

## Entorns de Desenvolupament



### 17. Sessions

## 17. Sessions

Una sessió és una forma d'**emmagatzemar la informació** (**variables**) per ser utilitzada **a través de múltiples pàgines**. A un visitant que accedeix a un lloc web se li assigna un **id únic**, també anomenat **id de sessió**. Aquest és **emmagatzemat en una galeta (cookie)** a la part del client o es **propaga en l'URL**.

L'ús de sessions permet **emmagatzemar** les dades entre peticions (accessos a les pàgines web d'un mateix lloc web) en l'**array \$\_SESSION**, que és un array de sistema, igual que **\$\_POST** o **\$\_GET**.

Quan un visitant accedisca a un lloc web, PHP comprovarà a través de la funció **session\_start()**, si s'ha enviat un id de sessió específic amb la petició. Si és així, es **recupera l'entorn** anteriorment **guardat**, amb les variables que s'hagen creat i els seus valors corresponents. El valor d'aquest id, es pot obtenir amb **session\_id()** i es pot canviar amb **session\_regenerate\_id()**.

### Què és una sessió de PHP?

Quan es treballa amb una aplicació, s'obre, es fan alguns canvis, i després es tanca. Això és molt semblant a una sessió. **L'equip local sap qui som**. Se sap quan s'inicia l'aplicació i quan s'acaba. No obstant això, a Internet hi ha un problema: **el servidor web no sap qui som** o el que fem, perquè la direcció HTTP no manté estat.

Les variables de sessió resolen aquest problema mitjançant l'emmagatzematge de la informació que l'usuari utilitzarà en diverses pàgines (per exemple, nom d'usuari, preferències de visualització de la pàgina, etc). **Per defecte**, les **variables de sessió duren fins que l'usuari tanca el navegador**.

Així que, les **variables de sessió** contenen informació d'un usuari particular (amb un id únic), i estan **disponibles en totes les pàgines** en una mateixa aplicació.

Totes les funcions disponibles per a treballar amb sessions es poden trobar [ací](#), en aquest document **veurem només les funcions bàsiques** per utilitzar-les.

## Iniciar una sessió de PHP

Una sessió s'inicia amb la funció `session_start()`.

Les variables de sessió s'estableixen amb la **variable global PHP**: `$_SESSION`.

Ara, crearem una nova pàgina anomenada "pagina1.php". En aquesta pàgina, es comença una nova sessió de PHP i s'estableixen algunes variables de sessió:

### Exemple:

```
<?php // Inici de la sessió
    session_start(); // recomanat: 1a instrucció en el document php!!!
    // Definim variables de sessió
    $_SESSION["nom"] = "Ernest";
    $_SESSION["cicle"] = "DAW";
    $_SESSION["ciutat"] = "Xàtiva";
?>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Pàgina 1 Sessions</title>
    <link rel="stylesheet" href="estils.css" >
</head>
<body>
<header id="cap">
<h1>Exemple Sessions 01</h1>
</header>
<main id="contingut">
<div class="tauler">
```

&lt;p&gt;

&lt;?php // les mostrem per pantalla

echo "S'han establert les variables de sessió: &lt;br&gt;";

echo "Valor de Nom: ".\$\_SESSION["nom"]." &lt;br&gt;";

echo "Valor de Cicle: ".\$\_SESSION["cicle"]." &lt;br&gt;";

echo "Valor de Ciutat: ".\$\_SESSION["ciutat"]." &lt;br&gt;";

?&gt;

&lt;/p&gt;

&lt;/div&gt;

&lt;/main&gt;

&lt;/body&gt;

&lt;/html&gt;



**Molt important:** La crida a la funció `session_start()` és recomanable que siga **la primera instrucció en el document**. **Abans de qualsevol etiqueta HTML.**

## Obtenir valors de variables de sessió PHP

A continuació, creem una **altra pàgina** anomenada "pagina2.php". Des d'aquesta pàgina, tindrem accés a la informació de la sessió que hem declarat a la primera pàgina ("pagina1.php").

Recordeu que les variables de sessió **no es passen individualment** a cada pàgina nova, sinó que es recuperen de la sessió que obrim al principi de cada pàgina (`session_start()`). Tots els valors de les variables de sessió s'emmagatzemen a la variable global `$_SESSION`:

```
<?php // Inici de la sessió
    session_start(); // primera instrucció en el document php!!!
    //si existeixen les variables, simplement les obtenim de $_SESSION
    //assignant-ne el valor a variables locals
    $nom_sessio=$_SESSION["nom"];
    $cicle_sessio=$_SESSION["cicle"];
    $ciutat_sessio=$_SESSION["ciutat"];
?>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Pàgina 2 Sessions</title>
<link rel="stylesheet" href="estils.css">
</head>
<body>
<header id="cap">
<h1>Exemple Sessions 02</h1>
</header>
<main id="contingut">
<div class="tauler">
<p>
<?php // les mostrem per pantalla
    echo "Hem obtingut les variables de sessió fixades en la pàgina
    anterior: <br/>";
    echo "Valor de Nom: ".$nom_sessio." <br>";
    echo "Valor de Cicle: ".$cicle_sessio." <br>";
    echo "Valor de Ciutat: ".$ciutat_sessio." <br>";
?>
</p>
```

```
</div>
</main>
</body>
</html>
```

