



Tehnologii Internet

CURSUL 09 – SERVERUL WEB

Universitatea Tehnică "Gheorghe Asachi" din Iași
Facultatea de Automatică și Calculatoare
Departamentul de Calculatoare
Specializarea Tehnologia informației

© 2017-2018 Adrian ALEXANDRESCU



Cuprins

1. Serverul web
2. Limbajul PHP
3. Cookie



1. Serverul web

1.1. Paradigma client-server

1.2. Serverul web – introducere

1.3. Limbaje client-side

1.4. Limbaje server-side

1.5. Exemple de servere web

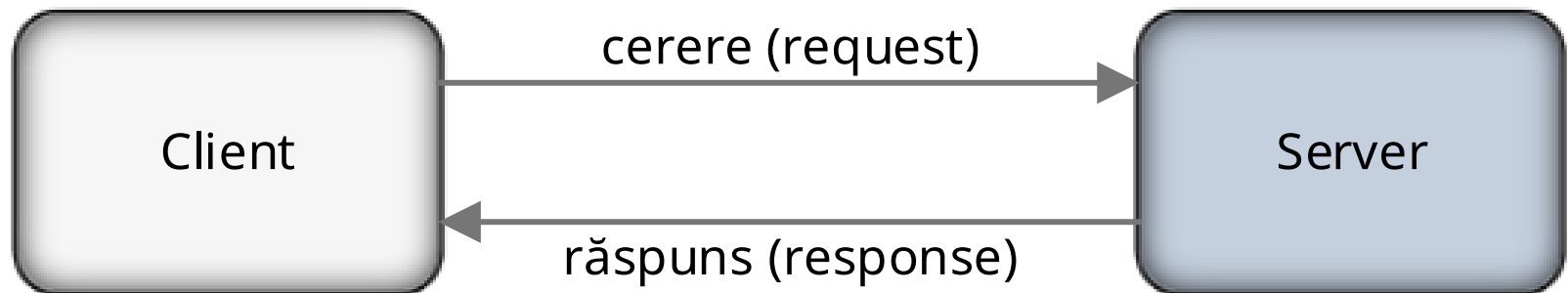
1.6. Statistici



1.1. Paradigma client-server

Paradigma client-server

- **Server** = instanță a unei aplicații care primește cereri și oferă răspunsuri
- **Client** = instanță care accesează serviciile puse la dispoziție de un server





1.2. Serverul web – introducere

- Serverul web este un server care folosește protocolul HTTP (HyperText Transfer Protocol)
- Portul implicit folosit de un server web este 80
- Informația de la un server web poate fi accesată prin intermediul URL-urilor (Uniform Resource Identifier)
- Exemplu:

`http://studenti.h23.ro/login`



1.3. Limbaje client-side

Clientul interpretează/execută codul

- HTML (.html)
- JavaScript (.js)
- CSS (.css)



1.4. Limbaje server-side

Serverul interpretează/execută codul

- PHP (.php)
- Java (.jsp) – Java Server Pages (JSP), Servlet
- ASP (.asp) și ASP.NET (.aspx) – Active Server Pages
- Server-side JavaScript (.ssjs, .js) – e.g., Node.js
- C (.c, .csp) prin CGI (Common Gateway Interface)
- Python (.py)
- Ruby (.rb, .rbw) – e.g., Ruby on Rails
- Go (.go)

