

299
300
301
302
303
304
305
306

```
$uniq_name = uniqid('');  
$path_fullsize = $thumbnails;  
$path_thumbnail = $thumbnails;  
  
// Resize and also save thumbnail  
$image = imagecreatefromjpeg($path_fullsize);
```

Κεφάλαιο 6°

Προγραμματισμός Εφαρμογών που εκτελούνται στον Εξυπηρετητή

Οι δυναμικές ιστοσελίδες, οι οποίες χρησιμοποιούν εφαρμογές που εκτελούνται στην πλευρά του εξυπηρετητή, α) λαμβάνουν τα δεδομένα που εισάγει ο χρήστης, β) τα επεξεργάζονται ή ανατρέχουν σε κάποια βάση δεδομένων η οποία είναι εγκατεστημένη στον εξυπηρετητή, γ) δημιουργούν μία προσαρμοσμένη ιστοσελίδα, την οποία και δ) παρουσιάζουν τελικά στο χρήστη. Το περιεχόμενο της παραγόμενης ιστοσελίδας εξαρτάται από την ερμηνεία του προγραμματιστικού σεναρίου. Σε αντίθεση με την τεχνολογία η οποία χρησιμοποιείται στον προγραμματισμό από την πλευρά του πελάτη όπως στην περίπτωση της JavaScript, ο προγραμματισμός από την πλευρά του εξυπηρετητή επιστρέφει προσαρμοσμένες ιστοσελίδες HTML. Οι ιστοσελίδες αυτές δεν περιέχουν άλλον κώδικα εκτός από τις ετικέτες HTML και μπορεί να εμφανίζονται σε όλους τους φυλλομετρητές, χωρίς να απαιτείται κάποια επιπλέον ρύθμιση σε αυτούς, όπως π.χ. η ενεργοποίηση της εκτέλεσης σεναρίων JavaScript.

Για την κατασκευή αυτών των προγραμμάτων μπορεί να χρησιμοποιηθούν διάφορες τεχνολογίες και γλώσσες σεναρίων. Οι πιο διαδεδομένες τεχνολογίες, οι οποίες αποτελούν αντικείμενο μελέτης του παρόντος κεφαλαίου, είναι οι: PHP, NodeJS και CGI (με PERL).



Η **PHP** (Hypertext Preprocessor) (php.net/) είναι μία ελεύθερα διαθέσιμη γλώσσα και διερμηνέας σεναρίων. Αρχικά χρησιμοποιήθηκε σε κεντρικούς υπολογιστές με περιβάλλον UNIX, αλλά μπορεί να χρησιμοποιηθεί και σε άλλα λειτουργικά συστήματα. Μία σελίδα HTML που περιλαμβάνει σεναρία PHP έχει τη χαρακτηριστική επέκταση ".php", ".php3", ".php4" ή ".phtml".

Η τεχνολογία **Node.js**, ή απλούστερα **Node** (nodejs.org), αποτελεί ένα περιβάλλον εκτέλεσης το οποίο επιτρέπει τη δημιουργία πολλών και διαφορετικών εργαλείων και εφαρμογών στην πλευρά του εξυπηρετητή με τη χρήση της Javascript. Δεν αποτελεί μια ακόμα γλώσσα προγραμματισμού, η οποία χρησιμοποιείται στην πλευρά

του εξυπηρετητή, αλλά μια πλατφόρμα/περιβάλλον ανάπτυξης εφαρμογών, οι οποίες έχουν δημιουργηθεί σε γλώσσα javascript. Η Javascript η οποία χρησιμοποιείται στη Node.js είναι σχεδιασμένη ώστε να λειτουργεί εκτός του περιβάλλοντος ενός φυλλομετρητή. Για το λόγο αυτό, έχουν 'αποκλειστεί' συναρτήσεις και δυνατότητές της, οι οποίες ήταν αυστηρά συνδεδεμένες με το περιβάλλον του φυλλομετρητή. Αντίθετα, έχουν προστεθεί άλλες, οι οποίες βοηθούν στη συνεργασία με το λειτουργικό σύστημα του εξυπηρετητή, π.χ. τη διαχείριση φακέλων και αρχείων ή ενέργειες του πρωτοκόλλου HTTP.

