

Server side scripting languages

8 Super global μεταβλητές στη PHP
Διαχείριση αρχείων

A large, stylized PHP logo consisting of the lowercase letters 'php' in a bold, black, sans-serif font, set against a blue oval background with a white outline. The logo is centered on the slide.

Β' Τεχνική Σχολή Λεμεσού
Σχολική χρονιά 2019 – 2020

Άριστος Πασιάς

Μεταβλητές Superglobals

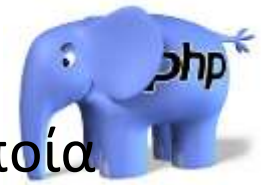


Είναι ενσωματωμένες μεταβλητές που είναι πάντα διαθέσιμες. Μπορείτε να έχετε πρόσβαση σε αυτές από οποιαδήποτε συνάρτηση, κλάση ή αρχείο χωρίς να χρειάζεται να κάνετε κάτι ιδιαίτερο

- `$GLOBALS`
- `$_SERVER`
- `$_REQUEST`
- `$_POST`
- `$_GET`
- `$_FILES`
- `$_ENV`
- `$_COOKIE`
- `$_SESSION`

Εδώ θα γίνει μια απλή αναφορά στις μεταβλητές αυτές. Τις κυριότερες από αυτές θα τις εξετάσουμε με περισσότερη προσοχή σε επόμενα μαθήματα.

PHP \$GLOBALS



Η \$GLOBALS είναι μια Superglobal μεταβλητή PHP, η οποία χρησιμοποιείται για την πρόσβαση σε GLOBAL μεταβλητές από οπουδήποτε στο σενάριο PHP. Η PHP αποθηκεύει όλες τις GLOBAL μεταβλητές σε έναν πίνακα που ονομάζεται \$GLOBALS [index]. Ο δείκτης περιέχει το όνομα της μεταβλητής.

```
<?php
$x = 75;
$y = 25;
function addition() {
    $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}
addition();
echo $z;
?>
```

PHP \$_SERVER



Η \$_SERVER είναι μια Superglobal μεταβλητή (Πίνακας) που περιέχει πληροφορίες σχετικά με τις κεφαλίδες, τις διαδρομές και τις θέσεις σεναρίων.

```
<?php
echo $_SERVER['PHP_SELF'];
echo "<br>"; // /php/demo_global_server.php
echo $_SERVER['SERVER_NAME'];
echo "<br>"; // www.w3schools.com
echo $_SERVER['HTTP_HOST'];
echo "<br>"; // www.w3schools.com
echo $_SERVER['SERVER_ADDR'];
echo "<br>"; // server IP Address
echo $_SERVER['HTTP_ACCEPT_CHARSET'];
echo "<br>"; // utf-8
echo $_SERVER['REMOTE_ADDR']; // Client IP Address
?>
```

Για περισσότερες επιλογές: <https://www.php.net/reserved.variables.server>

PHP \$_REQUEST, \$_POST, \$_GET



Χρησιμοποιούνται για τη συλλογή δεδομένων μετά την υποβολή μιας φόρμας HTML

```
<html>
<body>
<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  // collect value of input field
  $name = $_REQUEST['fname'];
  if (empty($name)) {
    echo "Name is empty";
  } else {
    echo $name;
  }
}
?>
</body></html>
```

Μια φόρμα με ένα πεδίο εισαγωγής και ένα κουμπί υποβολής. Κάνοντας κλικ στο "Υποβολή", τα δεδομένα φόρμας αποστέλλονται στο αρχείο που καθορίζεται στο χαρακτηριστικό δράσης της ετικέτας <form>. Σε αυτό το παράδειγμα, δείχνουμε αυτό το αρχείο για την επεξεργασία δεδομένων φόρμας. Αν θέλετε να χρησιμοποιήσετε άλλο αρχείο PHP, αντικαταστήστε το με το όνομα αρχείου.

\$_GET



Η μεταβλητή `$_GET` είναι ένας πίνακας από ονόματα μεταβλητών και τιμές που στέλνονται μέσω της μεθόδου HTTP GET. Χρησιμοποιείται για να συλλέξει τιμές από μια φόρμα που χρησιμοποιεί τη μέθοδο GET. Η πληροφορία που στέλνεται μέσω της HTTP GET μεθόδου είναι **ορατή** στον καθένα (φαίνεται στο address bar του browser!) και υπάρχει περιορισμός στην ποσότητα πληροφορίας που μπορεί να αποσταλεί (max 2000 χαρακτήρες).

