פרק 7 - משתנים

אם ניצור לעצמנו שמש אסוציאציות סביב המילה "תכנות", אחת הקרניים הראשונות תהיה "משתנים". משתנים (Variables) מאפשרים לשמור פיסת מידע לאורך התוכנית. אני לא יודע להצביע על שפת תכנות עילית שאינה מאפשרת הגדרה ושימוש במשתנים.

תחביר משתנים ב - PHP

עבודה עם משתנים ב PHP דומה מאוד לעבודה עם משתנים ב ActionScript פרט לעניין תחבירי שולי – ב PHP כל שם משתנה מתחיל בסימן הדולר ($). לוקח קצת זמן להתרגל לעניין אחרי עבודה עם ActionScript, אבל אחרי שמתרגלים, נוח מאוד לזהות את המשתנים בקוד אפילו בסריקה מהירה מעל דפי הסקריפט. אני אישית מוצא את זה מבדר שבתוכנה חינמית כמו PHP בחרו להשתמש דווקא בסימן הדולר כאחד מהסימנים השמורים של השפה.

שם משתנה יכול להיות מורכב מאותיות, מספרים וקו תחתון. התו הראשון (לאחר סימן הדולר) אינו יכול להיות מספר. בניגוד ל ActionScript ובדומה ל ActionScript 2, המשתנים ב PHP רגישים לגודל האות (Case sensitive) מה שאומר שמשתנה בשם $name שונה מהמשתנה $Name.

סוגי משתנים

PHP כמו ActionScript תומכת בארבעת סוגי המשתנים הבסיסיים – מספר שלם, מספר בעל נקודה צפה, מחרוזת ומשתנה בוליאני. לטוב ולרע, אין זו חובה להגדיר את סוג המשתנה, PHP מסיקה לבד את סוגו על פי ערכו וסוג זה יכול להשתנות במהלך הסקריפט בהתאם לערך שיוזן לו.

משתנים מסוג מחרוזת

עד שלא נשחק קצת עם משתנים לא נבין איך הם פועלים. בקוד הבא הגדרתי משתנה בשם $name והזנתי לו ערך, השם שלי:

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

```
<?php
$name = "Gil Cohen";
echo "My name is $name";
?>
```

בשורה הראשונה הכרזתי על פתיחת בלוק PHP. בשורה השנייה, היותר מעניינת, הגדרתי משתנה חדש בשם $name והזנתי לו את הערך "Gil Cohen". בזכות הערך שהוזן PHP תבין ש $name הוא משתנה מסוג מחרוזת. בשורה השלישית הצגתי לדפדפן את תכולת המשתנה $name עם קידומת קצרה. שים לב כי בכדי לשלוח את תוכן המשתנה ללקוח (במקרה הזה להציג את תוכן המשתנה לדפדפן) אין צורך בשום פעולה מיוחדת, כל שיש לעשות הוא לרשום את שם המשתנה.

אם ברצונך להציג רק את תוכן המשתנה, כלומר אתה לא מעוניין להציג את הקידומת, לא תצטרך להשתמש במירכאות. כלומר הקוד הבא יפעל יופי:

```
<?php
$name = "Gil Cohen";
echo $name;
?>
```

ניתן להציג שרשור מחרוזות ב PHP בעזרת אופרטור השרשור המיוצג על ידי סימן הנקודה (.). לדוגמא:

```
<?php
$name = "Gil Cohen";
$age = 15;
echo "$name, Come back in ".(18-$age)." Years!";
?>
```

כפי שניתן לראות מהדוגמא, ניתן לשרשר גם מספרים למחרוזת ו PHP תציג אותם בצורה נכונה.

משתנים מספריים

נעבור להגדרה של משתנה מספרי שלם. הפעם נעשה משהו קצת יותר מתוחכם. הבט בקוד הבא:

```
$english = 88;
$math = 100;

$average = ($english+$math)/2;
echo "You're average is $average";
```

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

כפי שתוכל לראות בשורה השנייה והשלישית הכרזתי על שני משתנים - $english ו $math המייצגים ציונים בין 0 ל-100. בשורה שהגדרתי את המשתנים דאגתי להזין להם ערך, לפעולה הזו בה מגדירים משתנה ומזינים לו ערך בפעם הראשונה קוראים אתחול. נהוג לומר - אתחלתי את המשתנה $math ל-100.

