

**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ**



**ΤΜΗΜΑ
ΠΛΗΡΟΦΟΡΙΚΗΣ &
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

5



Πανεπιστήμιο Ιωαννίνων

ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

2^ο Εξάμηνο 2019 – 2020



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Πρωτόκολλα & Αρχιτεκτονικές Δικτύων





PHP



Γενικά

Η PHP (Hypertext Preprocessor: Προεπεξεργαστής Υπερκειμένου) είναι μια γλώσσα προγραμματισμού για τη δημιουργία ιστοσελίδων με δυναμικό περιεχόμενο. Αυτό σημαίνει ότι ένα site αποκτά διαδραστικότητα π.χ: η εισαγωγή ενός username και ενός κωδικού σε μια σελίδα και η επαλήθευσή τους ώστε ο χρήστης να συνδεθεί σ' αυτό το site – για παράδειγμα σε ένα eshop!

Η PHP είναι τόσο ισχυρή που χρησιμοποιείται από το μεγαλύτερο σύστημα blog στο διαδίκτυο που είναι το WordPress αλλά και από το Facebook.

Ένα αρχείο PHP μπορεί να περιέχει κείμενο, κώδικα HTML, CSS, JavaScript και PHP, με τον κώδικα να εκτελείται στον server και το αποτέλεσμα να επιστρέφει στον browser ως απλές σελίδες HTML.



Τι άλλο μπορεί να κάνει η PHP

Η PHP μπορεί να δημιουργήσει, να ανοίξει, να κλείσει, να διαβάσει, να γράψει και να διαγράψει αρχεία σε έναν server. Μπορεί να στείλει και να λάβει cookies. Μπορεί επίσης να προσθέσει, να διαγράψει και να τροποποιήσει δεδομένα στη Βάση Δεδομένων σας. Ακόμη μπορεί να κρυπτογραφήσει δεδομένα.

Με την PHP δεν περιορίζεσαι στο να παράγεις HTML. Μπορείς να παράγεις εικόνες, αρχεία PDF, ακόμα και ταινίες Flash, καθώς και οποιοδήποτε κείμενο όπως XHTML και XML.



Γιατί να χρησιμοποιήσεις κανείς PHP

Η PHP τρέχει σε όλες τις πλατφόρμες όπως Windows, Linux, Unix, Mac OS X κ.λπ. Είναι συμβατή με σχεδόν όλους τους server που χρησιμοποιούνται σήμερα (Apache, IIS, κ.λπ.) Υποστηρίζει ένα ευρύ φάσμα απο Βάσεις Δεδομένων, είναι δωρεάν, είναι εύκολη να τη μάθει κανείς και τρέχει αποτελεσματικά στους server.



Λίγα ιστορικά στοιχεία

Η ανάπτυξη της PHP ξεκίνησε το 1994, όταν ο Rasmus Lerdorf έγραψε αρκετά CGI προγράμματα σε C, τα οποία χρησιμοποιούσε για να συντηρεί το προσωπικό του site.

Τα επέκτεινε για να δουλεύουν με φόρμες και να επικοινωνούν με βάσεις δεδομένων, και αυτή την υλοποίηση την ονόμασε Personal Home Page/Forms Interpreter – PHP/FI.

Πλέον PHP είναι το ακρωνύμιο για το Hypertext Preprocessor. Η γλώσσα εξελίσσεται διαρκώς και νέες εκδόσεις της βγαίνουν ανά τακτά χρονικά διαστήματα.

HTML – PHP

- ✓ Η HTML χρησιμοποιεί εντολές που ονομάζονται HTML tags δίνοντας εντολές στους Web browsers για το πώς να εμφανίζουν την κάθε ιστοσελίδα.
- ✓ Η PHP είναι μια γλώσσα προγραμματισμού η οποία επεκτείνει τις δυνατότητες της HTML δίνοντας τη δυνατότητα να δημιουργήσουμε δυναμικές ιστοσελίδες.

Τι είναι η RHP?

Πλεονεκτήματα της RHP

- Ευκολία στη χρήση
- Ανοιχτού κώδικα (Open Source)
- Υποστήριξη πολλαπλών πλατφόρμων
- Απόδοση
- Υποστήριξη Βάσεων Δεδομένων



Γράφοντας PHP

Για ν' αναγνωρίσει η PHP ότι ένα κομμάτι κώδικα μέσα σε μία σελίδα HTML είναι PHP πρέπει να περικλείεται ανάμεσα σε εισαγωγικά της μορφής `<?php ... ?>` ή της μορφής `<? ... ?>`. Προτείνεται το πρώτο για λόγους συμβατότητας.

