String Functions

STRING FUNCTIONS

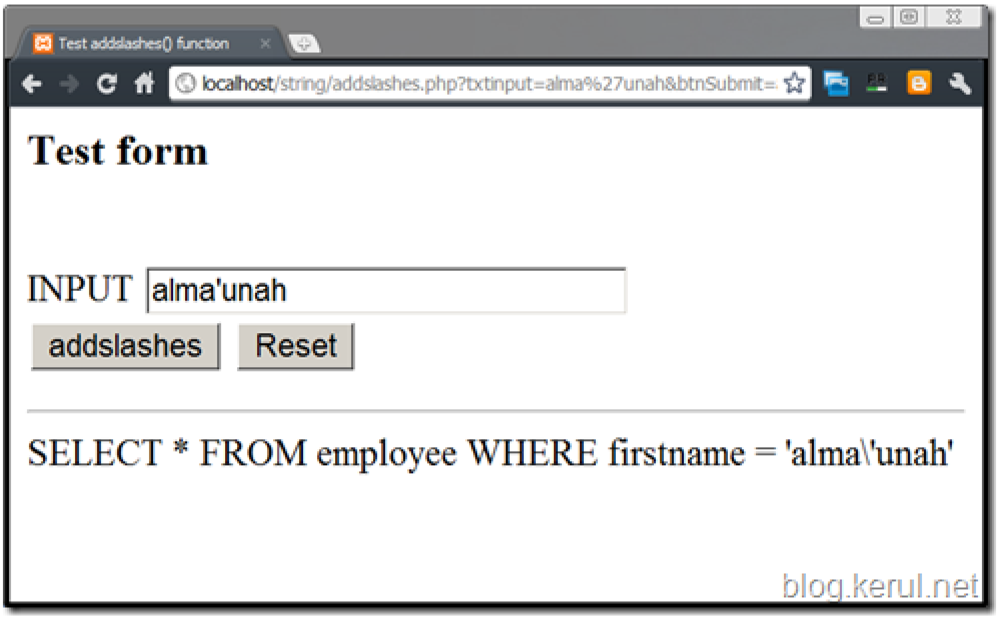
String functions are used in computer programming languages to manipulate a string or query information about a string (some do both).

Frequently used STRING functions

EXAMPLE 1
`addslashes` — Quote string with slashes

In the example, the addslashes function prevents the SQL string from being misinterpreted by the single quote in the middle of the search criteria.

```
<html>
<head><title>Test addslashes() function</title></head>
<body>
<h3>Test form </h3><br>
<form name="formtest"  method= "GET"  action="">
  INPUT
      <input name="txtinput"  type="text" size="30"><br>
      <input name="btnSubmit"  type="submit"  value="addslashes">
      <input name="btnReset"  type="reset"  value="Reset">
</form>
<hr>
<?php
    $in=$_GET["txtinput"];
    if ($in==NULL){
        echo "Pls enter a string with quote";
    }
    else{
        $sql="SELECT * FROM employee WHERE firstname =
'".addslashes($in)."'";
        echo $sql;
    }
?>

</body>
</html>
```



Test addslashes() function
localhost/string/addslashes.php?txtinput=alma%27unah&btnSubmit=
blog.kerul.net

Test form

INPUT alma'unah

addslashes Reset

SELECT * FROM employee WHERE firstname = 'alma\'unah'

Example for **`addslashes`** function

`stripslashes`– this is the inverse of `addslashes` where you remove backslashes from a string.