Η Node.js δημιουργήθηκε από τον Ryan Dahl το 2009, είναι ανοικτού κώδικα και σχεδιασμένη να συνεργάζεται με πολλαπλές πλατφόρμες λειτουργικών συστημάτων, όπως με: Linux, macOS, Solaris, FreeBSD, OpenBSD και Microsoft Windows (σε όλες τις νεότερες εκδόσεις). Τα αρχεία που παράγει έχουν επέκταση .js, όπως όλα τα αρχεία κώδικα javascript.

Η τεχνολογία **CGI** (Common Gateway Interface) είναι μία διαδεδομένη μέθοδος προγραμματισμού από την πλευρά του εξυπηρετητή. Αποτελεί ένα πρωτόκολλο επικοινωνίας ανάμεσα στους εξυπηρετητές του Ιστού (π.χ. Apache, IIS) και στα διάφορα εκτελέσιμα αρχεία και προγράμματα, τα οποία είναι εγκατεστημένα στον εξυπηρετητή (server). Το προϊόν αυτής της επικοινωνίας είναι κάποιες δυναμικά διαμορφωμένες ιστοσελίδες. Για την κατασκευή σεναρίων CGI μπορεί να χρησιμοποιηθούν πολλές γλώσσες γενικού προγραμματισμού, όπως η C, η C++ και η PERL, την οποία και θα μελετήσουμε. Η Perl (Practical Extraction και Report Language) (www.perl.org) είναι μία από τις πρώτες γλώσσες που χρησιμοποιήθηκαν για τον προγραμματισμό στον παγκόσμιο Ιστό. Ο αριθμός των έτοιμων βιβλιοθηκών προγραμμάτων που είναι διαθέσιμα αυτήν τη στιγμή την καθιστούν ένα ευπροσάρμοστο εργαλείο. Πολλοί άνθρωποι βρίσκουν δύσκολο τον προγραμματισμό σε Perl και αυτός είναι πιθανώς ο λόγος για τον οποίο έχουν γίνει δημοφιλέστερες άλλες, ευκολότερες και πιο εξειδικευμένες λύσεις. Εντούτοις, αποτελεί μια ολοκληρωμένη γλώσσα προγραμματισμού, η οποία μπορεί να χρησιμοποιηθεί και στην τεχνολογία CGI. Ο κώδικας της PERL δεν ενσωματώνεται σε αυτόν της σελίδας HTML, αλλά σε αυτόνομα αρχεία αποθηκευμένα στον εξυπηρετητή. Τα αρχεία αυτά καλούνται από το φυλλομετρητή του πελάτη μέσω της ιστοσελίδας και εκτελούνται στον εξυπηρετητή. Το αποτέλεσμα της εκτέλεσης δεν είναι τίποτα άλλο παρά κώδικας HTML, ο οποίος αποστέλλεται στον πελάτη και εμφανίζεται ως μια ιστοσελίδα. Τα αρχεία που περιέχουν προγράμματα σε Perl έχουν συνήθως επέκταση ".pl".

Η σύγκριση των τεχνολογιών είναι δύσκολη. Πολλοί προγραμματιστές προτιμούν τη μέθοδο που τυχάνει να ξέρουν καλύτερα ή τη μέθοδο που είναι ήδη καθιερωμένη στον οργανισμό στον οποίο εργάζονται ή σπουδάζουν. Η Perl μπορεί να απαιτεί περισσότερο χρόνο για την εκμάθησή της, σε σχέση με τις άλλες εναλλακτικές τεχνολογίες, εντούτοις, αποτελεί μία πλήρη γλώσσα προγραμματισμού για τον Ιστό και παρέχει μεγαλύτερη ευχέρεια υλοποίησης. Η PHP συνδυάζει μία ισχυρή γλώσσα

