

**Nota:** Notiunile teoretice se regasesc in cursul 10 de pe [www.invata-online.ro](http://www.invata-online.ro)

## I. Cookie-uri. Notiuni de baza

1. Accesati [www.google.ro](http://www.google.ro). Alegeti setarea Google.ro oferit în: Deutch. Vizualizati cookie-urile: *Tools --> Page Info --> Security --> View Cookies*.  
Inchideti browserul, redeschideti-l si accesati din nou [www.google.ro](http://www.google.ro). Ce observati? S-a pastrat setarea de limba?  
Stergeti manual cookie-urile si dati refresh in browser. Ce observati?  
Dezactivati cookie-urile in browser: *Tools --> Page Info --> Permission --> Set Cookies = Block* si incercati sa reparcurgeti etapele de mai sus.

**Concluzie:** Un **cookie** este o informatie de forma **nume=valoare** setata intr-un raspuns http (de catre serverul web) si stocata de catre browserul web pe calculatorul utilizatorului, pe harddisk, sub forma unui fisier de tip text (pe windows: **C:\documents and settings\your computers name\cookies**). De fiecare data cand utilizatorul web se intoarce la pagina web pentru care a fost setat acel cookie, browserul trimite inapoi catre server cookie-ul de forma **nume=valoare**; astfel, in aplicatia web se vor sti informatii despre utilizator, setate anterior.

### 2. Creati scriptul **creeaza\_cookie.php** in care

- veti crea un cookie cu urmatoorii parametrii: `$name='user'; $value='ion'; $expire=time()+24*60*60;//expira peste o zi.`
    - veti folosi functia **setcookie(\$name, \$value, \$expire)**
      - **\$name** - numele cookie-ului; devine ulterior cheie a superglobalului `$_COOKIE`
      - **\$value** - informatia pastrata de cookie; valoarea cookie-ului
      - **\$expire** - momentul de expirare al cookie-ului, exprimat in Unix Timestamp (nr de secunde de la 1 Ianuarie 1970). Adaugam numarul de secunde peste care vrem sa expire cookie-ul la rezultatul functiei **time()** (nr de secunde pana in prezent). Daca nu specificam acest parametru, si implicit el va fi 0, cookie-ul expira la inchiderea browser-ului.
  - daca a fost creat cookie-ul (isset(\$\_COOKIE['user'])), afisati informatia acestuia (print\_r(\$\_COOKIE))
    - informatiile din cookie sunt disponibile in array-ul asociativ `$_COOKIE`, variabila de tip superglobal.
  - verificati existenta cookie-ului si in browser - in Firefox (Windows, versiunea 1.9): *Tools --> Page Info --> Security --> View Cookies*
3. Creati scriptul **sterge\_cookie.php** in care veti sterge cookie-ul creat anterior in **creeaza\_cookie.php**.  
**Nota:** Un cookie se sterge folosind aceiasi parametri ca la setare, insa cu o data de expirare in trecut. E de preferat ca data de expirare sa fie mult in trecut, pentru ca posibil ca timpul de pe server sa fie diferit de timpul de pe client
  4. Creati cate un link care sa lege cele doua scripturi.

## II. Cookie-uri. Aplicatie – preferinte utilizator

### a) Creati scriptul **seteaza\_preferinte.php** in care:

1. Creati un formular prin care utilizatorului i se prezinta un meniu derulant din care poate selecta culoarea de fundal si un altul pentru a selecta culoarea fontului. Meniul afiseaza culorile in forma textuala, dar valorile sunt coduri hexa de culoare. Puteti folosi urmatorul array:  
`$culori=array('#ffffff'=>'Alb','#00cc00'=>'Verde','#0000ff'=>'Albastru','#cc0000'=>'Rosu','#000000'=>'Negru');`
2. Declarati variabila fanion care sa indice daca au fost expediate cookie-urile:  
`$cookies=FALSE;`
3. Daca formularul a fost trimis;