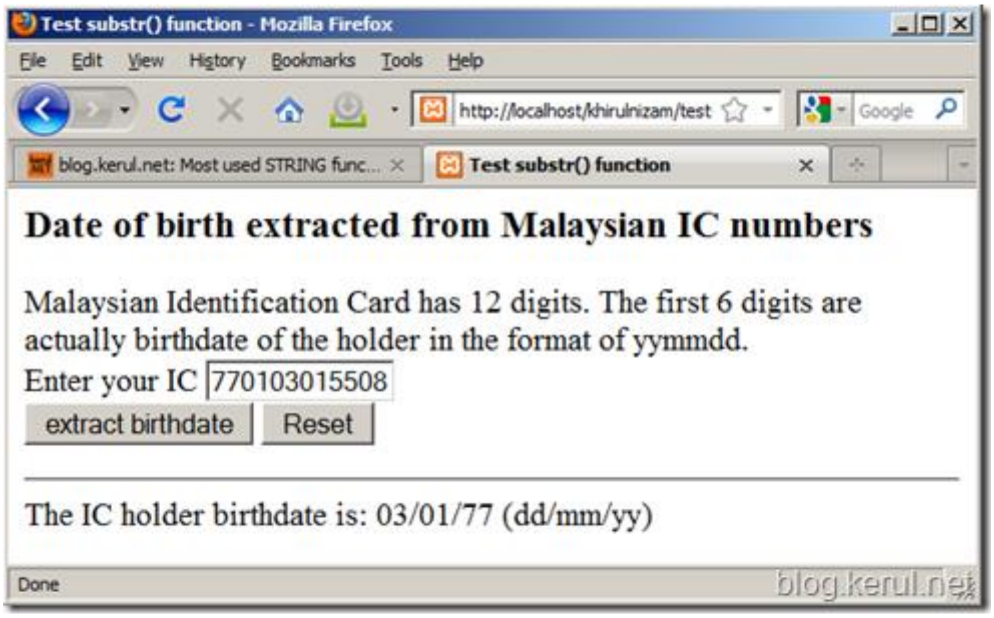
EXAMPLE 2

`strlen` — Get string length

`substr` — Return part of a string

In the example, we are trying to extract birthdate information from a Malaysian IC numbers.

```
<html>
<head><title>Test substr() function</title></head>
<body>
<h3>Date of birth extracted from Malaysian IC numbers </h3>
Malaysian Identification Card has 12 digits.
The first 6 digits are actually birthdate of the holder
in the format of yymmdd.
<br>
<form name="formtest"  method= "GET"  action="">
  Enter your IC
      <input name="txtic"  type="text" size="12" maxlength="12"><br>
      <input name="btnSubmit"  type="submit"  value="extract birthdate">
      <input name="btnReset"  type="reset"  value="Reset">
</form>
<hr>
<?php
    $ic=$_GET["txtic"];
    if ($ic!=NULL && ctype_digit($ic) && strlen($ic)==12){
        // $ic!=NULL - to check there's actually input in the textbox
        // ctype_digit($ic) - all characters are number
        // strlen($ic)==12 - to check there are 12 digits
        $year=substr($ic,0,2);//extract digit 1 and 2
        $month=substr($ic,2,2);//extract digit 3 and 4
        $day=substr($ic,4,2);//extract digit 5 and 6

        echo "The IC holder birthdate is: ";
        echo "$day/$month/$year (dd/mm/yy)";// date format dd/mm/yy
    }
    else{
        echo "Enter an IC number (must be in 12 digits format)";
    }
?>

</body>
</html>
```


Test substr() function - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/khirulnizam/test
Google
blog.kerul.net: Most used STRING func...
Test substr() function
Done
blog.kerul.net

Date of birth extracted from Malaysian IC numbers

Malaysian Identification Card has 12 digits. The first 6 digits are actually birthdate of the holder in the format of yymmdd.

Enter your IC 770103015508

extract birthdate Reset

The IC holder birthdate is: 03/01/77 (dd/mm/yy)

