

PHP

**Sesión, cookies, multiidioma,
ficheros y correo**

En esta parte trataremos:

- Uso de variables de sesión
- Uso de cookies
- Incorporar multiidioma en una página web
- Manejo de ficheros

VARIABLES DE SESION I

Podemos utilizar la supervariable `$_SESSION[]` para guardar datos en la sesión del navegador y que permanecerán mientras no lo cerremos

Para trabajar con variables de sesión la primera instrucción php debe ser `session_start()`

```
<?php
    session_start();
    $_SESSION['sesion'] = 'sesión activada';
?>
```

VARIABLES DE SESION II

Para comprobar si existe una variable de sesión

```
<?php
    session_start();

    if (isset($_SESSION['sesion'])) {
        echo $_SESSION['sesion'];
    } else {
        echo 'sesion no iniciada';
    }
?>
```

VARIABLES DE SESION III

Para borrar una variable de sesión

```
<?php
    session_start();

    if (isset($_SESSION['sesion'])) {
        unset($_SESSION['sesion']);
    }
?>
```

Si utilizamos `session_destroy()` borraremos todos los datos de la sesión

EJERCICIO

Confeccionar dos páginas web (dos archivos .php)

En el primero crearemos una variable de sesión

En el segundo comprobaremos que exista la variable de sesión y visualizaremos su contenido

Posteriormente la borraremos en este segundo archivo.

COOKIES

COOKIES I

Son archivos de texto almacenados en el ordenador del cliente y que contienen información que será utilizada por la página web para cuando el cliente se vuelva a conectar. Gracias a ellas un sitio web puede almacenar información acerca de tu perfil.

```
<?php
    session_start();
    $contenido = "Juan";
    setcookie("nombreCookie", $contenido, time()+3600);
    echo $_COOKIE['nombreCookie'];
?>
```

El array `$_COOKIE` contiene las cookies en el momento de hacer la petición y **éstas no serán visibles hasta la siguiente sesión.**

COOKIES II

La sintaxis es la siguiente:

```
<?php
    //indicamos que vamos a utilizar las variables de sesión
    session_start();
    //determinamos el contenido a guardar
    $contenido = "Juan";
    //guardamos el contenido en la cookie
    setcookie("nombreCookie", $contenido, time()+3600);
    echo $_COOKIE['nombreCookie'];
?>
```

Los parámetros de setcookie son los siguientes:

- Nombre que le daremos a la cookie
- Contenido a guardar
- Tiempo de expiración (normalmente se utiliza la hora actual más el tiempo en segundos)

EJERCICIO

Realizar un contador de visitas que se incremente en uno cada vez que visitamos la página:

- Iniciar la sesión
- Comprobar si existe la cookie con `isset`
- Si no existe crear una con el contador inicializado a cero
- Si existe, recuperar el valor del contador y sumarle uno
- Asignar el nuevo valor a la cookie

BORRAR COOKIES

Para borrar una cookie utilizaremos:

```
setcookie('nombreCookie', null, -1);
```

NOTA: Si utilizamos `unset($_COOKIE['nombreCookie'])` eliminaremos el dato de la cookie en la sesión pero no borraremos la misma

GUARDAR ARRAYS EN COOKIES I

En una cookie NO podemos guardar un array. Si intentamos guardar el siguiente array asociativo:

```
$paises=array('Francia' => 'París', 'Ecuador' => 'Quito', 'Gran Bretaña' => 'Londres', 'Alemania' => 'Berlin', 'Argentina' => 'Buenos Aires');
```

Obtenemos el siguiente error:

Warning: setcookie() expects parameter 2 to be string

Para evitarlo tenemos que convertir el array a formato texto utilizando el formato JSON:

```
$listaPaises=json_encode($paises);  
setcookie("paises", $listaPaises, time()+3600);
```

GUARDAR ARRAYS EN COOKIES II

Y, posteriormente, al leer la cookie tenemos que volver a convertir el texto en array:

```
$paises = json_decode($_COOKIE['paises'], true);
```

El parámetro booleano **true** indica que queremos convertir el contenido de la cookie en un array asociativo

INCORPORAR CÓDIGO EXTERNO

INCLUDE Y REQUIRE

Permiten incluir el código de otros ficheros en nuestra página web.

La diferencia entre `require()` e `include()` es que en caso de no encontrar el fichero a incluir `include()` produce un Warning mientras que `require()` produce un Error Fatal.

