



Burdur
Mehmet Akif Ersoy
Üniversitesi

MAKÜ|GMYO
GÖLHİSAR MESLEK YÜKSEKOKULU

İnternet Programcılığı I

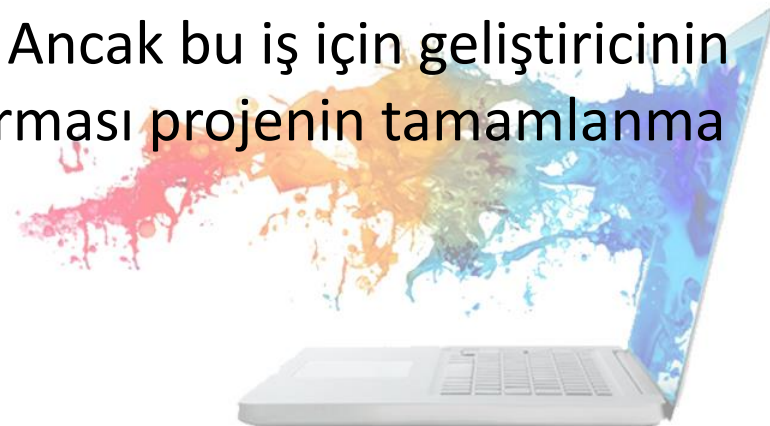
Öğr. Gör. Hüseyin Şengün
hsengun@mehmetakif.edu.tr

6. Hafta
Hazır Fonksiyonlar



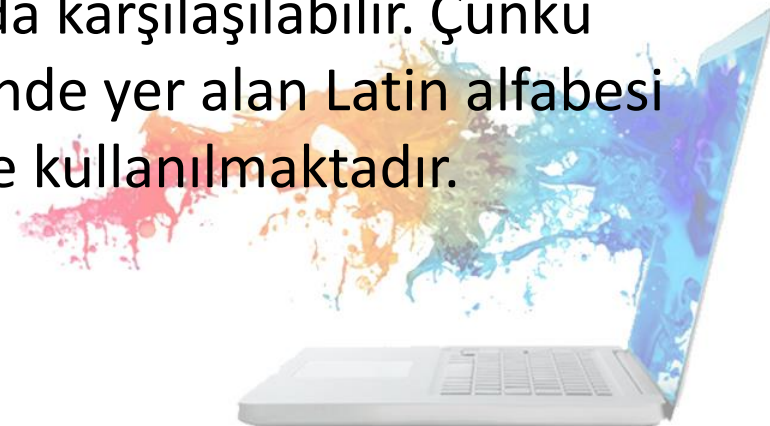
Fonksiyonlar

- Programcılar, PHP ile geliştirdikleri uygulamalarda genellikle sadece kendi hazırladıkları fonksiyonlardan ayrı olarak hazır fonksiyonları da kullanırlar. Bu fonksiyonlardan en yaygınları **veritabanı, metin, matematik ve dosya işlemlerini** yöneten fonksiyonlardır.
- Hazır fonksiyonlar ile metin değişkenleri kısaltılabilir, içinde geçen bir ifade tespit edilebilir veya bir ifade, başka bir değer ile değiştirilebilir. Bu tür metin işlemlerini hazır fonksiyonlar kullanmadan da yapmak mümkündür. Ancak bu iş için geliştiricinin ayıracağı süreyi başka iş kollarına aktarması projenin tamamlanma süresini azaltacaktır.



Fonksiyonlar

- Ayrıca hazır fonksiyonların yaygın olarak kullanıldığı bir diğer metin işlemleri ise şifreleme ve şifre çözme işlemleridir. Metin şifreleme işlemleri geri döndürülebilir/döndürülemez olarak iki farklı türe sahiptir. Bu türlerden geri döndürülebilir olanların şifreleme işlemine “**encode/encrypt**”; geri döndürme işlemine de “**decode/decrypt**” adı verilir.
- En yaygın kullanılan metin şifreleme işlemleri PHP dilindeki **base64_encode** ve **base64_decode** fonksiyonları ile yapılır. Bu fonksiyonlar e-posta iletimi sırasında da karşılaşılabılır. Çünkü base64 işlemleri sıklıkla metnin içerisinde yer alan Latin alfabesi harici karakterlerden kurtulmak için de kullanılmaktadır.