- creati cookie-urile 'bg\_color' si 'font\_color', fiecare avand drept valoare culoarea corespunzatoare preluata din formular
  - veti folosi functia **setcookie(\$name, \$value, \$expire)**
- dupa expedierea formularului, se expediază cookie-urile, deci veti atribui valoarea TRUE variabilei \$cookies
- 4. Daca au fost expediate cookie-urile (\$cookies==TRUE), transmiteti un mesaj corespunzator.
- 5. Incarcati scriptul pe server si verificati in browser ca au fost transmise cookie-urile (Firefox v. 1.9: Tools --> Page Info --> View Cookies)
- b) Creati scriptul **vezi\_preferinte.php** in care:
  1. veti crea codul CSS pentru a aplica preferintele utilizatorului
    - daca scriptul poate accesa un cookie cu numele **bg\_color** (isset(\$\_COOKIE['bg\_color'])), va tipari valoarea cookie-ului respectiv ca valoare pentru proprietatea background-color. Daca nu exista un astfel de cookie, PHP va tipari o culoare prestabilita (alb, reprezentata prin #ffffff)
    - daca scriptul poate accesa un cookie cu numele **font\_color** (isset(\$\_COOKIE['font\_color'])), va tipari valoarea cookie-ului respectiv ca valoare pentru proprietatea color. Daca nu exista un astfel de cookie, PHP va tipari o culoare prestabilita (negru, reprezentata prin #000000)
  2. veti crea un link catre **seteaza\_preferinte.php** (de asemenea, din **eteaza\_preferinte.php** veti avea un link catre **vezi\_preferinte.php**)
  3. incarcati scriptul pe server si verificati functionalitatea
- c) Reveniti la scriptul **seteaza\_preferinte.php** si modificati funtia setcookie() pentru 'bg\_color' astfel incat cookie-ul sa expire dupa 30 de secunde. Incarcati scriptul pe server si verificati inainte si dupa scurgerea celor 30 de secunde.
  - veti folosi functia setcookie() cu trei parametri: **setcookie(\$name, \$value, \$expire)**
  - **\$expire** - momentul de expirare al cookie-ului, exprimat in Unix Timestamp (nr de secunde de la 1 Ianuarie 1970). Adaugam numarul de secunde peste care vrem sa expire cookie-ul la rezultatul functiei time() (nr de secunde pana in prezent). Daca nu specificam acest parametru, si implicit el va fi 0, cookie-ul expira la inchiderea browser-ului.
- d) Creati scriptul **reset\_preferinte.php** care va sterge cookie-urile expediate si va afisa un mesaj corespunzator cu un link catre **vezi\_preferinte.php** care va afisa din nou culorile prestabilite.
  - Un cookie se sterge folosind aceiasi parametri ca la setare, inasa cu o data de expirare in trecut. E de preferat ca data de expirare sa fie mult in trecut, pentru ca posibil ca timpul de pe server sa fie diferit de timpul de pe client. In plus, valoarea ei poate fi setata ca sirul gol "" sau valoarea booleana FALSE.
  - Ex: \$day=24\*60\*60; setcookie('bg\_color',"",time()-30\*\$day);

---

### III. Sesiuni. Notiuni

Setari legate de sesiuni:

- *session.save\_path* – calea catre folderul in care se vor salva sesiunile
- *session.use\_cookies* – daca are valoarea 0, transmiterea ID-ului de sesiune de la o pagina la alta se poate face prin URL (GET)
- *session.use\_trans\_sid* – daca are valoarea 1 si session.use\_cookies=0, interpretorul PHP adauga automat id-ul de sesiune si valoarea sa la toate linkurile din scripturile in care pornim o sesiune; se poate seta si in script: **ini\_set('session.use\_trans\_sid', 1);**
- *session.use\_only\_cookies* – daca are valoarea 1, transmiterea ID-ului de sesiune de la o pagina la alta se poate face numai printr-un cookie, daca are valoarea 0, transmiterea ID-ului de sesiune de la o pagina la alta se face prin URL (GET). Modul de transmitere a ID-ului se poate seta si in script: **ini\_set('session.use\_only\_cookies', 1);**
- *session.name* – numele predefinit al unei sesiuni; se poate seta si in script cu **session\_name('nume\_sesiune');**