La función `require_once()` se comporta de manera similar a `require()`, con la única diferencia que si el código ha sido ya incluido, no se volverá a incluir.

```
<?php
    require('archivoExterno.php');
    require('http://www.archivoExterno.php'); //no es posible incluir una url
?>
```

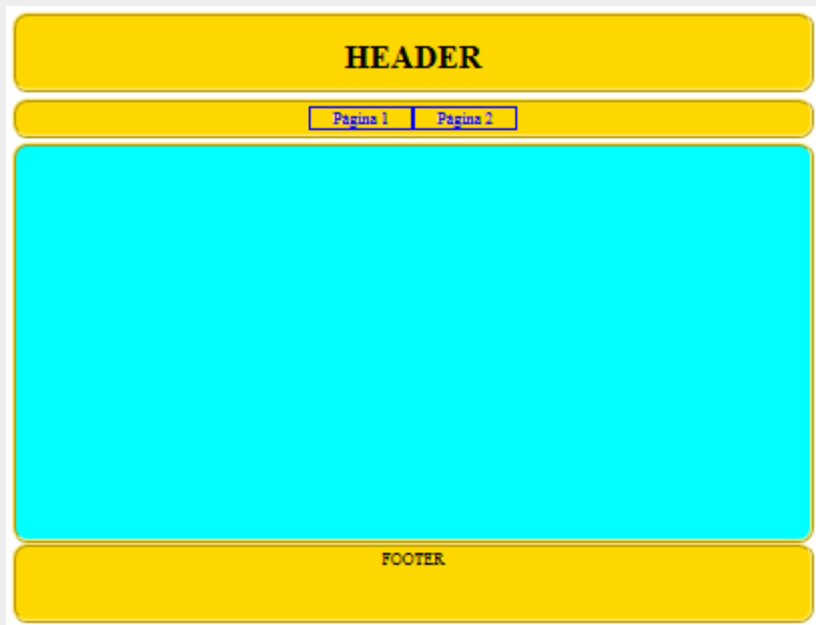
INCLUDE Y REQUIRE

Con estas funciones podemos confeccionar una página web de forma dinámica y son muy útiles para evitar repetir secciones de la página que son comunes en todo el website:

HEADER

EJERCICIO

Confeccionar una página web cuyas secciones: **header**, **nav**, y **footer** sean cargadas desde ficheros externos.



La barra de navegación tendrá dos opciones de enlace

EJERCICIO

Ahora activaremos los dos enlaces de la botonera de forma que el primero enlazará con la página principal y el segundo con otra página con contenido distinto pero con idénticas secciones **header**, **nav** y **footer**

Para realizar este ejercicio necesitaremos 5 ficheros:

- cabecera.html
- botonera.html
- pagina1.php
- pagina2.php
- footer.php



MULTIIDIOMA

IDIOMAS I

El ejercicio anterior lo vamos a transformar de forma que la página se muestre en dos idiomas:

1.- Substituiremos los textos de la página por variables php que, posteriormente almacenaremos en un fichero diccionario.

```
<section><?=$contenido?></section>
```

2.- Crearemos un fichero php que contendrá todas las variables anteriores en uno de los idiomas (por ejemplo castellano). Le daremos un nombre similar a `contenido_es.php`

```
<?php
    $contenido = 'contenido en el idioma 1';
?>
```

IDIOMAS II

3.- Replicaremos el fichero anterior para todos los idiomas de forma que, en cada uno de ellos, traduciremos los textos al idioma correspondiente. Estos ficheros los guardaremos con el mismo nombre que el fichero anterior pero con la terminación que corresponda al idioma que contiene (ejemplo: `contenido_ca.php`, `contenido_en.php`,...)

```
<?php
    $contenido = 'contenido en el idioma 2';
?>
```

4.- Ahora vamos a incorporar (normalmente en el `header`) los enlaces para que el usuario pueda seleccionar el idioma (en principio haremos que ambos enlaces apunten a la misma página)

```
<a href="pagina1.php">ES</a> | <a href="pagina1.php">CA</a>
```

IDIOMAS III

5.- En el atributo href vamos a incorporar el parámetro que enviaremos al servidor con el idioma seleccionado

```
<a href="pagina1.php?idioma=es">ES</a> | <a href="pagina1.php?  
idioma=ca">CA</a>
```