Fonksiyonlar

- Base64 ile şifrelenmiş bir metin yalnızca a-z, 0-9 ve diğer bazı karakterleri içerecektir. Bu durum geliştiricinin, ziyaretçiden aldığı metnin dili ne olursa olsun aynı işlemleri yapabilmesine olanak tanıyacaktır.
- Metin şifreleme işlemlerinde geri döndürülemez olan işlemler ise; “**hash**” olarak tanımlanır ve genelde verilerin doğrulanmasında kullanılır. Bu işlemlerin yaygın olanları, MD5 ve SHA fonksiyonlarıdır. Bu fonksiyonlara şifrelenmesi için verilen metin ne kadar uzun olursa olsun şifreleme sonrası her zaman sabit uzunlukta (sha için 64 karakter) ifade türetilir.



Fonksiyonlar

- **METİN İŞLEMLERİ**
- Çevrimiçi uygulama geliştirme işlemlerinde kullanıcılardan alınan verilerin miktarı normal uygulamalardan daha fazla olabilmektedir. Bu durumun en önemli kaynağı ise kullanıcıların sayısının fazlalığıdır. Kullanıcıların farklı sektörlerden olması ve girilebilecek verilerin standart bir biçime sahip olmamasından dolayı genellikle geliştirici veriyi işlemek zorunda kalmaktadır.
- Örneğin; e-posta adresi bilgisinin kullanıcıdan alan geliştirici; bu veri içerisinde “@” işaretini arayacak ve şekilsel doğruluğunu kontrol edecektir.
- PHP dilinde kullanılacak metin işleme fonksiyonlarının listesi <http://tr1.php.net/manual/tr/ref.strings.php> adresinden elde edilebilir.



Fonksiyonlar

- Yaygın olarak kullanılan PHP metin işleme fonksiyonları

Fonksiyon	Açıklama
<code>substr</code>	Bir metnin istenen kısmını kırparak verir ¹
¹ <code>mb_substr</code>	<i>"substr" fonksiyonunun UTF8 destekli sürümü</i>
<code>explode</code>	Verilen karaktere göre parçalayarak metni dizi haline dönüştürür
<code>str_replace</code>	Metin içindeki verilen karakteri bir başka karakterle değiştirir
<code>strpos</code>	İstenen karakterin bulunduğu sıra numarasını verir ²
² <code>mb_strpos</code>	<i>"strpos" fonksiyonunun UTF8 destekli sürümü</i>
<code>strlen</code>	Metnin kaç karakter olduğunu (uzunluğunu) verir ³
³ <code>mb_strlen</code>	<i>"strlen" fonksiyonunun UTF8 destekli sürümü</i>
<code>strrchr</code>	İstenen karakterin son eşleşmesini tespit eder ⁴
⁴ <code>mb_strrchr</code>	<i>"strrchr" fonksiyonunun UTF8 destekli sürümü</i>
<code>strrev</code>	Metni ters çevirir
<code>strtolower</code>	Metnin bütün karakterlerini küçültür
<code>strtoupper</code>	Metnin bütün karakterlerini büyültür



