פרק 17– הכללת קוד מקבצים חיצוניים

הקדמה

הכללת קוד מקבצים חיצוניים מאפשרת לנו לחלק את המשימה המוטלת על התוכנה שלנו לקבצים שונים ובכך לשמור על סדר, על מודולאריות ועל חלוקת עבודה. למה מודולאריות? ובכן נניח ויש לנו פונקציה שאנו משתמשים בה שוב ושוב כמו פונקציה שמחברת אותנו למסד נתונים ומטפלת בכל השגיאות במידה ועולה הצורך, במקרה כזה עדיף שהפונקציה תשב פעם אחת וכל התוכנות שאנו בונים יקראו לה, פתרון ה Cut & Paste לא יהיה יעיל כי אם נעתיק את הפונקציה לכל קובץ שנשתמש בו ויום אחד נגלה באג או נרצה לעדכן את הפונקציה אנו נצטרך לעשות זאת מספר רב של פעמים. הרעיון מאחורי מודולאריות הוא שימוש חוזר בקוד, פונקציות זו רמה אחת של מודולאריות שהרי בעזרת פונקציות אנו יכולים לכתוב קוד פעם אחת ולהשתמש בו שוב ושוב, הכללת קוד מקבצים אחרים זו רמה אחרת של מודולאריות – מודולאריות של פונקציות.

הפקודה include

הפקודה (לא פונקציה) include היא הפקודה הבסיסית להכללת קוד מקובץ אחד בקובץ אחר. אתה ודאי מוכר עם include מ ActionScript אך אנו נביט בדוגמא בכל זאת ולאחריה אציג את הפרטים הקטנים מאחורי include שכדאי לדעת.

הנורמה היא לקרוא לקבצים שאמורים להיכלל על ידי קבצי PHP בשם כלשהו עם הסיומת inc ולא php אם כי הדבר איננו בגדר חובה. נניח ויש בידינו שני קבצים – קובץ בשם index.php וקובץ אחר בשם myMath.inc. עוד נניח שהקובץ myMath.inc מכיל פונקציה בשם myMax שמקבלת כארגומנטים שני מספרים ומחזירה את הגדול מביניהם. הקוד הבא מראה את תוכנו של הקובץ myMath.inc

```
<?
function myMax($a, $b)
{
     if ($a > $b) return $a;
     return $b;
}

?>
```

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

אני רוצה שתשים לב לכך שהקובץ נראה כמו כל קובץ PHP וכל הקוד נמצא בין תגיות פתיחה וסגירה של PHP.

הקובץ index.php נראה כך:

```
<?
include "myMath.inc";
echo myMax(10,5);
?>
```

לרוב אני נוהג להשמיט בדוגמאות את תגיות הפתיחה והסגירה אבל הפרק בכדי להיות בטוח שאתה מבין שתמיד צריך להשתמש בתגיות פתיחה וסגירה אני אחרוג ממנהגי.

בשורה הראשונה ניתן לראות את הפקודה include בפעולה – בכדי לכלול קוד מקובץ חיצוני כל שצריך לעשות הוא לכתוב את הפקודה include ולאחריה מחרוזת המציינת את שם הקובץ אשר את תוכנו אנו כוללים. ההנחה היא כי הקובץ myMath.inc נמצא באותה ספרייה בה נמצא הקובץ index.php, אם נרצה לכלול קובץ מספרייה אחרת נצטרך לציין במפורש את הספרייה לדוגמא, אם הקובץ myMath.inc היה בספרייה שנקראת include בתוך הספרייה שבה נמצא index.php אז הקוד היה נראה כך:

```
<?
include "include/myMath.inc";
echo myMax(10,5);
?>
```

במידה והקובץ אותו אנו רוצים לכלול נמצא דווקא בספרייה חיצונית לקובץ אותו אנו מריצים, לדוגמא נניח והקובץ אותו אנו מריצים נקרא index.php והוא נמצא בספרייה folder אך מנסה לכלול קובץ חיצוני שנמצא בספרייה הראשית, אנו נוכל להיעזר בסינטקס הבא:

```
<?
include "../myMath.inc";
echo myMax(10,5);
?>
```

כמו כן אפשר לציין במפורש מאיזה URL לכלול את הקובץ, לדוגמא:

```
<?
include "http://www.flashoo.co.il/include/myMath.php";
echo myMax(10,5);
?>
```

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

לאחר שכללנו קובץ מסוים אנו יכולים לקרוא לכל הפונקציות שלו ולכל המשתנים שבו כאילו הוגדרנו מתוך הקובץ המקורי (זה שביצע קריאה ל include) אך יש כמה דברים שצריך לדעת בעבודה עם include:

אם נכלול קובץ מתוך גוף של פונקציה טווח ההכרה של כל המשתנים והפונקציות שבו יהיו נגישים רק מהפונקציה עצמה, כל לדוגמא הקוד הבא יציג הודעת שגיאה:

```
<?php
function myFunc()
{
     include "myMath.inc";
     echo myMax(10,5);
}

echo myMax(5,7);
?>
```

הבעיה בקוד הוא שה include מתבצע מגוף הפונקציה myFunc והפונקציה עצמה תקינה ופועלת היטב, אך ביציאה מהפונקציה myFunc הפונקציה myMax איננה מוגדרת שהרי ה include נעשה מגוף הפונקציה ולכן myMax היא פונקציה בטווח ההכרה של myFunc.

בעיה נוספת שעלולה להתעורר היא עם include בתוך משפט התניה, לדוגמא:

```
<?php
if ($condition) include "this.inc";
           else include "that.inc";
?>
```

למרות שלכאורה לא נראה שיש בעיה בקוד הזה, יש צורך ב PHP להציב include במשפטי התניה בתוך סוגריים מסולסלות, הקוד הבא הוא קוד תקין:

```
if ($condition)
{
     include "this.inc";
} else {
     include "that.inc";
}
```

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

הפקודה require

הפקודה require מתנהגת כמו הפקודה include פרט לשני מאפיינים, האחד – במידה והקובץ המבוקש להיכלל איננו נמצא require מפסיקה את הרצת הקוד. Include לעומת זאת תמשיך את פעילות הקוד גם במקרה בו הקובץ איננו נמצא. בגלל זה require מתאים יותר להכללת קבצים החיוניים לפעולת הסקריפט, אשר בלעדיהם אין דרך לבצע את השורות הבאות.

ההבדל השני נעוץ בכך ש include מכליל את כל הקובץ המבוקש לעומת require שמכלילה רק את מה שהיא רואה שהקוד מבקש, לדוגמא אם יש לנו קובץ myMath.inc בעל 100 פונקציות מתמטיות ואנו רוצים להשתמש רק באחדות מהפונקציות הללו עדיף לנו להשתמש ב require אשר לא טוען את כל הקובץ אלא רק את הפונקציות שאנו באמת משתמשים בהם – ההחלטה איזו פונקציה לטעון מתבצעת באופן אוטומטי ולנו כמפתחים אין כל עניין בדרך ביצועה (קופסא שחורה כבר אמרנו?).

באופן כללי אפשר לומר שהביצועים של include טובים יותר מבחינת מהירות מאשר ביצועי require במקרה הכללי, כנראה בגלל ש require צריכה לבדוק אילו פונקציות נקראות לעומת include שפשוט כוללת את כולן.

הפקודות include_once ו require_once

ברגע שננסה לבצע include או require יותר מפעם אחת לאותו הקובץ אנו נקבל הודעת שגיאה – על כן אנו צריכים להיזהר ולא לנסות לכלול את אותו הקובץ יותר מפעם אחת. הבעיה היא שיש מקרים בהם יש לנו include או require בהתניות או בפרויקטים גדולים בהם קבצים כוללים קבצים שכוללים קבצים... ואנו לא בטוחים לגמרי אם ההכללה תתבצע יותר מפעם אחת.

בכדי להיות בטוחים לגמרי שלא ניתקל בבעיה כזו עומדות לפנינו זוג פקודות בשם include_once ו require_once אשר פועלות בדיוק כמו include ו require בהתאמה אך לפני שהן כוללת קובץ הן בודקות האם הקובץ כבר נכלל – במקרה כזה הן לא כוללות אותו שוב – Better safe then sorry.

במקרה והקובץ כבר נכלל ו include_once או require_once דאגו כמובן לא לכלול אותו בפעם השנייה, הפקודה מחזירה true.

סיכום

הפרק הזה, ככל שקצר וקליל, כך חשוב והכרחי להבנה בכדי לעבוד על פרויקטים גדולים המתרפסים על יותר מקובץ אחד. בפרק הבא נביט על פונקציות שימושיות לעבודה עם מחרוזות.

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