בשורה הרביעית הגדרתי משתנה חדש ואתחלתי אותו לממוצע החשבוני של ערכי המשתנים $english ו $math. ברגע ש PHP רואה סימן השמה (=) היא מחשבת את הערך הנמצא בצידו הימני של סימן ההשמה ורק לאחר שהיא מגיעה לביטוי שאינו דורש עוד חישוב היא מציבה את ערכו במשתנה הנמצא מצידו השמאלי של סימן ההשמה. לכן הפעולה הראשונה ש PHP עושה היא לחבר את ערך המשתנה $english לערך המשתנה $math, את התוצאה היא מחלקת ב-2 ואת התוצאה כולה, המהווה בעצם את הממוצע של ערכי המשתנים הללו, היא מזינה אל $average. בשורה החמישית ערך המשתנה $average מועבר ללקוח.

הממוצע של 88 ו-100 הוא 94, אבל מה יקרה אם אני אשנה את הציון שלי באנגלית מ-88 ל-89? במקרה הזה הממוצע יעמוד על 94.5. המספר הזה אינו מספר שלם ולכן $average יהיה משתנה מטיפוס נקודה צפה. תחשוב על זה – עד ש PHP לא מחשבת איזה ערך יוזן ל $average לה או לך אין מושג איזה סוג משתנה $average יהיה.

משתנים בוליאניים

משתנה בוליאני הוא משתנה שערכו יכול להיות אחד משני ערכים – true או false. אם אינך יודע דבר בנושא אני מציע שתפנה למאמר שכתבתי - אל מעמקי הלוגיקה הבוליאנית הנמצא בכתובת הבאה:

- http://www.flashoo.co.il/articles/pdfs/008.pdf

הגדרת משתנה בוליאני זהה להגדרת כל משתנה אחר ועד שאתה לא מזין למשתנה ערך true או false PHP עדיין לא יודעת שמדובר במשתנה בוליאני. כמו כן האופרטורים המופעלים על משתנים בוליאניים זהים לאופרטורים בפלאש: ! לשלילה, & מייצג "גם" ו | מייצג "או". נסתכל על הדוגמא הבאה:

```
<?php
$A = true;
$B = false;
$C = !($A);
$D = ($A)&($B);
?>
```

בשורה השנייה הגדרנו משתנה בשם $A והזנו לו את הערך true. זה מספיק בשביל ש PHP תבין שמדובר במשתנה בוליאני. בשורה השלישית הגדרנו משתנה בוליאני נוסף בשם $B ואתחלנו אותו ל false. בשורה הרביעית הגדרנו משתנה בוליאני נוסף - $C ואתחלנו אותו להופכי של $A, מכיוון שערכו של $A בעת ביצוע השורה הזו הוא true ערכו של $C יהיה false.

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

בשורה החמישית והאחרונה הגדרנו משתנה נוסף בשם $D ואותו אתחלנו לתוצאת הביטוי הבוליאני A&B. מכיוון שערכו של $A בעת ביצוע השורה הוא true וערכו של $B הוא false ערכו של $D יהיה false.

אם נחזיר את ערכו של משתנה בוליאני ללקוח הלקוח לא יקבל מחרוזת "true" או "false" אלא יקבל את המחרוזת "1" עבור true ו-"0" עבור false. זו מעלה של PHP כי בדרך כלל החזרה של משתנה בוליאני לפלאש מיועדת למשפטי התניה והמחרוזת "true" תצריך אותנו לבצע יותר עבודה מאשר המחרוזת "1".

משתנים – מה משתנה?

לכל משתנה יש 3 מאפיינים – שם, ערך וטיפוס. ב PHP שם המשתנה לא יכול להשתנות אחרת איך היינו יכולים לזהות אותו. ערך המשתנה בודאי יכול להשתנות, כלומר ניתן להזין ערך בשורה כלשהי בקוד וערך אחר לאותו משתנה בשורה אחרת, אבל אתה כבר יודע את זה מ ActionScript. דבר נוסף שיכול להשתנות אצל משתנים ב PHP זה הטיפוס שלהם.

תאר לעצמך את התסריט הבא – השנה 2,160. מדענים מצאו תרופות לכל המחלות הקיימות, כל הדתות וכל האומות התאחדו לאומה אחת גדולה כך שאין מלחמות, ובגלל שאין מלחמות אין הרבה דאגות ותוחלת החיים של אדם ממוצע כל כך ארוכה ש"עד 120" זו קללה. הבעיה בתסריט הכל כך נחשק הזה היא שישנם כל כך הרבה אנשים על כדור הארץ שאין ברירה אלא להתחיל לקרוא לאנשים במקום בשמות (מחרוזות) במספרים. עוד דמיין לעצמך ש PHP עדיין קיימת ויש צורך לעדכן את כל התוכנות כך שבמקום טיפוס מחרוזת כל השמות יהיו מטיפוס מספר שלם. ובכן ב PHP אין בעיה, הבט בקוד הבא:

```
<?php
// Last updated 01/08/2004
$name = "Gil Cohen";
echo $name;
echo "<br>";
// Last updated: 22/02/2160
$name = 9311345;
echo $name;
?>
```

כפי שניתן לראות בשורה השלישית הוגדר משתנה בשם $name שערכו הוא "Gil Cohen" מה שהופך את $name למשתנה מטיפוס מחרוזת. ערך המשתנה מועבר לדפדפן בשורה 4, ובשורה 5 דאגנו לכך שפקודת ה echo הבאה תעביר לדפדפן מידע שיוצג בשורה הבאה (בעצם מה שעשינו היה להעביר תגית HTML שהדפדפן יפרש אותה כמו שרק הוא יודע). בשורה השלישית לפני אחרונה הצבנו ערך מספרי במשתנה $name מה שאומר ל PHP – לא רק הערך של $name השתנה אלא גם הטיפוס שלו, מעכשיו $name הוא משתנה מסוג מספר שלם. בשורה האחת לפני אחרונה דאגנו להציג את השם החדש.

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

בדוגמא האחרונה ראינו בעצם ש PHP מאפשרת למשתנים לשנות לא רק את ערכם אלא גם את הטיפוס שלהם. אני לא מציע לשחק עם זה יותר מדי, דאג לדעת מה אמור להיות ערך של משתנה ונסה שלא לקפוץ מטיפוס אחד לאחר, זה שאפשר זה לא תמיד אומר שכדאי.

משתנה כשם משתנה

תכונה של משתנים שקיימת ב PHP ולא קיימת ב ActionScript היא האפשרות להשתמש במשתנה כשם של משתנה אחר. לא יצא לי אף פעם ממש להיעזר בתכונה הזו אבל אני אציג אותה בכל זאת מפני שהיא באמת מגניבה. הבט בקוד הבא:

```
<?php
$gil = "03/12/1982";
$orit = "22/02/1982";
$albert = "15/03/1879";

$name = "albert";

echo $$name;
?>
```

בשלושת השורות הראשונות (שאחרי פתיחת קוד ה PHP) הגדרתי שלושה משתנים בשם $gil, $orit ו- $albert ואתחלתי אותם כתאריך הלידה של כל אחד מהם. בשורה החמישית הגדרתי משתנה בשם $name ואתחלתי אותו למחרוזת "albert".

בשורה השישית קורה דבר מעניין – העברתי לדפדפן את $$name. אבל מה פירוש התחביר הזה? ובכן, כש PHP רואה את סימן ה $ היא מבינה שהיא עומדת מול משתנה והיא מחליפה את שם המשתנה כולל סימן הדולר בערכו. במקרה הזה החלק המודגש ב **$$name** מומר למחרוזת המהווה את ערכו של המשתנה, כלומר PHP מסתכלת על $$name ורואה $albert. לאחר מכן היא חוזרת שוב על הפעולה והפעם ממירה את $albert בערך שלו שזה בעצם תאריך הלידה של אלברט (איינשטיין).

שוב, PHP מאוד דינאמית בכל הנוגע לשמות המשתנים שלה ואנחנו יכולים להשתמש במשתנה בכדי לייצג שם של משתנה אחר. למה זה טוב? כי ניתן לשנות את ערכו של המשתנה המכיל את שם המשתנה. הבט בקוד הבא ונסה לחשוב מה הדפדפן יציג אם נריץ את הקוד?

```
<?php
$a = "b";
$b = "c";
$c = "a";

echo $$$a;
?>
```

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

העברת ערכי משתנים לפלאש

תרגיל טוב יהיה להעביר ערך של משתנים שקיימים בקוד PHP לפלאש. אנו נבנה תוכנית קטנה ב PHP שמחזירה ללקוח את המיקום והרדיוס של עיגול. פלאש, המתפקדת כלקוח, תבקש את הנתונים הללו, ותציג עיגול מתאים. הקוד לא מורכב. ראשית נביט בקוד ה PHP:

```
<?php
$x = 140;
$y = 200;
$r = 80;

echo "x=$x&y=$y&r=$r";
?>
```

בשלושת השורות הראשונות בבלוק הקוד ניתן לראות הגדרה ואתחול של שלושה משתנים: $x, $y ו-$r. כפי שאולי ניחשת, $x ו-$y מייצגים את המיקום האופקי והאנכי בהתאם של מרכז העיגול ו $r מייצג את רדיוס העיגול.

השורה הרביעית מעבירה בעזרת הפקודה echo מחרוזת ללקוח, הלקוח במקרה שלנו הוא לא הדפדפן אלא פלאש ולכן אנו מדברים בשפה שפלאש מבינה, כלומר אנו מעבירים לפלאש ראשית את שם המשתנה ולאחריו את ערכו כאשר בין כל שני משתנים הצבתי את הסימן & שמורה לפלאש שעד כאן לערך של המשתנה הנוכחי, עוברים למשתנה הבא.

עבור הערכים הנוכחיים פלאש תקבל את המחרוזת:

x=140&y=200&r=80

בהנחה שבבמה בפלאש יושב מוביקליפ בשם circle כך קוד ה ActionScript שיושב בפריים הראשון נראה:

```
circle._visible = false;
circleData = new LoadVars();

circleData.onLoad = function()
{
     _root.circle._x = circleData.x;
     _root.circle._y = circleData.y;
     _root.circle._width = circleData.r;
     _root.circle._height = circleData.r;
     _root.circle._visible = true;
}

circleData.load("http://localhost/ch7/circle.php");
```

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

בשורה הראשונה דאגתי להעלים את מופע העיגול שהרי אני לא רוצה שהוא יוצג עד שלא נקבל את המידע לגבי היכן אנחנו אמורים למקמם אותו ומה הגודל של העיגול.

בשורת הקוד השנייה הגדרתי אובייקט חדש מהמחלקה LoadVars עליה הרחבנו בפרקים הקודמים. לאובייקט החדש קראתי circleData, שם המעיד על תפקיד האובייקט.

השורה השלישית היא פונקציה לטיפול באירוע onLoad של אובייקט ה LoadVars שהגדרנו קודם לכן, בפונקציה זו להזכירך אנחנו נשים את כל הקוד שאמור להתבצע ברגע שהנתונים יגיעו לפלאש. במקרה שלנו עדכנתי את המיקום והגודל של העיגול על פי הנתונים שהגיעו אלינו מתסריט ה PHP. עוד נזכיר כי הנתונים שהגיעו מתסריט ה PHP זמינים דרך מאפיינים חדשים שנוצרו לאובייקט circleData בעקבות בקשת הנתונים, במקרה שלנו נוספו ל circleData המאפיינים x, y ו-r. בסוף הפונקציה לטיפול באירוע onLoad שיניתי את ערך המאפיין visible_ ל true בכדי להציג את העיגול.

השורה האחרונה בקוד היא בעצם השורה שמבקשת מ PHP נתונים. במקרה הנוכחי תסריט השרת נמצא בקובץ בשם circle.php שנמצא בספרייה ch7 (chapter 7).

שים לב שלא יצא לנו תועלת משימוש במשתנים בדוגמא. יכולנו לעשות בדיוק את אותו הדבר בלי משתנים, אפילו בלי PHP, אבל כשנתחיל לדבר על מסדי נתונים או כשנגיע לקוד יותר מתוחכם נראה שבלי משתנים קשה מאוד עד בלתי אפשרי ליצור אפליקציה שימושית.

סיכום

במאמר זה לא התיימרתי ללמד אותך אודות משתנים אלא לקחת את הידע שלך במשתנים מ ActionScript ולהיעזר בו בכדי ללמד אותך אודות משתנים ב PHP. בפרק הבא נדון במשפטי התניה.

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