Examples for **`strlen`** and **`substr`** function

EXAMPLE 3

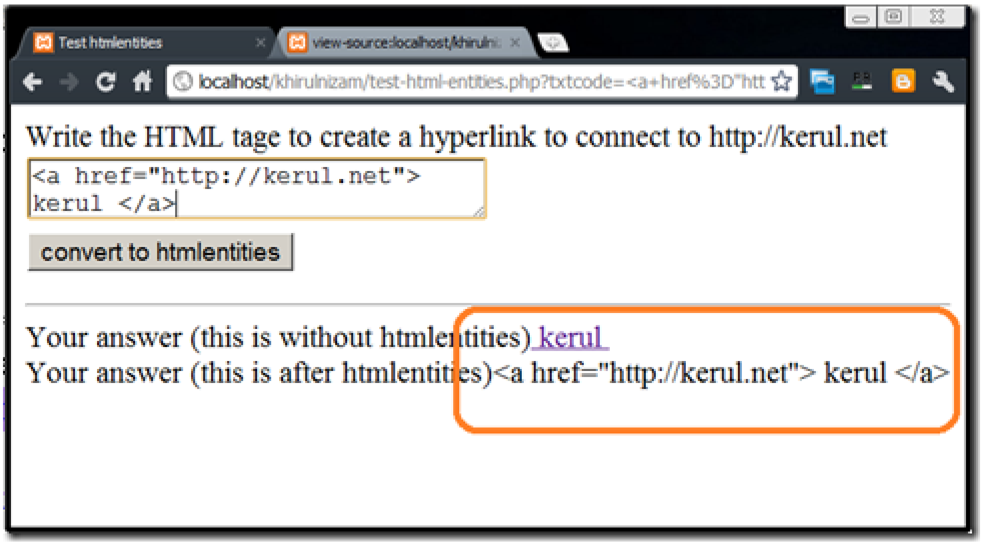
`htmlentities` — Convert all applicable characters to HTML entities

In the example, we are trying to print a HTML tag into the HTML page.

```
<html>
<head>
<title>Test htmlentities</title>
</head>

<body>
<form name="formtest"  method= "GET"  action="">
  Write the HTML tage to create a hyperlink to connect to
http://kerul.net<br>
      <textarea name="txtcode"></textarea>
      <br>
      <input name="btnSubmit" type="submit" value="convert to
htmlentities">
</form>
```

```
<hr>
<?php
    $code=$_GET["txtcode"];
    //echo " $ic <br>";
    if ($code!=NULL ){

        echo "Your answer (this is without htmlentities)";
        echo $code;
        echo "<br>\n";
        echo "Your answer (this is after htmlentities)";
        echo htmlentities($code);
        echo "<br>\n";


    }else{
        echo "Please provide the code";
    }
?>
</body>
</html>
```

The output;


Test htmlentities
view-source:localhost/khirulni
localhost/khirulnizam/test-html-entities.php?txtcode=<a+href%3D"htt

Write the HTML tage to create a hyperlink to connect to http://kerul.net

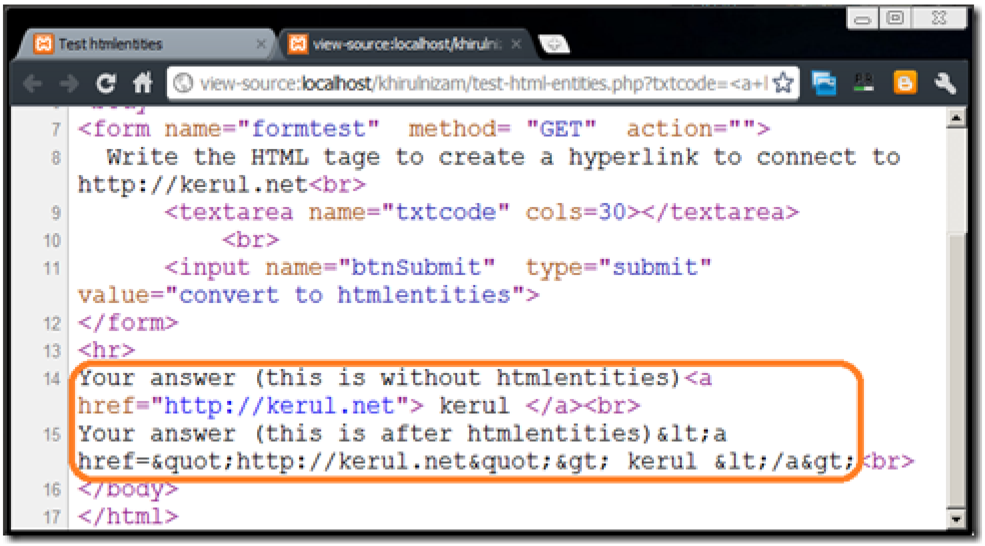
```
<a href="http://kerul.net">
kerul </a>
```

convert to htmlentities

Your answer (this is without htmlentities) kerul
Your answer (this is after htmlentities)<a href="http://kerul.net"> kerul </a>

