PROGRAMARE IN PHP

Seminar introductiv

Limbajul de programare PHP- scurta prezentare

- PHP este un limbaj de programare care se dovedeste o unealta puternica pentru e crea pagini Web dinamice si interactive
- PHP este gratuit si larg folosit ca o alternativa eficienta la competitori ca Microsoft ASP.
- In acest seminar PHP veti invata despre PHP, si cum sa executati scripturi PHP pe server
- creat de Rasmus Lerdorf in 1995 sub numele *Personal Home Page* (PHP)

Cunostiinte prealabile (**Pre-requisite)**

- Este foarte util sa aveti cunostiinte prealabile de:
- • HTML/XHTML
- • JavaScript

Ce este PHP ?

- PHP ca denumire actuala provine din: PHP:**H**ypertext **P**reprocessor

  PHP este un limbaj de scripting executat pe server , ca si ASP
- PHP suporta baze de date ca:MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc.
- PHP este un software open source
- PHP e gratuit pentru download si utilizare

Ce este un fisier PHP ?

- • fisierele PHP pot contine text, tag-uri HTML si script-uri
- • fisierele PHP sunt returnate in browser ca HTML
- • fisierele PHP au extensie ".php", ".php3", sau ".phtml"

De ce PHP?

- • PHP ruleaza pe diferite platforme (Windows, Linux, Unix, etc.)
- • PHP e compatibil cu servere Web ca: Apache, IIS, etc.
- • PHP e gratuit de download-at de la resursa oficiala PHP : www.php.net
- • PHP e usor de invatat si ruleaza eficient pe server

De unde incepem?
Instalarea PHP si serverul Web

- Pentru acces la server web cu suport PHP, puteti:
- • Instalati Apache (sau IIS) pe propriul server, instalati PHP, si ca baza de date MySQL: de exemplu pachetul wamp
- Sau gasiti un provider de web hosting cu suport de PHP si MySQL

Instalarea PHP si serverul Web(continuare)

- Pachetul wamp va fi folosit in continuare pentru platforme windows:
- Etape :
- 1)Download-ati Microsoft Visual C++ 2010 SP1 Redistributable Package (x86) de la Official Microsoft Download Center
- 2)Download-ati WampServer de pe SourceForge.net :
- WampServer Free software downloads at SourceForge.net;
- http://www.wampserver.com/en/#download-wrapper
- Aici folosim wamp 2.2E pentru 32 biti , ce include PHP 5.3; MySql 5.5.24 ; Apache 2.2.22

Sintaxa PHP

- **Sintaxa de Baza PHP**
- codul PHP e executat pe server, si rezultatul HTML este trimis la browser.
- un bloc de scripting PHP intotdeauna incepe cu **<?php** si se termina cu **?>**. Un bloc de scripting PHP poate fi plasat oriunde in document.
- Pe servers cu suport stenografic se poate starta un bloc de scripting cu<? si termina cu ?>.
- pentru maximum de compatibilitate, se recomanda forma standard
- <?php
- ?>

Sintaxa PHP (continuare)

- Un fisier PHP normal contine tag-uri HTML, ca un fisier HTML, si cod de scripting PHP.
- Dam un examplu de script simplu PHP ce trimite textul "Hello World" la browser:
- <html>
- <body>
- <?php echo "Hello World"; ?>
- </body>
- </html>

Sintaxa PHP (continuare)

- Fiecare linie de cod in PHP trebuie sa se termine cu o semicoloana.
- semicoloana e un separator si e folosita pentru a distinge un set de instructiuni de altul .
- Sunt doua instructiuni de baza pentru afisare text in PHP: **echo** si **print**.
- In examplul de mai sus am folosit echo pentru afisarea textului "Hello World".
- **Nota:** fisierul trebuie sa aibe extensie .php.
- daca fisierul are o extensie .html, codul PHP nu va fi executat.

Comentarii in PHP

- In PHP, folosim // pentru o linie de comentariu sau /* si */ pentru un bloc de comentarii mai mare.
- Exemplu:
- <html>
- <body>
- <?php
- //
- This is a comment
- /*
- This is a comment block
- 123
- 456
- */
- ?>
- </body>
- </html>

Variable PHP

- o variabila e utilizata pentru a memora o informatie ca siruri de tip text, numere sau matrici de date.
- Cand o variabla e declarata, poate fi folosita oriunde in script.
- variablele in PHP starteaza cu semnul $.
- Corecta declarare a unei variable in PHP este:
- $var_name = value;

Exemplu de variabile PHP

- cream o variabila ce contine un sir, si o variabila ce contine un numar:
- <?php
- $txt="Hello World!";
- $x=16;
- ? >

Reguli de denumire a variabilelor PHP

- un nume de variabila trebuie sa inceapa cu o litera sau cu underscore "_"
- • un nume de variabila poate doar contine charactere alpha-numerice si underscor (a-z, A-Z, 0-9, and _)
- • un nume de variabila nu trebuie sa contina spatii.
- daca un nume de variabila are mai mult de un cuvant, trebuie sa fie separat cu un underscore ($my_string), sau cu capitalizare ($myString)

Variable PHP de tip sir

- variable tip sir (string) sunt folosite pentru memorare si manipulare de text.
- variable tip sir sunt folosite pentru valori ce contin caractere .
- Vom vedea cele mai comune functii si operatori folositi pentru manipulare de siruri in PHP.
- dupa crearea sirului il vom manipula. Un sir poate fi folosit direct in functie sau memorat intr-o variabila.
- Iata un script PHP ce atribuie textul "Hello World" la o variabila sir numita $txt:

Variable PHP de tip sir exemplu

- <?php
- $txt="Hello World";
- echo $txt;
- ?>
- rezultatul rularii codului va fi:
- Hello World

Diferite functii si operatori pentru manipularea sirurilor.

- **Operator** de **Concatenare**
- E doar un operator de tip sir in PHP.
- operatorul de concatenare (.) e folosit pentru a pune doua valori de sir impreuna.
- Pentru a concatena doua variable sir impreuna folositi operatorul de concatenare :

Operator de Concatenare exemplu

- <?php
- $txt1=“Buna ziua!";
- $txt2=“Ce zi frumoasa!";
- echo $txt1 . " " . $txt2;
- ?>
- Rezultatul este
- Buna ziua! Ce zi frumoasa!

Operator de Concatenare exemplu-continuare

- Daca privim la cod vedem ca am folosit am folosit operatorul de concatenare de doua ori pentru ca a trebuit sa inseram un al treilea sir (un spatiu), pentru a separa cele doua siruri.

functia strlen()

- **functia** strlen() intoarce lungimea sirului.
- Gasim lungimea unui sir:
- <?php
- echo strlen("Hello world!");
- ?>
- Rezulta
- 12
- lungimea unui sir e adesea folosita in bucle de program sau alte functii, cand e important sa stim cand se termina sirul. (i.e. in bucla , vrem sa oprim bucla dupa ultimul caracter din sir).

Functia strpos()

- **functia** strpos() cauta caracter intr-un sir.
- Daca se gaseste potrivire se va returna pozitia primei potriviri , iar daca nu, se intoarce FALSE.
- Cautam sirul "world" in sirul nostru:
- <?php
- echo strpos("Hello world!","world");
- ?>
- Rezultatul : 6
- pozitia sirului "world" in sirul nostru e pozitia 6. The reason that it is 6 (and not 7), is that the first position in the string is 0, and not 1.

cautare in siruri cautare a tuturor aparitiilor unui sir folosind Offset

- Exemplu :

```
<?php
$numberedString =
"12345678901234567890123456789012345678890";
$fivePos = strpos($numberedString, "5");
echo "pozitia lui  5 in sir este  $fivePos";
$fivePos2 = strpos($numberedString, "5", $fivePos + 1);
echo "<br />pozitia celui de-al doilea 5 este  $fivePos2";
?>
Rezultatul este:
pozitia lui 5 in sir este 4
pozitia celui de-al doilea 5 este 14
```

Functii pentru capitalizarea in siruri php:
Conversia unui sir in Upper Case –: functia strtoupper

- <?php
- $originalString = "String Capitalizare 1234";
- $upperCase = strtoupper($originalString);
- echo "Old string - $originalString
";
- echo "New String - $upperCase";
- ?>
- Rezultat:
- Old string - String Capitalizare 1234
  New String - STRING CAPITALIZARE 1234

Conversia unui sir in Lower Case – : functia strtolower

- <?php
- $originalString = "String Capitalizare 1234";
- $lowerCase = strtolower($originalString);
- echo "Old string - $originalString
";
- echo "New String - $lowerCase";
- ?>
- Rezultat:
- Old string - String Capitalizare 1234
  New String - string capitalizare 1234

Capitalizarea primei Litere –: functia ucwords

- <?php
- $titleString = " titlu : hELP";
- $ucTitleString = ucwords($titleString);
- echo "Old title - $titleString
";
- echo "New title - $ucTitleString";
- ?>
- Rezultat:
- Old title - titlu : hELP
  New title - Titlu : HELP

Referinta Completa siruri PHP

- Pentru o completa referinta a tuturor functiilor de tip sir, http://www.w3schools.com/php/php_ref_string.asp
- si contine o scurta descriere, si exemple pentru fiecare functie

Variabile de tip matrici(array) in PHP

- **PHP : matrici**
- **matricile** memoreaza multiple valori in o singura variabila.
- Pentru o lista de obiecte (o lista de nume de masini), putem avea
- $cars1="Saab";
- $cars2="Volvo";
- $cars3="BMW";
- Daca vrem o bucla care sa gaseasca una specifica si pentru mai multe obiecte folosim array(matrice)

. Se acceseaza valorile prin referirea la un nume de array.

- fiecare element din array are un index si asa e usor accesat.
- In PHP, sunt trei tipuri de array-uri:
- • **Numerice** – cu index numeric
- • **Asociative** – fiecare cheie ID key e asociata cu o valoare
- • **Multidimensionala**- un array continand unul sau mai multe array-uri

Matrici(Array-uri) Numerice

- Acestea stocheaza fiecare element din array cu un index numeric.
- Sunt doua methode de creare :
- 1. index automatic asignat (index starteaza de la 0): $cars=array("Saab","Volvo","BMW","Toyota");
- 2. atribuire de index manual:
- $cars[0]="Saab"; $cars[1]="Volvo"; $cars[2]="BMW"; $cars[3]="Toyota
- In examplu se acceseaza valorile variabilei prin referirea la nume array si index: <?php
- $cars[0]="Saab"; $cars[1]="Volvo"; $cars[2]="BMW"; $cars[3]="Toyota";
- echo $cars[0] . " and " . $cars[1] . " are Swedish cars.";
- ?>
- Rezultat: Saab and Volvo are Swedish cars.

Array-uri Asociative

- Aici fiecarei chei ID I se asociaza o valoare.
- Cand stocam date despre valori cu nume specifice, un array numeric nu e intotdeauna cel mai potrivit.
- cu array asociativ putem folosi valori ca si chei si sa le atribuim valori.
- **Exemplul 1**
- Se atribuie varste la persoane diferite:
- $ages = array("Peter"=>32, "Quagmire"=>30, "Joe"=>34);

Array-uri Asociative(continuare)

- **Exemplul 2**
- La fel ca exemplul 1, dar in alt mod:
- $ages['Peter'] = "32";
- $ages['Quagmire'] = "30";
- $ages['Joe'] = "34";
- Cheile ID folosite in script astfel :
- <?php
- $ages['Peter'] = "32";
- $ages['Quagmire'] = "30";
- $ages['Joe'] = "34";
- echo "Peter are " . $ages['Peter'] . " ani.";
- ?>
- rezultat:
- Peter are 32 ani.

Array-uri Multidimensionale

- fiecare element in array-ul principal poate fi tot un array. Si fiecare element in sub-array poate fi un array, etc.
- **Exemplu**

- cream un array multidimensional, cu chei ID automatic atribuite :
- $familii = array ("Griffin"=>array (
- "Peter",
- "Lois",
- "Megan"),
- "Quagmire"=>array (
- "Glenn“
-),
- "Brown"=>array (
- "Cleveland",
- "Loretta",
- "Junior")

-);

Array-uri Multidimensionale continuare

- Array-ul arata astfel explicit cu ID-urile elementelor
- : Array ([Griffin] => Array (
- [0] => Peter
- [1] => Lois
- [2] => Megan
-)
- [Quagmire] => Array (
- 0] => Glenn
-)
- [Brown] => Array (
- [0] => Cleveland
- [1] => Loretta
- [2] => Junior))

Array-uri Multidimensionale continuare

- **Exemplul 2**
- Afisam o valoare din array:
- echo “e " . $familii['Griffin'][2] . " membru al familiei Griffin?";
- Se va afisa :
- e Megan membru al familiei Griffin?

- **Referinta Completa de PHP Array** http://www.w3schools.com/php/php_ref_array.asp contin scurte descrieri si exemple pentru fiecare functie

Operatori in PHP

- Operatorii sunt folositi pentru manipularea sau efectuare de operatii asupra variabilelor si valorilor.
- Exemple "=" , operator de atribuire si "." de concatenare .

Operatori aritmetici

Operator		**Exemple**
+	Adunare	2 + 4
-	Scadere	6 - 2
*	Multiplicare	5 * 3
/	impartire	15 / 3
%	Modulo	43 % 10

Operatori aritmetici exemple

- $adunare = 2 + 4;
- $scadere= 6 - 2;
- $multiplicare = 5 * 3;
- $impartire = 15 / 3;
- $modulo = 5 % 2;
- echo " adunare : 2 + 4 = ".$adunare."
";
- echo " scadere : 6 - 2 = ".$subtraction."
";
- echo " multiplicare: 5 * 3 = ".$multiplication."
";
- echo " impartire: 15 / 3 = ".$division."
";
- echo " modulo: 5 % 2 = " . $modulus
- . ". Modulo e restul de la impartire in acest caz 5 / 2, cu rest 1.";

Operatori de atribuire

- Pentru a seta o variabla egala cu o valoare ori seta o variabila cu valoarea altei variable.
- cu "=", ori caracterul egal. Exemplu:
  - • $my_var = 4;
  - • $another_var = $my_var ;
- $my_var si $another_var contin valoarea 4. Atribuirea poate fi folosita in conjunctie cu operatorii aritmetici.

operatori combinati: aritmetici si de atribuire

Operator	Exemplu	Operatiune Echivalenta
• +=	$x += 2;	$x = $x + 2;
• -=	$x -= 4;	$x = $x - 4;
• *=	$x *= 3;	$x = $x * 3;
• /=	$x /= 2;	$x = $x / 2;
• %=	$x %= 5;	$x = $x % 5;
• .=	$my_str.="hello";	$my_str = $my_str . "hello";

operatori de comparare

Operator	English	Exemplu	Rezultat
• ==	Equal To	$x == $y	false
• !=	Not Equal To	$x != $y	true
• <	Less Than	$x < $y	true
• >	Greater Than	$x > $y	false
• <=	Less Than or Equal To	$x <= $y	true
• >=	Greater Than or Equal To	$x >= $y	false

operatori logici

Operator	**Descriere**	**Exemple**
&&	and	x=6 y=3 (x < 10 && y > 1) rezulta true
||	or	x=6 y=3 (x==5 || y==5) rezulta false
!	not	x=6 y=3 !(x==y) rezulta true

Instructiuni PHP if...else

- Instructiunile Conditionale fac diferite actiuni bazate pe diferite conditii.
- **instructiunea if**
- Se executa un anumit cod daca o conditie e adevarata
- <html>
- <body>
- <?php
- $d=date("D");
- if ($d=="Fri") echo “Sa aveti un weekend placut!";
- ?>
- </body>
- </html>

instructiunea if...else

- **if....else** executa un cod daca o conditie e true si alt cod daca o conditie e false.
- **Sintaxa**
- if (*condition*) *code de executat pentru conditie true;* else *cod de executat pentru conditie false;*
- exemplu : afisez "Have a nice weekend!" daca ziua curenta e vineri(Friday), altfel afisez "Have a nice day!":
- <html>
- <body>
- <?php
- $d=date("D");
- if ($d=="Fri") echo "Have a nice weekend!";
- else echo "Have a nice day!";
- ?>
- </body>
- </html>
-

instructiunea if...elseif....else

- **if....elseif...else**
- selecteaza unul sau mai multe blocuri de cod pentru executie.
- if (*conditie*) *cod de executat daca conditia e true;* elseif (*conditie*) *cod de executat daca conditia e true;* else *cod de executat daca conditia e false;*
- **Exemplu**
- Se afiseaza "Have a nice weekend!" daca ziua curenta e vineri(Friday), si "Have a nice Sunday!" daca ziua curent e Sambata.
- altfel afisez "Have a nice day!":
- <html>
- <body>
- <?php
- $d=date("D");
- if ($d=="Fri") echo "Have a nice weekend!";
- elseif ($d=="Sun") echo "Have a nice Sunday!";
- else echo "Have a nice day!"; ?>
- </body>
- </html>

Instructiunea switch

- selecteaza unul sau mai multe blocuri de cod pentru executie.
- **Sintaxa:**
- switch (*n*) {case *label1:*
- *cod de executat daca n=label1;*
- break;
- case *label2:*
- *cod de executat daca n=label2;*
- break;
- default:
- *Cod de executat daca n e diferit de label1 si de label2;* }

Instructiunea switch

- **Exemplu**
- <html>
- <body>
- <?php
- $x=3;
- switch ($x) {case 1: echo "Numarul 1";
- break; case 2: echo "Numarul 2";
- break;
- case 3: echo "Numarul 3";
- break;
- default: echo "Nici un numar intre 1 si 3"; }?>
- </body> </html>

Bucle in PHP
bucle while

- bucle while
- executa un bloc de cod cand o conditie e true.
- While(conditie){*cod de executat*; }
- Exemplu:
- <html>
- <body>
- <?php
- $i=1;
- while($i<=5)
- {echo “numarul este " . $i . "
"; $i++; }
- ?>
- </body>
- </html>
- Bucla incepe cu i=1. bucla ruleaza cat timp i e mai mic sau egal cu 5. i creste cu 1 de fiecare data candleaza bucla.

bucle while

- Rezultat
- numarul este 1
- numarul este 2
- numarul este 3
- numarul este 4
- numarul este 5

bucle do....while

- Va executa intotdeauna blocul de cod o data , apoi va verifica conditia, si repeta bucla cat timp conditia e true.
- Sintaxa:
- do{*cod de executat;* } while (*conditie*);
- **Exemplu**
- Bucla incepe cu i=1. apoi incrementeaza i cu 1, si afiseaza ceva. Apoi conditia e verificata, si bucla va continua sa ruleze cat timp i e mai mic, ori egal cu 5:
- <html>
- <body>
- <?php
- $i=1;
- do{$i++; echo “numarul e " . $i . "
"; } while ($i<=5);
- ?>
- </body>
- </html>
- numarul e 2
- numarul e 3
- numarul e 4
- numarul e 5
- numarul e 6

bucle for

- Se foloseste cand se stie de cate ori trebuie rulata bucla
- for (*init; conditie; increment*) {*cod de executat;* }
- Exemplu
- Bucla starteaza cu i=1. Bucla continua sa ruleze pentru i mai mic sau egal to 5.
- i creste cu 1 la fiecare rulare a buclei
- <html>
- <body>
- <?php
- for ($i=1; $i<=5; $i++)
- {echo “nu " . $i . "Numarul este "; }
- ?>
- </body>
- </html>

bucle for

- Rezultat:
- Nu Numarul este 1
- Nu Numarul este 2
- Nu Numarul este 3
- Nu Numarul este 4
- Nu Numarul este 5

bucle foreach

- foreach loop e bucla pentru matrici(array-uri).
- foreach ($*array* as $*value*) {*cod de executat;* }
- Pentru fiecare iteratie de bucla, valoarea elementului curent din array e asignat la $value (si pointer de array e mutat cu unu) – asa ca la urmatoarea iteratie de bucla, te uiti la urmatoarea valoare din array.
- **Exemplu**
- O bucla ce afiseaza valorile unui array:
- <html>
- <body>
- <?php
- $x=array(“unu",“doi","trei");
- foreach ($x as $value) {echo $value . "
"; }
- ?>
- </body>
- </html>
- rezultat:unu doi trei

Functii PHP

Functii din PHP de tip BUILT-IN

- Puterea reala a PHP provine din functiiile sale.
- In PHP, sunt mai mult de 700 functii built-in (incluse).
- Referinta completa pentru acestea este la
- http://www.w3schools.com/php/default.asp
- o functie va fi executata de un apel al functiei

Crearea unei Functii PHP

- .
- **Sintaxa**
- functia *Numefunctie*() {*cod de executat*; }
- PHP functii indrumar:
- • dati functiei un nume ce reflecta ce face functia • numele functiei poate incepe cu o litera ori underscore (nu cu un numar)
- <html>
- <body>
- <?php
- function writeName()
- {
- echo “Mircea";
- }
- echo “prenumele meu este "; writeName();
- ?>
- </body>
- </html>

-

adaugare de parametrii in functii

- Parametrii sunt specificati dupa numele functiei, intre parenteze
- **Exemplul 1**
- scrie diferite prenume, cu acelasi nume de familie:
- <html>
- <body>
- <?php
- function writeName($fname)
- {
- echo $fname . " Ionescu”.
";
- }
- echo "My name is "; writeName(“Ion");
- echo "My sister's name is "; writeName(“Geta");
- echo "My brother's name is "; writeName(“Adrian");
- ?>
- </body>
- </html>
- Rezultat:
- My name is Ion Ionescu. My sister's name is Geta Ionescu. My brother's name is Adrian Ionescu.

adaugare de parametrii in functii

- **Exemplul 2**
- functie cu doi parametrii:
- <html>
- <body>
- <?php
- function writeName($fname,$punctuation)
- {
- echo $fname . " Ionescu" . $punctuation . "
";
- }echo "My name is "; writeName(“Ion",".");
- echo "My sister's name is ";
- writeName(“Geta","!");
- echo "My brother's name is ";
- writeName(“Adrian","?");
- ?>
- </body> </html>
- My name is Ion Ionescu. My sister's name is Geta Ionescu! My brother's name is Adrian Ionescu?

returnare de valori din functii PHP

- Pentru ca o functie sa returneze o valoare se foloseste instructiunea return.
- **Exemplu**
- <html>
- <body>
- <?php
- function add($x,$y)
- {$total=$x+$y;
- return $total;
- }
- echo "1 + 16 = " . add(1,16); ?>
- </body>
- </html>
- Rezultat:
- 1 + 16 = 17
-

Creare de Formulare in PHP
validarea formularelor

- variabilele PHP $_GET and $_POST preiau informatia din formulare, ca user input. Inputul de la utilizatori trebuie validat in browser de fiecare data cand e posibil. (cu client script). Validarea Browserului e mai rapida si reduce incarcarea serverului.
- consideram validarea serverului daca user input va fi inseart intr-o baza de date.
- functia (built-in) $_GET colecteaza valorile dintr-un formular trimise cu metoda="get".
- Informatia trimisa de la un formular cu metoda GET e vizibila tuturor (e display-ata in browser's pe address bar) si are limitari ale cantitatii de informatie de trimis (max. 100 caractere).
- **Exemplu** <form action="welcome.php" method="get"> Name: <input type="text" name="fname" /> Age: <input type="text" name="age" /> <input type="submit" /> </form>
- Cand utilizatorul face click pe butonul "Submit", URL-ul trimis la server poate fi ceva de genul: http://www.w3schools.com/welcome.php?fname=Peter&age=37

Creare de Formulare in PHP
validarea formularelor

- Fisierul "welcome.php" poate folosi acum functia $_GET pentru a prelua date din formular (numele campurilor formularului vor fi automat cheile din array-ul $_GET):
- Welcome <?php echo $_GET["fname"]; ?>.
- You are <?php echo $_GET["age"]; ?> years old!
- **Cand folosim metoda="get"?**
- **Cand folosim metoda="get** " in formulare HTML, toate numele variabilelor si valorile sunt afisate in URL.
- **Nota:** aceasta metoda nu se foloseste cand se trimit parole sau alte informatii sending tip variabilele apar in URL, e posibil sa facem bookmark aceasta pagina, uneori poate fi util.
- **Nota:** metoda aceasta nu este buna pentru variabile ce depasesc 100 caractere.

Functia $_POST

- Functia built-in $_POST colecteaza valori dintr-un formular trimise cu metoda="post".
- Informatia fiind invisibila altora si nu are limite cantitative.
- Nota: este 8 Mb max pentru POST by default (dar se poate modifica setand post_max_size in fisierul php.ini).
- **Exemplu**
- <form action="welcome.php" method="post"> Name: <input type="text" name="fname" /> Age: <input type="text" name="age" /> <input type="submit" /> </form>
- la click pe "Submit", URL arata asa
- http://www.w3schools.com/welcome.php
- "welcome.php" poate folosi $_POST pentru a colecta date din formular (numele campurilor formularului vor fi automat chei in array-ul $_POST):
- Welcome <?php echo $_POST["fname"]; ?>!
- You are <?php echo $_POST["age"]; ?> years old.
- Deoarece variabilele nu sunt afisate in URL, nu e posibil sa bookmark-am pagina.
-

Functia PHP $_REQUEST

- Functia PHP built-in $_REQUEST are continutul $_GET, $_POST, si $_COOKIE.
- $_REQUEST poate fi folosita pentru a colecta date din formular trimise atat cu metoda GET cat si cu metoda POST.
- Exemplu
- Welcome <?php echo $_REQUEST["fname"]; ?>!
 You are <?php echo $_REQUEST["age"]; ?> years old.

creare de fisier PHP – functia fopen

- $ourFileName = "testFile.txt";
- $ourFileHandle = fopen($ourFileName, 'w') or die("can't open file");
- fclose($ourFileHandle);
- Aici se creaza fisierul testFile.txt pentru scriere(‘w’)
- *file handle*, permite sa manipulam fisierul. salvam file handle in variabila *$ourFileHandle*

creare de fisier PHP – functia fopen

- **Un fisier poate fi deschis pentru :**
- **Read: 'r'** citire. Pointerul de fisier e la inceputul fisierului.
- • **Write: 'w' –**scriere,datele sunt sterse si se scriu date de la inceputul fisierului(se numeste si trunchierea fisierului)
- . pointerul e la inceputul fisierului.
- • **Append: 'a' :** pointerul e la sfarsitul fisierului unde se adauga datele

Functia fclose

- Dupa ce un fisier a fost inchis cu fclose e imposibil de a fi citit, scris sau apendat decat daca este din nou deschis cu fopen .

PHP – citirea fisierelor: Functia fread

- $myFile = "testFile.txt";
- $fh = fopen($myFile, 'r');
- $theData = fread($fh, 5);

citeste 5 caractere

- fclose($fh);
- echo $theData;
- Rezultat:
- Flopp

PHP – citirea fisierelor: Functia fgets

- $myFile = "testFile.txt";
- $fh = fopen($myFile, 'r');
- $theData = fgets($fh); citeste o linie din fisier
- fclose($fh);
- echo $theData;
- **testFile.txt :Continut:**

Floppy Jalopy

Scrierea in fisiere –functia fwrite

- $myFile = "testFile.txt";
- $fh = fopen($myFile, 'w') or die("can't open file");
- $stringData = "Bobby Bopper\n";
- fwrite($fh, $stringData);
- $stringData = "Tracy Tanner\n";
- fwrite($fh, $stringData);
- fclose($fh);
- Rezultat:
- **testFile.txt :**
- Bobby Bopper
- Tracy Tanner

PHP –scriere fisiere : suprascriere (Overwriting)

- $myFile = "testFile.txt";
- $fh = fopen($myFile, 'w') or die("can't open file");
- $stringData = "Floppy Jalopy\n";
- fwrite($fh, $stringData);
- $stringData = "Pointy Pinto\n";
- fwrite($fh, $stringData);
- fclose($fh);
- Rezultat:
- **testFile.txt :**
- Floppy Jalopy
- Pointy Pinto

stergerea fisierelor
functia **Unlink**

- $myFile = "testFile.txt";
- $fh = fopen($myFile, 'w') or die("can't open file");
- fclose($fh);
- $myFile = "testFile.txt";
- unlink($myFile);

PHP –scriere in fisier : Apendare Date

- **PHP Code:**
  - $myFile = "testFile.txt";
  - $fh = fopen($myFile, 'a') or die("can't open file");
  - $stringData = "New Stuff 1\n";
  - fwrite($fh, $stringData);
  - $stringData = "New Stuff 2\n";
  - fwrite($fh, $stringData);
- fclose($fh);

PHP –scriere in fisier : Apendare Date

- Continut al fisier testFile.txt :
- Floppy Jalopy
- Pointy Pinto
- New Stuff 1
- New Stuff 2

PHP - Upload de fisier in formular HTML

- Intr-un formular HTML existent adaugam codul PHP:
- <form enctype="multipart/form-data" action="uploader.php" method="POST">
- <input type="hidden" name="MAX_FILE_SIZE" value="100000" />
- Choose a file to upload: <input name="uploadedfile" type="file" />
- <input type="submit" value="Upload File" />
- </form>
- **Rezultat:**

- **Choose a file to upload: Upload File**
- Dupa ce se face click pe submit, data se posteaza pe server si user-ul se redirecteaza la *uploader.php*. Acest fisier PHP va procesa datele din formular.

uploader.php

- // unde va fi plasat fisierul
- $target_path = "uploads/";
- /* Adaug numele original al fisierului filename la tinta(target path).
- Resultatul este "uploads/filename.extension" */
- $target_path = $target_path . basename($_FILES['uploadedfile']['name']);
- $_FILES['uploadedfile']['tmp_name'];

PHP - File Upload: move_uploaded_file on

- $target_path = "uploads/";
- $target_path = $target_path . basename($_FILES['uploadedfile']['name']);
- if(move_uploaded_file($_FILES['uploadedfile']['tmp_name'], $target_path)) {
- echo "The file ". basename($_FILES['uploadedfile']['name']).
- " fisierul uploaded!";
- } else{
- echo "eroare la upload , re-incercati!";

PHP Date - Robust Data calendaristica si ora

- Functia PHP's *date()*
- <?php
- echo date("m/d/y");
- ?>
- Rezultat:
- 02/27/10

PHP date-timestamp

- timestamp– numar de secunde din
- ianuarie 1, 1970 la 00:00
- *mktime* functie ce creaza un timestamp pentru ziua de maine

```
<?php
$tomorrow = mktime(0, 0, 0, date("m"), date("d")+1, date("y"));
echo “maine este ".date("m/d/y", $tomorrow);

?>
```

- Rezultat:

- maine este 02/28/10

PHP Date - Reference

- **Formatare timestamp:optiuni**:

- **Important Full Date and Time:**
  - • **r**: Displays the full date, time and timezone offset. It is equivalent to manually entering *date("D, d M Y H:i:s O")*

- **Time:**
  - • **a**: am or pm depending on the time
  - • **A**: AM or PM depending on the time
  - • **g**: Hour without leading zeroes. Values are 1 through 12.
  - • **G**: Hour in 24-hour format without leading zeroes. Values are 0 through 23.
  - • **h**: Hour with leading zeroes. Values 01 through 12.
  - • **H**: Hour in 24-hour format with leading zeroes. Values 00 through 23.
  - • **i**: Minute with leading zeroes. Values 00 through 59.
  - • **s**: Seconds with leading zeroes. Values 00 through 59.

- **Day:**
  - • **d**: Day of the month with leading zeroes. Values are 01 through 31.
  - • **j**: Day of the month without leading zeroes. Values 1 through 31
  - • **D**: Day of the week abbreviations. Sun through Sat
  - • **l**: Day of the week. Values Sunday through Saturday
  - • **w**: Day of the week without leading zeroes. Values 0 through 6.
  - • **z**: Day of the year without leading zeroes. Values 0 through 365.

PHP Date - Reference

- **Month:**
  - • **m**: Month number with leading zeroes. Values 01 through 12
  - • **n**: Month number without leading zeroes. Values 1 through 12
  - • **M**: Abbreviation for the month. Values Jan through Dec
  - • **F**: Normal month representation. Values January through December.
  - • **t**: The number of days in the month. Values 28 through 31.

- **Year:**
  - • **L**: 1 if it's a leap year and 0 if it isn't.
  - • **Y**: A four digit year format
  - • **y**: A two digit year format. Values 00 through 99.

- **Other Formatting:**
  - • **U**: The number of seconds since the Unix Epoch (January 1, 1970)
  - • **O**: This represents the Timezone offset, which is the difference from Greenwich Meridian Time (GMT).

Sesiuni PHP

- o sesiune PHP permite memorarea de informatii ale userului pe server pentru uz ulterior (i.e. username, shopping cart, etc). Aceste informatii de sesiune sunt temporare si uzual sterse repede dupa ce user-ul a parasit website ce foloseste sesiunile.
- Sesiunile functioneaza prin crearea de unic identificator(UID),un numar si memoreaza variabilele bazate pe ID. Asta previne ca datele a doi useri sa creeze probleme cand viziteaza aceeasi pagina web.

Startare de Sesiune PHP

- When you want to store user data in a session use the $_SESSION associative array. This is where you both store and retrieve session data. In previous versions of PHP there were other ways to perform this store operation, but it has been updated and this is the correct way to do it.
- <?php
- session_start(); // start up your PHP session!
- ?>
- **memorare unei Variable de Sesiune**
- <?php
- session_start();
- $_SESSION['views'] = 1; // store session data
- echo "Pageviews = ". $_SESSION['views']; //retrieve data
- ?>
- Rezultat: Pageviews = 1

Sesiune PHP

Functia

isset

- Now that you know can easily store and retrieve data from the $_SESSION array, we can now explore some of the real functionality of sessions. When you create a variable and store it in a session, you probably want to use it in the future. However, before you use a session variable it is necessary that you check to see if it exists already!
- This is where PHP's isset function comes in handy. isset is a function that takes any variable you want to use and checks to see if it has been set. That is, it has already been assigned a value.
- With our previous example, we can create a very simple pageview counter by using isset to check if the pageview variable has already been created. If it has we can increment our counter. If it doesn't exist we can create a pageview counter and set it to one.
- <?php
- session_start();
- if(isset($_SESSION['views']))
- $_SESSION['views'] = $_SESSION['views']+ 1;
- else
- $_SESSION['views'] = 1;
- echo "views = ". $_SESSION['views'];
- ?>

curatarea(cleaning) si distrugerea(destroying)unei sesiuni PHP

- Although a session's data is temporary and does not require that you explicitly clean after yourself, you may wish to delete some data for your various tasks.
- Imagine that you were running an online business and a user used your website to buy your goods. The user has just completed a transaction on your website and you now want to remove everything from their shopping cart.
- **Code:**
  - <?php
  - session_start();
  - if(isset($_SESSION['cart']))
  - unset($_SESSION['cart']);
  - ?>
- You can also completely destroy the session entirely by calling the *session_destroy* function.
- **PHP Code:**
  - <?php
  - session_start();
  - session_destroy();
  - ?>
- Destroy will reset your session, so don't call that function unless you are entirely comfortable losing all your stored session data!

PHP Cookies

- Cookies have been around for quite some time on the internet. They were invented to allow webmaster's to store information about the user and their visit on the user's computer.
- At first they were feared by the general public because it was believed they were a serious privacy risk. Nowadays nearly everyone has cookies enabled on their browser, partly because there are worse things to worry about and partly because all of the "trustworthy" websites now use cookies.
- basics of **storing a cookie** and **retrieving a cookie**, as well as explaining the various options you can set with your cookie.

PHP Cookies

- **Creating Your First PHP Cookie**
- When you create a cookie, using the function *setcookie*, you must specify three arguments. These arguments are ***setcookie****(name, value, expiration)*:
- 1. **name**: The name of your cookie. You will use this name to later retrieve your cookie, so don't forget it!
- 2. **value**: The value that is stored in your cookie. Common values are username(string) and last visit(date).
- 3. **expiration**: The date when the cookie will expire and be deleted. If you do not set this expiration date, then it will be treated as a session cookie and be removed when the browser is restarted.
- In this example we will be creating a cookie that stores the user's last visit to measure how often people return to visit our webpage. We want to ignore people that take longer than two months to return to the site, so we will set the cookie's expiration date to two months in the future!

- **PHP Code:**
- <?php
- //Calculate 60 days in the future
- //seconds * minutes * hours * days + current time
- $inTwoMonths = 60 * 60 * 24 * 60 + time();
- setcookie(lastVisit, date("G:i - m/d/y"), $inTwoMonths);
- ?>
-

Retrieving Your Fresh Cookie

- If your cookie hasn't expired yet, let's retrieve it from the user's PC using the aptly named *$_COOKIE* associative array. The name of your stored cookie is the key and will let you retrieve your stored cookie value!
- **PHP Code:**
- <?php
- if(isset($_COOKIE['lastVisit']))
- $visit = $_COOKIE['lastVisit'];
- else echo "You've got some stale cookies!";
- echo "Your last visit was - ". $visit;
- ?>
- Rezultat: Your last visit was - 11:48 - 02/28/08
-

PHP HTML Formular Exemplu

- Use this example as a form walkthrough. We will briefly build an HTML form, and call the form data using PHP. PHP offers several methods for achieving this goal, so feel free to substitute alternative methods as you follow along. Our example will show you a method using a single .php file, combining both PHP and HTML in one simple text file, to retrieve the data and display the results. Below is a quick review of bullets, check boxes, text fields, and input fields and using them to build a form to retrieve some personal information about our user.
- **Building the HTML Form**
- Step 1 is to build the form document to retrieve user date. If you already experienced using HTML forms, this should be review, however, if not we recommend a brief visit through the Tizag HTML Forms Tutorial. The code below shows a simple html form document set up to retrieve some personal knowledge about our user.
- **Input Fields**
- Input fields are the simplest forms to grasp. As mentioned in the Forms Tutorial, just be sure to place the name attribute within the tags and specify a name for the field. Also be aware that for our form's action we have placed the $PHP_SELF super global to send our form to itself. We will be integrating more PHP code into our form as we continue on so be sure to save the file with a .php extension.

PHP HTML Formular Exemplu

- **Cod:**
- <html>
- <head>
- <title>Personal INFO</title>
- </head>
- <body> <form method="post" action="<?php echo $PHP_SELF;?>"> First Name:<input type="text" size="12" maxlength="12" name="Fname">:
 Last Name:<input type="text size="12" maxlength="36" name="Lname">:

Butoane tip Radio si Checkbox

butoanele radio sunt asociate cu atributul valoare(*value).* Textul plasat sub atributul valoare va fi afisat de browser cand variabila e apelata in PHP.

- Check box-urile necesita utilizarea unui array(matrice). PHP va plasa automat check boxe-urile intr-un array daca plasati [] ca paranteze la sfarsitul fiecarui nume.

Butoane tip Radio si Checkbox

- ... Gender::
- Male:<input type="radio" value="Male" name="gender">:
- Female:<input type="radio" value="Female" name="gender">:
- choose type of residence::
- Steak:<input type="checkbox" value="Steak" name="food[]">:
- Pizza:<input type="checkbox" value="Pizza" name="food[]">:
- Chicken:<input type="checkbox" value="Chicken" name="food[]">:

Campuri de tip Textareas

In realitate,campurile textarea sunt campuri de intrare supra dimensionate . Va trebui sa se tina seama,atributul *wrap si cum* fiecare tip de wrap va apare. PHP se bazeaza pe acest atribut pentru afisare de tip textarea.

- **Cod exemplu:**
  - ... <textarea rows="5" cols="20" name="quote" wrap="physical">Enter your favorite quote!</textarea>:

Liste tip Drop Down si de Selectie

Aceste doua formulare actioneaza similar cubutoanele radio si checkbox in ce priveste selectiile. Pentru a denumi un formular de selectie , plasati atributul *name intre* tag-urile de selectie de la inceputul formularuluisi apoi plasati valoarea corespunzatoare fiecarei optiuni.

- **Cod exemplu:**
  - ... Select a Level of Education:
 <select name="education"> <option value="Jr.High">Jr.High</option> <option value="HighSchool">HighSchool</option> <option value="College">College</option></select>:
 Select your favorite time of day::
 <select name="TofD" size="3"> <option value="Morning">Morning</option> <option value="Day">Day</option> <option value="Night">Night</option></select>:

Se afiseaza:

  - First Name: Last Name: Gender: Male: Female: Favorite Food: Steak: Pizza: Chicken: Enter your favorite quote! Select a Level of Education: Jr.High Select your favorite time of day: MorningDayNight

Buton tip: Submission(Submit)

- Pentru a putea apela acest buton din PHP adaugati un nume la el
- Cod exemplu:
- ... <input type="submit" value="submit" name="submit">
 </form>

Recuperarea(regasirea) datelor din Formular – Setarea de Variable

- In PHP exista un array ce apeleaza date din formular. Este un superglobal din PHP . **$_POST** recupereaza(regaseste) date din formular si se afiseaza in browser. Cel mai bun mod de a face asta e de a crea variable pentru fiecare element din formular,pentru a avea un output al datelor la initiativa noastra, utilizand numele date de noi variabilelor. Plasati urmatoarele linii de cod la inceputul fisierului formular cu sintaxa corecta PHP .
- **Cod exemplu:**
  - <?php $Fname = $_POST["Fname"]; $Lname = $_POST["Lname"]; $gender = $_POST["gender"]; $food = $_POST["food"]; $quote = $_POST["quote"]; $education = $_POST["education"]; $TofD = $_POST["TofD"]; ?>
- Astfel putem apela datele cu usurinta.
- Literele mari de sub atributul nume trebuie sa se potriveasca cu instructiunile de mai sus si se va evita folosirea de nume prea complicate pentru un debugging mai usor.

$PHP_SELF; - Submiterea datelor(submission)

- Pentru actiunea asupra datelor din formular(form action) vom apela, array-ul PHP's $PHP_SELFcare e setat de a se apela pe el insusi la submitere. Practic setam formularul pentru apelarea "formexample.php", pe el insusi. Iata
- Codul:
- ... $quote = $_POST["quote"]; $education = $_POST["education"]; $TofD = $_POST["TofD"]; ?> <html> <head> <title>Personal INFO</title> </head> <body> <form method="post" action="<?php echo $PHP_SELF;?>"> ...
- astfel am completat formularul pentru a fi gata sa primeasca date si sa afiseze rezultate.
- Dar trebuie ajustate lucrurile asa ca atunci cand datele s-au submis sa fim directati la rezultate. Tipic, avem un complet nou fisier .php ce primeste datele din formularul HTML .astfel utilizam o instructiune if statement pentru afisarea primului formular, si apoi rezultatele din formularul nostru dupa submitere.aceasta este o metoda practica cand introducem informatii in baze de date .

formularul completat pana acum este:

$PHP_SELF; submitere(Submission)

- Cod :
- <?php $Fname = $_POST["Fname"]; $Lname = $_POST["Lname"]; $gender = $_POST["gender"]; $food = $_POST["food"]; $quote = $_POST["quote"]; $education = $_POST["education"]; $TofD = $_POST["TofD"]; ?> <html> <head> <title>Personal INFO</title> </head> <body> <form method="post" action="<?php echo $PHP_SELF;?>"> First Name:<input type="text" size="12" maxlength="12" name="Fname">
 Last Name:<input type="text" size="12" maxlength="36" name="Lname">
 Gender:
 Male:<input type="radio" value="Male" name="gender">
 Female:<input type="radio" value="Female" name="gender">
 Please choose type of residence:
 Steak:<input type="checkbox" value="Steak" name="food[]">
 Pizza:<input type="checkbox" value="Pizza" name="food[]">
 Chicken:<input type="checkbox" value="Chicken" name="food[]">
 <textarea rows="5" cols="20" name="quote" wrap="physical">Enter your favorite quote!</textarea>
 Select a Level of Education:
 <select name="education"> <option value="Jr.High">Jr.High</option> <option value="HighSchool">HighSchool</option> <option value="College">College</option></select>
 Select your favorite time of day:
 <select name="TofD" size="3"> <option value="Morning">Morning</option> <option value="Day">Day</option> <option value="Night">Night</option></select>
 <input type="submit" value="submit" name="submit"> </form>

Afisarea Paginii

- Pana la acest punct am completat formularul cu actiunea corecta si submiterea. Cu inca putina programare vrem afisarea inainte si dupa un anumit eveniment ,inainte ca utilizatorul sa submita orice informatie.
- Mai intai le directam in formular si apoi, vom afisa resultatele folosind numele variabilelor noastre.
- PHP ofera un mod excelent de a crea acest efect cu
- instructiunea *if* . Plasati urmatoarele linii langa inceputul fisierului formexample.php .
- **Codul :**
  - <?php $Fname = $_POST["Fname"]; $Lname = $_POST["Lname"]; $gender = $_POST["gender"]; $food = $_POST["food"]; $quote = $_POST["quote"]; $education = $_POST["education"]; $TofD = $_POST["TofD"]; if (!isset($_POST['submit'])) { // if page is not submitted to itself echo the form ?>

Afisarea rezultatelor ca ecou (echo back of resuts)

- Hereaici, avem un ecou (echo back) al resultatelor linie cu line pentru a arata sintaxa de baza in actiune. Se foloseste clauza else a instructiunii *if pentru a* directa utilizatorul la sectiunea de rezultate.
- **Codul este:**
- ... <option value="Night">Night</option></select> <input type="submit" value="submit" name="submit"> </form> <? } else { echo "Hello, ".$Fname." ".$Lname.".
"; echo "You are ".$gender.", and you like "; foreach ($food as $f) { echo $f."
"; } echo "".$quote."
"; echo "You're favorite time is ".$TofD.", and you passed ".$education."!
"; } ?>

Iata codul complet:

- **Cod:**
- <?php $Fname = $_POST["Fname"]; $Lname = $_POST["Lname"]; $gender = $_POST["gender"]; $food = $_POST["food"]; $quote = $_POST["quote"]; $education = $_POST["education"]; $TofD = $_POST["TofD"]; if (!isset($_POST['submit'])) { // if page is not submitted to itself echo the form ?> <html> <head> <title>Personal INFO</title> </head> <body> <form method="post" action="<?php echo $PHP_SELF;?>"> First Name:<input type="text" size="12" maxlength="12" name="Fname">
 Last Name:<input type="text" size="12" maxlength="36" name="Lname">
 Gender:
 Male:<input type="radio" value="Male" name="gender">
 Female:<input type="radio" value="Female" name="gender">
 Please choose type of residence:
 Steak:<input type="checkbox" value="Steak" name="food[]">
 Pizza:<input type="checkbox" value="Pizza" name="food[]">
 Chicken:<input type="checkbox" value="Chicken" name="food[]">
 <textarea rows="5" cols="20" name="quote" wrap="physical">Enter your favorite quote!</textarea>
 Select a Level of Education:
 <select name="education"> <option value="Jr.High">Jr.High</option> <option value="HighSchool">HighSchool</option> <option value="College">College</option></select>
 Select your favorite time of day:
 <select name="TofD" size="3"> <option value="Morning">Morning</option> <option value="Day">Day</option> <option value="Night">Night</option></select>
 <input type="submit" value="submit" name="submit"> </form> <? } else { echo "Hello, ".$Fname." ".$Lname.".
"; echo "You are ".$gender.", and you like "; foreach ($food as $f) { echo $f."
"; } echo "".$quote."
"; echo "You're favorite time is ".$TofD.", and you passed?> ".$education."!
"; }