

פונקציות

[הגדרת פונקציה](#)

[מספר משתנה של פרמטרים](#)

[שליחת ארגומנטים By Reference](#)

[החזרת ערך By Reference](#)

[משתנים מקומיים וגלובליים](#)

הגדרת פונקציה

פונקציה היא קטע קוד שיש לו שם ושניתן לקרוא להפעלתו באמצעות שמו.

מגדירים פונקציה באמצעות המילה function, ציון שמה בתוספת סוגריים עגולות ותחילת הקוד שלה בתוך סוגריים מסולסלות.

```
function _____ ( )
{
    ...
}
```

קריאה להפעלתה של פונקציה נעשית באמצעות ציון שמה בתוספת סוגריים עגולות מייד לאחר מכן.

לדוגמא, הפונקציה הבאה מדפיס למסך Hello World כל פעם שקוראים להפעלתה.

```
<?php
function sayHello()
{
    echo "<BR/>Hello World!<BR/>";
}
sayHello();
sayHello();
sayHello();
?>
```

פונקציות ב-PHP אינן Case Sensitive. קריאה להפעלתה של פונקציה באמצעות ציון שמה ללא הקפדה על דיוק בכל הקשור לאותיות קטנות וגדולות עדיין תגרום להפעלתה. בדומה לכללים שחלים על שם של משתנה גם שמה של פונקציה יכול לכלול אותיות, ספרות וקו תחתי בלבד. כמו כן, שמה של פונקציה לא יכול להתחיל בסיפרא.

בדומה לשפות תיכנות אחרות גם PHP מאפשרת להגדיר פונקציה אשר מחזירה ערך באמצעות הפקודה return. להבדיל מ-Java ומ-C# אין צורך לציין את ה-type (הטיפוס) המוחזר בכותרת של הפונקציה. כאשר מתבצעת הפקודה return ה-PHP Engine מזהה את הטיפוס של הערך המוחזר בזמן ריצה ונוהג בהתאם.

הדוגמא הבאה כוללת פונקציה אשר מחשבת את סכום המספרים בין 1 ו-100 (כולל) ומחזירה אותו. הערך שהיא מחזירה מוכנס לתוך משתנה ולאחר מכן ערכו של המשתנה מודפס למסך.

```

<?php
function sum()
{
    $total=0;
    for($index=1; $index<=100; $index++)
    {
        $total+=$index;
    }
    return $total;
}
$number = sum();
echo "sum of 1..100 is $number";
?>

```

ניתן להוסיף לשורת הכותרת של הפונקציה את הגדרתם של פרמטרים. את הפרמטרים יש לרשום בתוך הסוגריים העגולות של הפונקציה. להבדיל משפות תיכנות אחרות אין צורך לציין את ה-type (הטיפוס) של כל אחד מהפרמטרים. הדוגמא הבאה מציגה פונקציה שבעת הפעלתה היא מקבלת שני ערכים, מחשבת את הסכום שלהם ומחזירה אותו.

```

<?php
function sum($numA,$numB)
{
    $total=0;
    $total = $numA + $numB;
    return $total;
}
$number = sum(24,44);
echo "sum of 24 and 44 is $number";
?>

```

מספר משתנה של פרמטרים

כאשר מגדירים פונקציה ניתן להגדיר עבור כל אחד מהפרמטרים ערך ברירת מחדל אשר יהיה ערכו של אותו פרמטר במקרה שלא נשלח עבורו ארגומנט. כאשר קוראים להפעלתה של פונקציה שהוגדרה עם פרמטרים ניתן לשלוח מספר שונה של ארגומנטים ובלבד שמספר הארגומנטים שאנו שולחים איננו קטן ממספר הפרמטרים שהוגדרו ללא ערך ברירת מחדל. כמו כן, יש למקם את הפרמטרים עם ערך ברירת המחדל בסוף רשימת הפרמטרים של הפונקציה.

הדוגמה הבאה כוללת פונקציה אשר מדפיסה למסך ברכה. הפונקציה מוגדרת עם שני פרמטרים. הפרמטר הראשון הוא שם האדם שמברכים והפרמטר השני הוא הברכה עצמה. הפרמטר השני מוגדר עם ערך ברירת המחדל "Good Morning".

```
<?php
function greet($name,$greeting="Good Morning")
{
    echo "<BR>".$greeting." ".$name;
}
greet("Moshe");
greet("David","Good Evening");
?>
```

הפלט של התכנית הוא:

```
Good Morning Moshe
Good Evening David
```

בשפת התיכנות PHP יש פונקציות מוכנות אשר מסייעות לכתיבת פונקציות עם מספר משתנה של פרמטרים.

```
func_num_args()
```

פונקציה זו מחזירה את מספר הארגומנטים שנשלחו אל הפונקציה.