Fonksiyonlar

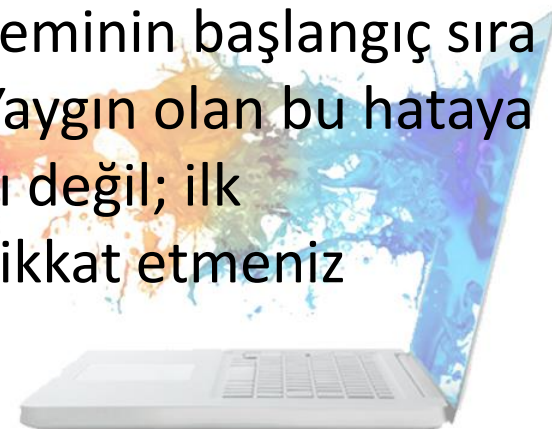
- PHP dilinde metin işlemleri yapmak için gerekli olan birçok fonksiyon verilmiş olmasına karşın; birçok fonksiyon Türkçe UTF8 desteğini bünyesinde barındırmamaktadır. Bu nedenle UTF8 destekli metin işlemleri fonksiyonları sonradan PHP diline dâhil edilmiştir.
- Bu fonksiyonların listesini <http://tr2.php.net/manual/tr/book.mbstring.php> adresinden elde edebilirsiniz.



Fonksiyonlar

- **KIRPMA İŞLEMLERİ**

- Çevrimiçi uygulamalar geliştirirken programcılar genellikle kullanıcılardan aldıkları veri üzerinde düzeltmeler yapmak durumunda kalır. Bu düzeltmelerin en sık kullanılanlarından birisi ise metnin belirli bir kısmının elde edilmesidir.
- Metin kırpma işlemleri kırılacak karakterin sıra numarası ve bu karakterden sonra kaç karakter alınacağını belirtmesi ile olmaktadır. Kırpma fonksiyonu iki sayı aldığından geliştiriciler genellikle öğrenme aşamasındayken kırpma işleminin başlangıç sıra numarası ile sonlanış sıra numarasını verirler. Yaygın olan bu hataya düşmemek için ikinci sayının sonlanış numarası değil; ilk karakterden sonraki ilerleme sayısı olduğuna dikkat etmeniz gerekmektedir.



Fonksiyonlar

- **KIRPMA İŞLEMLERİ**
- PHP dilinde metni kırpma için “**substr**” ve “**mb_substr**” adında iki fonksiyon bulunmaktadır. Bu fonksiyonlardan ilkinin kullanılması olursanız “ç,i” gibi Türkçe karakterlerin kırılması sorunlar yaşayabilirsiniz.
- Bu durumun nedeni UTF8 karakterlerinin bayt (byte) değerlerinin daha fazla olmasıdır. UTF8 olsun veya olmasın herhangi bir sorun yaşamamak için “**mb_substr**” fonksiyonunu kullanarak kırma işlemlerinizi gerçekleştirmeniz tavsiye edilir.



Fonksiyonlar

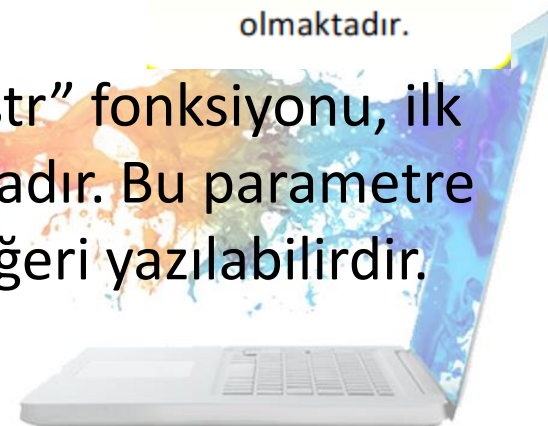
- **KIRPMA İŞLEMLERİ**
- Metin kırpma için kullanılan “substr” fonksiyonun genel görünümünü içeren bir örnek aşağıdaki gibi verilebilir.

```
1 <?php
2 $metin = "ATA-AÖF";
3 $kesim = substr( $metin, 0, 3 );
4
5 echo $kesim; //ATA
6 ?>
```



Metin kırpma işlemleri kırılacak karakterin sıra numarası ve bu karakterden sonra kaç karakter alınacağını belirtilmesi ile olmaktadır.

- Yukarıdaki örnekten anlaşılacağı gibi; “substr” fonksiyonu, ilk parametre olarak “\$metin” değişkenini almaktadır. Bu parametre için ayrıca doğrudan, “ATA-AOF” gibi, metin değeri yazılabilir.