Example for **`htmlentities`** function

See the HTML tag for the page,


Test htmlentities
view-source:localhost/khiruln
view-source:localhost/khirulnizam/test-html-entities.php?txtcode=<a+l

```
<form name="formtest"  method= "GET"  action="">
  Write the HTML tage to create a hyperlink to connect to
http://kerul.net<br>
      <textarea name="txtcode" cols=30></textarea>
          <br>
      <input name="btnSubmit"  type="submit"
value="convert to htmlentities">
</form>
<hr>
Your answer (this is without htmlentities)<a
href="http://kerul.net"> kerul </a><br>
Your answer (this is after htmlentities)&lt;a
href=&quot;http://kerul.net&quot;&gt; kerul &lt;/a&gt;<br>
</body>
</html>
```

HTML tag for the page

html_entity_decode — Convert all HTML entities to their applicable characters

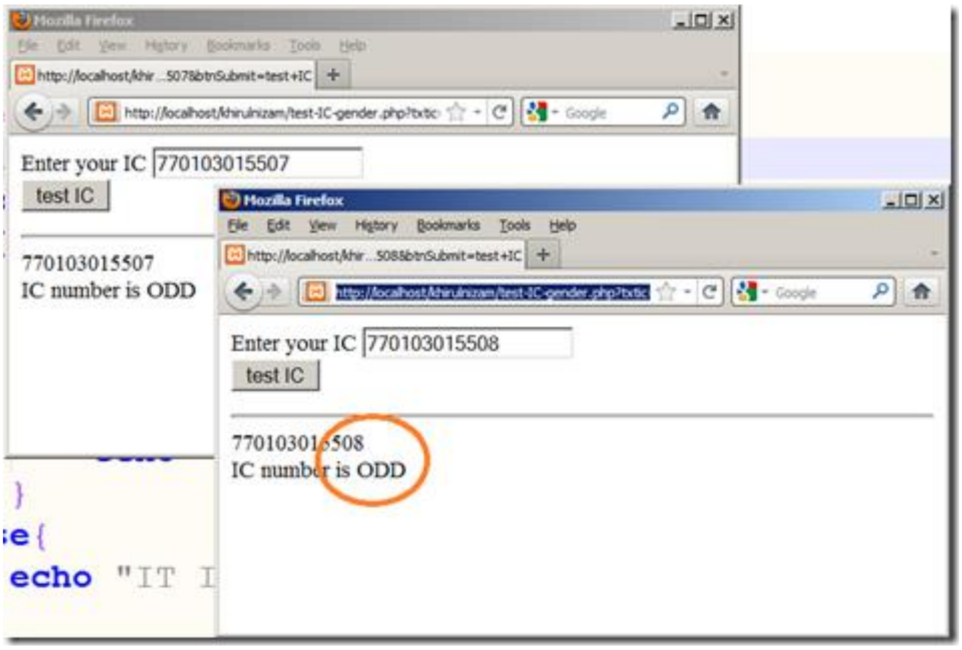
EXAMPLE 4 – String to number (big integer)

FYI, the maximum range for an integer value in PHP variable is 2147483647. I got this from the code;

```
<?php
echo PHP_INT_MAX; //to get the maximum integer value PHP variable can hold
?>
```

Guest what happen if I run this code to check Malaysian IC number whether it’s ODD or EVEN (input comes from a textbox – means it in STRING). Malaysian IC number consists of 12 digits number.

```
<form name="formtest"  method= "GET"  action="">
  Enter your IC
      <input name="txtic"  type="text"maxlength="12"><br>
      <input name="btnSubmit"  type="submit"  value="test IC">
</form>
<hr>
<?php
    $ic=$_GET["txtic"];
    echo " $ic <br>";
    if ($ic!=NULL && ctype_digit($ic) && strlen($ic)==12){
        if ($ic%2==0){//IC is even
            echo "IC number is EVEN";
        }
        else{
            echo "IC number is ODD";
        }
    }else{
        echo "IT IS NOT A VALID IC NUMBER";
    }
?>
```