συγγραφής σεναρίων, με σχετική ευκολία χρήσης, η οποία μπορεί να είναι η προφανής επιλογή για κάποιον αρχάριο προγραμματιστή, που χρειάζεται γρήγορα αποτελέσματα χωρίς πρόσθετες δαπάνες λογισμικού. Η Node.js πραγματοποιεί την επιθυμία, πολλών ίσως, να υπάρχει javascript παντού. Ως εκ τούτου με μια γλώσσα μπορεί να αναπτυχθεί εφαρμογή η οποία θα αξιοποιεί τις δυνατότητες του περιβάλλοντος όχι μόνον του χρήστη (φυλλομετρητή), αλλά και του εξυπηρετητή.

Ο Πίνακας 5.1 παρουσιάζει μία απλή σύγκριση αυτών των τεχνολογιών.

Πίνακας 5.1 – Σύγκριση τεχνολογιών PHP, Node.js και Perl

	Perl	PHP	Node.js
Κόστος	Δωρεάν	Δωρεάν	Δωρεάν
Ενσωμάτωση σε σελίδα HTML	Όχι	Ναι	Όχι
Ευκολία χρήσης	+	+++	++
Διεπαφή με βάση δεδομένων	+++	++	+
Χειρισμός κειμένου	+++	+++	+++
Χειρισμός Εικόνων	++	++	++
Είναι Γλώσσα Προγραμματισμού	Ναι	Ναι	Όχι

6.1 Η γλώσσα σεναρίων PHP

Η PHP είναι μία ισχυρή γλώσσα για τη συγγραφή σεναρίων που συνεργάζεται πολύ καλά με την HTML. Αποτελεί δε, ένα πολύ καλό εργαλείο για το σχεδιασμό δυναμικών ιστοσελίδων.

Ένα από τα σημαντικότερα χαρακτηριστικά της PHP είναι ότι πρόκειται για μια γλώσσα ανοιχτού κώδικα και διανέμεται ελεύθερα, το οποίο σημαίνει ότι κάθε υλικό αναφοράς της μπορεί να εντοπιστεί εύκολα και να χρησιμοποιηθεί από τον καθένα. Επίσης, μπορεί να χρησιμοποιηθεί με ευκολία σε διάφορα λειτουργικά συστήματα.

Για τον έλεγχο των προγραμμάτων PHP απαιτείται ένας εξυπηρετητής ιστού ο οποίος υποστηρίζει τη γλώσσα.

6.1.1 Τι ακριβώς είναι η PHP

Η PHP αποτελεί ένα πρόγραμμα που μπορεί να εγκατασταθεί είτε ως ενσωματωμένο επιπλέον λογισμικό (module) σε έναν εξυπηρετητή ιστού (Web Server), είτε ως αυτόνομος εκτελέσιμος κώδικας. Προτιμάται η πρώτη περίπτωση, αφού έτσι είναι γρηγορότερη η εκτέλεση και δεν επιβαρύνεται ο εξυπηρετητής με επιπλέον

απαιτήσεις σε μνήμη και χώρο στον δίσκο. Μπορεί να χρησιμοποιηθεί σε πολλές εκδόσεις των Apache, Microsoft IIS, Netscape Enterprise Server και άλλων πακέτων.

Η σύνταξη της γλώσσας PHP είναι παρόμοια με αυτήν της PERL και της C. Στην πραγματικότητα η PHP ξεκίνησε, στα τέλη του 1994, ως μία εύκολη Perl. Γράφτηκε από τον Rasmus Lerdorf. Τα επόμενα χρόνια εξελίχθηκε στην PHP/FI 2.0 και στην PHP/FI. Το καλοκαίρι του 1997, οι Zeev Suraski και Andi Gutmans έφτιαξαν έναν καινούριο επεξεργαστή, οποίος οδήγησε στην PHP 3.0. Η έκδοση PHP 3.0 όρισε τη σύνταξη για τις εκδόσεις 3 και 4. Πρόσφατα κυκλοφόρησε η έκδοση 8.