6.- Vamos a incorporar ahora el código php necesario para recoger este parámetro de la URL (y que llega, por tanto, por el método GET) y cargar el archivo con las variables que correspondan al idioma seleccionado

```
<?php  
    $idioma = "es"; //idioma por defecto  
    if (isset($_GET["idioma"])){  
        $idioma = $_GET["idioma"]; //asignamos el idioma seleccionado  
    }  
    include ("include/contenido_{$idioma}.php"); //cargamos el fichero del idioma
```

IDIOMAS IV

7.- Para proteger la página ante una manipulación del usuario de la URL introduciremos, antes de cargar el fichero del idioma, una validación que compruebe que el idioma que llega corresponda a uno para los que existe diccionario

```
$idiomasPermitidos=array('es', 'ca');
```

```
if (isset($_GET['idioma'])) {  
    if (in_array($_GET['idioma'], $idiomasPermitidos)) {  
        $idioma = $_GET["lang"]; //asignamos el idioma seleccionado  
    };  
}  
.../...
```

IDIOMAS: recordar el idioma

8.- Ahora guardaremos el idioma seleccionado en una variable de sesión para futuras entradas a la página

```
session_start(); //para acceder a las variables de sesión
```

```
$idioma='es'; //idioma por defecto
```

```
if (isset($_GET["idioma"])) { //si cambiamos de idioma..
```

```
    if (in_array($_GET['idioma'], $idiomasPermitidos)) {
```

```
        $idioma = $_GET["idioma"]; //asignamos el idioma seleccionado
```

```
        $_SESSION['idioma']= $idioma; //..guardamos el idioma en sesión
```

```
    }
```

```
} else if (isset($_SESSION["idioma"])) {
```

```
    $idioma=$_SESSION["idioma"]; //si hay un idioma en sesión lo asignamos
```

```
}
```


IDIOMAS: detectar idioma del navegador

9.- Y ahora vamos a detectar el idioma por defecto del navegador .

`session_start()` //para acceder a las variables de sesión

//seleccionar el idioma por defecto del navegador

`$idiomaNavegador=substr($_SERVER['HTTP_ACCEPT_LANGUAGE'], 0, 2);`

//si está dentro de los idiomas permitidos lo seleccionamos

`if (in_array($idiomaNavegador, $idiomasPermitidos)) {`

`$idioma=$idiomaNavegador;`

`} else {`

`$idioma='es';`

`}`

EJERCICIO

Añadir el uso de Cookies, para que el idioma seleccionado quede almacenado una vez cerrado el navegador.

HEADER

[ES](#) | [CA](#)

[Página 1](#) | [Página 2](#)

TEXTE

Ès un fet establert de forma evident que un lector es distraurà amb el contingut llegible d'una pàgina quan miri a la seva composició. El fet de fer servir Lorem Ipsum és perquè conté una distribució més o menys normal de lletres, de forma oposada a quan es fa servir "Contingut aquí, contingut aquí", aconseguint així que sembli català llegible. Molts paquets d'autoedició fan servir Lorem Ipsum com el seu model de text, i una cerca per "Lorem Ipsum" descobrirà molts llocs web encara a les baceroles. Diverses versions han anat evolucionant al llarg dels anys, a vegades per accident, a vegades a propòsit (inserir humor i coses per l'estil.

FOOTER

MULTIIDIOMA EN TODAS LAS PAGINAS

Ahora vamos a adaptar a multiidioma la página 2 del ejercicio.

1.- Añadimos el código php que controla el cambio de idioma en la segunda página

NOTA: Puesto que el código php a añadir es exactamente el mismo podemos incorporarlo en un fichero externo y añadirlo, a su vez con un include()

2.- Si no guardamos el idioma en variable de sesión o en cookie tendremos que pasar el idioma seleccionado en la página en los links de la barra de navegación.

```
<nav><a href="pagina1.php?idioma=<?=$idioma?>">Pagina 1</a><a href="pagina2.php?idioma=<?=$idioma?>">Pagina 2</a></nav>
```

MULTIIDIOMA EN TODAS LAS PAGINAS

Para evitar que, al cambiar de idioma en una página que no sea la principal se vuelva a ésta, vamos a montar los links del **header** para el cambio de idioma de forma dinámica:

1.- Obtenemos el nombre del archivo en el que nos encontramos

```
$archivo = basename($_SERVER['PHP_SELF']);
```

2.- Modificamos el atributo href de los enlaces de idioma del header

```
<a href="<?=$archivo?>?idioma=es">ES</a> | <a href="<?=$archivo?>?  
idioma=ca">CA</a>
```

FICHEROS

FICHEROS

Los ficheros nos permitirán guardar datos en el servidor de forma masiva y sin necesidad de utilizar bases de datos todo y que no son equivalentes a éstas.