Fonksiyonlar

- **KIRPMA İŞLEMLERİ**
- İkinci parametre olarak, kırpma işleminin başlayacağı karakterin sıra numarası verilmektedir. Örnekte yer alan sıfır (0) değeri, ilk karakterin önünden kırpma başlanması için verilmiştir.
- Üçüncü parametre olarak, kırılacak toplam karakter sayısı verilmektedir. Örneğimizde değer olarak üç (3) alınmasının sonucunda ilk üç karakterin geri dönmesi sağlanmıştır.

```
1 <?php
2 $metin = "ATA-AÖF";
3 $kesim = substr( $metin, 0, 3 );
4
5 echo $kesim; //ATA
6 ?>
```

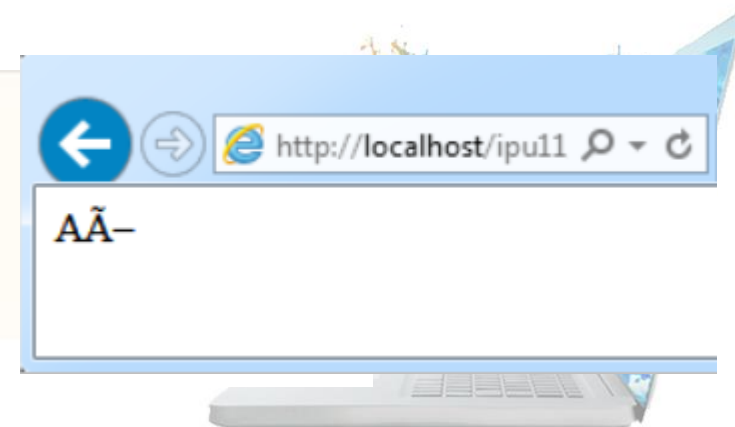


Fonksiyonlar

- **KIRPMA İŞLEMLERİ**

- Bir sonraki kullanım örneği olarak yine “ATA-AÖF” kelimesini ele alabiliriz. Kırpma işlemi sonucunda “AÖF” kelimesini elde etmek için tire (-) işaretinden sonra üç (3) karakter almamız gerekmektedir. Bu durumda “substr” fonksiyonunun ikinci parametresi (kırpmanın başlanacağı karakterin sıra numarası) değer olarak dört (4) almalıdır. İkinci parametre ise üç karakter istediğimizden dolayı üç (3) değerine sahip olmalıdır. Bu durumda fonksiyonumuzun görünümü,

```
1 <?php
2 $metin = "ATA-AÖF";
3 $kesim = substr( $metin, 4, 3 );
4
5 echo $kesim; //AÖF **beklenen
6 ?>
```

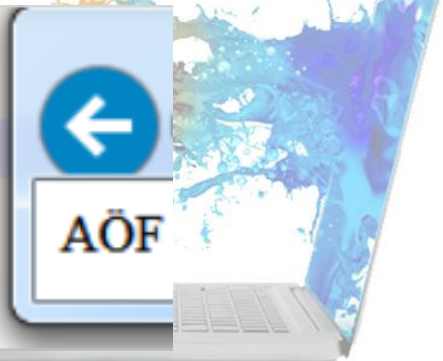


Fonksiyonlar

- **KIRPMA İŞLEMLERİ**

- Türkçe karakterleri sayma işlemi zahmetli gelirse; “**mb_substr**” fonksiyonunu da kullanılabilir. Bu fonksiyon “substr” fonksiyonu ile aynı parametreleri alır; sadece ek olarak hangi karakter kodlamasının kullanılacağı da fonksiyona belirtilir. Bizler UTF8 karakter kodlaması kullandığımız için parametre değerini bu şekilde vermeliyiz. Yukarıdaki örneğin “mb_substr” fonksiyonu kullanılarak yeniden kırpma işlemine alınması sonucunda oluşacak PHP kodlarının ve tarayıcı ekranının görüntüsü aşağıdaki gibi olacaktır.

```
1 <?php
2 $metin = "ATA-AÖF";
3 $kesim = mb_substr( $metin, 4, 3, "UTF8" );
4
5 echo $kesim; //AÖF
6 ?>
```