Si desitgem modificar una variable, **simplement la sobreescrivim**:

```
$_SESSION["nom"] = "Elena";
$_SESSION["cicle"] = "ASIX";
$_SESSION["ciutat"] = "Canals";
```

Quan volem eliminar les dades relacionades amb una sessió, primer hem d'eliminar totes les variables de sessió executant la funció `session_unset()` i finalment executar `session_destroy()` per destruir l'identificador de la sessió actual.

```
<?php
    session_start();
?>
<!DOCTYPE html>
<html>
<body>
<?php
    session_unset(); // elimina totes les variables de sessió
    session_destroy(); // destrueix la sessió
?>
</body>
</html>
```

## Tractament de dades en formularis

Es pot utilitzar un **fitxer php per processar les dades d'un formulari i després redirigir l'usuari** cap a una pàgina diferent (amb `header()`). En aquest cas, el fitxer php **no té perquè tindre una estructura html**, ja que **no va a ser mostrat per pantalla**.

Per exemple, suposem que en una pàgina tenim un formulari, i en funció d'un valor d'un dels seus camps redirigirà l'usuari cap a una pàgina o una altra.

En la pàgina inicial tindrem entre el seu codi aquest formulari:

```
<form id="formulari" method="POST" action="processaFormulari.php">
  <div>nom:<input id="nom" name="nom" type="text" ></div>
  <div>valor:<input id="valor" name="valor" type="number" ></div>
  <div>
    <input id="submitform" name="submitform" type="submit"
    value="Envia">
  </div>
</form>
```

El fitxer destinació de l'acció del formulari, **processaFormulari.php** s'encarrega d'obtenir els valors que l'usuari ha introduït als camps i en funció del seu contingut redirigir cap una o altra pàgina:

```
<?php
session_start(); //iniciem la sessió
$usuari = $_POST['usuari']; //obtenció dels valors d'usuari i valor
$valor = $_POST['valor'];
//introduïm usuari en una variable de sessió per usar-lo més endavant
$_SESSION['usuari']=$usuari;
//comprovem si el valor és positiu-zero o negatiu
if ($valor>=0){ //si es positiu o zero redirigim a la pàgina positiu.php
    header("Location: ./positiu.php");
    die(); //acaba l'execució de l'script
} else { //si es negatiu redirigim a la pàgina negatiu.php
    header("Location: ./negatiu.php");
    die(); //acaba l'execució de l'script
}
```

El valor del camp **usuari**, l'hem introduït en una variable de sessió **\$\_SESSION["usuari"]**, de manera que podrem utilitzar-lo en les pàgines destinació (**positiu.php** o **negatiu.php**) si en aquestes s'inicia la sessió amb **session\_start()**.

- ♦ La funció **header()** s'utilitza per enviar capçaleres HTTP sense format.

Ha cridar-se a **header()** **abans de mostrar res per pantalla**, etiquetes HTML, línies o espais en blanc des d'un fitxer o des de PHP, si no, pot donar un **error** (i **no redirigir correctament**, perquè les **capçaleres HTTP ja han sigut enviades** en el moment que es mostra un contingut).

Es controla mitjançant un **paràmetre** del fitxer de configuració **php.ini**: **output\_buffering**. És un mecanisme que controla **quantes dades** (excloent capçaleres i galetes) hauria de **mantenir PHP internament abans d'enviar** les dades **al client**.

- Si està **deshabilitat** (valor **Off**) **qualsevol eixida**, per xicoteta que siga (inclús una línia o un espai en blanc) **provocarà un error**.
- Si està **habilitat** (valor **On**) el **buffer no té límit**, però pot provocar que el funcionament no siga tampoc l'esperat.
- Si se li estableix un **valor enter**, el fixa com a la **màxima mida en bytes que es pot emmagatzemar abans d'enviar les dades al client**. Si se **supera**, es produirà també un **error**.

En moltes configuracions inicials de motor PHP, el valor per defecte en php.ini és:

**output\_buffering=4096**

Que es considera un valor adequat per **raons de rendiment**.

També es pot controlar el buffer utilitzant **funcions**:

- **ob\_start()** : quan volem que comence el buffer
- **ob\_end\_flush()**: quan volem enviar tot el buffer

**<html>**

**<?php**

**//es produirà un error: etiqueta html abans que header()**



```
header('Location: http://ieslluissimarro.org/');  
exit;
```

```
?>
```

Mostrarà:

**Warning: Cannot modify header information - headers already sent by ...**

Se li passa com a primer paràmetre la capçalera HTTP en format cadena. Hi ha **dos casos especials** en l'ús de **header**:

- 1) La capçalera que comença amb la cadena "HTTP /", utilitzat per a esbrinar el codi d'estat HTTP a enviar.

```
<?php
```

```
header("HTTP/1.1 404 Not Found");
```

```
?>
```

Altres exemples:

- 200 OK
- 401 Unauthorized
- 500 Internal Server Error
- 301 Moved Permanently
- 403 Forbidden
- 502 Bad Gateway

- 2) Capçalera **"Location:"** No solament envia la capçalera al navegador, sinó que també retorna el codi d'estat (302) REDIRECT al navegador i redirigeix a la URL indicada.

```
<?php
```

```
header("Location: http://www.google.com/");
```

```
//es redirecciona el navegador
```

```
die(); //acaba l'execució de l'script
```

```
?>
```