1. Creati scriptul **create\_session.php** in care:
  - veti deschide sesiunea
    - **`$t = session_start();`**
  - Daca sesiunea a fost pornita cu succes (\$t=TRUE)
    - veti afisa numele si ID-ul sesiunii: `session_name(), session_id();`
    - veti salva date in sesiune: `$_SESSION["username"] = 'ion'; $_SESSION["uid"] = 33;`
    - Vetii crea un link catre **view\_session.php**.
2. Creati scriptul **view\_session.php** in care:
  - veti (re)porni sesiunea: `$t = session_start();`
  - Daca sesiunea a fost (re)pornita cu succes (\$t=TRUE)
    - veti afisa numele si ID-ul sesiunii: `session_name(), session_id();`
    - veti afisa continutul variabilei superglobale `$_SESSION`: `print_r($_SESSION);`

#### IV. (optional)

1. Creati scriptul **pagina1.php** in care:
  - veti seta ca id-ul de sesiune sa se transmita numai prin cookie:
    - **`ini_set('session.use_only_cookies', 1);`**
  - veti seta un nume pentru sesiune, altul decat cel predefinit
    - **`session_name('mysession');`**
  - veti deschide sesiunea (**setarile de sesiune se fac inainte de deschiderea acesteia**)
    - **`$t = session_start();`**
  - Daca sesiunea a fost pornita cu succes (\$t=TRUE)
    - veti afisa numele si ID-ul sesiunii: `session_name(), session_id();`
    - veti salva date in sesiune: `$_SESSION["username"] = 'ion'; $_SESSION["uid"] = 33;`
  - Vetii crea un link catre **pagina2.php**.
2. Creati scriptul **pagina2.php** in care:
  - veti avea aceleasi setari de sesiune ca si **pagina1.php**
  - veti (re)porni sesiunea: `$t = session_start();`
  - Daca sesiunea a fost (re)pornita cu succes (\$t=TRUE)
    - veti afisa numele si ID-ul sesiunii: `session_name(), session_id();`
    - veti afisa continutul variabilei superglobale `$_SESSION`: `print_r($_SESSION);`
  - Vetii crea un link catre **pagina1.php**.
3. Transformati cele doua scripturi astfel:
  - setati durata de viata a cookie-ului la 5 secunde: **`session_set_cookie_params(5);`** (inainte de `session_start()`) - durata de viata se masoara din momentul crearii cookie-ului.
  - rulati scripturile pe server
  - prelungiti viata cookie-ului de sesiune cu 5 secunde de la ultimul click:  
**`setcookie(session_name(),session_id(),time()+5);`** (dupa `session_start()`)

#### V. Sesiuni. Miniaplicatie login

Creati o aplicatie care permite vizitatorului urmatoarele:

1. sa vizualizeze continutul destinat publicului
2. sa faca login;
3. in caz de login reusit
  - vizualizeaza continut particularizat
  - face logout

## Web2, curs08, Aplicatii

Puteti parcurge urmatoarele etape:

1. creati baza de date mylogin si importati mylogin.sql
  2. creati scriptul **login.php**
    - a) formular cu campurile *username*, *password*, *submit*; *username*, *password* campuri obligatorii
    - b) daca s-a apasat butonul submit
      1. preiau din formular username, password
      2. daca ambele sunt completate
        - caut in baza de date utilizatorul cu username si password
        - daca am gasit, pornesc sesiunea, memorez in \$\_SESSION['uid'] id-ul utilizatorului, redirect catre welcome.php
        - altfel mesaj de eroare
  3. creati scriptul **welcome.php**
    - a) daca nu s-a facut login (nu este setata var \$\_SESSION['uid'])
      - mesaj general de bun venit
      - link catre login.php
    - b) altfel
      - mesaj de bun venit + numele userului
      - link catre logout.php
  4. creati scriptul **logout.php**
    - a) sterge sesiunea
      1. session\_start();//nu se poate sterge sesiunea pana cand aceasta nu e activata
      2. unset(\$\_SESSION);//sterge variabila sesiunii
      3. session\_destroy();//elimina din server fisierele propriuizise ale sesiunii
    - b) redirectare catre welcome.php
- 