Προσοχή! Δεν μπορούμε να στέλνουμε passwords ή άλλη ευαίσθητη πληροφορία με τη GET. Όμως όταν θέλουμε να θέσουμε έναν σελιδοδείκτη είναι χρήσιμη η GET.

\$_POST



Η μεταβλητή `$_POST` είναι ένας πίνακας από ονόματα μεταβλητών και τιμές που στέλνονται μέσω της μεθόδου HTTP POST.

Χρησιμοποιείται για να συλλέξει τιμές από μια φόρμα που χρησιμοποιεί τη μέθοδο POST. Η πληροφορία που στέλνεται μέσω της POST μεθόδου είναι **αόρατη** και δεν έχει όριο μεγέθους για την πληροφορία που αποστέλλεται.

(Καθορίζεται μέσω της μεταβλητής `post_max_size` στο αρχείο `php.ini`)

Φόρμες στην PHP



```
<html>
  <body>
    <form action="examp1.php" method="POST">
      Name: <input type="text" name="name"/>
      Age : <input type="text" name="age"/>
      <input type="submit"/>
    </form>
  </body>
</html>
```

POST

```
<html> <body> Welcome
  <?php
    echo $_POST["name"];
  ?>
    . <br/>
    You are
  <?php
    echo $_POST["age"];
  ?>
    years old.
</body></html>
```

\$_POST και \$_GET



Τόσο το GET όσο και το POST δημιουργούν έναν πίνακα (π.χ. `Array (key0 => value0, key1 => value1, key32=> value2, ...)`). Ο οποίος περιέχει ζεύγη κλειδιών / τιμών (key/value), όπου keys είναι τα ονόματα των στοιχείων ελέγχου και value είναι τα δεδομένα εισόδου από το χρήστη.

Τόσο το GET όσο και το POST αντιμετωπίζονται ως `$_GET` και `$_POST`. Αυτά είναι τα superglobals, που σημαίνει ότι είναι πάντα προσβάσιμα από οποιαδήποτε λειτουργία, τάξη ή αρχείο χωρίς να χρειάζεται να κάνετε κάτι ιδιαίτερο.

Το `$_GET` είναι ένας πίνακας μεταβλητών που μεταφέρονται στο τρέχον σενάριο μέσω των παραμέτρων διεύθυνσης URL.

Το `$_POST` είναι ένας πίνακας μεταβλητών που μεταφέρονται στο τρέχον σενάριο μέσω της μεθόδου POST HTTP.

\$_POST και \$_GET



GET

- μπορούν να αποθηκευτούν προσωρινά
- παραμένουν στο ιστορικό του προγράμματος περιήγησης
- μπορούν να επισημανθούν με σελιδοδείκτη
- δεν θα πρέπει ποτέ να χρησιμοποιούνται για μεταφορά ευαίσθητων δεδομένων
- έχουν περιορισμούς μήκους

POST

- Δεν αποθηκεύονται προσωρινά
- Δεν παραμένουν στο ιστορικό του προγράμματος περιήγησης
- Δεν μπορούν να επισημανθούν με σελιδοδείκτη
- Μπορούν να χρησιμοποιούνται για μεταφορά ευαίσθητων δεδομένων
- Δεν έχουν περιορισμούς μήκους

Compare GET vs. POST (<https://www.w3schools.com/>)

GET	POST	
BACK button/Reload	Harmless	Data will be re-submitted (the browser should alert the user that the data are about to be re-submitted)
Bookmarked	Can be bookmarked	Cannot be bookmarked
Cached	Can be cached	Not cached
Encoding type	application/x-www-form-urlencoded	application/x-www-form-urlencoded or multipart/form-data. Use multipart encoding for binary data
History	Parameters remain in browser history	Parameters are not saved in browser history
Restrictions on data length	Yes, when sending data, the GET method adds the data to the URL; and the length of a URL is limited (maximum URL length is 2048 characters)	No restrictions
Restrictions on data type	Only ASCII characters allowed	No restrictions. Binary data is also allowed
Security	GET is less secure compared to POST because data sent is part of the URL Never use GET when sending passwords or other sensitive information!	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs
Visibility	Data is visible to everyone in the URL	Data is not displayed in the URL