Όταν κάποιος πελάτης (χρήστης) επισκεφτεί την ιστοσελίδα η οποία περιέχει κώδικα PHP, ο εξυπηρετητής εκτελεί τον κώδικα και αυτό που επιστρέφεται στο χρήστη είναι μία ιστοσελίδα HTML. Όλη τη δουλειά την κάνει ο εξυπηρετητής και όχι κάποιος φυλλομετρητής. Ο πελάτης (χρήστης) από την πλευρά του δεν χρειάζεται κάποιο πρόσθετο εργαλείο ή πρόγραμμα για να δει το αποτέλεσμα της PHP, επειδή όπως προαναφέρθηκε, το αποτέλεσμα φτάνει στο χρήστη ως κώδικας HTML.

Η PHP είναι μία γλώσσα για τη συγγραφή σεναρίων. Αυτό σημαίνει πως ο κώδικας, όπως και στην HTML, δεν χρειάζεται να μεταγλωττιστεί πριν χρησιμοποιηθεί.



Σημείωση: Επισκεφτείτε τη σελίδα www.php.net. Είναι ένα κέντρο ελέγχου, με υλικό αναφοράς, για τη γλώσσα και συμβουλές από χρήστες που χρησιμοποιούν και ασχολούνται με την PHP παγκόσμια.

Η PHP μπορεί να :

- Πάρει πληροφορία από φόρμες και να τη χρησιμοποιήσει με διάφορους τρόπους: να την αποθηκεύσει σε μία βάση δεδομένων, να δημιουργήσει εξαρτημένες/υποθετικές σελίδες που εξαρτώνται από τα περιεχόμενα της φόρμας, να τοποθετήσει cookies στο φυλλομετρητή του χρήστη και να στείλει ηλεκτρονικό ταχυδρομείο (e-mail).
- Πιστοποιήσει και να εντοπίσει χρήστες.
- Φιλοξενήσει συζητήσεις στην ιστοσελίδα.
- Προσαρμόσει την εμφάνιση μιας ιστοσελίδας στους διαφορετικούς φυλλομετρητές ή συσκευές, τους οποίους χρησιμοποιούν οι χρήστες.
- Δημοσιεύσει μία ολόκληρη ιστοσελίδα χρησιμοποιώντας μόνο μία φόρμα σχεδίου (server-side includes-style).
- Εξυπηρετήσει σελίδες XML και JSON.

6.1.2 Βασικοί κανόνες συγγραφής σεναρίων PHP

Πριν προχωρήσουμε παρακάτω στη μελέτη της PHP, θα χρειαστεί να ριξουμε μια γρήγορη ματιά στα στοιχεία της, ξεκινώντας από ένα παράδειγμα. Το παρακάτω πρόγραμμα περιέχεται στο αρχείο με όνομα "test.php". Όταν καλείται από έναν φυλλομετρητή, ο χρήστης απλά διαβάζει τη φράση: "Αυτό είναι ένα παράδειγμα!".

```
<?php
    print ("Αυτό είναι ένα παράδειγμα!");
?>
```

Όλα τα σεναρία PHP περικλείονται στις εντολές: <?php και ?>, οι οποίες είναι οι ετικέτες αρχής και τέλους του προγράμματος PHP. Το παραπάνω πρόγραμμα χρησιμοποιεί την εντολή `print` για να εκτυπώσει τη φράση στο "παράθυρο" του φυλλομετρητή.