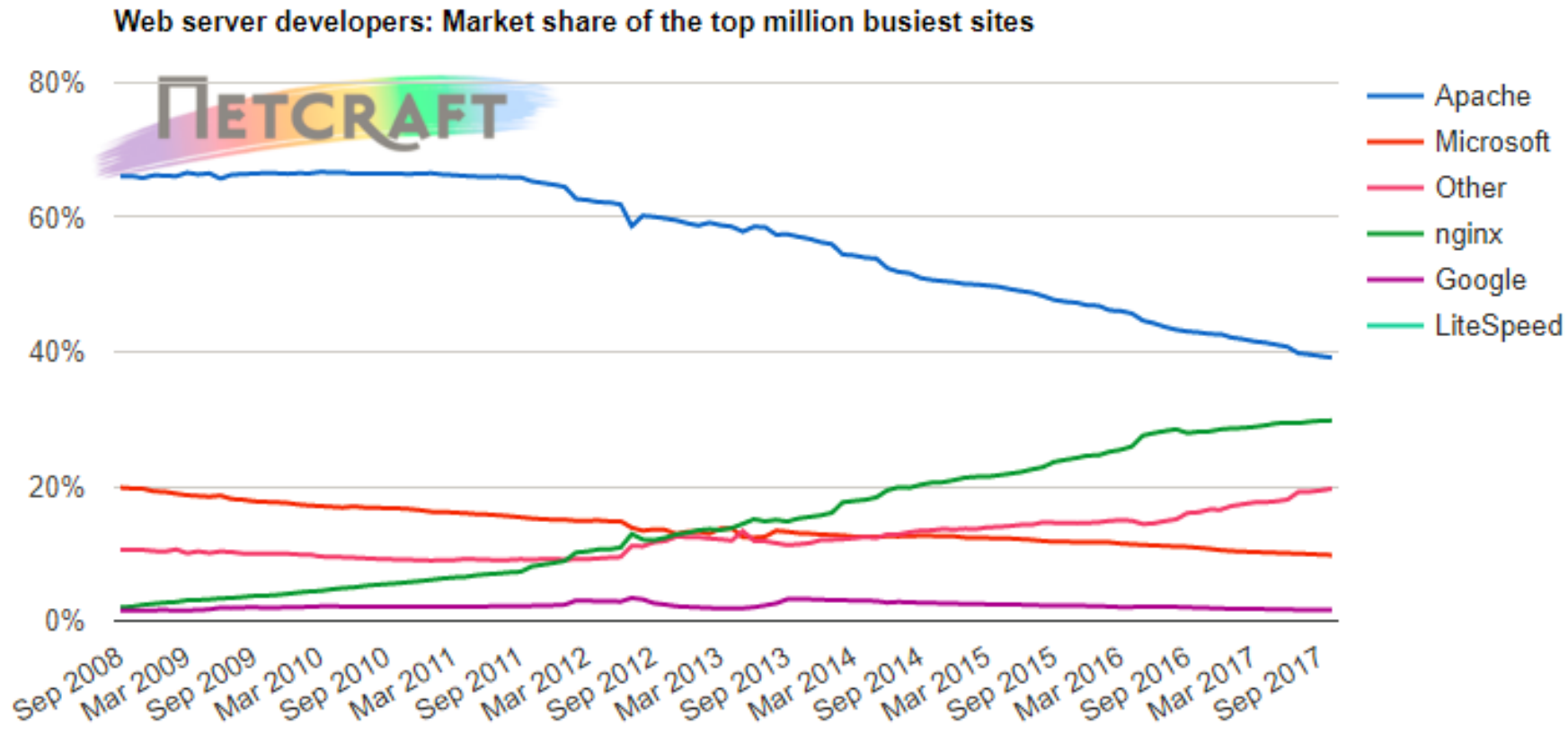


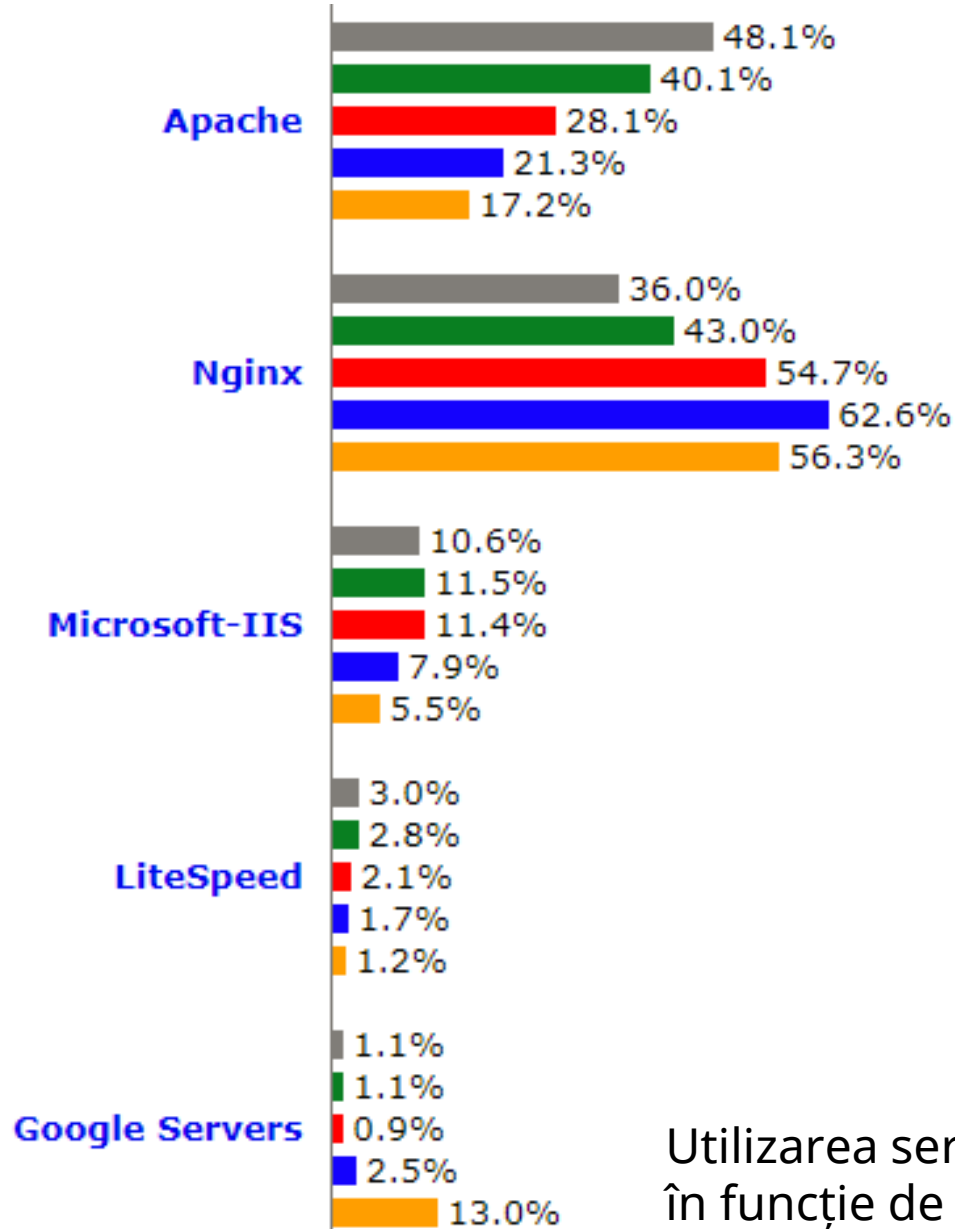
1.5. Exemple de serveur web

- Apache HTTP server
- Nginx
- Microsoft-IIS (Internet Information Services)
- Apache Tomcat
- LiteSpeed Web Server
- Google Web Server (GWS)
- Jetty
- Oracle WebLogic Server
- Mongoose



1.6. Statistici





Utilizarea serverelor web
în funcție de ranking

W3Techs.com, 19 November 2017

■ Overall ■ top 1,000,000 ■ top 100,000 ■ top 10,000 ■ top 1,000



1.6. Statistici

- Website ranking
- <http://www.alexa.com/topsites>



2. Limbajul PHP

2.1. Introducere

2.2. Istoric

2.3. Sintaxa PHP

2.4. Formulare PHP



2.1. Limbajul PHP - introducere

- Exemplu de cod/pagina PHP

| La server | La client |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre><!DOCTYPE html> <html> <body> <h1> Tehnologii internet </h1> <?php echo '<p>Hello World!</p>'; ?> </body> </html></pre> | <pre><!DOCTYPE html> <html> <body> <h1> Tehnologii internet </h1> <p>Hello World!</p> </body> </html></pre> |



2.1. Limbajul PHP - introducere

- PHP - **P**HP: **H**ypertext **P**reprocessor
- Limbaj de scripting open-source folosit în dezvoltarea site-urilor web
- Paginile PHP conțin cod HTML, CSS, JavaScript în care este inserat cod PHP.
- Într-o pagină poate alterna codul HTML cu codul PHP
- Codul PHP dintr-o pagină web este interpretat la server
- Un bloc PHP începe cu instrucțiunea de procesare `<?php` și se termină cu `?>`
- Fișierele PHP au extensia `.php`



