Pagini WEB cu PHP 4

Mihai Scorțaru, Claudiu Soroiu

În cadrul acestui articol vom continua prezentarea funcțiilor utilizate pentru prelucrarea și publicarea imaginilor dinamice și vom prezenta, de asemenea, funcțiile de acces la baze de date MySQL.

Scrierea textului pe imagini

În continuare vom prezenta câteva funcții utilizate pentru a scrie un text sau un caracter pe o imagine. Nu vom prezenta însă modul în care se poate încărca și folosi un anumit tip de caractere (*font*).

Funcția imagechar

Acestă funcție scrie pe o imagine primul caracter al unui șir de caractere.

Funcția `imagechar` are șase parametri:

- primul parametru este identificatorul de acces la imaginea pe care se va desena.
- cel de-al doilea parametru reprezintă identificatorul de acces la *font*-ul care va fi folosit pentru a scrie caracterul pe imagine. Acest identificator este de tip număr întreg și poate avea valorile predefinite 1, 2, 3, 4 sau 5 sau poate fi un număr returnat de o funcție care încarcă un font de pe disc.
- cel de-al treilea parametru reprezintă coordonata *x* a colțului stânga-sus de unde va începe scrierea caracterului;
- cel de-al patrulea parametru reprezintă coordonata *y* a colțului stânga-sus de unde va începe scrierea caracterului;
- cel de-al cincilea parametru reprezintă șirul de caractere din care va fi scris doar primul caracter;
- cel de-al șaselea parametru reprezintă culoarea cu care se va scrie pe imagine.

În continuare vă prezentăm un exemplu de utilizare a acestei funcții, iar în figura 1a) se poate observa efectul execuției *script*-ului.

```
<?PHP
  header("Content-type: image/png");
  $im = imagecreatefromjpeg("images/ginfo.jpg");
  $gray = imagecolorallocate($im, 127,127,127);
  imagechar($im, 5, 75, 55, 'a', $gray);
  imagepng($im);
  imagedestroy($im);
?>
```

Funcția imagecharup

Această funcție scrie un caracter vertical pe o imagine. Are aceeași parametri ca funcția anterioară.

În continuare vă prezentăm un exemplu de utilizare a acestei funcții, iar în figura 1b) se poate observa efectul execuției *script*-ului.

```
<?PHP
  header("Content-type: image/png");
  $im = imagecreatefromjpeg("images/ginfo.jpg");
  $gray = imagecolorallocate($im, 127,127,127);
  imagechar($im, 5, 70, 65, 'a', $gray);
  imagepng($im);
  imagedestroy($im);
?>
```

Funcția imagestring

Această funcție scrie un text pe o imagine și are aceeași parametri ca funcția `imagechar`.

În continuare vă prezentăm un exemplu de utilizare a acestei funcții, iar în figura 2a) se poate observa efectul execuției *script*-ului.

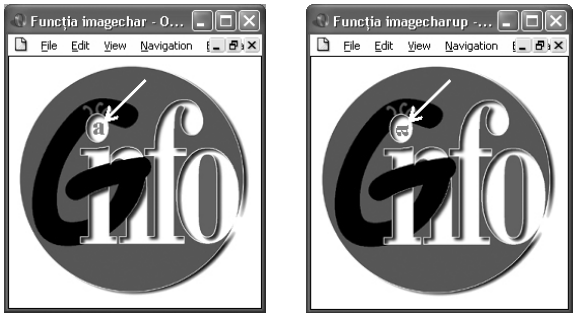

Figura 1: a) Funcția imagechar
b) Funcția imagecharup

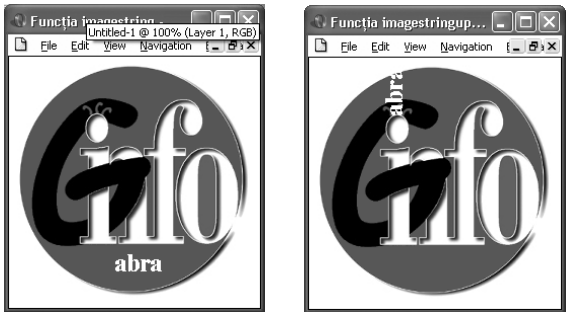

Figura 2: a) Funcția imagestring
b) Funcția imagestringup

```
<?PHP
  header("Content-type: image/png");
  $im = imagecreatefromjpeg("images/ginfo.jpg");
  $wh = imagecolorallocate($im, 255, 255, 255);
  imagestring($im, 5, 100, 190, 'abra', $wh);
  imagepng($im);
  imagedestroy($im);
?>
```

Funcția imagestringup

Această funcție scrie vertical un text pe o imagine și are aceeași parametri ca funcția `imagechar`.