Ο κώδικας PHP μπορεί να μπει σε οποιοδήποτε σημείο του κώδικα HTML (π.χ. εντός του head, εντός του body, μέσα σε ένα κελί ενός πίνακα που βρίσκεται στο body, κλπ.)

Παρά το γεγονός ότι ο κώδικας HTML και PHP συνυπάρχουν για τη σωστή εκτέλεση του προγράμματος PHP το αρχείο που περιέχει τον κώδικα (HTML και PHP) πρέπει να πάρει επέκταση `.php`

Σε κάθε περίπτωση, κάθε φορά που έχω έστω και μια γραμμή κώδικα php σε ένα αρχείο αυτό θα πρέπει να έχει επέκταση `.php` για να μπορεί να εκτελεστεί σωστά

Σχόλια

Στην PHP χρησιμοποιούμε `//` για σχόλια σε μια γραμμή και `/*` και `*/` για να εισάγουμε ένα μεγάλο μπλοκ σχολίων.

Όπως κάθε γλώσσα προγραμματισμού έτσι και η PHP αποτελείται από ένα σύνολο εντολών (όπως η εντολή `echo` η οποία είδαμε στο προηγούμενο παράδειγμα) τις οποίες χρησιμοποιούμε για να φτιάξουμε τα προγράμματα μας

Όλες οι εντολές στην PHP τερματίζουν με Ελληνικό ερωτηματικό για να ξεχωρίζουν από τις επόμενες. Μπορούμε μέσα στον κώδικα μας να εισάγουμε σχόλια (δηλαδή τμήμα κειμένου το οποίο στην ουσία εξηγεί τον κώδικα αλλά δεν εκτελείται – βοηθά απλά αυτόν που διαβάζει τον κώδικα να τον κατανοήσει ώστε να κάνει τις δικές του αλλαγές αν χρειαστεί)

Υπάρχουν δύο τρόποι να εισάγουμε σχόλια:

Σχόλια που εκτείνονται σε περισσότερες από μία γραμμές είναι οτιδήποτε περικλείεται από `/* ... */`

Σχόλια σε μια γραμμή είναι οτιδήποτε ακολουθεί το σύμβολο `//` μέχρι το τέλος της γραμμής


```
1 <html>
2   <head>
3     <title>PHP Example No 1</title>
4   </head>
5   <body>
6     <h2>
7       <?php
8         echo "Hello world"; // Auti einai mia entoli PHP. Prepei na termatizei me Elliniko erwtimatiko
9       ?>
10    </h2>
11  </body>
12 </html>
```

Ο κώδικας PHP (γραμμές 7-9) βρίσκεται ενσωματωμένος μέσα σε ένα tag <h2> εντός του body.

Υπάρχει μια εντολή (δηλαδή τμήμα κώδικα που τερματίζει με ελληνικό ερωτηματικό) μόνο κώδικα PHP η echo “Hello World”;

Η εντολή echo εμφανίζει στην οθόνη οτιδήποτε βρίσκεται εντός των εισαγωγικών που την ακολουθούν.


```

1 <html>
2   <head>
3     <title>PHP Example No 10</title>
4   </head>
5   <body>
6     <h2>
7       <?php
8         echo "Hello world</br>"; // Auti einai mia entoli PHP
9         $temp = "Today is ";    // Auti einai alli mia entoli
10        echo $temp . longdate(time()); // kai mia triti entoli
11        /*
12         Auto einai ena sxolio pou perilambanei polles grammes.
13         Ta sxolia den einai meros tou kwdika. H apousia tous
14         den allazei ti sumperifora tou porgarmmatos. Aplla kanei
15         ton kwdika mas pio euanagnwsto kai katanoito.
16        */
17
18        // Auto einai paradeigma sxoliou se mia grammi
19        function longdate($timestamp)
20        {
21            return date("l F jS Y", $timestamp);
22        }
23    ?>
24  </h2>
25 </body>
26 </html>

```

Ότι έχει χρώμα πράσινο είναι σχόλια
(δηλαδή δεν έχει καμία επίδραση στην
ουσιαστική λειτουργία του
προγράμματος)

Όπως φαίνεται στις γραμμές 8-10 τα
σχόλια γραμμής μπορούν να ακολουθούν
εντολές. Μπορούν επίσης να είναι σε
ξεχωριστή γραμμή όπως στην περίπτωση
της γραμμής 18

Στις γραμμές 11-16 έχουμε ένα σχόλιο
που εκτείνεται σε πολλαπλές γραμμές.