Fonksiyonlar

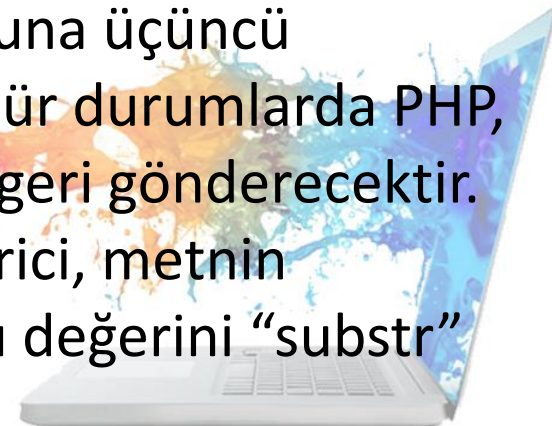
- **KIRPMA İŞLEMLERİ**
- PHP dilinde karakter kırpma işlemini daha iyi anlayabilmek için aşağıdaki örnekleri inceleyebilirsiniz. Örnekte yeşil olan metinler, ekranda oluşacak çıktıları temsil etmektedir.

```
1 <?php
2 $metin = "ATA-AOF";
3
4 echo substr( $metin, 0, 3 ); //ATA
5 echo substr( $metin, 0, 4 ); //ATA-
6 echo substr( $metin, 3, 1 ); //-
7 echo substr( $metin, 4, 3 ); //AOF
8 echo substr( $metin, 2, 4 ); //A-AO
9 echo substr( $metin, -3 ); //AOF
10 echo substr( $metin, -3, 2 ); //AO
11 echo substr( $metin, 3 ); //-AOF
12 echo substr( $metin, 4 ); //AOF
13 ?>
```



Fonksiyonlar

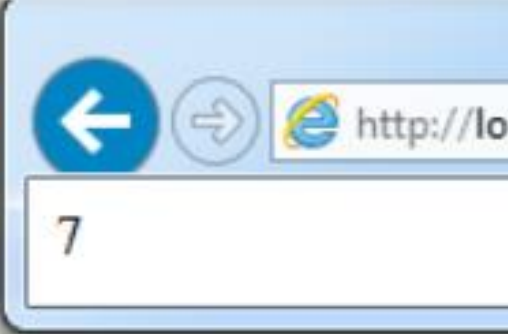
- **KIRPMA İŞLEMLERİ**
- Örnekte yer alan 9. ve 10. satırlardaki ikinci parametrenin negatif olmasından anlaşılacağı gibi, tersten kesme işlemi de yapılabilmektedir. Yani, ikinci parametrenin negatif olması halinde PHP, metni ters çevirip kırpma işlemini yapar, daha sonra sonucu tekrar ters çevirip kullanıcıya iletir. Bu işlev kullanıcıya, verilen metindeki son karakterleri kırpma için büyük kolaylık sağlamaktadır.
- Ayrıca; 11. ve 12. satırlarda “substr” fonksiyonuna üçüncü parametrelerin gönderilmediği görülebilir. Bu tür durumlarda PHP, verilen karakterden sonraki metnin tamamını geri gönderecektir. Aksi halde, metnin kalan kısmını isteyen geliştirici, metnin uzunluğunu “strlen” fonksiyonu ile ölçerek sayı değerini “substr” fonksiyonuna belirtmek zorunda kalırdı.



Fonksiyonlar

- **UZUNLUK HESAPLAMA**
- PHP ile bir metnin kaç karakterden oluştuğunun hesaplanması “**strlen**” fonksiyonu ile mümkündür. Bu fonksiyon parametre olarak sadece metin değerini almaktadır. Geri dönen cevap ise metin uzunluğunun sayısal değeridir. Örnek olarak aşağıdaki kod parçası ve ekran görüntüsü incelenebilir.

