

פרק 8 – משפטי התניה

הקדמה

אם משתנים זה הזיכרון של התוכנה שלנו אז משפטי התניה זה המוח. משפטי התניה זה הכלי המאפשר לנו כמפתחים לאפשר קבלת החלטות בתוכנה. בעזרת משפטי התניה אפשר לבצע קטע קוד מסוים רק אם תנאי מסוים מתקיים, לדוגמא לאפשר למשתמש להיכנס לחלק מסוים באתר רק אם יש בידו את הסיסמא. אפילו פעולה פשוטה יותר כמו חיפוש שם בתוך רשימת שמות היא פעולה שאינה אפשרית ללא היכולת לקבל החלטות.

אבל אתה כבר יודע את כל זה, זכור כי ההנחה היא כי אתה יודע ActionScript ואני לא אלאה אותך בפרטים, עם זאת הנושא חשוב ולפעמים אחטא ואעמיק.

תחביר משפט התניה בסיסי ב PHP

החדשות הטובות הן (ואין חדשות רעות) שמשפטי התניה ב PHP מבחינת תחביר וקונספט זהים כמעט לגמרי למשפטי התניה ב ActionScript. כמו ב ActionScript גם כאן אנו מתחילים את ההתניה במילה השמורה if שלאחריה בסוגריים נמצא התנאי אשר יכריע האם לבצע קוד קטע מסוים או לא. הנה דוגמא לשימוש במשפט התניה ב PHP:

```
<?php  
  
$age = 17;  
if ($age < 18) echo "Kid, stay out!";  
  
?>
```

בשורה הראשונה הגדרתי משתנה מסוג מספר שלם בשם \$age ואתחלתי אותו ל-17. השורה השנייה מציגה בפנינו משפט התניה אשר התנאי הנבדק הנמצא בין הסוגריים בודק אם ערך המשתנה \$age קטן מ-18. במקרה הזה התנאי הוא תנאי אמת ולכן הפקודה שיושבת מיד אחרי התנאי תבצע ו PHP תעביר ללקוח (נניח לדפדפן) מחרוזת המרחיקה ילדים.

בדיוק כמו ב ActionScript, אם נרצה לבצע יותר מפקודה אחת בתנאי שהתנאי מתקיים נצטרך להכניס את כל הפקודות לתוך בלוק קוד כפי שמודגם בקטע הקוד הבא:

```
<?php
if ($min > $max)
{
    $temp = $min;
    $min = $max;
    $max = $temp;
}
?>
```

בדוגמא הזו נניח וקיימים שני משתנים בשמות \$min ו-\$max שאת ערכם אנחנו לא יודעים, תוכל להניח שקיבלנו אותם ממסד נתונים, מפלאש או מטופס שהמשתמש הזין. בכל מקרה אנחנו רוצים לוודא שערך המשתנה \$max אכן גדול מערך המשתנה \$min, בכדי לעשות זאת אנחנו ראשית בודקים אם \$min גדול מ \$max, אם זה המצב אז יש צורך לתקן אותו, זאת נעשה בעזרת משתנה עזר כמודגם בקוד. הרעיון הוא לשמור את ערך המשתנה \$min במשתנה זמני בשם \$temp, לאחר מכן להציב את ערך המשתנה \$max במשתנה \$min ולבסוף להציב את ערך המשתנה \$temp (שמכיל בעצם את המספר הגדול מבין השניים) במשתנה \$max. בכל מקרה אנחנו צריכים שלוש פעולות בכדי לבצע את המשימה ולכן הצבנו את הפעולות הללו בתוך בלוק קוד מיד אחרי התנאי.

ניתן להחליף את הערכים של שני משתנים ב-2 פעולות אם משתמשים בפעולה הבינארית xor. אבל הדבר חורג ממסגרת וממטרת המאמר.



השוואה של שני ערכים במשפט התניה נעשה בעזרת האופרטור == בדיוק כמו בפלאש, לדוגמא:

```
<?php

$password = "1234";

if ($password == "1234")
{
    "Welcome webmaster";
}

?>
```

כפי שניתן לראות בשורה השלישית בקוד – ביצענו השוואה בין ערך המשתנה \$password לסיסמא הנכונה בכדי לקבל החלטה האם הגולש הוא מנהל האתר.

