



Full Stack- server- PHP

הרצאה 2

משתנים

פונקציות על מחרוזות

מערכים



משתנים



סוגי משתנים

- String
- Integer
- Float (floating point numbers - also called double)
- Boolean
- Array
- Object
- NULL
- Resource



PHP String

■ מחרוזת מוגדרת כרצף של תווים

■ על מנת להגדיר משתנה מסוג מחרוזת יש להגדיר אותו תחת גרשיים
(") או גרש בודד (")

```
<?php
```

```
$x = "Hello world!";
```

```
$y = 'Hello world!';
```

```
echo $x;
```

```
echo "<br>";
```

```
echo $y;
```

```
?>
```

פלט:

Hello world!

Hello world!



פונקציות על מחרוזות

- `strlen()` - Return the Length of a String
 - ```
<?php
echo strlen("Hello world!"); // outputs 12
?>
```
- `str_word_count()` - Count Words in a String
  - ```
<?php  
echo str_word_count("Hello world!"); // outputs 2  
?>
```
- `strrev()` - Reverse a String
 - ```
<?php
echo strrev("Hello world!"); // outputs !dlrow olleH
?>
```



# פעולות על מחרוזות- המשך

---

- strpos() - Search For a Text Within a String
  - <?php  
echo strpos("Hello world!", "world"); // outputs 6  
?>
- str\_replace() - Replace Text Within a String
  - <?php  
echo str\_replace("world", "Dolly", "Hello world!"); // outputs Hello Dolly!  
?>





# PHP Integer

---

■ משתנה מסוג integer מוגדר כמשתנה מספרי בטווח:

■ 2,147,486,648 עד -2,147,483,648 עבור 32b

■ 9223372036854775808 עד -9223372036854775808 עבור 64b

■ מספרים גדולי/קטנים מכך יאוחסנו כ-float

■ מספר שהגיע כתוצאה של מספרים שאחד מהם שברי, יאוחסן כ-float (10 = 2.5\*4)

■ לא יכול לכלול מספר שברי (עם נקודה)

■ יכול להיות מספר שלם חיובי או שלילי

הפונקציה `var_dump()` מקבלת משתנה

ומחזירה את טיפוס הנתונים שלו

בדוגמא יחזור: `int(5985)`

```
<?php
```

```
$x = 5985;
```

```
var_dump($x);
```

```
?>
```



# PHP Integer

---

## ■ קבועים מוגדרים:

- `PHP_INT_MAX` - The largest integer supported
- `PHP_INT_MIN` - The smallest integer supported
- `PHP_INT_SIZE` - The size of an integer in bytes

## ■ פונקציות עבור בדיקה אם משתנה הוא `integer`:

- `is_int()`
- `is_integer()` - alias of `is_int()`
- `is_long()` - alias of `is_int()`

פונקציות אלו מחזירות משתנה מסוג `boolean`





# PHP Float

---

■ Floating point number - נקודה צפה

```
<?php
```

```
$x = 10.365;
```

```
var_dump($x);
```

```
?>
```

פלט:

float(10.365)

פונקציות לבדיקה האם מספר הוא float (מחזירות boolean)

is\_float()

is\_double() - alias of is\_float()



# Float predefined constants

---

■ קבועים עבור float

- PHP\_FLOAT\_MAX - The largest representable floating point number
- PHP\_FLOAT\_MIN - The smallest representable positive floating point number
- - PHP\_FLOAT\_MAX - The smallest representable negative floating point number
- PHP\_FLOAT\_DIG - The number of decimal digits that can be rounded into a float and back without precision loss
- PHP\_FLOAT\_EPSILON - The smallest representable positive number  $x$ , so that  $x + 1.0 \neq 1.0$



# PHP Boolean

---

■ משתנה בוליאני יכול להכיל

True ■

False ■

```
<?php
$x = True;
var_dump($x);
?>
```

פלט:  
bool(true)