În continuare vă prezentăm un exemplu de utilizare a acestei funcții, iar în figura 2b) se poate observa efectul execuției *script*-ului.

```
<?PHP
  header("Content-type: image/png");
  $im = imagecreatefromjpeg("images/ginfo.jpg");
  $wh = imagecolorallocate($im, 255, 255, 255);
  imagestringup($im, 5, 70, 50, 'abra', $wh);
  imagepng($im);
  imagedestroy($im);
?>
```

Alte prelucrări ale imaginilor

În cadrul acestui paragraf vom prezenta alte funcții utilizate pentru a prelucra în mod dinamic imagini.

Funcția imagerotate

Această funcție realizează rotirea unei imagini în sens trigonometric cu un unghi primit ca parametru. Imaginea rezultată în urma rotirii va avea dimensiunile egale cu dimensiunile celui mai mic dreptunghi ale cărui laturi sunt paralele cu axele *Ox* și *Oy* și care conține imaginea rotită.

Funcția are trei parametri:

- primul reprezintă identificatorul de acces la imaginea;
- al doilea este de tip real și reprezintă unghiul cu care se va roti imaginea;
- al treilea parametru este de tip întreg și reprezintă culoarea cu care se va umple dreptunghiul care va conține imaginea rotită.

Funcția `imagerotate` returnează un identificator de acces la imaginea rotită.

În continuare vă prezentăm un exemplu de utilizare a acestei funcții, iar în figura 3 se poate observa efectul execuției *script*-ului.

```
<?PHP
  header("Content-type: image/png");
  $im = imagecreatefromjpeg("images/ginfo.jpg");
  $gray = imagecolorallocate($im, 192, 192, 192);
  $imrot = imagerotate($im, 45, $gray);
  imagepng($imrot);
  imagedestroy($imrot);
?>
```

Funcția imagetruecolorpalette

Această funcție realizează transformarea unei imagini care nu folosește o paletă de culori într-o imagine care folosește o paletă de culori.

Funcția `imagetruecolorpalette` are trei parametri:

- primul reprezintă identificatorul de acces al imaginii care va fi transformată;
- cel de-al doilea parametru este de tip logic și reprezintă modul în care se va face transformarea: dacă imaginea va fi fără zgomot (valoarea logică **FALSE** implică o aproximare mai slabă a culorilor); dacă imaginea va fi cu zgomot (valoarea **TRUE** implică o aproximare mai bună a culorilor).

Funcția jpeg2wbmp

Această funcție transformă o imagine din formatul *JPEG* în formatul *WBMP*.

Funcția are cinci parametri. Primul parametru reprezintă calea spre fișierul de tip *JPEG* care va fi transformat. Al doilea parametru reprezintă calea spre fișierul care va conține rezultatul transformării.

Următorii doi parametri sunt de tip întreg și reprezintă înălțimea și lățimea noii imagini, iar ultimul parametru, tot de tip întreg, reprezintă pasul cu care se va parcurge imaginea pentru a fi transformată.

Funcția png2wbmp

Această funcție transformă o imagine din formatul *PNG* în formatul *WBMP*. Are aceeași parametri ca funcția anterioară, cu observația că primul parametru reprezintă calea spre un fișier de tipul *PNG*.

Figura 3: Funcția imagerotate

Publicarea imaginilor

După cum s-a putut observa, în cadrul *script*-urilor prezentate ca exemplu, am folosit funcția `imagepng`. Această funcție este folosită pentru a scrie la ieșirea standard o imagine în formatul *PNG*.

În continuare vom prezenta câteva funcții utilizate pentru a scrie la ieșirea standard o imagine în diferite formate, dar mai întâi vom prezenta funcția `imageinterlace`, o funcție foarte importantă în cazul în care doriți ca *site*-ul vostru să aibă o navigabilitate bună, chiar dacă nu s-au încărcat integral toate elementele grafice prezente în cadrul lui.

Funcția imageinterlace

Această funcție setează proprietatea *interlace* al unei imagini la valoarea logică precizată ca parametru sau la valoarea logică **TRUE** dacă acest parametru lipsește.

Funcția are doi parametri. Primul parametru reprezintă identificatorul de acces la imaginea pentru care se dorește setarea proprietății *interlace*, iar al doilea parametru este de tip logic și reprezintă valoarea pe care o va avea această proprietate. Cel de-al doilea parametru poate lipsei, dar proprietatea *interlace* a unei imagini va primi valoarea **TRUE**.