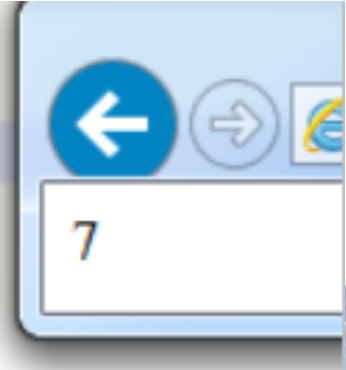
```
1 <?php
2 $metin = "ATA-AOF";
3 $sayim = strlen( $metin );
4
5 echo $sayim;
6 ?>
```



Fonksiyonlar

- **UZUNLUK HESAPLAMA**
- Türkçe karakterlerde ortaya çıkan bu sorunun çözümü için aşağıdaki örnekte olduğu gibi “**mb_strlen**” fonksiyonu kullanılabilir. Ayrıca; “mb_strlen” fonksiyonu, ikinci parametre olarak kaynak kodlaması olan “UTF8” değerini alacaktır.

```
1 <?php
2 $metin = "ATA-AÖF";
3 $sayim = mb_strlen( $metin, "UTF8" );
4
5 echo $sayim;
6 ?>
```



Fonksiyonlar

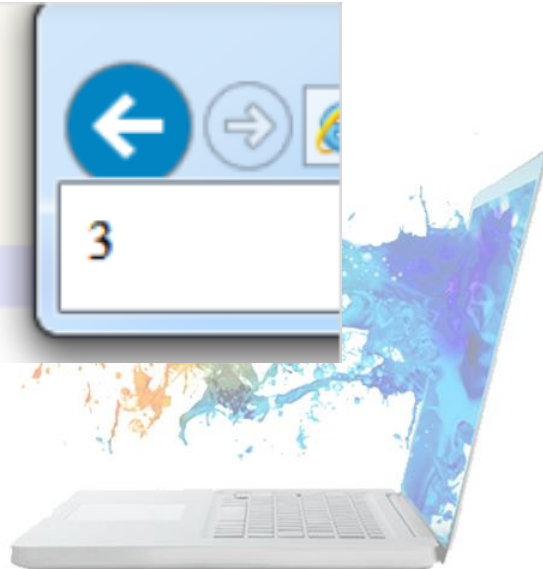
- **KONUM TESPİTLERİ**
- Geliştiriciler bazı durumlarda belirli bir karakterin, metin içinde kaçınıcı sırada olduğunu bulmak zorunda kalabilirler. Bu tür durumlar genellikle kırpma işlemlerinin dinamik olarak kullanıldığı zamanlarda ortaya çıkmaktadır.
- Örneğin; bir e-posta adresi içeren metnin “@” işaretinden önceki kısmını elde etmek isteyen geliştirici; öncelikle “@” işaretinin bulunduğu karakter sırasını bulmalı, daha sonra kırpma işlemine girmelidir.



Fonksiyonlar

- **KONUM TESPİTLERİ**
- Sıra numarasının bulunması için kullanılan fonksiyon “**strpos**”dur. Bu fonksiyon adını “string (metin)” ve “position (pozisyon)” kelimelerinin ilk üç harflerinin birleşiminden almaktadır. Fonksiyonun kullanımına örnek olarak aşağıdaki kod parçası ve ekran çıktısı verilebilir.

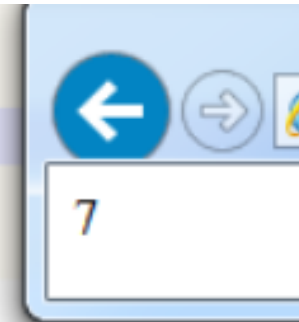
```
1 <?php
2 $metin = "ATA-AÖF";
3 $konum = strpos( $metin, "-" );
4
5 echo $konum;
6 ?>
```



Fonksiyonlar

- **KONUM TESPİTLERİ**
- Konum tespitinde Türkçe karakterlerin öneminin yanı sıra, karakterlerin büyük-küçük harf duyarlılıkları da vardır. Öncelikle bu durumu incelemek için, örneğimizde yer alan “F” harfinin konumunu tespit etmeye çalışabiliriz. Bu durumu temsil eden kod parçası ve ekran görüntüsü aşağıdaki gibi olacaktır.

```
1 <?php
2 $metin = "ATA-AÖF";
3 $konum = strpos( $metin, "F" );
4
5 echo $konum;
6 ?>
```