# PHP Infinity

---

- מספר שגדול מ- `PHP_FLOAT_MAX` מוגדר כאינסופי.
- פונקציות לבדיקה האם משתנה מכיל ערך סופי
  - `is_finite()`
  - `is_infinite()`
- הפונקציות מחזירות ערך של 1 במידה והמספר סופי  
אחרת מחזירה `false/nothing`



# PHP NaN

---

NaN- Not a Number ■

פונקציה שבודקת אם ערך הוא לא מספר ■

`is_nan()`

הפונקציה מחזירה (1) `true` אם הערך הוא לא מספר,  
אחרת מחזירה `false/nothing`



# Numerical Strings

---

```
<?php
$x = 5985;
var_dump(is_numeric($x));

$x = "5985";
var_dump(is_numeric($x));

$x = "59.85" + 100;
var_dump(is_numeric($x));

$x = "Hello";
var_dump(is_numeric($x));
?>
```

פלט:

```
bool(true)
bool(true)
bool(true)
bool(false)
```

■ מחרוזת מספרית

■ is\_numeric()

■ פונקציה שבודקת אם  
משתנה הוא מספרי.

■ הפונקציה מחזירה true אם  
המשתנה הוא מספר,  
אחרת מחזירה false



# מערכים

- נניח שצריך משתנים שיכילו מספרים שונים:

```
<? php
 $my_var1 = 2;
 $my_var2 = 4;
 $my_var3 = 8;
 $my_var4 = 16;
 $my_var5 = 24;
 $my_var6 = 32;
 $my_var7 = 64;
?>
```

- מה עושים? במקום ליצור שבעה משתנים שונים, אנו ניצור מערך או **array** באנגלית, הוא משתנה מסוג מיוחד שמכיל כמה נתונים.

```
<?php
 $my_arr = array(2,4,8,16,32,64,128);
?>
```



# מערכים

■ ההצהרה על איבר אחד במערך:

```
<?php
 $my_arr = array(2,4,8,16,32,64,128);
 print $my_arr[1]
?>
```

■ המספר שנמצא במקום 1 הוא 4.

■ אפשר לערבב סוגי נתונים במערך, למשל:

```
<?php
 $my_arr = array(2,"four",8,"sixteen",32,64,"one hundred and twenty eight");
 print $my_arr[1];
?>
```

■ התוצאה תהיה כמובן four





# מערך אסוציאטיבי

- עד עכשיו המערך היה מערך נומרי, כלומר – המפתח (key) היה מספר
- ב-PHP ניתן לייצר גם מערך שהמפתח שלו הוא מחרוזת טקסט
- לדוגמא:

```
<?php
 $my_arr = array('my_phone'=> 123456
 , 'my_sons_number'=> 3,
 'my_city' => 'Petah Tiqwa');
 print $my_arr['my_phone'];
?>
```

הפלט : 123456

- במקום להכניס ערכים לתוך סוגריים ולדאוג שיהיו פסיקים ביניהם, יש לפרט גם את המפתח ולעשות חץ שמצביע על הערך שלהם
- המפתח יכול להיות מחרוזת או מספר



# מערך אסוציאטיבי-המשך

---

- ניתן להשתמש במפתחות של מחרוזת טקסט וגם במספרים, למשל:
- לדוגמא:

```
<?php
 $my_arr = array('my_phone' => 123456,
 'my_sons_number' => 3,
 'my_city' => 'Petah Tiqwa',
 'my_car' => 'Hyundai',
 2 => 'two',);
 print $my_arr[2];
?>
```

הפלט : two

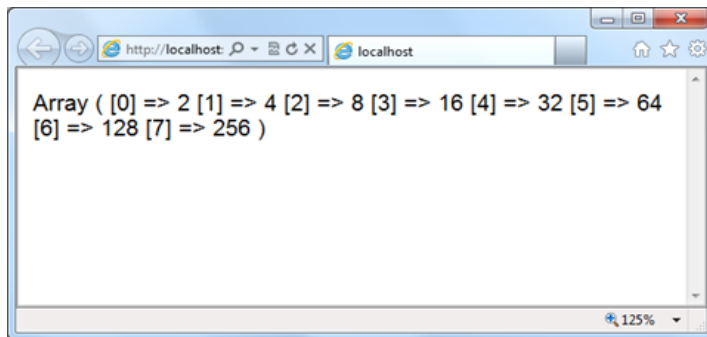
# הדפסת תוכן מערך

■ PHP מאפשרת להדפיס תוכן מערך באמצעות הפקודה `print_r`

■ מדפיסה את כל המערך, כאשר בתוך הסוגריים המרובעים יופיע המפתח, וחץ אל הערך.