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/khir...507&btnSubmit=test+IC
http://localhost/khirulnizam/test-IC-gender.php?txtic
Google
Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/khir...508&btnSubmit=test+IC
http://localhost/khirulnizam/test-IC-gender.php?txtic
Google
}
e{
echo "IT I

Enter your IC 770103015507

test IC

770103015507
IC number is ODD

Enter your IC 770103015508

test IC

770103015508
IC number is ODD

It will always tell you the number is ODD…

Why? Not quite sure why. Yet the solution is simple. Add 0 to the $ic as in the following code, and problem solved. They said it something to do with changing the integer variable into unsigned..

```
<form name="formtest"  method= "GET"  action="">
  Enter your IC
      <input name="txtic"  type="text"maxlength="12"><br>
      <input name="btnSubmit"  type="submit"  value="test IC">
</form>
<hr>
<?php
    $ic=$_GET["txtic"];
    $ic=$ic+0;
    echo " $ic <br>";
    if ($ic!=NULL && ctype_digit($ic) && strlen($ic)==12){
        if ($ic%2==0){//IC is even
            echo "IC number is EVEN";
        }
        else{
            echo "IC number is ODD";
        }
    }else{
        echo "IT IS NOT A VALID IC NUMBER";
    } ?>
```

EXAMPLE 5

array_count_values() - returns an array using the values of array as keys and their frequency in array as values.

```
<?php
$array = array(1, "hello", 1, "php", "hello");
print_r(array_count_values($array));
?>
```

The output for above example will

```
Array
(
    [1] => 2
    [hello] => 2
    [php] => 1
)
```

The following is example for word

EXAMPLE 6

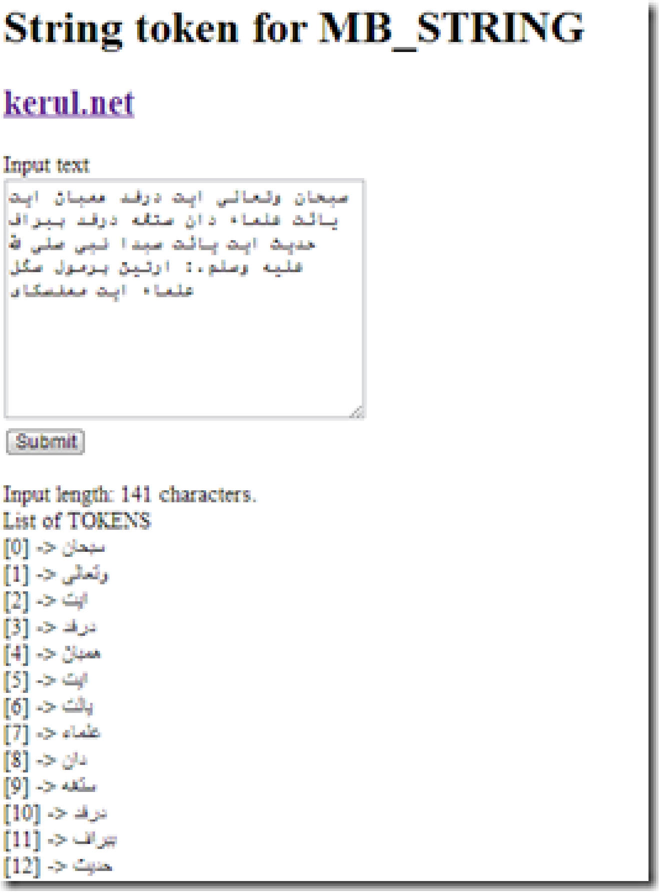
mb-strlen() - When counting the length of an UTF-8 string.

```
if (mb_strlen($name, 'UTF-8') < 3) {
    $error .= 'Name is required. Minimum of 3 characters required';
}
```

EXAMPLE 7

mb_strtok() - String tokenizer for Multibyte

This is a simple function to implement some kind of `mb_strtok()` in PHP. As maybe you all are aware the `mb_strtok` function does not available for multibyte string (aka Unicode string). So this is my attempt to solve the problem. Anyway, there are bugs where the program halt if the input text is too long (how long? not sure yet). Maybe you could improve to provide better result.



