC11 ile uyumlu, php-5.6.34-Win32-VC11-x86 (php-5.6.34-Win32-VC11-x86.zip) sürümünü indirin.
2. "php-5.6.34-Win32-VC11-x86.zip
3. "D:/webserver/php" dizininde PHP yapılandırma ayarlarının bulunduğu "php.ini-development" dosyasını bulup adını "php.ini" olarak değiştirin.
4. "php.ini" dosyasını bir metin düzenleyici ile açıp aşağıdaki satırları düzenleyin:  
     
   **extension\_dir = "ext"** yerine  
   **extension\_dir = "d:/webserver/php/ext"** yazın.  
     
   Aşağıdaki satırların başında bulunan noktalı virgülleri (;) kaldırın. MySQL veritabanı bağlantısı pasiflikten çıkıp aktif hale gelir.   
     
   **extension=php\_mysql.dll**  
   **extension=php\_mysqli.dll**  
     
   **[Date]** 'i bulup Türkiye saati için:   
   **date.timezone = "Etc/GMT-3"** ifadesini ekleyin. Eğer date.timezone ifadesi mevcut ve başında (;) varsa kaldırın.

#### **Apache ve PHP'nin Bağlanması**

1. Apache ile Php entegrasyonu için "D:/webserver/Apache24/conf" dizinindeki "httpd.conf" dosyasını yeniden bir metin düzenleyiciyle açın.  
     
   **LoadModule** ile başlayan satırların en altına aşağıdaki satırı ekleyin.   
   **LoadModule php5\_module d:/webserver/php/php5apache2\_** Bu, Apache ile PHP iletişimini sağlar.  
     
   **DirectoryIndex index.html** satırında sitelerin anasayfa tespitinde varsayılan index dosyaları belirtilir, bu satıra index.php ifadesi eklenip   
   **DirectoryIndex index.html index.php** haline getirilir. Bu, sitedeki .php uzantılı index dosyalarını da anasayfa olarak tanıtmış oluyor.  
     
   **<IfModule mime\_module>** grubu içerisine   
   **AddHandler application/x-httpd-php .php** satırını ekleyerek php uzantılı dosyaları Apache'ye tanıtın.  
     
   Apache'nin, php.ini dosyasındaki ayarlara ulaşabilmesi için dosyanın bulunduğu dizini "httpd.conf" dosyasının en altına şu şekilde ekleyin.   
   **PHPIniDir "d:/webserver/php"**

#### **Apache'nin Windows Servisi Olarak Kurulumu**

1. Nihayet "D:/webserver/Apache24/bin" dizinindeki "httpd.exe"yi çalıştırın. Apache, her seferinde bu şekilde çalıştırılmak yerine Windows Servisi olarak kurabilir. Bunun için Komut İstemi(cmd.exe)'ni yönetici olarak çalıştırın. Burada;
2. **cd /** komutuyla C:/ ana dizinine dönün ve sonrasında **D:** ifadesini yazarak "d" diskine geçiş yapın.
3. **cd D:/webserver/Apache24/bin** komutunu yazın ve "httpd.exe" dosyasının bulunduğu dizine ulaşın.
4. Ekrana dizin satırı geldikten sonra **httpd.exe -k install -n "Apache HTTP Server"** komutunu yazın ve Apache'yi Windows Servisi olarak kurun. Kurulumu kontrol etmek için başlat menüsünde "Hizmetler"i açın. Gelen hizmetler listesinde "Apache2.4" satırına sağ tıklıyın ve "Başlat"ı seçerek hizmeti başlatın. Eğer hatasız çalışır ve "Başlangıç Türü" de "Otomatik" ise Windows Servisi hazır demektir. Artık bilgisayar yeniden başladığında da "Apache" otomatik olarak çalışmaya başlayacaktır.

#### **Apache ve PHP Kurulumunun Test Edilmesi**

1. Kurulumunuzun çalıştığını kontrol etmek için "D:/www" dizini içerisinde bir metin düzenleyiciyle "test.php" adında yeni dosya oluşturun, ve dosya içerisine aşağıdaki PHP kodlarını yazın.   
   **<?php echo phpinfo(); ?>**
2. Bir tarayıcı açıp adres çubuğuna "localhost/test.php" yazıp "enter" tuşuna basın. Sayfa sorunsuz açılıyor ve PHP'nin özelliklerini gösteriyorsa kurulum başarıyla tamamlanmış demektir.
3. Sitelerinizi "D:/www" dizininde oluşturabilirsiniz. Mevcut sitelerinizi görmek için tarayıcınızın adres çubuğuna "localhost" yazıp "enter" tuşuna basın, bu listeden sitelerinizi tarayıcı üzerinde test edebilirsiniz.

soru4--

--TEBRİKLER--SAYI 50

soru5--

<?php

$tarih = time();

echo(date("d/m/Y",$tarih)."\n");

?>