■ דוגמא:

```
<?php
 $my_arr = array(2,4,8,16,32,64,128, 256);
 print_r($my_arr);
?>
```



■ הפלט:



# הוספת איבר למערך

■ כדי להוסיף איבר הפנייה למערך תהיה עם סוגריים מרובעים

■ האיבר יתווסף בסוף המערך

```
<?php
 $my_arr = array(2,4,8,16,32,64,12);
 $my_arr[] = 256;
 print_r($my_arr);
?>
```

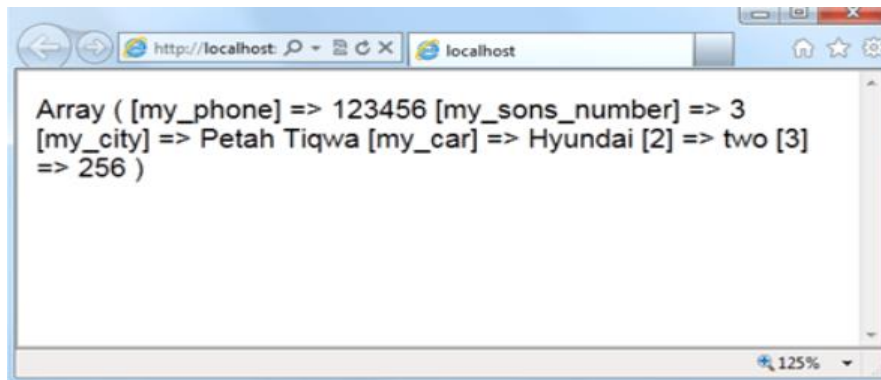
```
Array ([0] => 2 [1] => 4 [2] => 8 [3] => 16 [4] => 32 [5] => 64
[6] => 128 [7] => 256)
```

■ הפלט:

- עבור מערך אסוציאטיבי המפתח יהיה המקום 0 אלא אם יש במערך key שהינו מספר ואז הוא יהא גדול ב- 1 מה- key המספרי האחרון

```
<?php
 $my_arr = array('my_phone' => 123456,
 'my_sons_number' => 3,
 'my_city' => 'Petah Tiqwa',
 'my_car' => 'Hyundai',
 2=>'two',);
 $my_arr[] = 256;
 print $my_arr;
?>
```

הפלט:



- מה שקרה הוא הוספת מפתח 3 עם ערך 256. למה 3? כי יש לנו מפתח 2 במערך.



➤ ואם רוצים להוסיף ערך דווקא למפתח 0 (המפתח הראשון) ? זה דווקא פשוט:

```
<?php
 $my_arr = array('my_phone' => 123456,
 'my_sons_number' => 3,
 'my_city' => 'Petah Tiqwa',
 'my_car' => 'Hyundai',
 2 => 'two',);
 $my_arr[0] = 256;
 print $my_arr;
?>
```

➤ במקום סתם סוגריים מרובעות, אנו מציינים את מספר הערך שרוצים להוסיף.



# PHP Array

---

- מערך יכול להכיל מספר ערכים במשתנה אחד, ערכים יופרדו ע"י פסיק

פלט: הפונקציה `var_dump()` מחזירה את סוג המשתנה וערכו

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
var_dump($cars);
?>
```

```
array(3) {
 [0]=>
 string(5) "Volvo"
 [1]=>
 string(3) "BMW"
 [2]=>
 string(6) "Toyota"
}
```



# PHP Object

---

- אובייקט הוא מופע של מחלקה (class)
- כאשר אובייקט נוצר הוא יורש את כל התכונות מהמחלקה אך לכל אובייקט יכולים להיות ערכים שונים.



# מחלקה ואובייקט - דוגמא