Ce însemnă de fapt că o imagine are proprietatea interlace? Ei bine, în cazul în care aveți o viteză de acces la *Internet* mică sau imaginea pe care doriți să o încărcați este mare, nu este nevoie să așteptați încărcarea întregii imagini pentru a vedea ce este în ea. O astfel de imagine se încarcă adaptiv, și în momentul în care s-a încheiat transferul imaginii către *browser*-ul utilizatorului, pe imagine se pot observa toate detaliile. De fapt, teoretic vorbind, o imagine este împărțită în subimagini de aceeași dimensiune și calitate diferită (din ce în ce mai bună), care suprapuse dau ca rezultat imaginea inițială. Practic nu se întâmplă chiar așa, dar efectul obținut este cel precizat anterior.

Publicare în diverse formate

Pentru a publica o imagine generată dinamic se poate utiliza una dintre funcțiile: `imagegif`, `imagepng`, `imagejpeg` sau `imagewbmp`. Aceste funcții primesc ca parametru un identificator de imagine și tipăresc la ieșirea standard imaginea în format *GIF*, *PNG*, *JPEG* sau *WBMP*.

Primele trei funcții se folosesc pentru a genera o imagine care să poată fi vizualizată într-un *browser* încărcat de pe un calculator, iar ultima se folosește pentru ca imaginea generată să poată fi vizualizată în cadrul unui *browser* care suportă *WAP*.

MySQL

În cele ce urmează vom prezenta cele mai utilizate funcții pentru a accesa și interoga baze de date *MySQL* din cadrul *script*-urilor *PHP*.

Funcția mysql_connect

Această funcție realizează conectarea la un *server MySQL*.

Funcția `mysql_connect` primește cinci parametri opționali:

- primul reprezintă numele *server*-ului *MySQL* și dacă lipsește se încearcă conectarea la calculatorul curent.
- cel de-al doilea reprezintă numele utilizatorului și în caz că acesta lipsește se folosește valoarea `root` ca nume de utilizator.
- al treilea parametru reprezintă parola utilizatorului și în caz că acesta lipsește se folosește șirul vid în momentul conectării la *server*.
- al patrulea parametru este de tip logic și are rolul de a indica motorului *PHP* că în cazul în care există o altă conexiune la același *server* să se mai creeze încă una în loc să se reutilizeze cea existentă.
- ultimul parametru este de tip întreg și reprezintă proprietățile pe care le va avea conexiunea. Acest ultim parametru poate fi o combinație a valorilor:
  - `MYSQL_CLIENT_COMPRESS` – indică *server*-ului că va trebui să trimită date comprimate;
  - `MYSQL_CLIENT_IGNORE_SPACE` – indică *server*-ului *MySQL* faptul că va trebui să ignore spațiile care urmează după numele funcțiilor;
  - `MYSQL_CLIENT_INTERACTIVE` – indică *server*-ului *MySQL* să nu închidă conexiunea după scurgerea unui anumit interval de timp în care *script*-ul nu a emis cereri către acesta.

Funcția `mysql_connect` returnează valoarea logică **FALSE** în cazul în care a eșuat conectarea la *server*-ul de baze de date și un identificator de acces la conexiunea către *server*-ul *MySQL* în cazul în care conectarea a reușit.

În continuare prezentăm un exemplu de utilizare a acestei funcții, iar în figura 4 poate fi observat efectul execuției acestui *script*:

```
<HTML>
  <HEAD>
    <TITLE>
      Exemplu mysql_connect
    </TITLE>
  </HEAD>
  <BODY>
  <?PHP
    $conn = mysql_connect("blue", "ginfo",
                                  "pginfo");
    if ($conn === FALSE)
      echo "Conectare eşuată.";
    else
      echo "Conexiune stabilită.";
    mysql_close($conn);
  ?>
  </BODY>
</HTML>
```

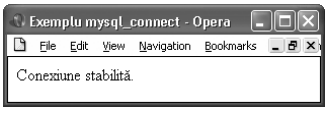

Figura 4: Funcția mysql_connect

Funcția mysql_close

Această funcție primește ca parametru un identificator de acces la o conexiune spre un *server MySQL* și realizează închiderea acesteia.

Am folosit această funcție în exemplul anterior, chiar înainte de terminarea *script*-ului.

Funcția mysql_ping

Funcția `mysql_ping` verifică dacă *server*-ul de *MySQL* a închis conexiunea sau nu. Aceasta nu are nici un parametru și returnează valoarea logică **TRUE** în cazul în care conexiunea nu a fost închisă și valoarea logică **FALSE** în caz contrar.