2.1. Limbajul PHP - introducere

Capabilități

- Generarea dinamică a conținutului
- Lucrul cu fișierele de la server (creare, citire, scriere, ștergere)
- Prelucrarea datelor obținute în urma completării și trimiterii formularelor HTML
- Manipularea datelor din baze de date
- Controlarea accesului utilizatorilor
- Criptarea datelor
- Trimiterea și primirea cookie-urilor



2.2. Limbajul PHP - istoric

- 1994 – Rasmus Lerdorf pune bazele limbajului
- Personal Home Page/Forms Interpreter (PHP/FI)
- 1995 – v1.0 – Personal Home Page Tools (PHP Tools)
- 1997 – v2.0
- 1998 – v3.0
- 2000 – v4.0 – Zend engine
- 2004 – v5.0 – Zend engine II
- 2014 – v5.6 – Zend engine II
- 2015 – v7.0 – Zend engine III
- Ultima versiune v7.2 (30.11.2017)



2.3. Sintaxa PHP

- O secvență de cod PHP începe cu instrucțiunea de procesare `<?php` și se termină cu `?>`
- Instrucțiunile PHP se termină cu ;
- Toate cuvintele cheie (e.g., if, else, for, echo, ...), funcțiile (inclusiv cele definite de utilizator) și clasele NU sunt case-sensitive
- Numele de variabilă sunt case-sensitive



2.3. Sintaxa PHP

```
<!DOCTYPE html>
<html>
<body>
    <h1>Tehnologii internet</h1>
    <p><?php echo 'Hello World!'; ?></p>
</body>
</html>
```



2.3. Sintaxa PHP

Comentarii

- Sunt ca în C/C++/Java
- Pentru a comenta o singură linie de cod: `//`
- Pentru a comenta mai multe linii de cod: `/*` `*/`
- Pentru a comenta o singură linie de cod: `#`



2.3. Sintaxa PHP

Variabile

- Sunt similare cu cele din JavaScript
- NU există o modalitate de a declara o variabilă, în schimb aceasta este creată în momentul primei utilizări
- Toate variabilele încep cu simbolul **\$**
- Numele de variabilă sunt case-sensitive



2.3. Sintaxa PHP

Tipuri de variabile

- **Locale** – accesibile doar în funcția în care au fost create
- **Globale** – accesibile doar în exteriorul funcțiilor
- **Static** – variabilă locală care nu este ștearsă la ieșirea din funcție
- Accesarea unei variabile globale într-o funcție:
 - `global $nume_var;`
 - `$GLOBALS['nume_var']` sau `$GLOBALS["nume_var"]`



2.3. Sintaxa PHP

Constante

- Sunt globale
- Nu își modifică valoarea
- `define(nume, valoare, caseInsensitive);`



2.3. Sintaxa PHP

<?php

```
define("UNU", 1, true); // constantă
$x = 10;                // var globală
$y = 20;                // var globală
function f() {          // definirea funcției f
    static $w = 1;      // var statică
    $z = 30;            // var locală
    $z = $z + $w;
    global $y;          // accesarea unei var globale
    $y = $y + $GLOBALS['x'];
    $w = $y + $z + unu;
    echo "<p>ok: $y, $z, $w - eroare: $x</p>";
}
f(); f(); f();          // apelarea de 3 ori a funcției f
// aici nu se pot accesa $w și $z
echo "<p>Var x inafara functiei: $x</p>";
```

?>



2.3. Sintaxa PHP

Tipuri de dată

- String – între apostrofuri sau ghilimele
- Integer – 4 octeți
- Float – maxim 1.8e308, 14 zecimale
- Boolean – true sau false
- Array – `$fructe = array("mere", "pere", "zmeura");`
- Object – instanță a unei clase
- NULL
- Resource



2.3. Sintaxa PHP

Afișarea

- Instrucțiunea `echo`
 - Nu returnează nimic
 - Acceptă mai mulți parametri
 - `echo val1, val2, val3;`
- Instrucțiunea `print`
 - Returnează 1
 - Are un singur parametru
 - `print val;`
- Instrucțiunea `echo` este mai rapidă