### VI. Sesiuni. Cos de cumparaturi

Creati o aplicatie care afiseaza un catalog de produse (descriere si pret) si permite utilizatorului sa duca in cosul de cumparaturi produsele pe care doreste sa le cumpere. Utilizatorul poate vizualiza cosul de cumparaturi: descrierea produsului, pretul unitar, cantitatea, pretul total pe produs si valoarea totala a produselor, dupa care poate continua cumparaturile sau poate face una dintre urmatoarele operatii: actualizeaza cosul de cumparaturi (modifica cantitatea pentru anumite produse), sterge un produs sau sterge cosul de cumparaturi. Pentru realizarea aplicatiei, puteti parcurge urmatoarele etape:

1. Creati scriptul **produse.php** in care initializati array-ul \$produse

```
$produse = array(
    array('id' => '1', 'desc' => 'Dictionar englez-roman', 'pret' => 24.95),
    array('id' => '2', 'desc' => 'Songs of the Goldfish (set 2CD)', 'pret' => 100),
    array('id' => '3', 'desc' => 'PHP pentru World Wide Web', 'pret' => 35),
    array('id' => '4', 'desc' => 'Dictionar roman-englez', 'pret' => 39.95)
);
```
2. Creati scriptul **catalog.php** care va include **produse.php** si va afisa sub forma de tabel lista de produse – descriere si pret. In dreptul fiecarui produs se va afisa si un buton **Cumpara**. Scriptul va contine si un link “Vezi cosul de cumparaturi” care va transmite query stringul “cos”.  
**Nota:** in formularul care contine butonul Cumpara, veti crea si un camp de tip **hidden** care va memora in atributul value id-ul produsului selectat. Se va folosi POST ca metoda de transmitere a formularului.
3. Creati scriptul **index.php**.
  1. pentru a avea acces la lista de produse, se va include scriptul **produse.php**  
`include "produse.php";`
  2. pentru a memora lista de produse pe care utilizatorul le adauga in cosul de cumparaturi pe

toata perioada in care acesta viziteaza site-ul, se va initializa o sesiune si se vor initializa datele sesiunii: `$_SESSION['cos_id']` care va contine id-ul produselor din cosul de cumparaturi si `$_SESSION['cos_cant']` care va memora cantitatea pentru fiecare produs din cos (`$_SESSION['cos_cant'][$id]` – numarul de unitati pentru produsul cu id-ul `$id`)

3. pentru a vizualiza catalogul de produse, la sfarsitul scriptului se va include scriptul **catalog.php**: `include "catalog.php";`
4. Dupa initializarea sesiunii si a datelor acesteia, daca se selecteaza butonul Cumpara, se va adauga in array-ul `$_SESSION['cos_id']` id-ul produsului selectat daca acesta nu exista inca si se va initializa `$_SESSION['cos_cant'][$id]` cu 1 sau se va incrementa `$_SESSION['cos_cant'][$id]` cu 1 daca produsul exista deja.
 

```
if(isset($_POST['cumpara'])){
    $id=$_POST['id'];
    if(!in_array($id,$_SESSION['cos_id'])){
        $_SESSION['cos_id'][]=$id;
        $_SESSION['cos_cant'][$id]=1;
    }
    else{
        $_SESSION['cos_cant'][$id]++;
    }
}
```
5. Daca utilizatorul a ales sa vizualizeze cosul de cumparaturi (a dat click pe link si s-a transmis prin get query stringul `cos`), se va afisa cosul de cumparaturi:
 

```
if(isset($_GET['cos'])){
    include "cos_cumparaturi.php";
    exit();
}
```

Se va crea scriptul **cos\_cumparaturi.php** care va afisa sub forma de tabel lista de produse din cosul de cumparaturi. Tabelul va fi inclus intr-un formular cu `action="?update"` si `method="post"`. Formularul va contine si butonul “Actualizeaza” la apasarea caruia se va transmite prin `$_POST` datele pentru actualizarea cosului de cumparaturi.