String token for MB_STRING

kerul.net

Input text

سبحان وتعالى ايت درڤد ممبان ايت
يالث علماء دان ستله درڤد بيراڤ
حديث ايت يالث سبدا نبي صلى ﷲ
عليه وسلم : ارتيڽ برسول سکل
علماء ايت معلمکان

Input length: 141 characters.
List of TOKENS
[0] -> سبحان
[1] -> وتعالى
[2] -> ايت
[3] -> درڤد
[4] -> ممبان
[5] -> ايت
[6] -> يالث
[7] -> علماء
[8] -> دان
[9] -> ستله
[10] -> درڤد
[11] -> بيراڤ
[12] -> حديث

String tokenization

The PHP code *mb_strtok.php* ;

```
<html>
    <head>
        <title>String token for MB_STRING</title>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
    </head>
    <body>
        <h1>String token for MB_STRING</h1>
      <h2><a href="http://kerul.net">kerul.net</a></h2>

<form method="GET" ACTION="">
Input text <br>
<textarea name="txtinput" cols=30 rows=10></textarea>
<br>
<input type="submit" >
</form>

<?php
$in=$_GET["txtinput"];
$inputlen=mb_strlen($in, 'UTF-8');
echo ("Input length: $inputlen characters. <br>\n");

$tokens=mb_strtok(" /n/t?\'.", $in);
echo ("List of TOKENS<br>\n");
//echo $tokens;
for($i=0; $i<count($tokens); $i++){
    echo ("[$i] -> ".$tokens[$i] ." <br> \n");
}

function mb_strtok($delimiters, $str=NULL)
{
    static $pos = 0; // Keep track of the position on the string for each
subsequent call.
    static $string = "";
    static $listtoken=array();
    // If a new string is passed, reset the static parameters.
    if($str!=NULL)
    {
        $pos = 0;
        $string = $str;
    }

    // Initialize the token.
    $token = "";

    while ($pos < mb_strlen($string,'UTF-8'))//loop till end of input
string
    {

        $char = mb_substr($string, $pos, 1);//fetch one character, pos =
char position
        $pos++;
        //echo ("Char at $pos => $char <br>\n");//trace character at
position

        if(mb_strpos($delimiters, $char)===FALSE)//if character is not
delimeter
        {
            $token .= $char;//put character in the token node
        }
        else
        {
           //if arrive at delimeter, push token to listtoken
           array_push($listtoken, $token);
           $token="";//clear the token node
        }
    }
    // return the list of tokens
    if ($listtoken!="")
    {
        return $listtoken;
    }
    else
    {
        return false;
    }
}
?>
  </body>
</html>
```

