פרק 11 – לולאת while

הקדמה

לולאת for, אותה פגשנו בפרק הקודם, מתאימה בעיקר למקרים בהם אנו יודעים מראש כמה פעמים יש לבצע משימה. לולאת while לעומת זאת נועדה בעיקר למקרים בהם אנו רוצים לבצע משימה כל עוד תנאי מסוים מתקיים מבלי לדעת מראש כמה פעמים האיטרציה תחול. הדגש הוא לא על מספר הפעמים אלא על התנאי. באופן מעשי אפשר להחליף כל לולאת for בלולאת while וכל לולאת while בלולאת for ואנחנו נראה איך עושים את זה, רק בשביל התרגול, בהמשך הפרק.

תחביר לולאת while

לולאת while ב PHP דומה מבחינת תחביר וקונספט ללולאת while ב ActionScript. הצץ בקוד הבא:

```
$sum = 0;
$a = 1;

while($a<=100)
{
     $sum += $a;
     $a++;
}

echo "1+2+...+100=$sum";
```

המטרה של הקוד היא לחבר את כל המספרים השלמים מ-1 עד 100 ולהציג את התוצאה. בשתי השורות הראשונות הגדרנו שני משתנים - $sum שיאגור את הסכום, הוא אותחל ל-0 כי הסכום לפני שסיכמנו משהו הוא 0. המשתנה $a הוא המספר הנוכחי אותו יש לחבר, מכיוון שהחלטנו שאנחנו מחברים את כל המספרים בין 1 ל-100 טבעי להתחיל עם 1 (למרות שאפשר גם להתחיל עם 100 ולרדת עד ל-1).

בשורה השלישית אנו נתקלים בלולאת ה while הראשונה שלנו ב-PHP, כמו ב ActionScript הלולאה מזכירה מאוד את התחביר של משפט התניה, אנו כותבים את המילה השמורה while ולאחריה בסוגריים את התנאי שכל עוד הוא יתקיים גוף הלולאה הנמצא בבלוק אחרי התנאי ימשיך לרוץ. בכל פעם שבלוק הקוד יסתיים התנאי יוערך מחדש, כלומר PHP שוב תבדוק אם $a קטן או שווה מ 100.

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

בגוף הלולאה אנו מוסיפים ל $sum את המספר הנוכחי $a ושורה לאחר מכן מקדמים את $a כך שבפעם הבאה נוסיף ל $sum את המספר העוקב.

הלולאה תסתיים כאשר $a יהיה גדול או שווה מ 100. אחרי גוף הלולאה נעזרתי ב echo בכדי להציג לדפדפן את התוצאה.

יותר טבעי לבצע את המשימה בעזרת לולאת for. כעת נביט בדוגמא שלולאת while פשוט טבעית יותר וצצה אינספור פעמים בתסריטים:

```
$done = false;

while (!$done)
{
     $done = getDataFromDataBase();
}
```

בדוגמא הזו אנו נמשיך לקרוא לפונקציה דמיונית בשם getDataFromDataBase עד שהיא תחזיר true, למה? כי אם הפונקציה תחזיר true אז בהערכה הבאה של $done אם $done יהיה true אז $done! יהיה false והתנאי יהיה כוזב. במקרה כזה אין לנו מושג מתי הלולאה תסתיים בעיקר כי זה תלוי בפונקציה אחרת, מושלם ללולאת while.

תנאים מרוכבים בלולאת while

בדיוק כמו במשפט if גם התנאי במשפט while יכול להיות מורכב בעזרת הקשרים "וגם" ו"או". לדוגמא, אם נשפץ את הדוגמא הקודמת:

```
$done = false;
$userAborted = false;

while ((!$done) && (!$userAborted))
{
     $done = getDataFromDataBase();
}
```

בדוגמא הזו שיננו את תנאי הלולאה כך שהלולאה תמשיך כל עוד המשימה לא הסתיימה וגם המשתמש לא ביטל. לכאורה נדמה כי היינו צריכים להשתמש ב"או" במקום ב"וגם" אבל אם תחשוב על זה קצת תראה ש"וגם" הוא הקשר המתאים.

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

לולאת do...while – נעשה ונשמע

ישנה וריאציה נוספת ללולאת while והיא לולאת do while. לולאת do while דומה ללולאת while פרט להבדל משמעותי בקונספט – לולאת while עשויה לא לבצע את בלוק הקוד שלה אף לא פעם אחת וזאת במקרה שהתנאי כוזב, לולאת do while קודם כל מבצעת את בלוק ורק אחר כך בודקת את התנאי, כמו שאמרו חכמינו – נעשה ונשמע. הבט בקוד הבא:

```
$i = 40;

do
{
     echo $i;
     $i++;
} while ($i<30);
```

בקוד יצרנו משתנה חדש בשם $i ואתחלנו אותו ל-40. מיד לאחר מכן אנו רואים לולאת do while. הלולאה מתחילה במילה השמורה do, לאחריה בלוק הקוד אותו יש לבצע, ולבסוף את התנאי. בין אם התנאי אמת או כוזב גוף הלולאה יתבצע בפעם הראשונה. לכן למרות ש $i הוא 40 וכמובן איננו קטן מ-30 בלוק הקוד של לולאת ה do while כן יתבצע והדפדפן יציג את המספר 40 וערכו של $i יגדל ב-1.

אם היינו מנסים את זה בלולאת while הדפדפן לא היה מציג דבר, הקוד נראה כך:

```
$i = 40;

while($i<30)
{
     echo $i;
     $i++;
}
```

קינון לולאות ודוגמא – חיבור מספרים ראשוניים

ניתן לקנן כל לולאה בתוך כל לולאה ורואים את זה לא מעט פעמים בקוד: באלגוריתמים בסיסיים כמו מיון רשימה, בהוצאה של טבלה ממסד נתונים (למרות שכרגע אתה לא אמור לדעת מה זה), ובאופן כללי בכל מה שקשור למערכים דו ממדיים.

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

לפני שנעבור לדוגמא הבאה אני אזכיר בקצרה מה זה מספר ראשוני. מספר ראשוני הוא מספר שמתחלק אך ורק בעצמו ובאחד. 5 לדוגמה הוא מספר ראשוני כי הוא מתחלק אך ורק ב-1 ובעצמו. 2 אף הוא מספר ראשוני, הוא המספר הראשוני הקטן ביותר והוא המספר הראשוני הזוגי היחיד, זאת מפני שכל מספר זוגי אחר מתחלק גם ב-2, בניגוד להגדרת הראשוניות.

אחד איננו מספר ראשוני למרות שהוא מתחלק בעצמו ובאחד. זה עניין של הגדרה, אבל הוא מסתדר יפה עם הציפיות שלנו במתמטיקה, נושא שלא נכנס בו יותר מדי לעומק במאמר.

הנה המספרים הראשוניים הראשונים:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, …

הסתכל על הקוד הבא, המחבר את כל המספרים הראשוניים שנמצאים בין 1 ל-$n.

```
$sum = 0;
$i = 2;
$n = 100;

while ($i<$n)
{
     $prime = true;

     for($j=2; $j<$i; $j++)
     {
          if ($i%$j == 0) $prime = false;
     }

     if ($prime) $sum += $i;

     $i++;
}

echo $sum;
```

כנראה קטע קוד מורכב יותר מכל קטע קוד שראינו עד כה אבל אין מה להיבהל. הרעיון הוא לעבור על כל המספרים מ-1 ועוד n, לבדוק עבור כל מספר אם הוא ראשוני, אם כן להוסיף אותו לסכום המצטבר.

בשורה הראשונה אתחלנו את הסכום המצטבר ל-0 שהרי עדיין לא הספקנו לסכם שום דבר. בשורה השנייה אתחלנו את $i ל-2, זה המספר הראשון שנבדוק. אנחנו לא באמת מתחילים מ-1 כי 1 אינו ראשוני, 2 הוא המספר הראשוני הראשון ולכן זה מקום טוב להתחיל בו. בשורה

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

השלישית הגדרנו את $n שקובע עד איפה אנחנו מסכמים. בדוגמא קבעתי את $n ל-100, כלומר לסכם את כל המספרים הראשוניים בין 1 ל-100.

הלולאה הראשונה היא לולאת while שעדיף היה להחליפה בלולאת for, אבל אנחנו בפרק הלא נכון. הלולאה הזו אחראית לעבור מספר אחר מספר. הפקודה הראשונה בתוך הלולאה היא לקבוע את ערכו של המשתנה הבוליאני $prime ל true, הסיבה היא ש $prime יקבע האם המספר הנדבק הוא ראשוני או לא, לפני שנבדוק אנו מניחים כי המספר אכן ראשוני.

מיד לאחר מכן מתבצעת הבדיקה, וזאת בעזרת לולאת for שמקוננת בתוך לולאת ה while. לולאת ה for בודקת אם קיים איזשהו מספר בין 2 ל $i (לא כולל $i) שהמספר הנוכחי, $i, מתחלק בו, אם כן אז $i, על פי הגדרת הראשוניות אינו ראשוני והתסריט מכוון את $prime ל false. ביציאה מלולאת ה for אנו נתקלים במשפט התניה שבודק את $prime ערכו אמת, אם כן אז ערכו של $i מתווסף ל $sum.

ביציאה מלולאת ה while מחכה לנו פקודת echo שמציגה לדפדפן את הסכום המבוקש.

כתרגיל אני מציע לך לבנות תוכנית הדומה לזו פרט לכך שאין היא מסכמת את כל המספרים הראשוניים מ-1 עד $n אלא מסכמת את $n המספרים הראשוניים.

המרת לולאת for ב while ולהיפך

כפי שאמרתי בתחילת המאמר, ניתן להמיר כל לולאת while בלולאת for ולהיפך. נסתכל על התחביר הכללי של לולאת while:

```
while(/*condition*/)
{
     /*code*/
}
```

במקרה זה התנאי הוא /*condition*/ שיכול להיות כל תנאי וכל הקוד מיוצג על ידי /*code*/. יכולנו להשתמש ב for ולקבל בדיוק את אותן התוצאות באופן הבא:

```
for(;/*condition*/;)
{
     /*code*/
}
```

כל שעשינו בכדי להמיר את לולאת ה while ללולאת for היה לא לציין אתחול או פעולת קידום אינדקס, קיבלנו לולאת while לכל צורך מעשי.

איך נמיר לולאת for ללולאת while, ובכן הדבר דורש קצת יותר עבודה מפני שבלולאת for יש יותר מידע מאשר בלולאת while. הבא במקרה הכללי של לולאת for:

```
for (/*init*/ ; /*condition*/ ; /*increment*/)
{
     /*code*/
}
```

הפעם כל שלב האתחול מיוצג על ידי /*init*/, התנאי שוב על ידי /*condition*/ ושלב קידום האינדקס מיוצג על ידי /*increment*/. לולאת while שקולה תראה כך:

```
/*init*/
while (/*condition*/)
{
     /*code*/
     /*increment*/
}
```

בעצם ביצענו את שלב האתחול לפני הכניסה ללולאה, בדיוק כמו ש for עושה, לאחר מכן יצרנו לולאת while בעל אותו התנאי של לולאת ה for. בגוף לולאת ה while מתבצע אותו הקוד של לולאת ה for אך בתוספת הקוד /*increment*/ וזאת מפני שלולאת for מבצעת את שלב קידום האינדקס מיד אחרי שסיימה את בלוק הקוד.

נביט בדוגמא – הבט בקוד הבא:

```
for($i=0; $i<10; $i++)
{
     echo "$i<br>";
}
```

זוהי לולאת for פשוטה שמציגה את כל המספרים בין 0 ל-10 (לא כולל 10) לדפדפן שורה אחר שורה. ההמרה ללולאת while יראה כך:

```
$i = 0;
while($i<10)
{
     echo "$i<br>";
     $i++;
}
```

כמו בהסבר הכללי – שלב האתחול $i=0 מתבצע לפני הלולאה, התנאי של לולאת ה for מומר לתנאי עבור לולאת ה while, כל הקוד של לולאת ה for נכנס ללולאת ה while אך בתוספת של שלב עדכון האינדקס $i++.

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

למעשה ההמרה כל כך פשוטה שאפשר לבנות תוכנית PHP פשוטה שעושה זאת.

אני רוצה להזכיר שההמרות הללו בוצעו רק בכדי להציץ קצת יותר אל תוך הקרביים של מבנה הלולאות, אין כל צורך מעשי בהמרה של לולאה אחת לאחרת פרט לצורך פדגוגי. אני אסתייג – אני לא רואה צורך כזה.

כתרגול אני מציע לך לנסות ולהמיר לולאת do while ללולאת for.

סיכום

הפרק הבא יהיה הפרק האחרון הדן בנושא לולאות, בו נדון במילות המפתח break, exit ו continue. אנו כן נחזור לדון בלולאה מעניינת ומאוד שימושית בשם foreach בדיון על מערכים. בכלל – מערכים ולולאות הולכים ביחד כמו נישואים בכפר סביון והסכמי ממון.

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