Funcția mysql_create_db

Această funcție este utilizată pentru a crea o nouă bază de date pe *server*-ul *MySQL*.

Funcția are doi parametri. Primul este de tip șir de caractere și reprezintă numele bazei de date care va fi creată, iar cel de-al doilea parametru reprezintă identificatorul de acces la conexiunea către *server*-ul *MySQL*.

Funcția returnează valoarea logică **TRUE** în cazul în care a reușit să creeze baza de date și valoarea logică **FALSE** în caz contrar.

Funcția mysql_drop_db

Această funcție este folosită pentru a șterge o bază de date.

Funcția `mysql_drop_db` are doi parametri. Primul este de tip șir de caractere și reprezintă numele bazei de date care va fi ștearsă, iar cel de-al doilea parametru reprezintă identificatorul de acces la conexiunea către *server*-ul *MySQL*.

Funcția returnează valoarea logică **TRUE** în cazul în care s-a reușit ștergerea bazei de date și valoarea logică **FALSE** în caz contrar.

Funcția mysql_select_db

Această funcție setează baza de date pentru o conexiune către un *server MySQL* pentru interogările care vor urma.

Funcția `mysql_select_db` are doi parametri. Primul este de tip șir de caractere și reprezintă numele bazei de date care va fi folosită pentru interogările ulterioare, iar cel de-al doilea parametru reprezintă identificatorul de acces la conexiunea către *server*-ul *MySQL*.

Funcția returnează valoarea logică **TRUE** în cazul în care s-a reușit setarea bazei de date și valoarea logică **FALSE** în cazul în care baza de date nu există sau utilizatorul nu are drepturi de acces la aceasta.

Funcția mysql_query

Această funcție se folosește pentru a interoga o anumită bază de date. Funcția `mysql_query` are trei parametri:

- primul este de tip șir de caractere și reprezintă cererea emisă *server*-ului *MySQL*;
- al doilea parametru reprezintă identificatorul de acces al unei conexiuni către *server*;
- parametru este opțional și de tip întreg și reprezintă modul în care va fi returnat rezultatul. Ultimul parametru poate avea valorile `MYSQL_USE_RESULT` sau `MYSQL_STORE_RESULT`.

În cazul în care ultimul parametru lipsește, pentru interogare se va folosi valoarea `MYSQL_STORE_RESULT`, iar rezultatul primit va fi stocat în zone tampon până când va fi utilizat. Dacă, pentru ultimul parametru, se folosește valoarea `MYSQL_USE_RESULT` , atunci rezultatul primit nu va fi stocat, deoarece va fi procesat imediat după apelul funcției.

Utilizarea funcției folosind valoarea `MYSQL_USE_RESULT` pentru ultimul parametru are același efect cu aplicarea funcției `mysql_unbuffered_query` fără a folosi ultimul parametru.

Funcția returnează o valoare de tip *resursă* în cazul în care a fost apelată folosind o interogare care presupune primirea unui rezultat cum sunt comenzile: `SELECT`, `SHOW`, `EXPLAIN` și `DESCRIBE` și nu a apărut nici o eroare, iar în cazul apariției unei erori, este returnată valoarea logică **FALSE**.

În cazul în care funcția `mysql_query` este apelată pentru o interogare care nu presupune primirea unui rezultat, este returnată valoarea logică **TRUE** și valoarea logică **FALSE** în cazul apariției unei erori.

În continuare prezentăm un exemplu de utilizare a acestei funcții, iar în figura 5 poate fi observat efectul execuției acestui *script*:

```
<HTML>
  <HEAD>
    <TITLE>
      Exemplu MySQL
    </TITLE>
  </HEAD>
  <BODY>
  <?PHP
    $conn = mysql_connect("blue", "ginfo",
                                  "pginfo");
    if ($conn === FALSE)
      die("Conectare eşuată.");
    mysql_select_db("ginfo", $conn);
    $res = mysql_query("select * from users
              order by prenume, nume", $conn);
    echo "<TABLE border = 1>";
    echo "<TR><TD align = center>Prenume</TD>";
    echo "<TD align = center>Nume</TD>";
    echo "<TD align = center>Email</TD></TR>";
    while ($lin = mysql_fetch_array($res,
                                MYSQL_BOTH)){
      echo "<TR><TD>".$lin["prenume"]."</TD>";
      echo "<TD>".$lin["nume"]."</TD>";
      echo "<TD><A href = \"mailto:".
                          $lin["email"]."\">";
```

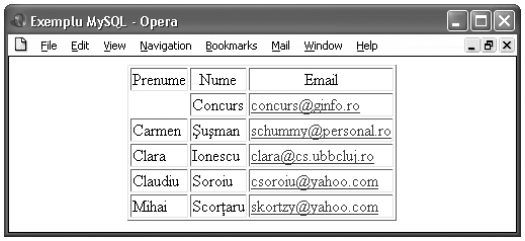

Figura 5: Exemplu de utilizare MySQL

```
      echo $lin["email"]."</A></TD></TR>";
    }
    echo "</TABLE>";
    mysql_free_result($res);
    mysql_close($conn);
  ?>
  </BODY>
</HTML>
```

Funcția mysql_unbuffered_query

Această funcție se folosește pentru a interoga o anumită bază de date și are aceeași parametri cu funcția anterioară.