2.3. Sintaxa PHP

Funcții

- `var_dump($nume_var)` – returnează tipul și valoarea variabilei
- `isset($nume_var)` – determină dacă o variabilă este setată și nu este null
- `empty($nume_var)` – determină dacă o variabilă este un șir vid, false, array(), null, 0 sau nesetată
- `is_null($nume_var)` – returnează true doar dacă variabila este null



2.3. Sintaxa PHP

Șiruri de caractere - String

Specificarea unui șir de caractere:

- Între apostrofuri
 - e.g., `'un sir $x'` va rezulta șirul: `un sir $x`
- Între ghilimele - șirul este interpretat
 - e.g., `"un sir $x"` va rezulta șirul: `un sir 0`
- Sintaxa heredoc (după `<<<` id-ul poate fi între ghilimele)
- Sintaxa nowdoc (după `<<<` id-ul este între apostrofuri)
 - `<<<IDENTIF`
`mai_multe_linii`
`IDENTIF;`



2.3. Sintaxa PHP

Funcții pentru manipularea șirurilor de caractere

- `strlen(șir)` – lungimea șirului
- `str_word_count(șir)` – numărul de cuvinte
- `strpos(șir, șirDeCautat)` – poziția unui șir în alt șir
- `substr(șir, start, lungime)` – subșir
- `htmlspecialchars(șir, ...)` – face conversia unui șir care conține caracterele predefinite & ' " < > la un șir cu entități HTML (&#amp; " ' < >)
- `preg_match(pattern, șir)` – expresii regulate
- `strtok(șir, token)` – împarte șirul în subșiruri delimitate de token



2.3. Sintaxa PHP

Operatori - *între paranteze este pusă precedența*

- Operatori de grupare (0)
 - () – controlarea precedenței
- Operatori de acces (1)
 - [] – accesarea elementelor unui array
 - new, clone
- Operatorul de ridicare la putere (2)
 - ** – exponențierea: \$a ** \$b



2.3. Sintaxa PHP

Operatori

- Operatori postfixați și prefixați (3)
 - `$a++`, `$a--` – incrementare și decrementare
 - `++$a`, `--$a` – incrementare și decrementare
- Operatorul `instanceof` (4)
 - `instanceof` – testează dacă un obiect este instanță a unei clase
- Operatori unari (*operator expresie*) (5)
 - `+`, `-`, `~`, `!`



2.3. Sintaxa PHP

Operatori

- Operatori aritmetici (6 și 7)
 - `*`, `/`, `%` `+`, `-`, `.` (*op . concatenează string-uri*)
- Operatori binari de shift-are (8)
 - `<<`, `>>`
- Operatori relaționali (9)
 - `<`, `>`, `<=`, `>=`



2.3. Sintaxa PHP

Operatori

- Operatori de egalitate (10)
 - ==, !=, <> – convertește operanzii dacă nu sunt de același tip și apoi aplică comparația strictă
 - ===, !== – compară operanzii fără conversia de tip
- Operatori binari (11, 12, 13)
 - &, ^, |
- Operatori logici (14, 15)
 - &&, ||



2.3. Sintaxa PHP

- Operatorul condițional (ternar) (16)
 - *(condiție)?dacă_este_adevărată:dacă_este_falsă*
- Operatori de atribuire (17)
 - `= += -= *= **= /= .= %= &= |= ^= <<= >>= =>`
- Operatori logici (18, 19, 20)
 - `and, xor, or`
- Operatorul virgulă (21)
 - `,` - permite evaluarea mai multor expresii și returnează rezultatul dat de ultima expresie



2.3. Sintaxa PHP

Intruțiuni

- Instrucțiunile (cuvintele cheie) următoare se comportă la fel ca în limbajele C/C++
 - if, else, switch, case, break
 - for, while, do... while
- Instrucțiunea `elseif` – în loc de `else if`
- Instrucțiunea `foreach($vector as $elem) { }`

```
$fructe = array("mere", "pere", "zmeura");  
foreach ($fructe as $f) {  
    echo "$f <br>";  
}
```



