

# SESIONES

Programando con PHP

# Introducción

- ❑ Hasta ahora, las variables sólo existen en el archivo que se está ejecutando.
- ❑ Cuando se carga una página distinta, los valores de las variables se pierden.
- ❑ La única forma de mantenerlos era:
  - ▣ Pasarlos por la URL (header)
  - ▣ Escribirlos en cookies
  - ▣ Pasarlos por formularios (hidden)

# Introducción

- ❑ Ninguno de los tres métodos es del todo infalible.
- ❑ Necesitamos variables que no mueran mientras la sesión esté abierta y que sea accesible desde cualquier punto de la aplicación Web.
- ❑ PHP permite guardar variables llamadas variables de sesión.
- ❑ Una vez definidas podrán ser utilizadas mientras la sesión dure y desde cualquiera de los script de la aplicación.



# Variables SESSION

# Variables SESSION

- ❑ Ejemplo:
  - ▣ Un carrito de la compra puede ser un array en el que se guarde para cada artículo pedido la cantidad solicitada.
  - ▣ Cada usuario debe tener su propio carrito.
- ❑ Son específicas del usuario: puede darse el caso de que varios usuarios tengan variables SESSION del mismo tipo pero con valores distintos.
- ❑ Cada sesión tiene su propio identificador que será único.

# Variables SESSION

- ❑ Cuando un visitante accede a nuestro sitio Web, se le asigna un id único ➔ id de sesión
- ❑ Se almacena en una cookie en la parte del cliente
- ❑ Se va propagando en la cabecera del documento
- ❑ El manejo de sesiones permite almacenar datos entre peticiones en el array superglobal `$_SESSION`



# Funciones de manejo de sesiones

# Manejo de Sesiones

- ❑ Podemos decir que las sesiones son la forma de guardar información específica de cada usuario mientras éste se mueve por el sitio Web
- ❑ Cada usuario que entra en el sitio abre una sesión que será independiente de la sesión de otro usuario.
- ❑ En una sesión se puede guardar cualquier tipo de dato:
  - ▣ Nombre de usuario
  - ▣ Preferencias de visualización
  - ▣ Páginas por las que ha pasado
  - ▣ Etc...



# Manejo de Sesiones

## □ session\_start()

- ▣ Crea una sesión o reanuda la sesión actual
- ▣ Al llamar a session\_start() pueden pasar dos cosas:
  - Se crea el identificador de la sesión (si no existía)
  - Se reanuda la sesión asociada usuario
- ▣ La función devuelve
  - TRUE si se ha iniciado la sesión correctamente
  - FALSE si por algún motivo la sesión no se pudo iniciar.

# Manejo de Sesiones

- ❑ Una vez creada la sesión se puede utilizar el array `$_SESSION`.
- ❑ `$_SESSION` es un array asociativo que al crearse está vacío.
- ❑ Se pueden ir adjuntando nuevas posiciones al array.
- ❑ `$_SESSION` estará accesible mientras no se cierre sesión:
  - ▣ De forma explícita
  - ▣ Cerrando el navegador

# Manejo de Sesiones

- `Session_id()`
  - ▣ Devuelve el id de la sesión actual.
- `session_name([nombre])`
  - ▣ Si no se le pasa el parámetro “nombre”, devuelve el nombre de la sesión.
  - ▣ Si se le pasa el parámetro “nombre”, devuelve el nombre antiguo de la sesión y asigna el nuevo nombre.
  - ▣ Cada vez que se llama a la función `session_start`, el nombre de la sesión se resetea.

# Ejemplo 1

- ❑ Crear una sesión en un documento php y darle un nombre.
- ❑ Crear un valor “idioma” que se guardará como valor de sesión y mostrar:
  - ▣ Id de la sesión
  - ▣ Nombre de la sesión
  - ▣ Idioma almacenado
- ❑ En otro documento, consultar los datos de la sesión y mostrarlos en una tabla.

# Manejo de Sesiones

## □ Session\_status ()

▣ Devuelve el estado actual de la sesión.