There is another one, this time the separator (.,;:) will be stored in the list of token (listtoken).

```
<html>
    <head>
        <title>String token for MB_STRING</title>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
    </head>
    <body>
        <h1>String token for MB_STRING</h1>
       <h2><a href="http://kerul.net">kerul.net</a></h2>

    <form method="GET" ACTION="">
    Input text <br>
    <textarea name="txtinput" cols=30 rows=10></textarea>
    <br>
    <input type="submit" >
    </form>

    <?php
    $in=$_GET["txtinput"];
    $inputlen=mb_strlen($in, 'UTF-8');
    echo ("Input length: $inputlen characters. <br>\n");

    $tokens=mb_strtok(" /n/t/f", $in);//delimeter by whitespace only
    echo ("List of TOKENS<br>\n");
    //echo $tokens;
    for($i=0; $i<count($tokens); $i++){
        echo ("[$i] -> ".$tokens[$i] ." <br> \n");
    }

    function mb_strtok($delimiters, $str=NULL)
    {
        static $pos = 0; // Keep track of the position on the string for
each subsequent call.
        static $string = "";
        static $listtoken=array();
        // If a new string is passed, reset the static parameters.
        if($str!=NULL)
        {
            $pos = 0;
            $string = $str;
        }

        // Initialize the token.
        $token = "";

        while ($pos < mb_strlen($string,'UTF-8'))//loop till end of input
string
        {

            $char = mb_substr($string, $pos, 1, 'UTF-8');//fetch one
character, pos = char position

            echo ("Char at $pos => $char <br>\n");//trace character at
position


            if(mb_strpos($delimiters, $char)===FALSE)//if character is not
delimeter
```

```
        {
            if($char=="." || $char==";"||$char==":"||$char==","){
                echo "Token detected $token <br>\n";
                array_push($listtoken, $char);
           //$token="";//clear the token node
            }else{
                $token .= $char;//put character in the token node
            }
        }
        else
        {
           //if arrive at delimeter, push token to listtoken
           echo "Token detected $token <br>\n";
           array_push($listtoken, $token);
           $token="";//clear the token node
        }
        $pos++;
    }
    return $listtoken;
    // return the list of tokens
    if ($listtoken!="")
    {
        return $listtoken;
    }
    else
    {
        return false;
    }

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

Other string functions

`similar_text` — Calculate the similarity between two strings

`md5` — Calculate the md5 hash of a string

`money_format` — Formats a number as a currency string

`echo` — Output one or more strings

`printf` — Output a formatted string

`str_word_count` — Return information about words used in a string

`strtok` — Tokenize string

`trim` — Strip *whitespaces (or other characters) from the beginning and end of a string

`ucwords` — Uppercase the first character of each word in a string

*Whitespaces are hidden characters such as space, tab and newline.

EXERCISE – String Functions

Exercise 1: Accept a date in dd/mm/yyyy format and convert into ISO date (yyyy-mm-dd). Make sure the input format is correct before converting to another format.

Answer:

```
<form method="get" action="">
Date <input type="text" name="txtdate">
    format (dd/mm/yyyy)<br>
    <input type="submit">
</form>
<?php
$date=$_GET['txtdate'];
if (strlen($date)==10){

    $day=substr($date, 0, 2);
    $month=substr($date, 3, 2);
    $year=substr($date, 6, 4);

    if(ctype_digit($day)&&ctype_digit($month)&&ctype_digit($year)){
        //display
        echo "Input date: $date<br>";
        echo "ISO date: $year-$month-$day";
    }
    else{
        echo "Date format not valid <br>";
    }
}
else{
    echo "Date format not valid <br>";
}
?>
```

Exercise 2: Accept a date in ISO format (yyyy-mm-dd) and convert to dd MONTH yyyy. Make sure the input format is correct before converting to another format.
Eg: Accept 2011-07-14 as the input, and output 14 JULY 2011

Answer:

```
<form method="GET" action="">
Task: Convert ISO date (yyyy-mm-dd) into dd NAME_OF_MONTH yyyy <br>
Input ISO date<input type="text" maxlength=10 name="txtisodate">
(format: yyyy-mm-dd)<br>
<input type="submit" value="Convert">

</form>
<hr>
Output
<br>

<?php
$date=$_GET['txtisodate'];
if (strlen($date)==10){

    $year=substr($date, 0, 4);
    $month=substr($date, 5, 2);
    $day=substr($date, 8, 2);
    if(ctype_digit($day)&&ctype_digit($month)&&ctype_digit($year)){
        //display
        echo "Input ISO date: $date<br>";
        if($month=="01")
            $monthname="January";
        else if ($month=="01")
            $monthname="February";
        else if ($month=="02")
            $monthname="February";
        else if ($month=="03")
            $monthname="March";
        else if ($month=="04")
            $monthname="April";
        else if ($month=="05")
            $monthname="May";
        else if ($month=="06")
            $monthname="June";
        else if ($month=="07")
            $monthname="July";
        else if ($month=="08")
            $monthname="August";
        else if ($month=="09")
            $monthname="September";
        else if ($month=="10")
            $monthname="October";
        else if ($month=="11")
            $monthname="November";
        else if ($month=="12")
            $monthname="December";
        else
            $month="NOT_VALID_VALUE";
        echo "After conversion: $day $monthname $year";

    }
    else{
        echo "Date format not valid <br>";
    }
}
else{
    echo "Date format not valid <br>";
}
?>
```