בהרצת הקוד הבא:

```
<?php
function doSomething()
{
    echo "<BR>num of args is ".func_num_args();
}
doSomething();
doSomething("abc");
doSomething(1,2,3);
```

```
?>
```

נקבל את הפלט הבא:

```
num of args is 0
num of args is 1
num of args is 3
```

```
func_get_arg($index)
```

פונקציה זו מחזירה את הארגומנט שמספר האינדקס שלו נשלח אליה.

בהרצת הקוד הבא:

```
<?php
function doSomething()
{
    $num_of_args = func_num_args();
    for($index=0; $index<$num_of_args; $index++)
    {
        echo "<BR>argument num.$index is ".func_get_arg($index);
    }
}
doSomething(1,2,3);
?>
```

הפלט שנקבל הוא:

```
argument num.0 is 1
argument num.1 is 2
argument num.2 is 3
```

```
func_get_args()
```

פונקציה זו מחזירה `reference` למערך שמכיל את הארגומנטים שנשלחו אליה.

בהרצת הקוד הבא:

```
<?php
function doSomething()
{
    $num_of_args = func_num_args();
    $vec_of_args = func_get_args();
    for($index=0; $index<$num_of_args; $index++)
```

```
{
    echo "<BR>argument num.$index is ".$vec_of_args[$index];
}
doSomething(1,2,3);
?>
```

נקבל את הפלט הבא:

```
argument num.0 is 1
argument num.1 is 2
argument num.2 is 3
```

שליחת ארגומנטים By Reference

שפת התיכנות PHP מאפשרת לשלוח ארגומנט By Reference באמצעות הוספת האופרטור '&' לפרמטר המתאים. המשמעות של שליחת ארגומנט By Reference היא שמה שבפועל נשלח הוא ה-reference למשתנה שאותו מנסים לשלוח כארגומנט (במקום הערך שבתוכו) ומסיבה זו שינויים שיתבצעו בתוך המתודה יבואו לידי ביטוי מחוץ לפונקציה במשתנה שנשלח By Reference כארגומנט לפונקציה.

בהרצת התוכנית הבאה:

```
<?php
function doSomething(&$num)
{
    $num++;
}
$varA = 2;
echo "<BR>before... ".$varA;
doSomething($varA);
echo "<BR>after... ".$varA;
?>
```

הפלט שנקבל הוא:

```
before... 2
after... 3
```

החזרת ערך By Reference

פונקציות ב-PHP מחזירות ערך By Value (ברירת מחדל). הוספת האופרטור '&' לשם הפונקציה תגרום להחזרת הערך By Reference. שימושי במיוחד במקרים שבהם הפונקציה מחזירה resource כגון חיבור לבסיס נתונים.

```
function &getConnection()  
{  
...  
...  
return connection;  
}
```


משתנים מקומיים וגלובליים

משתנים גלובליים הם משתנים שטווח ההכרה שלהם הוא התכנית כולה. משתנה גלובלי הוא כל משתנה שמצהירים עליו מחוץ לגבולות של פונקציה ומחוץ לגבולות של class.

משתנה מקומי הוא משתנה שמוגדר בתוך גבולות של פונקציה ושטווח ההכרה שלו הוא הפונקציה בלבד. כשהפונקציה מסתיימת המשתנה המקומי מסיים את חייו.

קיימות שתי דרכים שבהן ניתן לגשת למשתנה גלובלי בתוך פונקציה. דרך אחת היא באמצעות הפקודה global. יש לרשום את הפקודה global ומייד אחריה את שם המשתנה הגלובלי שאליו רוצים לגשת. ללא שימוש בפקודה global לא ניתן יהיה לגשת בתוך גבולותיה של פונקציה למשתנה הגלובלי.

```
<?php
$my_global_num = 12;
function doSomething()
{
    global $my_global_num;
    echo $my_global_num;
}
doSomething();
?>
```

דרך נוספת לגשת למשתנה גלובלי בתוך גבולותיה של פונקציה היא להשתמש במערך \$GLOBALS ושליחת השם של המשתנה (הגלובלי) ללא סימן ה-\$ (שאליו רוצים לגשת בתור ה-index של האיבר שאליו רוצים לגשת במערך \$GLOBALS).

הדוגמא הבאה זהה בפעולתה לדוגמא הקודמת.

```
<?php
$my_global_num = 12;
function doSomething()
{
    echo $GLOBALS['my_global_num'];
}
doSomething();
?>
```