- PHP\_SESSION\_DISABLED si el cliente tiene las sesiones deshabilitadas
- PHP\_SESSION\_NONE si el cliente tiene las sesiones habilitadas pero no existe ninguna.
- PHP\_SESSION\_ACTIVE si el cliente tiene las sesiones habilitadas y existe una.

# Manejo de Sesiones

- ❑ `Session_encode()`
  - ▣ Serializa toda la información almacenada actualmente en `$_SESSION`
  - ▣ De esta forma se puede guardar toda la información como una única cadena.

```
<?php
```

```
    session_start();
```

```
login_ok|b:1;nombre|s:5:"Maria";edad|i:34;
```

```
$_SESSION['login_ok'] = true;
```

```
$_SESSION['nombre'] = 'Maria';
```

```
$_SESSION['edad'] = 34;
```

```
echo session_encode();
```

```
?>
```

# Manejo de Sesiones

- Session\_decode(datos)
  - ▣ Realiza la acción inversa.
  - ▣ Recibe los datos serializados de una sesión y rellena con ellos la variable superglobal \$\_SESSION

# Manejo de Sesiones

## 1. Codificación

```
<?php
    session_start();

    $_SESSION['login_ok'] = true;
    $_SESSION['nombre'] = 'Maria';
    $_SESSION['edad'] = 34;

    $datos = session_encode();

    setcookie('sesion', $datos);
?>
```

## Visualización

1  
Maria  
34

## 2. Decodificación

```
<?php
    session_start();

    if(isset($_COOKIE['sesion']))
    {
        session_decode($_COOKIE['sesion']);

        echo $_SESSION['login_ok'];
        echo "<br>";
        echo $_SESSION['nombre'];
        echo "<br>";
        echo $_SESSION['edad'];
        echo "<br>";
    }
?>
```



# Manejo de Sesiones

- `Session_destroy()`
  - ▣ Destruye la sesión actual ➔ id de sesión.
  - ▣ No destruye las variables globales asociadas con la sesión.
  - ▣ Para volver a utilizarlas simplemente hay que acceder al array `$_SESSION`

```
<?php
```

```
    session_start();
```

```
$_SESSION['dato1']='Hola';
```

```
$_SESSION['dato2']='Adios';
```

```
$_SESSION['dato3']=26;
```

```
echo "Mostrando Id de sesión ";
```

```
echo session_id();
```

```
session_destroy();
```

```
echo "<br>Mostrando Id de sesión después de destruir: ";
```

```
echo session_id();
```

```
echo "<br>Datos de la sesión: <br>";
```

```
echo $_SESSION['dato1'];
```

```
echo "<br>";
```

```
echo $_SESSION['dato2'];
```

```
echo "<br>";
```

```
echo $_SESSION['dato3'];
```

```
echo "<br>";
```

```
?>
```

# Manejo de Sesiones

Mostrando Id de sesión nb4tagcrkqgmvg6a5u1ivns4t3

Mostrando Id de sesión después de destruir:

Datos de la sesión: Hola

Adios

26

```
<?php
```

```
session_start();
```

```
$_SESSION['dato1']='Hola';
```

```
$_SESSION['dato2']='Adios';
```

```
$_SESSION['dato3']=26;
```

```
echo "Mostrando Id de sesión ";
```

```
echo session_id();
```

```
$_SESSION=array();
```

```
session_destroy();
```

```
echo "<br>Mostrando Id de sesión después de destruir: ";
```

```
echo session_id();
```

```
echo "<br>Datos de la sesión: <br>";
```

```
echo $_SESSION['dato1'];
```

```
echo "<br>";
```

```
echo $_SESSION['dato2'];
```

```
echo "<br>";
```

```
echo $_SESSION['dato3'];
```

```
echo "<br>";
```

```
?>
```

# Manejo de Sesiones

Mostrando Id de sesión nb4tagcrkqgmvg6a5ulivns4t3

Mostrando Id de sesión después de destruir:

Datos de la sesión:

**Notice:** Undefined index: dato1 in **C:\wamp\www\destruir.php** on line **19**

**Notice:** Undefined index: dato2 in **C:\wamp\www\destruir.php** on line **21**

**Notice:** Undefined index: dato3 in **C:\wamp\www\destruir.php** on line **23**

# SESIONES

Programando con PHP