2.3. Sintaxa PHP

Funcții

- Similar cu funcțiile din JavaScript
- Funcțiile pot avea parametrii implicați
- Numele de funcții NU sunt case-sensitive



2.3. Sintaxa PHP

Vectori – array()

- Vectori indexați

```
$fructe = array("mere", "pere", "zmeura");  
$fructe[0] = "capsuni";  
count($fructe); // 3
```

- Vectori multidimensionali

```
$fructe = array(  
    array("mere", 3),  
    array("pere", 2),  
    array("zmeura", 15)  
);
```



2.3. Sintaxa PHP

Vectori - array()

- Vectori asociativi

```
$varste = array (  
    "Ion" => "25", "Maria" => "23"  
);  
$varste ['Ion'] = "35";  
foreach ( $varste as $cheie => $valoare ) {  
    echo "cheie=" . $cheie .  
        ", valoare=" . $valoare;  
    echo "<br>";  
}
```



2.3. Sintaxa PHP

Funcții de sortare a unui vector

- `sort()` – sortare crescătoare
- `rsort()` – sortare descrescătoare

Sortarea vectorilor asociativi:

- `asort()` – sortare crescătoare după valoare
- `ksort()` – sortare crescătoare după cheie
- `arsort()` – sortare descrescătoare după valoare
- `krsort()` – sortare descrescătoare după cheie

Nu se creează un vector nou sortat



2.3. Sintaxa PHP

Variabile globale – Superglobals

Sunt vectori asociativi

- **\$GLOBALS** – accesarea variabilelor globale de oriunde
- **\$_SERVER** – informații referitoare la header-e HTTP, căi și locații (e.g., `$_SERVER['REMOTE_ADDR']`)
- **\$_ENV** – variabile de mediu
- **\$_COOKIE** – cookie-uri HTTP
- **\$_SESSION** – variabile de sesiune



2.3. Sintaxa PHP

Variabile globale – Superglobals

- **\$_POST** – accesarea datelor trimise prin metoda POST în urma completării unui formular HTML
- **\$_GET** – similar cu \$_POST, doar că este folosită metoda GET
- **\$_FILES** –care conține fișiere uploadat-e prin metoda POST
- **\$_REQUEST** – conține informațiile din \$_POST, \$_GET, \$_COOKIE



2.4. Fișiere în PHP

Includerea conținutului unui fișier PHP în alt fișier PHP

- `include 'numeFișier';`
- `require 'numeFișier';`
- Dacă fișierul nu există, instrucțiunea `include` va da un warning și scriptul va continua, iar instrucțiunea `require` va genera eroare și scriptul își va opri execuția
- Similar cu `#include` din C/C++



2.4. Fișiere în PHP

Inserarea întregului conținut al unui fișier într-un fișier PHP

- `readfile("numeFișier");`
- Similar cu instrucțiunile `include` și `require`, cu observația că nu este parsat (interpretat la server) conținutul fișierului inserat



2.4. Fișiere în PHP

Inserarea conținutului unui fișier într-un fișier PHP

- `fopen("numeFișier", "mod");`
 - Returnează un obiect care reprezintă resursa
 - Moduri: `r`, `w`, `a`, `x`, `r+`, `w+`, `a+`, `x+`
- `fread(var, nrOcteți);`
- `fclose(var);`
- `fgets(var);` // linie de text
- `fgetc(var);` // caracter
- `feof(var);` // sfârșitul fișierului
- `fwrite(var, șir);`



2.4. Fișiere în PHP

Încărcarea unui fișier la server dintr-un formular HTML

- În fișierul `php.ini`, trebuie activată directiva:
 - `file_uploads = On`
- Alte directive:
 - `upload_max_filesize`, `upload_tmp_dir`,
`post_max_size`, `max_input_time`



2.4. Fișiere în PHP

Încărcarea unui fișier la server dintr-un formular HTML

- Tag-ul form trebuie să aibă attributele:
 - `<form action="upload.php" method="post" enctype="multipart/form-data">`
- Pentru a selecta fișierul de încărcat:
 - `<input type="file" name="unFisier" id="unFisier">`



