

Cursul 10

PHP - partea a II-a

1. Lucrul cu formulare in PHP

- Sunt utilizate variabilele `$_GET`, `$_POST` și `$_REQUEST` – vectori asociativi
- Elementele din formularele HTML sunt disponibile în mod automat script-urilor PHP
- Trebuie evitată încărcarea server-ului cu validarea datelor din formulare
- Server-ul va fi folosit pentru validarea datelor atunci când acestea sunt introduse într-o bază de date
- `$_REQUEST` stochează conținutul variabilelor superglobale `$_GET`, `$_POST` și `$_COOKIE`

form.html

```
<html>
<body>

<form action="test.php" method="get">
Nume: <input type="text" name="nume" /><br />
Prenume: <input type="text" name="prenume" /><br />
Varsta: <input type="text" name="varsta" /><br />
<input value="Trimite" type="submit" />
</form>

</body>
</html>
```

test.php

```
<html>
```

```
<body>
```

```
Numele este:      <?php echo $_GET["nume"]; ?><br />
```

```
Prenumele este: <?php echo $_GET["prenume"]; ?><br />
```

```
Varsta este:     <?php echo $_GET["varsta"]; ?><br />
```

```
</body>
```

```
</html>
```

1.1. Metoda GET

- Este utilizată variabila \$_GET – vector asociativ (numele câmpurilor din formular reprezintă indecși în vector)
- Informația trimisă este afișată în bara de adrese
- Nu trebuie folosită la transmiterea datelor cu caracter privat (parole etc.)
- Cantitatea de informație trimisă este limitată: maxim 100 caractere
- Adresa paginii rezultate în urma transmiterii datelor din formular poate fi salvată și reacesată

```
http://localhost/test.php?nume=Ionescu&prenume=Ion&varsta=21
```

1.2. Metoda POST

- Este utilizată variabila `$_POST` – vector asociativ (numele câmpurilor din formular reprezintă indecși în vector)
- Informația trimisă este invizibilă
- Cantitatea de informație trimisă este relativ nelimitată (în mod predefinit 8Mb)
- Adresa paginii rezultate în urma transmiterii datelor din formular **NU** poate fi salvată și reacesată

form.html

```
<html>
<body>

<form action="test.php" method="post">
Nume:      <input type="text" name="nume" /><br />
Prenume:   <input type="text" name="prenume" /><br />
Varsta:    <input type="text" name="varsta" /><br />
Ocupatia: <input type="text" name="ocupatia" /><br />
<input value="Trimite" type="submit" />
</form>

</body>
</html>
```

test.html

```
<html>
<body>

Numele este: <?php echo $_POST["nume"]; ?><br />
Prenumele este: <?php echo $_POST["prenume"]; ?><br />
Varsta este: <?php echo $_POST["varsta"]; ?><br />
Ocupatia este <?php echo $_POST["ocupatia"]; ?>????<br />

</body>
</html>
```

Informația din bara de adrese

```
http://localhost/test.php
```


2. SSI – Server Side Includes

- Conținutul unui fișier PHP poate fi inserat în alt fișier PHP înainte ca acesta să fie executat de către server
- Funcții folosite:
 - ***include()*** – la apariția unei erori este afișat un mesaj de avertizare dar execuția script-ului este continuată
 - ***require()*** – la apariția unei erori este afișat un mesaj de eroare iar execuția script-ului este oprită (*este recomandată utilizarea sa, deoarece un script nu ar trebui să-și continue execuția în cazul apariției unei erori*)
- Este permisă astfel reutilizarea codului sursă (funcții, elemente de tip meniu, “header”, “footer” etc.)

Exemplul 1

```
<html>
<body>

<?php include("header.php"); ?>

<h1>Titlul paragrafului</h1>
<p>Continut paragraf.</p>

<?php include("footer.php"); ?>

</body>
</html>
```

Exemplul 2

menu.php

```
<a href = "index.php">Pagina principală</a>
<a href = "cursuri.php">Cursuri</a>
<a href = "laboratoare.php">Laboratoare</a>
<a href = "evaluare.php">Evaluare</a>
<a href = "utile.php">Utile</a>
<a href = "contact.php">Contact</a>
```

index.php

```
<html>
<body>

<div class="menu">
<?php include("menu.php"); ?>
</div>

<h1>Titlul paragrafului de introducere</h1>
<p>Paragraf introducere.</p>

</body></html>
```

Codul sursă al paginii afișate de navigator pentru index.php

```
<html>
<body>

<div class="meniu">
<a href = "index.php">Pagina principala</a>
<a href = "cursuri.php">Cursuri</a>
<a href = "laboratoare.php">Laboratoare</a>
<a href = "evaluare.php">Evaluare</a>
<a href = "utile.php">Utile</a>
<a href = "contact.php">Contact</a>
</div>

<h1>Titlul paragrafului de introducere</h1>
<p>Paragraf introducere.</p>

</body>
</html>
```

3. Lucrul cu fișiere în PHP

- Funcțiile destinate lucrului cu fișiere fac parte din nucleul PHP
- Funcții mai importante:
 - ***fopen*** – deschide un fișier (returnează 0 dacă deschiderea eșuează)
 - ***fclose*** – închide un fișier
 - ***feof*** – verifică ajungerea la sfârșitul fișierului
 - ***fgets*** – citește conținutul unui fișier linie cu linie
 - ***fgetc*** – citește conținutul unui fișier caracter cu caracter
 - ***file_exists*** – verifică existența unui fișier
 - ***fread*** – returnează șirul de caractere citit
 - ***fwrite*** – scrie un șir de caractere într-un fișier
 - ***fscanf*** – citire formatată dintr-un fișier

- Moduri de deschidere a unui fișier

Mod	Descriere
r	Citire, de la începutul fișierului
r+	Citire/scriere, de la începutul fișierului
w	Scriere, deschide și șterge conținutul fișierului existent sau crează un fișier nou dacă el nu există
w+	Citire/scriere, deschide și șterge conținutul fișierului existent sau crează un fișier nou dacă el nu există
a	Adăugare, deschide și scrie la sfârșitul fișierului sau crează fișierul în cazul în care el nu există
a+	Citire/adăugare
x	Scriere, crează un fișier nou.
x+	Citire/scriere, crează un fișier nou.

Exemplu deschidere fișier

```
<html>
<body>

<?php
$file=fopen("test.txt","r");

while(!feof($file))
{
    echo fgets($file). "<br />";
}

fclose($file);
?>

</body>
</html>
```