Fonksiyonlar

- **KONUM TESPİTLERİ**
- Şimdi ise aynı örneği “f” karakterinin konumunun tespiti için yeniden uyarlayalım. Küçük harf kullanıldığı zaman “strlen” fonksiyonun sonucunu görebilmek için gerekli kod ve oluşacak ekran görüntüsü aşağıdaki gibi olacaktır.

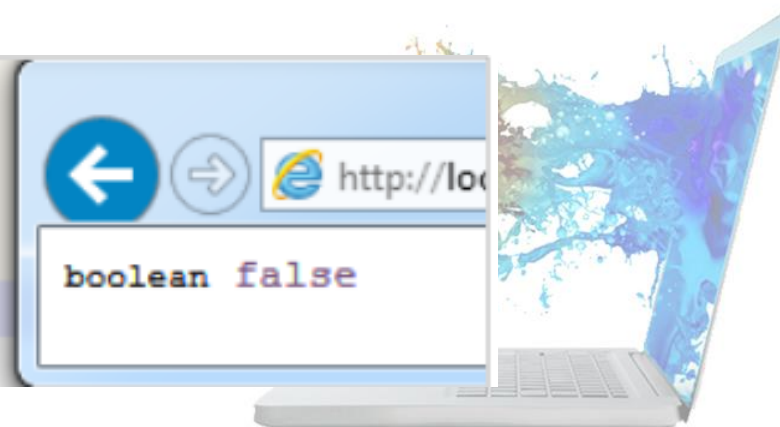
```
2 $metin = "ATA-AÖF";  
3 $konum = strpos( $metin, "f" );  
4  
5 echo $konum;
```



Fonksiyonlar

- **KONUM TESPİTLERİ**
- Örnekte tarayıcı ekranında herhangi bir sonuç oluşmadığı görülebilir. Bu durumun nedeni, metin içerisinde küçük “f” karakterinin bulunmamasıdır. Örnekten anlaşılacağı gibi; fonksiyon eşleşme bulamadığı durumlarda “false” değeri geri göndermektedir. Bu sonucu tarayıcı ekranında görebilmek için “**echo**” komutu yerine “**var_dump()**” fonksiyonu ile çıktı alabiliriz. Bu durumda ekran görüntüsü ve kod parçası aşağıdaki gibi olacaktır.

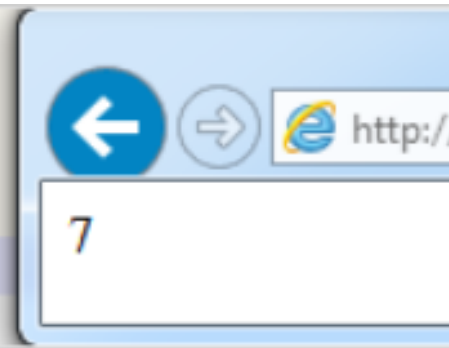
```
1 <?php
2 $metin = "ATA-AÖF";
3 $konum = strpos( $metin, "f" );
4
5 var_dump( $konum );
6 ?>
```



Fonksiyonlar

- **KONUM TESPİTLERİ**
- Konum tespiti yapılırken “**strpos**” fonksiyonunu büyük-küçük harf duyarız (case-insensitive) haline getirmek mümkün olmasa da “**stripos**” fonksiyonu ile aynı işlemi gerçekleştirmek mümkündür.
- PHP fonksiyonlarında araya giren “i” harfi genel olarak büyük-küçük harf duyarız (case-insensitive) işlevini temsil etmektedir.

```
1 <?php
2 $metin = "ATA-AÖF";
3 $konum = stripos( $metin, "f" );
4
5 echo $konum;
6 ?>
```