Ο παραπάνω κώδικας γράφεται εντός της σελίδας HTML η οποία ονομάζεται `αρχείο1.php`, ως εξής:

```
<html>
<head><title> Παράδειγμα </title></head>
<body>
<p style="color: red;">Ο κώδικας PHP δημιουργεί μια
σελίδα που λέει:</p>
<p>
    <?php
        print ("Αυτό είναι ένα παράδειγμα!");
    ?>
</p>
</body>
</html>
```

Η HTML διαχειρίζεται τον κώδικα ως απλή HTML, αλλά οτιδήποτε βρίσκεται μέσα στα <?php και ?> θα τα επεξεργαστεί η PHP.



Σημείωση: Υπάρχουν στο Διαδίκτυο διάφοροι ιστότοποι όπου οι χρήστες μπορούν εύκολα να δοκιμάσουν απλό κώδικα PHP. Ένας από αυτούς είναι και ο <http://sandbox.onlinephpfunctions.com/>, όπου οι χρήστες μπορούν να 'δοκιμάσουν' τα παραδείγματα του βιβλίου.

Οι βασικοί κανόνες συγγραφής της PHP παρουσιάζονται στις ακόλουθες παραγράφους.

Ορισμός κώδικα

Κάθε τμήμα κώδικα PHP ξεκινάει με το "<?php" (ή σύντομα με το "<?" αν το αναγνωρίζει ο εξυπηρετητής).

Τελειώνουμε τον κώδικα PHP γράφοντας ">" στο τέλος του.

Ονομασία αρχείων

Για να λειτουργήσει ένα πρόγραμμα σε PHP, θα πρέπει, το αρχείο στο οποίο περιλαμβάνεται ο κώδικας ή οποιοδήποτε αρχείο καλείται μέσα στον κώδικα, να έχει την κατάληξη .php (παλιότερες εκδόσεις χρησιμοποιούσαν τις επεκτάσεις .php3 και .html). Όπως και στην HTML, τα αρχεία αποθηκεύονται ως απλό κείμενο ().



Σημείωση: Είναι δυνατόν ένα αρχείο να περιέχει κώδικα php αλλά να έχει διατηρήσει την επέκταση .html. Αυτό πραγματοποιείται με κατάλληλη διαφοροποίηση του αρχείου .htaccess του εξυπηρετητή Ιστού.

Εισαγωγή σχολίων

Στην PHP κάθε μονή γραμμή σχολίου ξεκινάει με το "//". Αν θέλουμε να γράψουμε ένα μπλοκ σχολίων ή να βγάλουμε προσωρινά εκτός επεξεργασίας κάποιο κομμάτι κώδικα μπορούμε να το βάλουμε εντός των συμβόλων "/*" και "*/":

```
<?php
// Αυτές οι γραμμές θα αγνοηθούν.
// Αποτελούν σχόλια
print ("Αυτό είναι ένα παράδειγμα!");
/*
και αυτές οι γραμμές θα αγνοηθούν. Αποτελούν ένα τμήμα
σχολίων.
*/
?>
```

Διαχωρισμός εντολών

Κάθε γραμμή εντολών τελειώνει με το σύμβολο του ερωτηματικού ";".

Όπως και στην HTML, η μορφοποίηση του κώδικα της PHP, δηλαδή τα κενά, οι αλλαγές γραμμής, κτλ., δεν επηρεάζει το αποτέλεσμα. Έτσι, ένα σύνολο από εντολές θα μπορούσαν να βρίσκονται όλες στην ίδια γραμμή, αρκεί να μην παραλείπονται τα ερωτηματικά. Εξάιρεση αποτελεί η μορφοποίηση που έχει εφαρμοστεί στα τμήματα του κώδικα τα οποία ορίζουν στο φυλλομετρητή πώς θα εμφανίσει την ιστοσελίδα.

Για παράδειγμα ο κώδικας:

```
<?php
    print ("Αυτό είναι ένα παράδειγμα!");
?>
```

θα έχει το ίδιο αποτέλεσμα στην οθόνη του χρήστη με τον παρακάτω:

```
<?php print ("Αυτό είναι ένα παράδειγμα!"); ?>
```

Συναρτήσεις εξόδου

Μέχρι στιγμής για την εκτύπωση κάποιου κειμένου στην οθόνη του φυλλομετρητή έχει χρησιμοποιηθεί η συνάρτηση `print`.