Όπως βλέπουμε ξεκινάει με τον
συνδυασμό συμβόλων /* και τερματίζει με
το συνδυασμό */



Οι μεταβλητές είναι χώρος αποθήκευσης τιμών κατά τη διάρκεια εκτέλεσης ενός προγράμματος.

Οι τιμές που αποθηκεύονται στις μεταβλητές κατά τη διάρκεια εκτέλεσης ενός προγράμματος αλλάζουν (για αυτό ονομάζονται και μεταβλητές)

Ο χώρος αποθήκευσης των τιμών προσδιορίζεται συμβολικά από το όνομα της μεταβλητής

Στην PHP τα ονόματα των μεταβλητών αρχίζουν υποχρεωτικά με το σύμβολο **\$**

Οι μεταβλητές παίρνουν συνήθως αριθμητικές τιμές είτε τιμές κειμένου (string).

Η ανάθεση τιμών σε μεταβλητές γίνεται με τον τελεστή **=**

Παράδειγμα

\$myNumber = 3; (Ανάθεση στην μεταβλητή με όνομα \$myNumber της αριθμητικής τιμής 3)

\$yourName = "Soula"; (Ανάθεση στην μεταβλητή με όνομα \$ yourName του κειμένου Soula

Για ανάθεση τιμών κειμένου το κείμενο πρέπει να εσωκλείεται σε εισαγωγικά **"**)



- Στην PHP μια μεταβλητή δεν χρειάζεται να δηλωθεί πριν χρησιμοποιηθεί.
- Η PHP μετατρέπει αυτόματα μια μεταβλητή στο σωστό τύπο δεδομένων, αναλόγως με το πως τη χρησιμοποιούμε.

Με τη συνάρτηση **print_r()** τυπώνει την τιμή μιας μεταβλητής σε human readable μορφή
var_dump() μπορούμε να τυπώσουμε τόσο την τιμή μιας μεταβλητής όσο και τον τύπο της.

```
fcomplex RCmul(float x, fcomplex a)
{
    fcomplex c;
    c.r = x*a.r;
    c.i = x*a.i;
    return c;
}

fcomplex Conjg(fcomplex z)
{
    fcomplex c;
    c.r = z.r;
    c.i = -z.i;
    return c;
}

fcomplex Cinv(fcomplex z)
{
    fcomplex c;
    float s = 1.0 / (z.r*z.r + z.i*z.i);
    c.r = z.r * s;
    c.i = -z.i * s;
    return c;
}

fcomplex Csqrt(fcomplex z)
{
    fcomplex c;
    float w;
    if ((z.r == 0.0) && (z.i == 0.0)) {
        c.r = 0.0;
        c.i = 0.0;
        return c;
    } else {
        w = sqrt((sqrt(z.r*z.r + z.i*z.i)));
        if (z.r >= 0.0) {
            c.r = w;
            c.i = z.i / (2.0*w);
        } else {
            c.r = -w;
            c.i = z.i / (2.0*w);
        }
        return c;
    }
}
```


Για την καλύτερη κατανόηση των μεταβλητών θεωρήστε τις ως «δοχεία».

Το όνομα τους αναγράφεται έξω από το δοχείο (π.χ. «λάδι», «αλάτι», «σπίρτα», «μηνύματα»)

Η τιμή τους είναι αυτό που περιέχεται μέσα στο δοχείο (ποσότητα λαδιού, αλατιού, αριθμός σπέρτων, χαρτάκι με γραμμένο κάποιο μήνυμα πάνω σε αυτό)

Κάθε φορά που χρησιμοποιούμε κάποιο δοχείο το περιεχόμενο του μεταβάλλεται (αλλάζει). Προσθέτουμε λάδι στο δοχείο λαδιού, αφαιρούμε αλάτι βάζοντας το στο φαγητό, χρησιμοποιούμε σπίρτα για άναμμα φωτιάς, αφαιρούμε το παλιό μήνυμα και βάζουμε κάποιο καινούργιο, κ.ο.κ.