Tabelul va contine urmatoarele coloane:

1. sterge – link care va transmite query stringul “sterge” si id-ul produsului ce va fi sters
2. descriere produs (se va prelua din `$produse` pe baza id-ului din `$_SESSION['cos_id']`)
3. pret unitar (se va prelua din `$produse` pe baza id-ului din `$_SESSION['cos_id']`)
4. cantitate – input de tip text (se va prelua din `$_SESSION['cos_cant']`)
  1. name pentru acest camp va fi **cantitate[]** pentru ca vrem sa se transmita prin POST un array cu cantitatea asociata fiecarui produs
  2. vom avea aici si un camp hidden cu `name="id[]"` pentru a prelua si id-urile produselor
5. pret total = pret unitar \* cantitate

La sfarsitul tabelului se va afisa linia **Total comanda** care va contine valoarea totala a comenzii.

6. Reveniti la scriptul **catalog.php** si afisati numarul de produse din cosul de cumparaturi:
 

```
echo '<p>Cosul dvs. de cumparaturi contine '. nrProduse($_SESSION['cos_cant']). ' produse.</p>';
```

`nrProduse($cantitate)` – insumeaza valorile di array-ul `$cantitate`.

7. Reveniti la scriptul **cos\_cumparaturi.php** si adaugati un link “Continua cumparaturile” care va afisa `index.php` fara query string si un buton de tip submit “Goleste cosul” care va “goli” cosul de cumparaturi.
8. Reveniti la **index.php** si tratati:
  - situatia in care se apasa butonul “Goleste cosul” - se va sterge sesiunea si se va redirecta catre scriptul curent cu query stringul `cos` (se va afisa cosul de cumparaturi, care acum va fi gol)
  - situatia in care se apasa butonul “Actualizeaza” - se vor prelua datele transmise prin POST (array-ul `$_POST['id']` si array-ul `$_POST['cantitate']`) si pe baza lor se va actualiza

```
$_SESSION['cos_cant']:  
    if(isset($_POST['actualizeaza'])) {  
        //preiau datele din $_POST  
        $cos_id = $_POST['id'];  
        $cos_cant = $_POST['cantitate'];  
  
        //actualizez $_SESSION['cos_cant'];  
        $nr = count($cos_id);  
        for($i=0; $i<$nr; $i++){  
            $id = $cos_id[$i];  
            $cant[$id] = $cos_cant[$i];  
        }  
  
        $_SESSION['cos_cant'] = $cant;  
        include "cos_cumparaturi.php";  
        exit();  
    }  
}
```

- situatia in care se alege sa se stergea un produs din cosul de cumparaturi – se preia id-ul produsului din \$\_GET, se sterge din \$\_SESSION['cos\_id'] si \$\_SESSION['cos\_cant'] inregistrarea asociata produsului.

```
if(isset($_GET['sterge'])) {  
    $id = $_GET['id']; //id-ul produsului care se va sterge  
    $cheie = array_search($id, $_SESSION['cos_id']); //caut cheia produsului  
    unset($_SESSION['cos_id'][$cheie]);  
    unset($_SESSION['cos_cant'][$id]);  
  
    include "cos_cumparaturi.php";  
    exit();  
}
```

9. Setati durata de viata a cooki-ului de sesiune la 30 de zile, astfel incat cosul de cumparaturi sa existe si dupa inchiderea browserului, timp de o luna. In **index.php**:

```
ini_set('session.use_only_cookies', 1);
```

```
$zi = 60*60*24;  
session_start();  
setcookie(session_name(), session_id(), time() + 30*$zi, "/"); //prelungeste viata  
cookie-ului de la ultimul click
```