Η τυπική συνάρτηση φαίνεται ως εξής:

```
print ();
```

Η `print` είναι η συνάρτηση και τα δεδομένα που επεξεργάζεται βρίσκονται μέσα στις παρενθέσεις. Εναλλακτικά της `print` θα μπορούσε να χρησιμοποιηθεί η `echo`. Οι δυο συναρτήσεις μοιάζουν αρκετά, καθώς και οι δύο αποτελούν δομικά στοιχεία της γλώσσας PHP και χρησιμοποιούνται ώστε να εκτυπώνουν δεδομένα στην οθόνη. Ενώ και οι δύο είναι συναρτήσεις, εντούτοις αποτελούν εξαίρεση καθώς λειτουργούν με ή χωρίς τις παρενθέσεις. Οπότε, μπορεί να βρίσκονται με τη μορφή `print ()` και `echo ()` ή `print` και `echo`.

Η διαφορά τους έγκειται στον τρόπο λειτουργίας τους. Η `print` δέχεται μια παράμετρο ως δεδομένο εισόδου και επιστρέφει κάποια τιμή (που είναι πάντα ίση με 1). Ενώ η `echo` μπορεί να έχει περισσότερες της μιας παραμέτρους εισόδου και δεν επιστρέφει κάποια τιμή, το οποίο την καθιστά σχετικά γρηγορότερη. Αντίθετα, εξαιτίας της μη επιστροφής τιμής, η `echo` δεν μπορεί να χρησιμοποιηθεί ως τμήμα πολύπλοκων διαδικασιών.

Το παρακάτω παράδειγμα παρουσιάζει τη χρήση της `echo` με δύο παραμέτρους εισόδου. Οι παράμετροι διαχωρίζονται με κόμμα.

```
echo "κάποιο κείμενο 1", " κάποιο κείμενο 2";
```



Σημείωση: Η παρακάτω γραμμή κώδικα αποτελεί παράδειγμα πολύπλοκων διαδικασιών, στις οποίες δεν μπορεί να χρησιμοποιηθεί η συνάρτηση `echo`.

```
$b ? print "Είναι TRUE" : print "Είναι FALSE";
```

Το παραπάνω αποτελεί μια ερώτηση για την τιμή της μεταβλητής `$b`. Εάν είναι αληθής (*true*) τότε θα εκτυπωθεί το πρώτο λεκτικό, δηλαδή η φράση "Είναι TRUE". Στην αντίθετη περίπτωση, δηλαδή η μεταβλητή `$b` είναι ψευδής (*false*), τότε θα εκτυπωθεί το δεύτερο λεκτικό.

Η `print_r` αποτελεί μια λίγο διαφορετική συνάρτηση εξόδου. Η `print_r` δέχεται δύο παραμέτρους εισόδου. Η μία είναι η μεταβλητή για την οποία εκτυπώνει πληροφορίες, εύκολα κατανοητές από τον άνθρωπο. Αν η παράμετρος εισόδου της

είναι απλή μεταβλητή, τότε εκτυπώνει την τιμή της. Διαφορετικά εάν πρόκειται, δηλαδή, για μεταβλητή πίνακα, τότε εκτυπώνει τη θέση και την τιμή του κάθε στοιχείου του. Η δεύτερη παράμετρος εισόδου της `print_r` καθορίζει αν η συνάρτηση θα επιστρέψει κάποια τιμή ή απλά θα κάνει εκτύπωση στην οθόνη του χρήστη. Παραδείγματα της χρήσης της `print_r` θα δούμε όταν εξετάσουμε τις μεταβλητές και κυρίως τις μεταβλητές πίνακα.



Σημείωση: Η εντολή `<?=$variable?>` αποτελεί μια σύντομη εκδοχή της `echo` και εκτυπώνει την τιμή της μεταβλητής στην οθόνη του φυλλομετρητή.