Τα ονόματα των μεταβλητών μας καλό είναι να εξηγεί το περιεχόμενο τους.

Θα μπορούσαμε για παράδειγμα να ορίσουμε μια μεταβλητή με το όνομα \$age και να αποθηκεύαμε σε αυτή το όνομα κάποιου. Είναι επιτρεπτό αλλά θα ήταν καλύτερα να χρησιμοποιήσουμε για όνομα μεταβλητής το \$name.

Το πιο πάνω παράδειγμα είναι ισοδύναμο με το να γράψουμε σε ένα δοχείο όνομα «λάδι» και να προσθέσουμε μέσα ξύδι! Εμείς μπορεί να το ξέρουμε και να μην μπερδευόμαστε. Αν όμως κάποιος έρθει στο σπίτι μας τι θα νομίζει ότι περιέχει το δοχείο με όνομα «λάδι»;

Αντίστοιχα τα σωστά ονόματα μεταβλητών κάνουν τον κώδικα μας πιο ευανάγνωστο σε όποιον θέλει να το διαβάσει και να τον χρησιμοποιήσει.

```

1 <html>
2   <head>
3     <title>PHP Example No 3</title>
4   </head>
5   <body>
6     <h2>
7       <?php
8         $minima = "Hello there!";           // Αναδеси stin metabliti $minima tou keimenou "Hello there!"
9         $imerominia = date("D, d M Y H:i:s"); // Αναδеси stin metabliti $imerominia tis trexousas imerominias
10        // H entoli date einai mia enswmatwmani stin PHP entoli i opoia mas dinei tin trexousa imeromina
11        $num = 3;                           // Αναδеси stin metabliti $num tis ari8mitkis timis 3
12
13        // Ektypwsi twm timwn twm metablitwn
14        echo $minima;                       // emfanizetai to periexomeno tis metablitis $minima
15        echo "<br>";                         // stin epomeni grammi
16        echo "Date and time: ";
17        echo $imerominia;                  // emfanizei to periexomeno tis metablitis $imerominia
18        echo "<br>";                         // stin epomeni grammi
19        echo "The content of variable num is ";
20        echo $num;                         // emfanizei to periexomeno tis metablitis $num
21      ?>
22    </h2>
23  </body>
24 </html>

```

Στο πιο πάνω παράδειγμα έχουμε τρεις μεταβλητές (\$minima, \$imerominia, \$num) στις οποίες αναθέτουμε

διαφορετικές τιμές

Η ανάθεση γίνεται με τη βοήθεια του τελεστή = (βλέπε γραμμές 8, 9, 11)

Οι τιμές των μεταβλητών εμφανίζονται στην οθόνη μας με τη βοήθεια εντολών echo (βλέπε γραμμές 14, 17, 20).

**strlen()**

Προέρχεται από τα αρχικά των λέξεων string length και δείχνει το μήκος ενός αλφαριθμητικού

str_word_count()

Μετράει τον αριθμό των λέξεων

strrev()

Εμφανίζει την λέξη ανάποδα

strpos ()

Αναζητά ένα συγκεκριμένο κείμενο μέσα σε μια συμβολοσειρά. Εάν βρεθεί, η συνάρτηση επιστρέφει τη θέση χαρακτήρων

str_replace ()

Αντικατάσταση κειμένου εντός συμβολοσειράς

Τελεστές

Οι τελεστές επιδρούν στο περιεχόμενο των μεταβλητών για να αλλάξουν τις τιμές τους είναι ελέγχουν το περιεχόμενο τους.

Υπάρχουν τέσσερις κατηγορίες τελεστών στην PHP:

Αριθμητικοί (οι γνωστοί μας αριθμητικοί τελεστές πρόσθεσης (+), αφαίρεσης (-), πολλαπλασιασμού (*), διαίρεσης (/), καθώς και το υπόλοιπο ακέραιας (%), η αύξηση (++), και η μείωση (--))

Ανάθεσης (το περιεχόμενο να γίνει ίσο με (=), όσο ήταν συν αυτό που βρίσκεται δεξιά από τον τελεστή(+=), όσο ήταν μείον αυτό που βρίσκεται δεξιά από τον τελεστή(-=) κ.ο.κ)

Σύγκρισης (το περιεχόμενο της μεταβλητής είναι ίσο με αυτό που βρίσκεται στα δεξιά του τελεστή (==)?, το περιεχόμενο της μεταβλητής είναι μεγαλύτερο από αυτό που βρίσκεται στα δεξιά του τελεστή (>)?, κ.ο.κ)

Λογικοί (ΚΑΙ -AND (&&), 'Η -OR (||), 'ΟΧΙ -NOT (!))