2.4. Fișiere în PHP

Încărcarea unui fișier la server dintr-un formular

- Variabila globală `$_FILES` – vector asociativ
- `$_FILES['unFișier']['cheie']`
- Cheia poate fi:
 - `name` – numele fișierului de la client
 - `type` – tip MIME (e.g., `image/gif`)
 - `size` – dimensiunea fișierului (în octeți)
 - `tmp_name` – numele temporar al fișierului de la server (unde este stocat temporar)
 - `error` – codul de eroare
 - `UPLOAD_ERR_OK`, `UPLOAD_ERR_INI_SIZE`,
`UPLOAD_ERR_NO_FILE`, `UPLOAD_ERR_CANT_WRITE`, ...



2.4. Fișiere în PHP

Încărcarea unui fișier la server dintr-un formular

- Informații despre o cale
 - `pathinfo(cale, opțiuni)`;
 - Op: `PATHINFO_DIRNAME`, `PATHINFO_BASENAME`, `PATHINFO_EXTENSION`, `PATHINFO_FILENAME`
- Numele părții de final dintr-o cale
 - `basename(cale, sufix)`;



2.4. Fișiere în PHP

Încărcarea unui fișier la server dintr-un formular

- Verificarea dacă un fișier există la server
 - `file_exists(cale);`
- Mutarea unui fișier încărcat la o nouă locație
 - `move_uploaded_file(fișier, destinație);`



2.4. Fișiere în PHP

Încărcarea unui fișier la server dintr-un formular HTML prin metoda PUT

- Se citește de la intrarea standard (stdin)

```
<?php
```

```
$fin = fopen("php://input", "r");  
$fout = fopen("f.out", "w");  
while ($data = fread($fin, 1024))  
    fwrite($fout, $data);  
fclose($fout);  
fclose($fin);
```

```
?>
```



3. Cookie

Cookie

- Informație trimisă de serverul web către client (browser) care este stocată de client și trimisă la server când utilizatorul accesează site-ul respectiv
- Fiecare cookie are un timp de expirare dat în secunde (e.g., `time()+86400`)
- Dacă timpul de expirare este omis sau este 0, cookie-ul va expira la sfârșitul sesiunii (închiderea browser-ului)



3. Cookie

Utilizare

- Managementul sesiunilor
- Personalizarea site-ului
- Tracking – urmărirea comportamentului utilizatorului pe un anumit site



3. Cookie

Creare și utilizarea

- Răspunsul primit de la serverul web conține header-ul: `Set-Cookie`
- Cererile următoare către serverul web respectiv vor conține header-ul: `Cookie`

Exemplu:

- Server → Client
 - Set-Cookie: `SID=31d4d96e407aad43`
- Client → Server
 - Cookie: `SID=31d4d96e407aad43`



3. Cookie

Sintaxa Cookie

Cookie: *nume=valoare* [*; nume2=val2* [...]]

Exemplu

Cookie: UID=6fb0703e-81.196.26.146-1367091656; UIDR=1398630478



3. Cookie

Proprietatea cookie din DOM (JavaScript)

- `document.cookie`
- Permite setarea unor cookie-uri prin specificarea unui șir de caractere cu sintaxa de la header-ul Set-Cookie
- Returnează un șir de caractere care reprezintă cookie-urile din documentul curent având sintaxa de la header-ul Cookie
- Pentru a șterge un cookie trebuie setată o dată de expirare din trecut



3. Cookie

Cookie-uri cu PHP

- Crearea, modificarea și ștergerea unui cookie se fac cu ajutorul funcției `setcookie`

`setcookie(nume, valoare, datăExpirare, cale, domeniu, secure, httponly);`

- Cookie-urile se pot obține cu ajutorul variabilei globale `$_COOKIE[nume]`



Bibliografie

- <http://www.w3schools.com/php/default.asp>
- <http://httpd.apache.org/>
- <http://php.net/>
- <http://php.net/manual/en/>
- <http://php.net/manual/en/features.file-upload.php>
- <http://php.net/manual/en/features.file-upload.errors.php>
- <http://computer.howstuffworks.com/cookie.htm>
- <http://www.nczonline.net/blog/2009/05/05/http-cookies-explained/>
- http://www.w3schools.com/js/js_cookies.asp