Κλήσεις προγραμμάτων PHP

Υπάρχουν δύο διαφορετικές προσεγγίσεις στην εισαγωγή/ κλήση κώδικα PHP μέσα από ένα αρχείο HTML.

Μικρά τμήματα κώδικα `php` μπορεί να ενσωματωθούν με τις ετικέτες HTML όπως παρουσιάζεται στο παρακάτω παράδειγμα.

```
<head></head>
<body>
    Γεια, σήμερα είναι <?php echo date('l, d-m-Y'); ?>.
</body>
</html>
```

Το οποίο θα έχει ως αποτέλεσμα την εκτύπωση της φράσης «Γεια, σήμερα είναι Wednesday, 29-09-2021», χρησιμοποιώντας την τρέχουσα ημερομηνία του υπολογιστή.

Οτιδήποτε δεν περιέχεται εντός των συμβόλων του `<?php` και `?>` αγνοείται από τον επεξεργαστή της PHP. Στο παραπάνω παράδειγμα, το τμήμα του κώδικα PHP χρησιμοποιεί τη συνάρτηση `date` ώστε να μορφοποιήσει και να εκτυπώσει την τοπική ημερομηνία στην οθόνη του χρήστη. Ο συγκεκριμένος ορισμός της `date` θα εκτυπώσει το πλήρες όνομα της μέρας: δυο ψηφία για την ημέρα, δύο ψηφία για το μήνα και τέσσερα ψηφία για το έτος.

Εάν τα τμήματα του κώδικα τα οποία πρέπει να ενσωματωθούν είναι αρκετά, και ίσως πολύπλοκα, τότε υπάρχει η δυνατότητα να αποδοθεί όλη η ιστοσελίδα με εντολές PHP. Έτσι το παραπάνω παράδειγμα γίνεται:

```
<?php
    echo "<html>";
    echo "<head></head>";
    echo "<body>";
    echo "Γεια, σήμερα είναι ";
```



```

echo date('l, d-m-Y');
echo ".";
echo "</body>";
echo "</html>";
?>

```

Το αποτέλεσμα που θα επιστραφεί στο φυλλομετρητή του χρήστη θα είναι το κείμενο:

```

<html><head></head><body> Γεια, σήμερα είναι Wednesday, 29-09-2021. </body></html>

```

Το οποίο πρόκειται, τελικά, για μια ιστοσελίδα που θα εμφανίσει στην οθόνη του φυλλομετρητή τη φράση «Γεια, σήμερα είναι Wednesday, 29-09-2021.».

Γενικά είναι αποδεκτό ότι ο πρώτος τρόπος είναι γρηγορότερος. Εντούτοις, πολλοί υποστηρίζουν ότι ο κόπος της συντήρησης μιας ιστοσελίδας που έχει εμβόλιμα αρκετά τμήματα κώδικα PHP αντισταθμίζει το πλεονέκτημα της ταχύτητας εκτέλεσης. Τελικά, αποτελεί επιλογή του προγραμματιστή ποιον τρόπο θα ακολουθήσει.

Ως πιο προχωρημένη τεχνική θεωρείται η χρήση των εντολών `include` και `require` για την εισαγωγή ενός ολόκληρου αρχείου PHP μέσα σε ένα άλλο. Έτσι, έστω ότι το αρχείο με όνομα `test.php` περιέχει τα παρακάτω:

```

<?php echo date('l, d-m-Y'); ?>

```

Έστω ότι χρειάζεται να ενσωματώσουμε το αρχείο αυτό στην ιστοσελίδα, τότε θα πρέπει να γραφεί το παρακάτω:

```

<head></head>
<body>
Γεια, σήμερα είναι <?php include 'test.php' ?>.
</body>
</html>

```

Η εναλλακτικά:

```

<head></head>
<body>
Γεια, σήμερα είναι <?php require 'test.php' ?>.
</body>
</html>

```

Οι παραπάνω εντολές αποτελούν εξαιρετικά χρήσιμα εργαλεία. Για παράδειγμα, ο ορισμός ενός μενού πλοήγησης που επαναλαμβάνεται σε αρκετές ιστοσελίδες θα μπορούσε να ενσωματωθεί με έναν από αυτούς τους δύο τρόπους. Επιπλέον, έτσι μπορεί να ενσωματωθεί κώδικας που αριθμεί αρκετές γραμμές.