```
<?php
class Car {
 public $color;
 public $model;
 public function __construct($color, $model) {
 $this->color = $color;
 $this->model = $model;
 }
 public function message() {
 return "My car is a " . $this->color . " " . $this->model . "!";
 }
}

$myCar = new Car("black", "Volvo");
echo $myCar -> message();
echo "
";
$myCar = new Car("red", "Toyota");
echo $myCar -> message();
?>
```

פלט:

My car is a black Volvo!

My car is a red Toyota!



# PHP NULL

---

- סוג משתנה שיכול להיות רק NULL
- למשתנה מסוג זה אין ערך למעט NULL
- משתנים שמוגדרים ללא ערך מקבלים אוטומטית ערך NULL

```
<?php
$x = "Hello world!";
$x = null;
var_dump($x);
?>
```

פלט:  
NULL



# Casting- String/Float → Integer

---

- המרה של מספר מסוג float/string ל-integer מתבצעת ע"י (int) לפני המשתנה

```
<?php
```

```
// Cast float to int
```

```
$x = 23465.768;
```

פלט:

```
$int_cast = (int)$x;
```

23465

```
echo $int_cast;
```

23465

```
echo "
";
```

```
// Cast string to int
```

```
$x = "23465.768";
```

```
$int_cast = (int)$x;
```

```
echo $int_cast;
```

```
?>
```



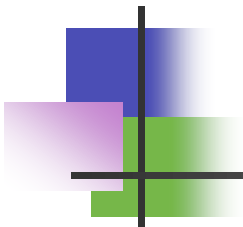
# דוגמא 1

---

■ צרו קובץ hello.php בעל התוכן הבא:

```
<html>
<head>
 <title>PHP Test</title>
</head>
<body>
 <?php echo '<p>Hello World</p>'; ?>
</body>
</html>
```

הריצו אותו



הקובץ מתורגם ע"י PHP ושולח לדפדפן את הקוד הבא:

```
<html>
<head>
 <title>PHP Test</title>
</head>
<body>
<p>Hello World</p>
</body>
</html>
```



## דוגמא 2- שימוש במשתנים

---

- משתנים מתחילים ב-\$
- משתנים מסוג מחרוזת יתחמו בגרשיים
- שם משתנה יתחיל באות או ב-\_\_
- שם משתנה לא יתחיל במספר
- שם משתנה יכול להכיל רק אותיות/מספרים/\_\_\_
- אותיות גדולות/קטנות משנות בשם המשתנה

```
<?php
$txt = "PHP";
echo "I love $txt!";
?>
```

הריצו

הפלט שמתקבל:

I love PHP!



## דוגמא 3- רגישות אותיות- פקודות

---

```
<?php
ECHO "Hello World!
" ;
echo "Hello World!
" ;
EcHo "Hello World!
" ;
?>
```

כל המשפטים הינן זהים



## דוגמא 4- רגישות אותיות- משתנים

---

```
<?php
$color = "red";
echo "My car is " . $color . "
";
echo "My house is " . $COLOR . "
";
echo "My boat is " . $coLoR . "
";
?>
```

רק השורה הראשונה תדפיס את תוכן המשתנה color





## דוגמא 4- הערות

■ הערות נכתבות ע"י // או # או /\*comment\*/