Οι αριθμητικοί και οι τελεστές ανάθεσης αλλάζουν τις τιμές των μεταβλητών ενώ οι τελεστές σύγκρισης και οι λογικοί τελεστές έχουν ως αποτέλεσμα true (1) ή false (0) (δηλαδή σωστό ή λάθος).

Παράδειγμα:

Έστω οι μεταβλητές με τις ακόλουθες αρχικές τιμές:

$\$x = 14$; $\$y = 4$;

Το αποτέλεσμα των πιο κάτω εντολών θα είναι αυτό στην παρένθεση

$\$z = \$x + \$y$; (η τιμή της μεταβλητής $\$z$ θα γίνει ίση με το άθροισμα των τιμών των μεταβλητών $\$x$ και $\$y$ δηλαδή 18)

$\$z = \$x - \$y$; (η τιμή της μεταβλητής $\$z$ θα γίνει ίση με 10)

$\$z = \$x / \$y$; (η τιμή της μεταβλητής $\$z$ θα γίνει ίση με 3.5)

$\$z = \$x * \$y$; (η τιμή της μεταβλητής $\$z$ θα γίνει ίση με 56)

$\$z = \$x \% \$y$; (η τιμή της μεταβλητής $\$z$ θα γίνει ίση με 2 δηλαδή το υπόλοιπο της διαίρεσης $14 / 4 \Rightarrow$ πηλίκο 3, υπόλοιπο 2)



Έστω οι μεταβλητές με τις ακόλουθες αρχικές τιμές:

`$x = 14; $name = "Soula "; surname = "Soulidou";`

Το αποτέλεσμα των πιο κάτω εντολών θα είναι αυτό στην παρένθεση

`$x = 13.2;` (η τιμή της μεταβλητής `$x` θα γίνει ίση με 13.2)

`$name = "Soula Soulidou";` (η τιμή της μεταβλητής `$ name` θα γίνει ίση με Soula Soulidou)

`$x +=5` (η τιμή της μεταβλητής `$x` θα γίνει όσο ήταν πριν (14) συν 5, άρα 19)

`$x -=3` (η τιμή της μεταβλητής `$x` θα γίνει όσο ήταν πριν (14) μείον 3, άρα 11)

`$x *=4` (η τιμή της μεταβλητής `$x` θα γίνει 4 φορές όσο ήταν πριν (14), άρα 56)

`$x /=4` (η τιμή της μεταβλητής `$x` θα γίνει όσο ήταν πριν (14) διαιρεμένο διά 4, άρα 7)

`$fullName = $name . $surname;` (ο τελεστής `.` χρησιμοποιείται μόνο για strings και τα συνενώνει (τα βάζει σε παράθεση) – επομένως η μεταβλητή `$fullName` θα πάρει την τιμή Soula Soulidou)

Προσοχή! Οι πιο πάνω εντολές θεωρούμε ότι εκτελούνται ανεξάρτητα και όχι σε διαδοχή.

Αν τις είχαμε βάλει σε σειρά μέσα σε ένα πρόγραμμά τα αποτελέσματα θα ήταν τελείως διαφορετικά διότι έχουμε κάνει πολλαπλές αλλαγές τιμών.



Τελεστές Σύγκρισης:

Οι τελεστές σύγκρισης χρησιμοποιούνται συνήθως ως συνθήκες στο πλαίσιο εντολών διακλάδωσης (if, while, κ.ο.κ)

Σύνταξη εντολής if:

if (συνθήκη) do something else do something else;

Η ερμηνεία της πιο πάνω εντολής είναι: Αν η συνθήκη είναι σωστή –αν ισχύει δηλαδή εκτέλεσε τις εντολές «do something» αλλιώς εκτέλεσε τις εντολές «do something else»

Έστω οι μεταβλητές με τις ακόλουθες αρχικές τιμές:

`$x = 14; $y=4;`

Το αποτέλεσμα της εκτύπωσης προκύπτει αφού γίνει έλεγχος της συνθήκης (όπου χρησιμοποιούνται οι τελεστές σύγκρισης)

`if ($x==$y) echo "x and y are equal";`

(η συνθήκη (`$x==$y`) δίνει λάθος (false, 0) διότι οι τιμές των μεταβλητών `$x` και `$y` είναι διαφορετικές. Αν ήταν ίσες θα εκτυπωνόταν το μήνυμα "x and y are equal" εξαιτίας της εκτέλεσης της εντολής `echo "x and y are equal"`. Επειδή η συνθήκη (`$x==$y`) δεν ισχύει η εντολή `echo "x and y are equal"` δεν εκτελείται)



Το αποτέλεσμα της εκτύπωσης προκύπτει αφού γίνει έλεγχος της συνθήκης (όπου χρησιμοποιούνται οι τελεστές σύγκρισης)

if (\$x>\$y) echo "x is greater than y" else echo "x is smaller than y";

(η συνθήκη (\$x>\$y) είναι ορθή (true, 1) διότι η τιμή της μεταβλητής \$x είναι μεγαλύτερη από την τιμή της μεταβλητής \$y επομένως θα εκτυπωθεί το μήνυμα x is greater than y διότι θα εκτελεστεί η εντολή echo "x is greater than y")

if (\$x<\$y) echo "x is smaller than y" else echo "y is smaller than x";

(η συνθήκη (\$x<\$y) δίνει λάθος (false, 0) διότι η τιμή της μεταβλητής \$x είναι μεγαλύτερη από την τιμή της μεταβλητής \$y επομένως θα εκτελεστεί η εντολή μετά το **else (echo "y is smaller than x")** και θα εκτυπωθεί το μήνυμα y is smaller than x)

if (\$x!=\$y) echo "x and y are different" else echo "x and y are equal";

(η συνθήκη (\$x!=\$y) είναι ορθή (true, 1) διότι οι τιμές των μεταβλητών \$x και \$y είναι διαφορετικές (ο τελεστής != σημαίνει διαφορετικό) επομένως θα εκτελεστεί η εντολή **echo "x and y are different"** και θα εκτυπωθεί το μήνυμα x and y are different)



Λογικοί Τελεστές

Οι λογικοί τελεστές χρησιμοποιούνται συνήθως για συνδυάζουν συνθήκες στο πλαίσιο εντολών διακλάδωσης (if, while, κ.ο.κ)

if (συνθήκη 1) && (συνθήκη 2) do something else do something else;

Η ερμηνεία της πιο πάνω εντολής είναι: Αν ισχύουν οι συνθήκες 1 και 2 τότε εκτέλεσε τις εντολές «do something» αλλιώς εκτέλεσε τις εντολές «do something else»

Ο λογικός τελεστής && είναι ισοδύναμος με το «ΚΑΙ»

Έστω οι μεταβλητές με τις ακόλουθες αρχικές τιμές:

\$x = 14; \$y=4; \$z=8

Το αποτέλεσμα της εκτύπωσης προκύπτει αφού γίνει έλεγχος του συνδυασμού των συνθηκών

if (\$x>\$y) && (\$x>\$z) echo "x is the greatest among x, y, z";

(ο τελεστής && απαιτεί να ισχύουν και οι δύο συνθήκες (\$x>\$y), (\$x>\$z) ταυτόχρονα. Αν ισχύουν (στη συγκεκριμένη περίπτωση ισχύουν) θα εκτελεστεί η εντολή echo "x is the greatest among x, y, z" και θα εκτυπωθεί το μήνυμα "x is the greatest among x, y, z" αν δεν ισχύουν ταυτόχρονα δεν θα εμφανιστεί κανένα μήνυμα)

Το αποτέλεσμα της εκτύπωσης προκύπτει αφού γίνει έλεγχος του συνδυασμού των συνθηκών

`if ($z>$x) || ($z>$y) echo "z is not the smallest among x, y, z";`

(ο τελεστής `||` απαιτεί να ισχύουν κάποια από τις δύο συνθήκες $(\$z>\$x)$, $(\$z>\$y)$ ή και οι δύο. Αν ισχύει κάποια από τις δύο (στη συγκεκριμένη περίπτωση ισχύει η $(\$z>\$y)$) θα εκτελεστεί η εντολή `echo "z is not the smallest among x, y, z"` και θα εκτυπωθεί το μήνυμα "z is not the smallest among x, y, z". Αν δεν ισχύει καμία από τις δύο δεν θα εμφανιστεί κανένα μήνυμα)

`if ($z>$x) xor ($z>$y) echo "z is the median of x, y, z";`

(ο τελεστής `xor` απαιτεί να ισχύει μόνο μια από τις δύο συνθήκες $(\$z>\$x)$, $(\$z>\$y)$ αλλά όχι και οι δύο. Αν ισχύει κάποια από τις δύο (στη συγκεκριμένη περίπτωση ισχύει η $(\$z>\$y)$) θα εκτελεστεί η εντολή `echo "z is the median of x, y, z"` και θα εκτυπωθεί το μήνυμα "z is the median of x, y, z". Αν δεν ισχύει καμία ή αν ισχύουν και οι δύο δεν θα εμφανιστεί κανένα μήνυμα)

Συναρτήσεις

Σε κάποιες περιπτώσεις μια σειρά από εντολές χρειάζεται να εκτελεστεί πολλές φορές μέσα στο πρόγραμμά μας (π.χ. σε ένα πρόγραμμα ταξινόμησης χρειάζεται πολλές φορές να βρούμε τον μεγαλύτερο από δύο αριθμούς)

Σε τέτοιες περιπτώσεις αντί να επαναλαμβάνουμε τον κώδικα πολλές φορές κατασκευάζουμε μια συνάρτηση την οποία καλούμε κάθε φορά που χρειάζεται

Οι συναρτήσεις ορίζονται με τη χρήση της κωδικής λέξης `function` έχουν ένα όνομα (όνομα συνάρτησης), δέχονται ορίσματα και επιστρέφουν μια τιμή:

```
function maxOfTwo($a,$b)
```

```
{ $mx=0;
```

```
  if ($a>$b) $mx=$a else $mx=$b;
```

```
  return $mx; }
```

Στο πιο πάνω παράδειγμα το όνομα της συνάρτησης είναι `maxOfTwo` δέχεται δύο ορίσματα (τις μεταβλητές `$a,$b`) και επιστρέφει την τιμή της μεταβλητής στην οποία έχουμε αποθηκεύσει την μεγαλύτερη από τις τιμές των μεταβλητών `$a,$b`.

Πιο κάτω δίνεται ο τρόπος με τον οποίο καλούμε τη συνάρτηση `maxOfTwo` μέσα στον κώδικα PHP

```
$c = maxOfTwo(9,13); (η μεταβλητή $c θα πάρει την τιμή 13)
```

```
$c = maxOfTwo($op1, $op2); (η μεταβλητή $c θα πάρει την μεγαλύτερη από τις  
τιμές των μεταβλητών $op1, $op2)
```




```
1 <html>
2 <head>
3 <title>PHP Example No 4</title>
4 </head>
5 <body>
6 <h2>
7 <?php
8     function maxOfTwo($a,$b)
9     { $mx=0;
10      if ($a>$b) $mx = $a; else $mx = $b;
11      return $mx; }
12
13     echo "Start computing the max<br>";
14     $num1 = 13; $num2 = 18; $num3 = 10; $num4 = 9;
15     // Bres ton megalytero an zeygi apo tous ari8mous
16     $mx1 = maxOfTwo($num1,$num2);
17     $mx2 = maxOfTwo($num3,$num4);
18     // compare the maximums
19     $max = maxOfTwo($mx1,$mx2);
20     echo "The maximum of $num1, $num2, $num3, $num4 is $max";
21     ?>
22 </h2>
23 </body>
24 </html>
```

Τι κάνει το πρόγραμμα;



```
1 <html>
2 <head>
3 <title>PHP Example No 4</title>
4 </head>
5 <body>
6 <h2>
7 <?php
8     function maxOfTwo($a,$b)
9     { $mx=0;
10      if ($a>$b) $mx = $a; else $mx = $b;
11      return $mx; }
12
13     echo "Start computing the max<br>";
14     $num1 = 13; $num2 = 18; $num3 = 10; $num4 = 9;
15     // Bres ton megalytero an zeygi apo tous arithmous
16     $mx1 = maxOfTwo($num1,$num2);
17     $mx2 = maxOfTwo($num3,$num4);
18     // compare the maximums
19     $max = maxOfTwo($mx1,$mx2);
20     echo "The maximum of $num1, $num2, $num3, $num4 is $max";
21 ?>
22 </h2>
23 </body>
24 </html>
```

Η συνάρτηση **maxOfTwo** καλείται τρεις φορές (γραμμές 16, 17 και 19) για τον υπολογισμό του μέγιστου από τέσσερις αριθμούς

Στη γραμμή 16 βρίσκουμε το μέγιστο από τους δύο πρώτους αριθμούς και το αναθέτουμε στη μεταβλητή \$mx1

Στη γραμμή 17 βρίσκουμε το μέγιστο από τους δύο τελευταίους αριθμούς και το αναθέτουμε στη μεταβλητή \$mx2

Στη γραμμή 19 βρίσκουμε το μέγιστο από τα \$mx1, \$mx2 και τα αναθέτουμε στη μεταβλητή \$max η οποία και είναι η τελική μέγιστη τιμή



Τα προγράμματα PHP έχουν πραγματικά ενδιαφέρον όταν επεξεργάζονται δεδομένα που εισάγει ο χρήστης.

Ο απλούστερος τρόπος για χειρισμό τέτοιων περιπτώσεων είναι εισαγωγή των στοιχείων μέσω μιας φόρμας σε ένα αρχείο html και η επεξεργασία τους και η προβολή των αποτελεσμάτων γίνεται από ένα αρχείο PHP

Να φτιάξετε μια φόρμα στην οποία ο χρήστης να εισάγει 4 αριθμούς και το πρόγραμμα να επιστρέφει τον μικρότερο μαζί με ένα σχετικό μήνυμα

```
fcomplex RCmul(float x, fcomplex a)
{
    fcomplex c;
    c.r=x*a.r;
    if ((z.i == 0.0)) {
        c.i=x*a.i;
        return c;
    }
    c.i=0.0;
    c.i=0.0;
}

fcomplex Cinv( fcomplex z)
{
    fcomplex c;
    float s = 1.0 / (z.r*z.r + z.i*z.i);
    if (z.r >= 0.0) {
        c.r = z.r * s;
        c.i = -z.i * s;
    } else {
        w = sqrt((sqrt(z.r*z.r + z.i*z.i) - z.r));
        if (z.r >= 0.0) {
            c.r=w;
            c.i=-z.i/(2.0*w);
        }
    }
}
```



Αρχείο εισαγωγής των στοιχείων

```
1 <html>
2   <head>
3     <title>PHP Example No 5</title>
4     <link href="WBTEexamples.css" rel="stylesheet" type="text/css">
5   </head>
6
7   <body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0" >
8     <h1>Παράδειγμα υπολογισμού σε στοιχεία που εισάγει ο χρήστης</h1>
9     <p>Παρακαλώ εισάγετε τους αριθμούς στα πιο κάτω πεδία</p>
10
11     <!-- This is the area of the form -->
12     <form action="lec10_ex05.php" method="post">
13       Number 1: <input type="text" name="num1" />
14       Number 2: <input type="text" name="num2" />
15       Number 3: <input type="text" name="num3" />
16       Number 4: <input type="text" name="num4" />
17       <input type="submit" value = "Compute minimum"/>
18     </form>
19   </body>
20 </html>
```




Αρχείο επεξεργασίας των στοιχείων

```
1 <html>
2 <head>
3 <title>PHP Example No 5</title>
4 <link href="WBTEexamples.css" rel="stylesheet" type="text/css">
5 </head>
6
7 <body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0" >
8 <h1> Αποτελέσματα της επεξεργασίας</h1>
9
10 <!-- (This is the area to put your code) -->
11 <?php
12 //Define the function that computes the minimum of two numbers
13 function minOfTwo($a,$b)
14 { $minim=0;
15   if ($a<$b) $minim =$a; else $minim =$b;
16   return $minim; }
17
18 $num1 = $_POST["num1"]; // Take the first number form the form
19 $num2 = $_POST["num2"]; // Take the second number form the form
20 $num3 = $_POST["num3"]; // Take the third number form the form
21 $num4 = $_POST["num4"]; // Take the fourth number form the form
22
23 $mn1 = minOfTwo($num1,$num2);
24 $mn2 = minOfTwo($num3,$num4);
25 // compare the minimums
26 $mn = minOfTwo($mn1,$mn2);
27 echo "The minimum of $num1, $num2, $num3, $num4 is $mn";
28 ?>
29 <!-- (End of editable area) -->
30 </body>
31 </html>
```

5