פרק 18 – פונקציות שימושיות לעבודה עם מחרוזות

הקדמה

פרק זה הוא הפרק האחרון בחלק הדן ב PHP, הפרק הבא יכיר אותנו עם ההיסטוריה של שפת SQL ועם MySQL, אבל עד אז כדאי שנלמד כיצד לעבוד עם מחרוזות, שהרי כאלו נקבל המון מבסיסי נתונים.

בפרק הנוכחי נלמד על חמישה פונקציות לעבודה עם מחרוזות, אלו חמישה מתוך מספר רב יותר של פונקציות אשר תיעוד להן ניתן למצוא באתר הבית של PHP הנמצא בכתובת http://www.php.net. אנו נתמקד בחמשת הפונקציות הללו מכיוון שהן מהשימושיות יותר. אופי הפרק הוא מאוד קליל, בסך הכל נלמד כיצד הפונקציות פועלות ומה תפקידן ונסכם בדוגמא מעשית מעניינת.

הפונקציה strlen

הפונקציה strlen מקבלת כפרמטר מחרוזת ומחזירה את מספר התווים במחרוזת, זוהי ללא ספק הפונקציה השימושית ביותר לטיפול במחרוזות. הבט בקוד הבא בכדי לראות אותה בפעולה:

```
$name = "If you are going to San Francisco";
echo strlen($name);
```

בשורה הראשונה הגדרנו משתנה מסוג מחרוזת בשם $name ואתחלנו את ערכו לשורה משיר שאני שומע כרגע. בשורה השנייה נעזרנו בפונקציה strlen בכדי לקבל את מספר התווים ב $name, במילים אחרות אנו מקבלים את אורך המחרוזת. את האורך שחזר מ strlen העברנו ל echo שדאגה להציג אותו בדפדפן, התוצאה היא אגב – 32.

חשוב לציין ש strlen סופרת גם רווחים שנמצאים מצידי המחרוזת, הבט בקוד הבא:

```
$str = "   I'm in deep space   ";
echo strlen($str);
```

כל הזכויות שמורות לגיל כהן, http://www.flashoo.co.il

הפעם המחרוזת מוקפת על ידי תווי רווח, strlen סופרת גם אותם, והתוצאה שתוצג לדפדפן היא 23.

הפונקציה trim

לעיתים נקבל טקסט מהמשתמש ונרצה לוודא כי הטקסט לא מוקף רווחים כמו בדוגמא הקודמת, במקרה הזה נעזר בפונקציה trim אשר "מנקה" את התווים הללו מהמחרוזת ומחזירה מחרוזת נקייה. Trim לא משנה את המחרוזת שהגיע אליה ולכן אם נרצה לשנות את המחרוזת שאנו מעבירים נצטרך להציב את הערך ש trim מחזירה אל המחרוזת הזו. הבט בדוגמא הבאה אם הדבר לא ברור, וגם אם כן.

```
$str = "     I'm in deep space     ";
$str = trim($str);
echo strlen($str);
```

בשורה הראשונה אנו רואים שוב את המחרוזת $str מאותחלת למחרוזת עם רווחים המקיפים אותה. בשורה השנייה אנו רואים את trim בפעולה, הפונקציה מקבלת כפרמטר את המחרוזת אותה יש לנקות ומחזירה מחרוזת נקייה, את המחרוזת ש trim מחזירה העברנו ל $str, שהוא בעצם המשתנה המחזיק את המחרוזת המקורית שלנו, בכך "עקפנו" את העובדה ש trim לא משנה את המחרוזת המועברת אליה. בשורה האחרונה קראנו ל strlen בכדי שתציג את האורך של המחרוזת החדשה.

הסיבה שלא קראנו ל echo להציג את המחרוזת החדשה היא שגם אם היינו מציגים את המחרוזת הישנה הרווחים לא היו מופעים בגלל אופייה של HTML.

הפונקציות strstr ו- stristr

לעיתים נרצה לבדוק אם מחרוזת מסוימת מכילה מחרוזת אחרת. ב PHP יש כמה דרכים לעשות את זה, אפשר לבנות פונקציה שעושה זאת, אבל קיימת פונקציה ב PHP שעושה את העבודה בשבילנו, וכנראה מהר יותר מהדרך בה אנו היינו מיישמים את האלגוריתם.

הפונקציה strstr מקבלת שתי מחרוזות, הראשונה היא המחרוזת בה יש לחפש והשנייה היא המחרוזת אותה יש לחפש. הפונקציה מחזירה true במקרה שהמחרוזת השנייה אכן מוכלת בראשונה, כלומר במקרה בו המחרוזת השנייה היא תת-מחרוזת של הראשונה. בכל מקרה אחר strstr מחזירה false. הבט בקוד הבא:

כל הזכויות שמורות לגיל כהן, http://www.flashoo.co.il

```
$a = "They are looking for me";
if (strstr($a, "me") != false)
{
     echo "They found me";
} else {
     echo "They didn't find me";
}
```

בשורה הראשונה הגדרנו מחרוזת בשם $a ואתחלנו אותה. בשורה השנייה נעזרנו במשפט התניה שהתנאי שלו מורכב מקריאה ל- strstr המקבלת כפרמטרים את $a ואת המחרוזת "me". אם המחרוזת "me" היא תת-מחרוזת של $a אז strstr לא תחזיר false (אבל היא גם לא תחזיר true, היא תחזיר "me"), בכל מקרה אחר היא תחזיר false. שאר משפט ההתניה בנוי בהתאם לכך שאם "me" היא תת מחרוזת של $a תוצג ההודעה "They found me", אחרת תוצג ההודעה "They didn't find me".

באתר של PHP כתוב שעדיף להשתמש בפונקציה בשם strpos מפני שהיא מהירה יותר מ strstr. אני מניח שהסיבה לכך היא ש strstr מחזירה את כל המחרוזת אשר מתחילה מתת-המחרוזת המבוקשת, לדוגמא אם נחפש אחר תת המחרוזת "ab" בתוך המחרוזת "shabat" אז strstr תחזיר לנו "abat". בכל מקרה אני חייב להודות שהתרגלתי ל strstr והחלטתי ללמד מהניסיון האישי שלי.

לסיום אציין שהפונקציה strstr היא case-sensitive מה שאומר שהיא מתייחסת לתו "A" כתו שונה מהתו "a" וכך לגבי כל אות לועזית. במידה ולא נרצה להבדיל בין התווים ולהתייחס ל "A" כמו ל "a" נוכל להשתמש בפונקציה stristr שפועלת בדיוק כמו strstr רק שהיא case-insensitive – כלומר stristr מתייחסת לאות לועזית קטנה כמו לאות לועזית גדולה.

הפונקציה substr

הפונקציה הבאה שנלמד עליה מחזירה לנו תת-מחרוזת ממחרוזת נתונה כאשר אנו בוחרים היכן "לקטוע" אותה. הפונקציה מקבלת שלושה פרמטרים, הראשון הוא המחרוזת ממנה אנו רוצים לשלוף תת-מחרוזת. הפרמטר השני הוא נקודת ההתחלה כאשר התו הראשון מספרו הוא 0, התו השני 1 וכו'. הפרמטר השני הוא פרמטר אופציונאלי אשר מציין את אורך המחרוזת אותה אנו רוצים לקבל. אם לא נספק ל PHP ערך עבור פרמטר זה PHP תחזיר לנו תת-מחרוזת מהנקודה שקבענו בפרמטר השני ועד לסוף המחרוזת. הבט בדוגמא הבאה:

```
echo substr("What if god was one of us", 16, 3);
```

בשורת קוד זו קראנו לפונקציה substr. הפרמטר הראשון – המחרוזת ממנה אנו רוצים לשלוף תת-מחרוזת היא שם השיר "What if god was one of us", הפרמטר השני אומר ל PHP להתחיל את המחרוזת המוחזרת מהתו ה-16, זה התו "o" במילה "one", הפרמטר השלישי מציין את אורך המחרוזת הרצויה, במקרה שלנו אורך המחרוזת המוחזרת יהיה 3, ולכן הפלט יהיה המחרוזת "one". בכדי לשפשף אותך קצת הבט בקוד הבא:

כל הזכויות שמורות לגיל כהן, http://www.flashoo.co.il

```
$str = "flashOO";
$length = strlen($str);

for($i=0; $i<$length; $i++)
{
     echo $str."<br>";
     $str = substr($str, 1);
}
```

בשורה הראשונה של הקוד הגדרנו משתנה מסוג מחרוזת ואתחלנו את ערכו, בשורה השנייה הגדרנו משתנה בשם $length ואתחלנו אותו להכיל את האורך של $str בעזרת הפונקציה strlen עליה למדנו באחד הסעיפים הקודמים בפרק.

בשורה השלישית אנו נתקלים בלולאת for שרצה מאפס עד ל $length, כלומר היא רצה כמספר התווים שבמחרוזת $str. בגוף הלולאה אנו רואים ש echo מדפיסה את המחרוזת $str וסימן מעבר שורה. בשורה השנייה בגוף הלולאה אנו נעזרים ב substr כדי לחתוך מ $str את תת-המחרוזת של $str המתחילה בתו השני (שמיקומו 1 כי התו הראשון מיקומו 0) ומסתיימת בסוף $str. בעצם בכל איטרציה אנו מוחקים את התו הראשון ב $str. הפלט יראה כך:

flashOO
lashOO
ashOO
shOO
hOO
OO
O

אם היינו רוצים למחוק בכל פעם את התו האחרון בניגוד למחיקת התו הראשון הקוד היה נראה כך:

```
$str = "flashOO";
$length = strlen($str);

for($i=0; $i<$length; $i++)
{
     echo $str."<br>";
     $str = substr($str, 0, $length-$i-1);
}
```

השינוי היחידי נעשה בשורה האחרונה שבגוף לולאה – בפרמטר השלישי ש substr מקבלת הצבנו את הביטוי $length-$i-1 – נסה לחשוב מדוע. את הפרמטר השני קבענו לאפס מפני שאנו רוצים לחתוך את המחרוזת כך שתתחיל מתחילת המחרוזת המקורית. הפלט יראה כך:

כל הזכויות שמורות לגיל כהן, http://www.flashoo.co.il

flashOO
flashO
flash
flas
fla
fl
f

נסה כתרגול לחשוב על דרך בה המחרוזת תקוצץ משני התווים הקיצוניים, כלומר על דרך בה הפלט יראה כך:

flashOO
lashO
ash
s

הפונקציה str_replace

הפונקציה str_replace היא פונקציה מאוד שימושית, הפונקציה מאפשרת להחליף תת-מחרוזת אחת באחרת בתוך מחרוזת נתונה. הפונקציה מקבלת כפרמטר ראשון את תת המחרוזת אותה יש להחליף, הפרמטר השני הוא תת המחרוזת אותה יש להציב במקום תת המחרוזת שהתקבלה על ידי הפרמטר הראשון, והפרמטר השלישי הוא המחרוזת עליה יש לעבוד. הפונקציה לא נוגעת במחרוזת שהועברה אליה. את המחרוזת המעודכנת נקבל כערך חוזר מהפונקציה. במקרה בו תת המחרוזת קיימת מופיעה יותר מפעם אחת הפונקציה str_replace תחליף כל מופע ומופע בתת מחרוזת החדשה שהעברנו דרך הפרמטר השני.

אני כמוך לא מבין למה בחלק מהפונקציות של PHP המחרוזת עליה אנו עובדים באה כפרמטר אחרון ובחלק כפרמטר ראשון, לא מצאתי סיבה בתיעוד של הפונקציה.

הקוד הבא מדגים שימוש ב str_replace, לאחר מכן אנו נבנה פונקציה שמקבלת מחרוזת ומחליפה את תווי הסמיילים שנגדיר לה בתמונות ומחזירה מחרוזת שבהצגה שלה לדפדפן תציג תמונות של סמיילים. כך פועלים כל עורכי הת'רדים בפורומים שונים מבוססי PHP.

```
$str = "I love mother and father but mother more!";
$str = str_replace("mother", "flashOO", $str);
$str = str_replace("father", "flash", $str);
echo $str;
```

בשורה הראשונה הגדרנו משתנה בשם $str ואתחלנו את ערכו. בשורה השנייה והשלישית קראנו ל str_replace, בשורה השנייה החלפנו כל מופע של "mother" שנמצא ב $str במחרוזת "flashOO", ובשורה השלישית החלפנו כל מופע של "father" במחרוזת "flash". התוצאה:

I love flashOO and flash but flashOO more!

ניתן לראות שכל מופע של "mother" הוחלף ב "flashOO" וכל מופע של "father" הוחלף ב "flash".

אבל מה יקרה אם נרצה להחליף 10 תת-מחרוזת ב 10 אחרים? האם נצטרך לבצע 10 קריאות שונות ל str_replace? ובכן באופן בו השתמשנו ב str_replace כרגע התשובה חיובית, אבל יש דרך אחרת. במקום להעביר כפרמטר ראשון ושני מחרוזות ניתן להעביר מערכים, המערך שיועבר כפרמטר ראשון יכיל מחרוזות שאנו רוצים להחליף במחרוזות שנמצאות במערך שמועבר כפרמטר שני, כך כל מופע של המחרוזת בתא הראשון במערך הראשון תוחלף על ידי המחרוזת שנמצאת בתא הראשון במערך השני, וכל הלאה תא אחר תא. הבה נכתוב את הקוד הקודם מחדש תוך שימוש בתחביר המערכים:

```
$str = "I love mother and father but mother more!";

$a = array("mother", "father");
$b = array("flashOO", "flash");

echo str_replace($a, $b, $str);
```

בשורה הראשונה שוב הגדרנו משתנה מחרוזת בשם $str ואתחלנו אותו לאותה מחרוזת כמו בדוגמא הקודמת. בשורה השנייה הגדרנו מערך בשם $a ובו, כאיברים במערך, כל המחרוזות שאנו רוצים להחליף. בשורה השלישית הגדרנו מערך בשם $b ובו כל המחרוזות שאותם נציב במקום אלו מהמערך הראשון בהתאמה. כשאני אומר בהתאמה אני מתכוון שהראשון יחליף את הראשון, השני יחליף את השני וכו'.

בשורה הרביעית קראנו ל str_replace עם שלושה פרמטרים. הפרמטר הראשון הוא המערך $a, המערך המכיל את המחרוזות שאנו רוצים להחליף. הפרמטר השני הוא המערך $b, המערך המכיל את המחרוזות אותן נציב במקום אלו שהחלפנו. הפרמטר השלישי הוא המחרוזת עליה נעבוד, $str בדוגמא זו. הפונקציה str_replace תחזיר מחרוזת מעודכנת בה כל המחרוזות הוחלפו ואותה נציג בעזרת echo לדפדפן. התוצאה היא בדיוק כמו קודם.

I love flashOO and flash but flashOO more!

דוגמא מעשית – החלפת סמיילים

בעורכי ת'רדים רבים בפורומים מבוססי PHP ובכלל, ישנה אופציה להציב סמיילים על ידי לחיצה על כפתור, הכפתור מציב מחרוזת "שמורה" בהודעה. אפליקצית הפורומים דואגת להעביר את ההודעה שהמשתמש כתב דרך פונקציה שבדרך כלל נקראת parse שדואגת לנתח את ההודעה ולהחליף תתי מחרוזת "שמורים" בתתי מחרוזת אחרים.

בדוגמא הבאה אנו נביט בקוד מעשי שמבצע את הפעולה וננסה לנתח אותו ולהבין כיצד הוא פועל. הרעיון הוא לקחת מחרוזת כפרמטר על פי ייחוס כך שנוכל לשנות את ערכה, להחליף את

כל המחרוזות ה"שמורות" לסמיילים בתמונות סמיילי, לשנות קצת את הפונט ולקבל משהו שאפשר להציג על המסך.