```
<?php
```

```
// This is a single-line comment
```

```
This is also a single-line comment
```

```
/*
```

```
This is a multiple-lines comment block
that spans over multiple
lines
```

```
*/
```

```
?>
```



# תרגילים

---

■ מה הפלט עבור על מקרה

תרגיל 1

```
<?php
$txt = "lala.com";
echo "I love $txt!";
?>
```

תרגיל 2

```
<?php
$txt = "lala.com";
echo "I love " . $txt . "!";
?>
```



## תרגיל 3

```
<?php
$x = 5;
$y = 4;
echo $x + $y;
?>
```

## תרגיל 4

```
<?php
$x = 5;
$y = 4;
echo '$x + $y';
?>
```



# פתרונות

---

I love lala .1

I love lala .2

9 .3

$\$x + \$y$  .4

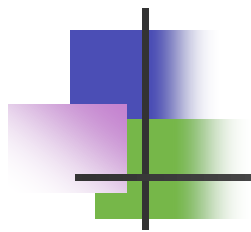


# קבועים

- עד כה המידע הוכנס למשתנים, אבל ניתן גם להשתמש בקבועים אשר מכילים מידע שאמור לא להשתנות כלל ואסור לו להשתנות.
- משתמשים בהם בדרך כלל על מנת להמנע ממספרים או מחרוזות טקסט שמופיעות בתוך הקוד כדי שהקוד יהיה קריא וקל לשינויים.
- לדוגמא:

```
?>php
define("SECRET_PASSWORD", "mellon");
print SECRET_PASSWORD ;
?>
```

- יודפס: mellon
- כללים:
- שם קבוע באותיות רישיות בלבד
- מוגדר בתחילת המסמך
- אין צורך להשתמש ב-\$



# **פונקציות לעבודה עם מערכים**



# מערכים

- מערכים הם סוג של משתנה שיכול להחזיק מספר פריטים ביחד

```
$arrayName= array(item1,item2,item3...)
```

- לדוגמא:

```
$names = array("Moshe", "Chen", "Gavriel");
```

- ניתן להוסיף פריטים למערכים

- כדי לגשת לפריטים במערך נשתמש במספר האינדקס, לדוגמא:

```
echo $names[0]
```

פלט: Moshe



# מערכים- ערכים ומפתחות

---

- למערכים יש מפתחות- keys וערכים- values
- מוצג בצורה הבאה: ערך => מפתח
- על מנת לראות את המערך בצורתו המפורשת משתמשים ב- `print_r()`

```
echo "<pre>";
print_r($names);
echo "</pre>";
```

פלט:

```
Array
(
 [0] => Moshe
 [1] => Shlomo
 [2] => Gavriel
)
```





# count

---

■ הפונקציה count מחזירה את מספר הפריטים  
במערך נתון

```
$cars = array('Talbot', 'Sussita', 'Tata');
echo count($cars);
```

בדוגמא יודפס למסך המספר 3



# פונקציות לחיפוש במערכים

---

- הפונקציה `in_array` מחפשת פריט במערך ומחזירה `TRUE` אם הפריט נמצא, אחרת מחזירה `FALSE` אם הפריט לא קיים

```
$cars = array('Talbot', 'Sussita', 'Tata');
if(in_array('Sussita', $cars)){
 echo 'Yes';
} else {
 echo 'No';
}
```

פלט: Yes



# פונקציות לחיפוש במערכים

---

- הפונקציה `array_search` מחפשת פריט במערך ומחזירה את מיקום הפריט. אם הפריט לא נמצא הפונקציה מחזירה `FALSE`
- ```
$cars = array('Talbot', 'Sussita', 'Tata');  
echo array_search('Sussita', $cars);
```

פלט: 1



פונקציות להוספת פריטים למערך

■ הפונקציה `array_push` מוסיף פריט או מספר פריטים לסופו של המערך

```
$cars = array('Talbot', 'Sussita', 'Tata');  
array_push($cars, 'Delta', 'Volvo');  
var_dump($cars);
```

פלט:

```
array(5) {  
    [0]=> string(6) "Talbot"  
    [1]=> string(7) "Sussita"  
    [2]=> string(4) "Tata"  
    [3]=> string(5) "Delta"  
    [4]=> string(5) "Volvo" }
```



הוספת פריט למערך

■ ניתן להוסיף פריט למערך בלי לציין את מספר האינדקס

■ `$names[] = "David";`



פונקציה להוצאת פריטים מהמערך

■ הפונקציה `array_pop` מסיר את הפריט האחרון במערך ומחזיר את הפריט שהוסר מהמערך

```
echo array_pop($cars); // Volvo  
var_dump($cars);
```

פלט:

```
array(5) {  
    [0]=> string(6) "Talbot"  
    [1]=> string(7) "Sussita"  
    [2]=> string(4) "Tata"  
    [3]=> string(5) "Delta"  
    [4]=> string(5) "Volvo" }
```



שינוי פריט במערך

■ כדי לשנות את ערכם של פריטים במערך, ניגש
לפריטים עם האינדקס שלהם

■ לדוגמא:

■ `$names[1]="Yigal";`



מחיקת פריטים ומערכים

- מחיקת פריטים ומערכים נעשית באמצעות `.unset()`

- נשתמש ב-`Unset` כדי למחוק את הפריט הראשון מהמערך `$names`

```
Unset($nnames[0]);
```

- מחיקת המערך כולו נעשית ע"י:

```
Unset($names);
```




מערכים אסוציאטיביים (שיוכיים)

■ במערכים אלו המפתחות הם מחרוזות לדוגמא:

```
$names = array(  
"n1" => "Moshe",  
"n2" => "Shlomo",  
"n3" => "Gavriel");
```

שליפת הפריט הראשון:

```
Echo $names["n1"];
```

פלט: Moshe



מערך רב- ממדי

- מקובל לקנן מערכים בתוך מערכים וע"י כך ליצור מערכים רב ממדיים.

לדוגמא:

```
$names = array("n1" => array(),  
    "n2" => array(),  
    "n3" => array()  
);
```



הוספת פריטים למערכים הפנימיים

```
$names = array(  
  "n1" => array(  
    "name" => "Moshe",  
    "number" => 11  
  ),  
  "n2" => array(  
    "name" => "Shlomo",  
    "number" => 6  
  ),  
  "n3" => array(  
    "name" => "Gavriel",  
    "number" => 67  
  )  
);
```

```
echo $names["n2"]["number"];
```

כדי לגשת לפריט number במערך n2:



הוספה והדפסה במערך רב ממדי

■ הוספת פריט למערך רב ממדי

```
$names["n4"] = array("name" => "David", "number" => 9);
```

■ הדפסת המערך באמצעות print_r()

```
echo "<pre>";  
print_r($names);  
echo "</pre>";
```

פלט:

```
Array  
(  
  [n1] => Array  
    (  
      [name] => Moshe  
      [number] => 11  
    )  
  [n2] => Array  
    (  
      [name] => Shlomo  
      [number] => 6  
    )  
)
```

```
  [n3] => Array  
    (  
      [name] => Gavriel  
      [number] => 67  
    )  
  [n4] => Array  
    (  
      [name] => David  
      [number] => 9  
    )  
)
```

הדפסת מערך יחיד מתוך מערך רב ממדי

- ניתן להדפיס כל אחד מהמערכים הפנימיים
- לדוגמא:

```
echo "<pre>";  
print_r($names["n4"]);  
echo "</pre>";
```

פלט:

```
Array  
(  
    [name] => David  
    [number] => 9  
)
```

פונקציית each

■ מחזירה את המפתח והערך הנוכחי ומזיזה את המצביע קדימה

```
<?php
$people = array("Peter", "Joe", "Glenn", "Cleveland");
print_r (each($people));
print_r (each($people));
?>
```

פלט:

```
Array
(
    [1] => Peter
    [value] => Peter
    [0] => 0
    [key] => 0
)
Array
(
    [1] => Joe
    [value] => Joe
    [0] => 1
    [key] => 1
)
```



פונקציית diff

■ array_diff - פונקציה שמשווה בין שני מערכים ומחזירה את ההבדלים

```
<?php
$a1=array("a"=>"red","b"=>"green","c"=>"blue","d"=>"yellow");
$a2=array("e"=>"red","f"=>"green","g"=>"blue");

$result=array_diff($a1,$a2);
print_r($result);
?>
```

פלט:

```
Array ( [d] => yellow )
```



פונקציית unique

■ array_unique - מסירה ערכים כפולים מהמערך

```
<?php  
$a=array( "a"=>"red", "b"=>"green", "c"=>  
"red");  
print_r(array_unique($a));  
?>
```

פלט:

Array ([a] => red [b] => green)