Diferența dintre această funcție și `mysql_query` este aceea că, în cazul funcției `mysql_unbuffered_query`, dacă ultimul parametru lipsește, se folosește la interogarea bazei de date valoarea `MYSQL_USE_RESULT`.

Funcția mysql_real_escape_string

Această funcție este utilă în momentul în care se dorește introducerea în cadrul unei interogări a unui șir de caractere care nu poate fi interpretat corect de către *server*-ul *MySQL* și realizează transformarea șirului într-unul care poate fi interpretat.

Funcția `mysql_real_escape_string` are doi parametri. Primul este de tip șir de caractere și reprezintă șirul care va fi transformat, iar cel de-al doilea reprezintă identificatorul de acces la o conexiune către un *server MySQL*.

Funcția returnează un șir de caractere care poate fi interpretat corect de către *server*-ul *MySQL*.

De exemplu, dacă apelăm această funcție pentru șirul de caractere `D'apoi` vom obține rezultatul `D\'apoi`, deoarece caracterul `"'"` (apostrof) are o altă semnificație pentru *server*-ul *MySQL* dacă nu este precedat de caracterul `"\"`.

Funcția mysql_free_result

Această funcție se folosește pentru a elibera memoria alocată stocării unui rezultat primit în urma unei interogări de la un *server MySQL*.

Funcția `mysql_free_result` are un singur parametru, și anume o variabilă de tip *resursă* a cărei valoare a fost setată folosind una dintre funcțiile `mysql_query` sau `mysql_unbuffered_query`.

Funcția returnează valoarea logică **TRUE** în cazul în care a reușit să elibereze memoria utilizată de variabila de tip *resursă* primită ca parametru și valoarea logică **FALSE** în caz contrar.

Funcția mysql_num_rows

Această funcție returnează numărul de înregistrări conținute de către un rezultat primit de la *server*-ul *MySQL*.

Funcția `mysql_num_rows` are un singur parametru, și anume o variabilă de tipul *resursă* a cărei valoare a fost setată folosind una dintre funcțiile `mysql_query` sau `mysql_unbuffered_query`.

Funcția returnează un număr întreg care reprezintă numărul de înregistrări conținute de variabila de tip *resursă* primită ca parametru.

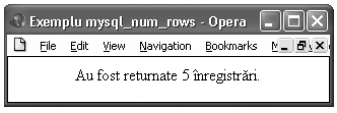

Figura 6: Funcția mysql_num_rows

În continuare prezentăm un exemplu de utilizare a acestei funcții, iar în figura 6 poate fi observat efectul execuției acestui *script*:

```
<HTML>
  <HEAD>
    <TITLE>
      Exemplu mysql_num_rows
    </TITLE>
  </HEAD>
  <BODY>
  <?PHP
    $conn = mysql_connect("blue", "ginfo",
                                  "pginfo");
    if ($conn === FALSE)
      die("Conectare eşuată.");
    mysql_select_db("ginfo", $conn);
    $res = mysql_query("select * from users
              order by prenume, nume", $conn);
    $num_rows = mysql_num_rows($res);
    echo "Au fost returnate $num_rows
                          înregistrări.";
    mysql_free_result($res);
    mysql_close($conn);
  ?>
  </BODY>
</HTML>
```

Funcția mysql_info

Această funcție returnează un șir de caractere care conține informații referitoare la ultima interogare a unei baze de date în urma căreia nu s-a primit nici un rezultat, cum este cazul funcțiilor `INSERT` sau `UPDATE`.

Funcția `mysql_info` are un singur parametru care reprezintă identificatorul de acces al unei conexiuni spre un *server MySQL*.

Va urma...

În numărul următor vom continua prezentarea funcțiilor referitoare la accesarea bazelor de date *MySQL* și vom începe prezentarea funcțiilor de poștă electronică folosite de programatorii *web* pentru a-i anunța pe eventualii utilizatori ai *site*-ului lor de apariția unor evenimente.