הקוד אינו מורכב כלל, והוא חוזר קצת על פונקציות, העברה על פי ייחוס ו str_replace. הבט:

```
$smiliesString = array("[tongue]","[huh]","[blink]");
$smiliesImages = array("<img src='tongue.gif'>",
                       "<img src='huh.gif'>",
                       "<img src='blink.gif'>");

function parse(&$message)
{
     global $smiliesString;
     global $smiliesImages;

     $message = "<font face='verdana'
                 size='2px'>".$message;

     $message = str_replace($smiliesString,
                            $smiliesImages, $message);

     $message .= "</font>";
}

$message  = "He stated talking about flash so I did
             [huh]<br>";
$message .= "and then he [blink] me so I [tongue] him
             back<p>";

echo $message;
parse($message);
echo $message;
```

בזוג השורות הראשונות הגדרנו שני מערכים, הראשון מכיל את המחרוזות השמורות "[huh]", "[tongue]" ו- "[blink]", באפליקציה אמיתית ישנם מספר רב יותר של סמיילים אבל הרעיון נשאר אותו רעיון. המערך השני מכיל את כל תגיות ה HTML הנועדות להחליף את המחרוזות השמורות ולהציג במקומם קבצי gif מתאימים.

לאחר הגדרת המחרוזות אנו רואים פונקציה בשם parse שמקבלת כפרמטר, על פי ייחוס, מחרוזת בשם $message – זוהי ההודעה שהמשתמש כתב ואליה אנו צריכים להכניס את

כל הזכויות שמורות לגיל כהן, http://www.flashoo.co.il

הסמיילים כתמונות. הפונקציה מבקשת גישה אל זוג המערכים בעזרת המילה השמורה global. הסיבה שלא הגדרנו את המערכים בתוך גוף הפונקציה היא שאלו ישמשו פונקציות נוספות כמו פונקציה עריכה ופונקציה שמציגה את הכפתורים השונים בהוספת תגובה או ת'רד חדש, לכן במקום ליצור מספר עותקים של אותו מידע אנו יוצרים אחד ומבקשים גישה אליו מכל פונקציה שנעזרת בו.

הפונקציה מוסיפה לתחילת המחרוזת הגדרת פונט בשורה השלישית לגוף הפונקציה, ובשורה החמישית סוגרת את תג הפונט. בין השורות הללו אנו קוראים לכוכב הערב str_replace ומבקשים להחליף במחרוזת $message כל מחרוזת מהמערך $smiliesString במחרוזת המתאימה לו מהמערך $smiliesImages, זוהי בעצם הפעולה העיקרית בקוד.
בשורות שלאחר הגדרת הפונקציה אנו מגדירים משתנה בשם $str ומאתחלים אותו להודעה שהמשתמש יכול היה לכתוב, אנו מציגים את ההודעה הזו, קוראים ל parse ומעבירים אליה כפרמטר את $str ואז שוב מציגים את ההודעה, הפעם ההודעה המעודכנת. הפלט נראה כך:

He started talking about flash so I did [huh]
and then he [blink] me so I [tongue] him back

He started talking about flash so I did
and then he me so I him back

המופעים של "[huh]" וחבריו הוחלפו בתמונות סמיילי מקסימות.

סיכום

ישנן עשרות רבות של פונקציות לטיפול במחרוזות, אנו דברנו על חמש עיקריות. עם העבודה שלך ב PHP אתה תצטרך לבצע פעולה מסוימת שכנראה מספר מפתחים היו צריכים לבצע, במקרה הזה כנראה מאוד שיש פונקציה שתבצע עבורך את העבודה, לכן אל תתבייש, קפוץ ל http://www.php.net וחפש אחר שם הפונקציה. מפתחי PHP מהווים קהילה מדהימה, חולקת מידע, ותומכת בקוד פתוח, אל תתבייש לשאול שאלות בפורומים אם אינך מוצא את מבוקשך באתר php.net.

פרק זה חותם את החלק הראשון בספר. החלק הבא מדבר על SQL ועל MySQL, אנו כמובן נחזור ונכתוב עוד לא מעט קוד ב PHP בכדי להתחבר למסד נתונים, אבל נכון לפרקים הבאים נשנה את הפוקוס ונלמד על התחביר של שפת SQL – החדשות הטובות שמדובר רק בפקודה אחת, החדשות העוד יותר טובות הן שעל הפקודה הזו נכתבו עשרות ספרים.

כל הזכויות שמורות לגיל כהן, http://www.flashoo.co.il