Η βασικότερη διαφορά ανάμεσα στην `include` και στην `require` αποτελεί το γεγονός ότι εάν το αρχείο (`test.php`) δεν βρεθεί κατά την κλήση της ιστοσελίδας, τότε η μεν πρώτη (`include`) θα εκτυπώσει προειδοποιητικό μήνυμα (`warning`), αλλά θα συνεχίσει να εκτελεί το πρόγραμμα, η δε δεύτερη (`require`) θα εκτυπώσει μήνυμα λάθους (`error`) και θα σταματήσει την εκτέλεση.

6.1.3 Ένα πρόγραμμα PHP

Γράφουμε το παρακάτω τμήμα κώδικα σε ένα αρχείο:

```
<html>
<body>
<?php
    echo ("Αυτό είναι ένα παράδειγμα!");
?>
</body>
</html>
```

Αποθηκεύουμε το αρχείο με κάποιο όνομα, π.χ. `test` (το όνομα του αρχείου δεν πρέπει να έχει κενά) με την κατάληξη `.php` στον κατάλογο `root` του εξυπηρετητή ιστού (για μια εγκατάσταση WAMP με το πακέτο XAMPP είναι συνήθως ο κατάλογος `"htdocs"` μέσα στον κατάλογο `"C:\xampp\htdocs"`).

Το επόμενο βήμα είναι να ανοιχθεί το αρχείο μέσα από το φυλλομετρητή. Επειδή για την εκτέλεση του κώδικα PHP απαιτείται ο εξυπηρετητής ιστού, θα πρέπει να ανοιχθεί το αρχείο μέσω ενός url, το οποίο θα προβάλλει το σωστό αρχείο στο φυλλομετρητή. Το url αυτό έχει τη μορφή `http://localhost/test.php`, όπου ο όρος `localhost` αντιπροσωπεύει το όνομα του τοπικού υπολογιστή. Εναλλακτικά, ο όρος `localhost` θα μπορούσε να αντικατασταθεί με τη διεύθυνση IP `127.0.0.1`, η οποία, συνηθέστερα, αντιπροσωπεύει τον τοπικό υπολογιστή (`localhost` διεύθυνση). Με τον τρόπο αυτό μπορεί κανείς να δει στο φυλλομετρητή του το παραπάνω παράδειγμα.

Εστω ότι υπάρχει πρόσβαση σε κάποιον εξωτερικό / απομακρυσμένο εξυπηρετητή. Σε αυτή την περίπτωση θα χρειαζόταν να μεταφορτωθεί το αρχείο του παραδείγματος, δηλαδή το `test.php`, με τη χρήση κάποιας εφαρμογής που 'τρέχει' το πρωτόκολλο FTP. Με αυτόν τον τρόπο θα μεταφερθεί το αρχείο `test.php` από το τοπικό μηχάνημα του προγραμματιστή στο απομακρυσμένο, δηλαδή στον εξυπηρετητή. Ο κατάλογος στον οποίο θα μεταφορτωνόταν το αρχείο θα ήταν ορισμένος στις ρυθμίσεις του εξυπηρετητή. Δηλαδή δεν θα ήταν απαραίτητα ο `htdocs`.

Μηνύματα λάθους

Αν δεν λειτουργήσει το παραπάνω παράδειγμα ή κάποιο πρόγραμμα που θα επιχειρήσουμε να τρέξουμε, θα δούμε ένα μήνυμα λάθους όπως αυτό: