

# Informatică Aplicată 1 – Îndrumar de Laborator

## Capitolul 3 – Formulare

### 1) Introducere

Prima etapă în crearea de formulare Web constă în proiectarea acestora. Astfel, e bine ca să nu se treacă direct la scrierea codului sursă al formularelor, ci, mai întâi, cu creionul pe hârtie să se proiecteze formularul. Proiectarea constă în: alegerea controalelor HTML necesare, selectarea poziției lor în pagină, diverse modalități de aranjare, alegerea etichetelor controalelor etc. Ordinea controalelor trebuie să fie cea a completării de către utilizator. Este recomandată alinierea controalelor, și gruparea celor din aceeași categorie. O aranjare eficientă a elementelor unui formular poate fi realizată plasând aceste elemente într-un tabel HTML.

Tag-ul HTML pentru formulare este FORM:

```
<FORM METHOD="metoda" ACTION="url">
```

Atributul *method* specifică metoda de trimitere a datelor din formular către server; aceasta poate fi GET sau POST. Se recomandă metoda POST pentru a putea trimite mai multe date, și pentru ca datele trimise să nu fie vizibile (ca în cazul metodei GET, care atașează datele la adresa fișierului apelat ce conține scriptul PHP).

Orice formular este încadrat între: <FORM> .... </FORM>

Putem avea mai multe formulare într-o pagină .HTML sau .PHP, dar *nu este permisă includerea unui formular în altul* (formularele trebuie să fie separate în pagină).

Atributul *action* specifică adresa scriptului (PHP de exemplu) care va prelua datele de la formular.

### 2) Principalele tipuri de controale HTML pentru formulare

#### a) Caseta text

```
<INPUT TYPE="TEXT" NAME="un_nume">
```

Va crea o casetă de introducere de text simplu. Atributul **NAME** specifică **variabila** care va conține textul introdus. Numele variabilei trebuie să respecte regulile specifice unei variabile PHP, de exemplu să nu conțină spații.

Atributele TYPE și NAME sunt obligatorii. Există și atribute opționale, de exemplu:

```
<INPUT TYPE="TEXT" SIZE="n1" MAXLENGTH="n2" NAME="nume" VALUE="val">
```

SIZE reprezintă dimensiunea casetei text, MAXLENGTH este numărul maxim de caractere permise, VALUE poate fi un text deja introdus în casetă.

Exemplu:

```
<INPUT TYPE="TEXT" SIZE=10 MAXLENGTH=20 NAME="nume">
```

Orice formular trebuie să conțină un buton de trimitere a datelor, care se definește cu:

```
<INPUT TYPE="SUBMIT" VALUE="OK">
```

Atributul *value* poate fi orice text și reprezintă eticheta butonului (ce va apărea scris pe acesta).

**Notă:** La apăsarea butonului SUBMIT (OK în exemplul de mai sus) se vor trimite către scriptul PHP specificat la ACTION variabilele definite în formular, adică cele specificate la atributul NAME în fiecare căsuță din formular. Se vor trimite ca perechi: nume variabilă și valoare variabilă (de exemplu textul introdus în caseta text).

*Exemplu:*

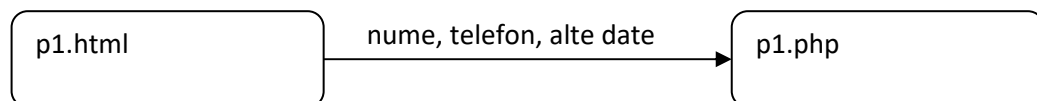
Creați un fișier HTML cu numele *p1.html* și următorul conținut:

```
<HTML>
<HEAD>
  <TITLE> Numele si telefonul utilizatorului </TITLE>
</HEAD>
<BODY>
  <h3>Introduceti numele si nr. de telefon</h3>
  <FORM METHOD="POST" ACTION="p1.php">
    Nume:<BR>
    <INPUT TYPE="TEXT" SIZE=15 MAXLENGTH=20 NAME="nume">
    <BR>
    Telefon:<BR>
    <INPUT TYPE="TEXT" SIZE=15 MAXLENGTH=15 NAME="telefon">
    <BR><BR>
    <INPUT TYPE="SUBMIT" VALUE="Trimite">
  </FORM>
</BODY>
</HTML>
```

Apoi creați un fișier PHP cu numele *p1.php* și următorul conținut:

```
<?php
    $nume = $_POST["nume"];
    $telefon = $_POST["telefon"];
    echo "Valori introduse in formular: <br>";
    echo "Numele este: $nume <br>";
    echo "Telefonul este: $telefon <br>";
?>
```

Primul fișier (*p1.html*) care conține formularul va transmite datele (nume, telefon, dar și alte date transmise automat precum adresa IP) către scriptul PHP specificat la rubrica ACTION. Aceste date vor fi trimise la apăsarea butonului „Trimite”.



**În PHP avem mai multe metode de acces la variabilele trimise de formulare:**

- metoda veche (până la versiunea PHP 4.2) permitea ca variabilele să fie vizibile direct în scriptul PHP apelat (de exemplu puteam vedea \$nume și \$telefon). S-a considerat că această facilitate poate crea probleme de securitate, și de aceea în fișierul PHP.INI a fost setată valoarea: **register\_globals = Off**. Dacă setăm register\_globals = On, vom putea vedea direct variabilele în scriptul PHP.

- preluarea prin intermediul funcției **import\_request\_variables**.

Exemplu de utilizare: import\_request\_variables("gp");

"gp" = variabile trimise prin metoda GET sau POST.

- **metoda recomandată**: folosirea sintaxei **\$\_POST["nume\_var"]**; sau **\$\_GET["nume\_var"]**;

Adică, sistemul oferă un vector cu numele **\$\_POST** sau **\$\_GET** (în funcție de metoda specificată la METHOD), cheile (indicii) vectorului reprezentând numele variabilelor, de exemplu "telefon".

În exemplul anterior (*p1.php*) am creat niște variabile locale (\$nume, \$telefon), cu același nume (NAME) ca și controalele formularului, pentru a face mai clar programul.

În afară de variabilele trimise de formulare, către scriptul PHP se mai trimit automat (de către browserul de Web) niște variabile de stare, predefinite, de exemplu:

- HTTP\_USER\_AGENT = numele browserului, versiunea, și versiunea sistemului de operare
- HTTP\_REFERER = adresa URL a formularului HTML
- REMOTE\_ADDR = adresa IP a clientului
- SERVER\_NAME = adresa URL a serverului care execută scriptul PHP

### **Exercițiu:**

Completați scriptul p1.php de mai sus pentru a afișa și aceste variabile de stare prezentate mai sus. Limbajul PHP oferă următoarea sintaxă pentru accesul la aceste variabile: \$\_SERVER["nume\_variabila\_de\_stare"].

#### *b) Zone de introducere de texte*

Putem crea o zonă specială pentru a introduce mai multe linii de text (precum un comentariu, mesaj etc.) cu ajutorul controlului TEXTAREA:

```
<TEXTAREA NAME="un_nume" ROWS=nr_lin COLS=nr_col WRAP="wrap">
</TEXTAREA>
```

**Atenție:** acest marcaj este de tip pereche, existând și marcajul </TEXTAREA>

Între cele două marcaje putem (opțional) scrie un text predefinit.

Atributele NAME și ROWS sunt obligatorii, celelalte fiind opționale. ROWS = numărul de linii de text vizibile (textul introdus poate fi mai lung și atunci apar bare de defilare verticală). COLS=numărul de coloane de text vizibile. Atributul WRAP determină trecerea la linie nouă a textului la marginea din dreapta a casetei. Acest comportament poate fi suprimat setând WRAP="OFF". Trimiterea textului poate fi făcută exact așa cum este afișată, pe mai multe coloane, setând WRAP="HARD". Setând WRAP="SOFT", chiar dacă textul introdus a fost afișat pe mai multe linii, el va fi trimis ca o singură linie (sigur, în cazul în care nu apăsăm <Enter>).

#### *c) Casete text pentru introducerea de parole*

Marcajul de tip parolă oferă mascarea caracterelor introduse de la tastatură. Diferă de controlul de tip casetă text doar prin tipul „PASSWORD”. Exemplu de utilizare:

```
<INPUT TYPE="PASSWORD" NAME="parola" SIZE=10 MAXLENGTH=10 >
```

#### *d) Casete de validare*

Dacă în formular avem nevoie de bifat anumite opțiuni, preferințe etc. putem folosi casete de tip *checkbox*, cu următoarea sintaxă:

```
<INPUT TYPE="CHECKBOX" NAME="alegere_1" CHECKED VALUE="un text">
```

Parametrii *checked* și *value* sunt opționali.

Dacă parametrul *checked* apare, căsuța va fi bifată în mod prestabilit. Dacă trecem atributul *value*, valoarea sau textul scris (de exemplu „un text”) va fi valoarea variabilei (de exemplu) \$alegere\_1 în cazul în care căsuța a fost bifată. Dacă ometem atributul *value*, variabila asociată controlului de tip checkbox în cazul bifării acestuia va avea valoarea „on”. Căsuțele nebifate *nu sunt trimise* deloc către server.

Putem crea și un grup de casete de validare folosind ca nume un vector. De exemplu: NAME="fructe[]". Presupunem că vectorul poate conține una sau mai multe denumiri de fructe alese, pentru fiecare denumire fiind necesar un marcaj INPUT distinct, dar având același nume.

#### *e) Butoane radio*

Spre deosebire de casetele de validare, butoanele radio apar de obicei în grupuri (cel puțin 2 butoane) și permit selectarea unei singure opțiuni din mai multe variante.

Sintaxa este similară cu cea a casetelor de validare, cu excepția tipului: TYPE="RADIO". Un grup de butoane radio se construiește pur și simplu alegând același nume pentru variabilă (parametrul NAME este comun unui grup de butoane radio).

*Exemplu:*

```
Cerul este:<br>
<INPUT TYPE="RADIO" NAME="acoperire_nori" VALUE="senin" CHECKED> Senin<BR>
<INPUT TYPE="RADIO" NAME="acoperire_nori" VALUE="variabil"> Variabil<BR>
<INPUT TYPE="RADIO" NAME="acoperire_nori" VALUE="innorat"> Innorat<BR>
```

**Exercițiu:** testați trimiterea datelor de la butoanele radio de mai sus prin afișarea opțiunii alese în scriptul PHP apelat.

#### *f) Meniuri verticale*

Directiva SELECT permite construirea de meniuri verticale din care utilizatorul poate alege una sau mai multe opțiuni. Sintaxa acestui control este:

```
<SELECT [MULTIPLE] NAME="nume" [SIZE="nr_linii"] >
  <OPTION [VALUE="valoare_1"] [SELECTED]> Text_afisat_1 </OPTION>
  <OPTION [VALUE="valoare_2"] [SELECTED]> Text_afisat_2 </OPTION>
  ...
</SELECT>
```

**Parantezele pătrate** din sintaxa de mai sus încadrează elemente opționale și *nu sunt elemente ale limbajului HTML (ele nu trebuie să apară în cod)*. Atributul opțional MULTIPLE permite selecția mai multor opțiuni o dată, prin apăsarea tastei CTRL și alegerea cu mouse-ul a opțiunilor respective. VALUE specifică valoarea ce va fi trimisă; dacă se omite, va fi trimis chiar textul afișat. SIZE determină aspectul meniului. Astfel, dacă SIZE=1 sau dacă este omis, meniul va fi o căsuță de selecție clasică. Pentru valori ale SIZE mai mari decât 2, meniul va deveni o listă de selecție afișată desfășurată. Opțiunea SELECTED determină pre-selectarea unui anumit element din meniu.

**Atenție**, pentru a putea trimite mai multe valori selectate (folosind opțiunea *multiple*), variabila trebuie să fie un vector, de exemplu putem scrie: NAME="nume[]".

#### **Exercițiu:**

Realizați un meniu cu posibilitatea alegerii unei singure opțiuni din mai multe posibile, apoi un meniu care oferă alegeri multiple. De exemplu: marca de mașină preferată, numele celor 12 luni ale anului etc. Afișați datele selectate într-un script PHP.

#### *g) Crearea unui buton de reset*

Există posibilitatea de a amplasa pe formular un buton de reset, adică cu efect de ștergere a tuturor valorilor introduse în formular. Sintaxa este următoarea:

```
<INPUT TYPE=RESET [VALUE ="text"]>
```

Dacă omitem parametrul *value*, pe buton va scrie pur și simplu „Reset”.

#### *h) Control de tip imagine*

Putem include o imagine (o hartă de exemplu) și selecta un punct din acea imagine cu mouse-ul. Coordonatele punctului ales vor fi trimise către scriptul PHP ca elemente ale unui vector. Exemplu:

```
<INPUT TYPE=IMAGE NAME="image[]" SRC="poza.jpg" width=300>
```

Imaginea va avea funcția de buton de trimitere a datelor din formular.

### i) Câmpuri ascunse

Câmpurile ascunse (sau hidden) nu sunt afișate în pagină. Ele permit trimiterea unor variabile și a valorilor acestora către scriptul PHP. Pot fi o metodă de a construi formulare interactive: datele introduse în primul formular sunt trecute în câmpuri ascunse în al doilea formular. Sintaxa este:

```
<INPUT TYPE=HIDDEN NAME="nume_variabila" VALUE="valoare_variabila">
```

### 3) Scripturi PHP auto-apelante

Până acum am scris formularele în fișiere HTML care apelează un script PHP aflat într-un fișier separat, cu numele specificat la atributul ACTION. Există și posibilitatea de a amplasa și formularul, și scriptul apelat în cadrul aceluiași fișier, cu extensia .PHP.

*Exemplu:*

Salvați codul de mai jos într-un fișier .php, de exemplu calc.php, și încărcați adresa lui (de exemplu: localhost/calc.php) în browser.

```
<html>
<head><title>Calculator simplu</title></head>
<body>
<h3>Calculator simplu</h3>
<p>
<FORM method=post action="calc.php">
  <INPUT TYPE="TEXT" NAME="op1" VALUE="<?php $op1=@$_POST["op1"]; echo $op1 ?>"> +
  <INPUT TYPE="TEXT" NAME="op2" VALUE="<?php $op2=@$_POST["op2"]; echo $op2 ?> ">
  <INPUT TYPE=SUBMIT VALUE="=">
  <INPUT TYPE=TEXT NAME=rezultat VALUE="
    <?php
      $op1=@$_POST["op1"]; $op2=@$_POST["op2"];
      echo $op1+$op2;
    ?> " disabled>
</FORM>
</body>
</html>
```

Mecanismul de funcționare al acestui mic calculator este următorul: attributele VALUE ale căsuțelor de text sunt scripturi PHP. Ultima instrucțiune a acestor scripturi, de exemplu *echo \$op1*, va seta practic atributul VALUE, funcția *echo* având rolul de a afișa, sau în acest caz returna o valoare.

Simbolul @ plasat ca prefix al unei variabile PHP transmite interpretorului să ignore faptul că acea variabilă este nedefinită. Dacă nu plasăm @, interpretorul va returna un mesaj de eroare.

Variabilele \$\_POST["op1"] și \$\_POST["op2"] sunt nedefinite la prima rulare a scriptului PHP, deoarece vectorul \$\_POST se formează doar după ce se trimit datele către formular prin apăsarea butonului de tip „Submit”.

Putem să utilizăm un câmp ascuns (hidden) pentru a implementa un contor al numărului de operații efectuate (al numărului de apeluri) inserând în formularul de mai sus codul următor:

```
<INPUT TYPE=HIDDEN NAME="nr" VALUE="<?php $nr=@$_POST["nr"]+1 ; echo $nr ?>">
```

Și inserând după sfârșitul formularului (după </FORM>) codul următor:

```
<?php
echo "<br> Nr. rulari = ", @$_POST["nr"], "<br>";
?>
```

#### 4) Verificarea datelor introduse în formulare

Pentru a preveni erorile sau omisiunile în completarea formularelor putem adăuga instrucțiuni de validare a datelor. Un exemplu este următorul:

Fișierul *p3.html*:

```
<HTML>
<HEAD>
  <TITLE> Numele, telefonul si emailul utilizatorului </TITLE>
</HEAD>
<BODY>
  <h3>Introduceti datele</h3>
  <FORM METHOD="POST" ACTION="p3.php">
    Nume:<BR>
    <INPUT TYPE="TEXT" SIZE=15 MAXLENGTH=20 NAME="nume">
    <BR>
    Telefon:<BR>
    <INPUT TYPE="TEXT" SIZE=15 MAXLENGTH=15 NAME="telefon">
    <BR>
    E-mail:<BR>
    <INPUT TYPE="TEXT" SIZE=15 MAXLENGTH=15 NAME="email">
    <BR>
    <BR>
    <INPUT TYPE="SUBMIT" VALUE="Trimite"> <INPUT TYPE="RESET">
  </FORM>
</BODY>
</HTML>
```

Fișierul apelat, *p3.php*:

```
<?php
$nume = $_POST["nume"]; $telefon = $_POST["telefon"]; $email = $_POST["email"];

echo "<h3>Valori introduse in formular</h3>";

if (!trim($nume))
  echo "<BR><font color=red> Numele este obligatoriu. </font>";
else
  echo "Numele introdus este: $nume <br>";

if (!trim($telefon))
  echo "<BR> <font color=red> Telefonul este obligatoriu. </font>";
else
  echo "<BR> Telefonul este $telefon <br>";

if (!filter_var($email, FILTER_VALIDATE_EMAIL))
  echo "<BR><font color=red> Adresa de e-mail nu este valida </font>";
else
  echo "<BR> E-mailul este $email <br>";
?>
```

Funcția *trim* elimină spațiile de la începutul și sfârșitul unui șir de caractere. Dacă șirul este vid sau format doar din spații, atunci *trim* va returna NULL. Iar !NULL este TRUE.

Funcția *filter\_var* poate valida un șir de caractere după anumite criterii: adresă de email, număr întreg, adresă IP etc.

## Probleme propuse

- 1) Realizați un formular care să preia următoarele date: Prenume, Nume, e-mail, nivel studii, limbi străine cunoscute, adresa completă, cod poștal, oraș, țară.  
Nivelul de studii să conțină 3 butoane radio: Medii, Superioare, Doctorat.  
Limbile străine să fie casete de validare: Engleză, Franceză, Germană.  
Țara să fie un meniu de selecție care să conțină: România, Bulgaria, Republica Moldova.  
Realizați și scriptul PHP care să afișeze datele completate de utilizator în formular.
- 2) Realizați un formular care să conțină un tag de tip text pentru o variabilă numită „raza”. Formularul să conțină de asemenea un meniu de selecție cu două opțiuni: 1- calculează aria cerc și 2- calculează volum sferă. Realizați și scriptul PHP apelat de formular, care primind raza să calculeze și să afișeze aria cercului de rază R sau volumul sferei de rază R, conform selecției din formular.
- 3) Creați un convertor inch / cm sub forma unui script PHP auto-apelant.  
Notă: 1 inch = 2,54 cm
- 4) Adăugați la scriptul de la punctul anterior un meniu de selecție: mm / cm / metri și calculați rezultatul în funcție de selecție.
- 5) Construiți un calculator de operații simple sub forma unui script PHP auto-apelant. Calculatorul să permită introducerea a două numere, și alegerea (prin buton radio) a operației de efectuat, între: +, -, \* și /.