אם היינו כותבים רק סימן שוויון אחד במקום שניים (= במקום ==) היינו נתקלים בבאג שנתקלתי בו עשרות פעמים – הערך "1234" היה מוזן ל \$password וכל הביטוי היה מוערך כפסוק אמת,

כתוצאה מכך המשתמש, גם אם אין בידו את הסיסמא הנכונה, היה יכול לגשת לאתר כאילו היה מנהל האתר – לא רק באג מרגיז אלא גם פרצה מגוחכת לגנב במקרים מסוימים.

מילת המפתח else

כמו ב ActionScript גם ב PHP ניתן לבצע בלוק קוד מסוים במקרה בו תנאי מתקיים וקטע קוד אחר במקרה בו התנאי איננו מתקיים. גם התחביר זהה לתחביר של ActionScript, בכל זאת נביט בדוגמא:

```
<?php
$age = 15;
$name = "Gil";

if ($age >= 18) {
    echo "Welcome $name";
}
else
{
    echo "$name, Come back in ".(18-$age)." years";
}
?>
```

בשתי השורות הראשונות הגדרנו שני משתנים - \$age שערכו במקרה הזה הוא 15 ו \$name שערכו במקרה הזה הוא "Gil". בשורה השלישית אנו נתקלים במשפט if שהתנאי אותו הוא מאמת הוא האם ערך המשתנה \$age גדול או שווה ל-18, במידה והתנאי מתקיים הוא מציג ברכת שלום דינאמית, דינאמית מהבחינה הזו שגם שם המשתמש מוצג.

במקרה והתנאי איננו מתקיים, בלוק הקוד שאחרי מילת המפתח else מורץ על ידי PHP. במקרה הזה PHP אומרת למשתמש לחזור עוד ככה וככה שנים. אם לפרט יותר – אם המשתמש הוא בן 15 כמו בדוגמא אז התסריט יודיע למשתמש לחזור בעוד 3 שנים.

מילת המפתח elseif

מילת המפתח elseif היא קצת פחות מוכרת למפתחים ב ActionScript מפני שהיא לא קיימת שם אבל היא מתפקדת בדיוק כמו שהיית מצפה ממנה. כעיקרון elseif מאפשרת לנו להריץ בלוק קוד במקרה בו תנאי מסוים לא מתקיים אבל תנאי אחר כן.

חשוב על הדוגמא הבאה – מערכת בחינת באוניברסיטה המחשבת את הציון הסופי של סטודנט. הציון הסופי נקבע על פי מבחן המהווה 70% מהציון הסופי ושיעורי בית המהווים מגן של 30%.

מה הכוונה במגן? הכוונה היא שאם ציון שיעורי הבית נמוך מציון המבחן אז הם לא נחשבים וציון המבחן הוא הציון הסופי. קוד שייתן מענה לבעיה הוא:

```
if ($test > $hw_grade)
{
    $grade = $test;
}
else
{
    $grade = 0.7*$test+0.3*$hw_grade;
}
```

שים לב כי החסרנו את הגדרת המשתנים \$test, \$hw_grade ו-\$grade מפני שאנו מתייחסים לקטע הקוד הזה כאל חלק מתוכנית שלפניה יש מספר פעולות (נניח הוצאת הנתונים מבסיס נתונים) ואחריה פעולות נוספות (כמו שליחת מייל להודיע לסטודנט ועדכון בסיס הנתונים בציון הסופי).



בשורה הראשונה בדקנו האם ציון המבחן \$test גדול מציון שיעורי הבית \$hw_grade, אם כן הציון הסופי נקבע רק על פי המבחן, אחרת הציון נקבע לפי 70% מבחן ו-30% מגן.

כעת נוסיף קושי נוסף לבעיה – ישנם 5 משימות שיעורי בית, אם הסטודנט מגיש 4 מהם לפחות רק אז יש לו אפשרות למגן של 30% על פי ציון שיעורי הבית, אם הוא מגיש פחות אז יש לו מגן של 10% בלבד. הקוד הפותר את הבעיה הזו ייראה כך:

```
if ($test > $hw_grade)
{
    $grade = $test;
}
elseif ($hw_submit >= 4)
{
    $grade = 0.7*$test+0.3*$hw_grade;
}
else
{
    $grade = 0.9*$test+0.1*$hw_grade;
}
```

בתחילה אנו בודקים, כמו בדוגמא הקודמת, האם ציון המבחן גדול מציון שיעורי הבית, אם כן אז הציון הסופי הוא פשוט ציון המבחן, אחרת יש לנו 2 אפשרויות, בשורה 5 אנו מטפלים באפשרות הראשונה – האפשרות הראשונה היא שהגשנו לפחות 4 עבודות בית, במקרה הזה הם ישמשו כמגן של 30%, אחרת – כלומר במקרה והגשנו פחות מ-4 עבודות בית ציון עבודות הבית יהווה מגן של 10%.

השתמשנו במילה השמורה elseif בשורה 4 בכדי לציין שלא מספיק לנו שהתנאי הראשון כוזב, אנחנו צריכים לוודא תנאי נוסף.

יכולנו, במקום להשתמש במילה השמורה elseif פשוט להשתמש בצד המילים השמורות else if כפי שנהוג ב ActionScript והיינו מקבלים בדיוק את אותה התוצאה. אני נוהג להשתמש בצמד המילים השמורות פשוט מפני שאני רגיל, אבל הבחירה שלך.

סוגי אופרטורים של התניה

למרות שאתה כבר יודע זאת מ ActionScript אני רוצה לסכם את הסוגים השונים של האופרטורים השונים של ההתניה.

- גדול מ (>)
- קטן מ (<)
- גדול או שווה מ (>=)
- קטן או שווה מ (<=)
- שווה ל (==)
- שונה מ (!=)

קשרים לוגיים

כמו ב ActionScript גם ב PHP ניתן להרכיב פסוק בוליאני מפסוקים בוליאניים קצרים יותר בעזרת הקשרים הלוגיים. להזכירך הקשרים הלוגיים הם:

- וגם (and) שמיוצג על ידי &&.
- או (or) שמיוצג על ידי ||.
- לא (not) שמיוצג על ידי !.

נביט בדוגמא:

```
if ($user_name == "Amir" && $pword == "89939485")
{
    // Go into the administer section
}
else
{
    // Report an error
}
```

כפי שניתן לראות השתמשנו בקשר "וגם" (&&) בכדי ליצור פסוק מורכב. הפסוק המורכב יהיה פסוק אמת אם ורק אם שני התנאים יתקיימו, כלומר גם אם שם המשתמש נכון וגם אם הסיסמא נכונה. בכל מקרה אחר, בעזרת מילת המפתח else, תוצג למשתמש הודעה.

בחן את עצמך - וודא שאתה יודע מה PHP שולח ללקוח.

```
$a = true;
$b = false;

if ( ($a)&&($b) || (!($a))&&(!($b)) )
    echo "yes";
else
    echo "no";
```

בסוף הפרק ישנה דרך מעניינת לפתרון.

שים לב שלא השתמשנו בתוחמי הקוד { } לאחר משפט ההתניה ובחלק ה else של משפט ההתניה וזאת מפני שכל שאנו צריכים לבצע מסתכם בשורה אחת עבור כל תנאי.

טופס Login – דוגמא בניחוח פלאש

נסיים את הפרק בתוכנית קצרה המשלבת פלאש ו-PHP בדגש על משפטי התניה. הרעיון מאחורי התוכנית הוא יצירת ממשק בפלאש לטופס Login למערכת דמונית כלשהי, כאשר בדיקת שם המשתמש והסיסמא נעשית בצד השרת בעזרת PHP. משמאל תוכל לראות תצלום מסך של הממשק.



לאחר שהמשתמש מזין את שם המשתמש והסיסמא ולוחץ על הכפתור Submit המידע שהוא הזין עובר ל PHP לבדיקה. PHP מחזירה לפלאש הודעה לגבי האם שם המשתמש והסיסמא נכונים, פלאש בתורה מציגה הודעה בהתאם.

כפי שמיד תראה בקוד ה PHP אנחנו נתקלים בפעם הראשונה במצב בו פלאש מעבירה מידע ל PHP ולצורך העברת הנתונים הזאת אנו משתמשים ב"כלי" שלא ראינו עד עכשיו. הבט בקוד, ההסבר יבוא אחריו.

```
<?php

$rUName = "hanan";
$rPWord = "fukxigwb";

$gUName = $_GET["uname"];
$gPWord = $_GET["pword"];

if (($rUName == $gUName) && ($rPWord == $gPWord))
{
    echo "result=1";
} else {
    echo "result=0";
}

?>
```

התסריט שלעיל יושב בקובץ בשם login.php בספרייה www/login/. בזוג השורות הראשונות הגדרתי שני משתנים שמהווים את שם המשתמש והסיסמא האמיתיים (האות r מייצגת את המילה real).

זוג השורות הבא דורש הסבר – לראשונה אנו נתקלים במערך ב PHP ועוד לא סתם מערך. המערך \$_GET, מבלי להיכנס לעובי הקורה, מכיל את המידע שהועבר ל PHP בשורת ה URL. במקרה שלנו פלאש תעביר ל PHP בשורת ה URL את שם המשתמש והסיסמא שהמשתמש הזין לתיבות הטקסט, את המידע הזה פלאש מעבירה ל PHP תחת השמות uname לשם המשתמש ו pword לסיסמא. בשורות הללו העברתי את המידע מהמערך למשתנים אחרים בכדי למנוע את התארכותה של שורת הקוד הבאה – משפט ההתניה.

משפט ההתניה בודק בעזרת שימוש בקשר "וגם" (&&) את תקפותם של שני התנאים – שם המשתמש המקורי זהה לשם המשתמש שהוזן על ידי המשתמש והסיסמא האמיתית זהה לסיסמא שהוזנה על ידי המשתמש. אם שני התנאים מתקיימים PHP מחזירה לפלאש משתנה בשם result בעל הערך 1, אחרת היא מחזירה את אותו משתנה אך עם הערך 0.

נסתכל כעת על קוד ה ActionScript (הקוד נמצא בעמוד הבא).

בשורה הראשונה השתמשתי ב stop() בכדי לעצור את הסרטון מלרוץ. שורה לאחר מכן הגדרתי אובייקט חדש מסוג LoadVars בשם login בעזרתו נתקשר עם תסריט ה PHP שלנו. בשורות הבאות ניתן לראות פונקציה לטיפול באירוע onLoad של האובייקט login. פונקציה זו נקראת כידוע מיד כשכל המידע הוזרם מ PHP לפלאש ובמקרה שלנו היא בודקת את ערכו של המאפיין result של login, בעצם המאפיין שתסריט ה PHP שלנו ייצר. אם ערך מאפיין זה הוא 0, זהו סימן שהמשתמש הזין שם משתמש וסיסמא שאינם חוקיים, בעזרת ActionScript אנחנו מעבירים אותו לפריים מס' 3, זהו פריים בו מוצג הודעת Invalid password. אחרת (שים לב

לשימוש בצמד המילים השמורות (else if) במידה וערך result הוא 1 המשתמש מקבל הודעת שלום.

בלוק הקוד האחרון מגדיר מטפל אירוע onRelease למוביקליפ btnSubmit. אירוע זה נקרא מתי שהמשתמש לוחץ (או יותר נכון מסיים את הלחיצה) על הלחצן Submit. ברגע שזה קורה אנו קוראים לשיטה load של login עם כתובת ה URL של תסריט ה PHP שלנו ועם המשתנים בפורמט שם-ערך, משתנים אלו ייכנסו היישר למערך \$_GET ב-PHP.

```
stop();
login = new LoadVars();

login.onLoad = function()
{
    if (login.result == 0)
    {
        gotoAndStop(3);
    } else if (login.result == 1) {
        gotoAndStop(2);
    }
}

btnSubmit.onRelease = function()
{
    url = "http://localhost/login/login.php";
    uname = "uname=" + _root.gUName.text;
    pword = "pword=" + _root.gPWord.text;
    login.load(url+"?" +uname+"&" +pword);
}
```

סיכום

בפרק זה למדנו אודות משפטי התניה, למזלנו לא היה הרבה להוסיף כי התחביר והקונספט של משפטי התניה ב PHP כמעט זהה לתחביר ולקונספט ב ActionScript.

בקשר לחידה שנשאלה קודם בפרק. אם תקרא בעברית את הפסוק תקבל את המשפט הבא: או ש a אמיתי ו-b אמיתי או ש a שקרי ו b שקרי, במילים אחרות אם a ו-b זהים אז הפסוק כולו אמת, אחרת הפסוק כוזב. במקרה שלנו a ו-b שונים ולכן הביטוי כולו יוערך כשקר. שים לב שניתן להחליף את כל הביטוי בביטוי:

```
if ($a == $b)
```