Fonksiyonlar

- **KONUM TESPİTLERİ**
- Türkçe karakterlerin bulunduğu metin içi konum tespit işlemlerinde “**mb_strpos**” ve “**mb_stripos**” fonksiyonları kullanılabilir.
- Ancak; bu fonksiyonları kullanırken aramaya başlanacak noktanın belirtilmesi zorunludur. Yukarıda yapılan işlemleri tekrar edecek olursak; aşağıdaki sonuçları elde ederiz.

```
1  <?php
2  $metin = "ATA-AÖF";
3  $konum = mb_strpos( $metin, "Ö", 0, "UTF8" );
4
5  echo $konum;
6  ?>
```



Fonksiyonlar

- **MATEMATİKSEL İŞLEMLER**
- PHP ile birçok matematiksel işlem yapmak mümkündür. Basit dört işlemleri doğrudan yapabilen PHP, trigonometrik ve logaritmik birçok işlev için bünyesinde hazır hesaplama fonksiyonları barındırmaktadır.
- Bu fonksiyonların tam listesine <http://tr2.php.net/manual/tr/ref.math.php> adresinden ulaşılabilir.



Fonksiyonlar

- **MATEMATİKSEL İŞLEMLER**
- Aşağıdaki tabloda ise sık kullanılan matematiksel fonksiyonlar ve açıklamaları yer almaktadır.

Fonksiyon	Açıklama
abs	Sayının gerçek değerini (negatif olanları pozitive çevirir) verir.
sin	Verilen açının sinüs değerini geri gönderir
cos	Verilen açının kosinüs değerini geri gönderir
floor	Aşağı yuvarlama yapar
round	Tam sayıya yuvarlar
ceil	Yukarı yuvarlama yapar
pi	Pi değerini verir
max	Verilen iki sayıdan maksimum olanını geri gönderir
min	Verilen iki sayıdan minimum olanını geri gönderir
sqrt	Verilen sayının karekökünü geri gönderir



Fonksiyonlar

- **MATEMATİKSEL İŞLEMLER**

- **round** Fonksiyonu

- `<?php`

```
function yuvarla( $a)  
{  
    echo round($a);  
}  
yuvarla(3.21);
```

```
?>
```

3



Fonksiyonlar

- **MATEMATİKSEL İŞLEMLER**
- **POW** Fonksiyonu: üs alma işlemi yapar.

- `<?php`

```
    echo pow(2,8);
```

256

```
?>
```

- **MAX , MIN** Fonksiyonu

- `<?php`

```
    echo max(5,9,1,45,15,25);
```

```
    echo min (5,9,1,45,15,25);
```

45

1

```
?>
```



Ödev

- 1- Mail adresindeki @ işaretinin kaçınıcı karakter olduğunu ekrana yazdıran kod?
- 2- Girilen Mail adresindeki @ işaretinden önceki kullanıcı adı bilgisini ekrana yazdıran kodu yazınız?
- 3- Toplam mail adresinde kaç karakter olduğunu fonksiyon içinde bulup ekrana yazdıran kod?
- 4- @ işaretinden sonraki domain adresini kırpıp ekrana yazdıran kodu ekleyiniz?
- Mail adresi: hsengun@mehmetakif.edu.tr



Cevap

```
<?php
```

```
$metin ="huseyinsengun1515@mehmetakif.edu.tr";
```

```
$konum=mb_strpos($metin,"@",0,"UTF8");
```

```
echo "$konum<br>";
```

```
$kesilen=mb_substr($metin,0,$konum,"UTF8");
```

```
echo "$kesilen<br>";
```

```
$sayim=mb_strlen($metin,"UTF8");
```

```
echo "$sayim<br>";
```

```
$konum++;
```

```
$domain=substr($metin,$konum);
```

```
echo $domain;
```

```
?>
```





Kaynaklar

- PHP Kılavuzu - <http://php.net/manual/tr/index.php>
- PHP Tutorials - <https://www.w3schools.com/php/default.asp>
- PHP Videolu Temel Dersler İçin: Youtube/Php Türkiye
- <https://php-mysql.org/>
- <http://www.hsengun.com>