Podemos guardar cualquier fichero de texto con la extensión que necesitemos para su uso futuro (.php, .html, .txt, .csv, ...)

En php existen unas funciones para lectura y escritura rápida en ficheros

ACCESO RAPIDO A FICHEROS I

Si tenemos el fichero *fichero.txt*

Hola mundo
Qué tal estás
Yo voy tirando

`$contenido=file_get_contents('fichero.txt')` → devuelve el contenido del fichero en una cadena.

`$caracteres=readfile('fichero.txt')` → muestra directamente el contenido del fichero y, además, devuelve el número de caracteres del mismo.

`$arrayLineas=file('fichero.txt')` → devuelve el contenido del fichero en un array en donde cada elemento corresponde a una línea del fichero.

ACCESO RAPIDO A FICHEROS II

`file_put_contents('fichero.txt', $nuevoContenido)` → escribe el contenido especificado en el segundo parámetro en el fichero indicado y sustituyendo el contenido anterior. Si el fichero no existe se crea automáticamente.

Si queremos insertar un salto de línea podemos utilizar el carácter especial `\n`

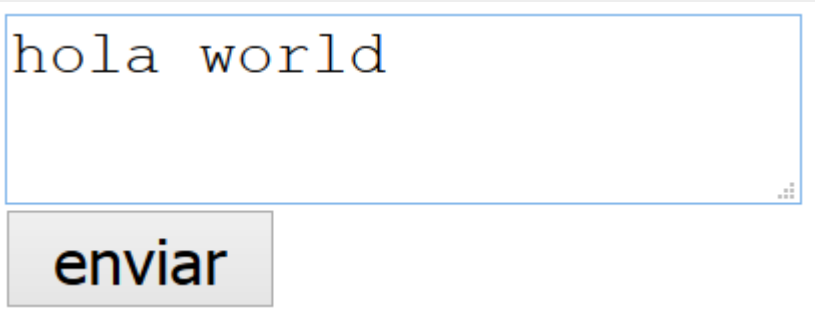
Para saber si un fichero existe podemos utilizar la función:

`file_exists('fichero.txt')` → devuelve true o false

EJERCICIO

Confeccionar un formulario con un `textarea`. Al pulsar el botón de enviar guardaremos el contenido en un fichero.

Al volver a entrar en la página comprobaremos si existe el fichero y, en caso afirmativo, cargaremos automáticamente el contenido del fichero dentro del `textarea`



hola world

enviar

OJO: no dejar espacios en blanco dentro de la etiqueta `textarea` o éstos se mostrarán al visualizar la página

EJERCICIO LIBRO DE VISITAS

Hacer el siguiente libro de visitas:

Enviar

Comentarios:

David Alcolea escribió el 22-4-16:
readfile() lo podemos poner donde lo necesitemos

David Alcolea escribió el 22-4-16:
Para la fecha utilizar: date("j-n-y");

CORREO

FUNCION MAIL()

En php disponemos de la función mail() para el envío de correos electrónicos siempre y cuando dispongamos de un servidor de correo instalado.

Su sintaxis es la siguiente:

mail(destinatario, asunto, mensaje, cabeceras);

El parámetro ***cabeceras*** es opcional y en él se indican parámetros adicionales tales como:

- Remitente: **From**
- Copia: **Cc**
- Copia oculta: **Bcc**

FUNCION MAIL(): EJEMPLO

Vamos a enviar un correo completo con copia a una segunda dirección de correo e indicando el remitente del mismo.

```
$destinatario='correo@mail.com';  
$remitente='nombre@dominio.com';  
$asunto='Correo de prueba';  
$mensaje='Texto del correo';  
$cabeceras = 'From: '.$remitente."\r\n" .  
              'Reply-To: '.$remitente."\r\n" .  
              'MIME-Version: 1.0'."\r\n" .  
              'Content-Type: text/html; charset=UTF-8';  
  
if (mail($destinatario, $asunto, $mensaje, $cabeceras)) {  
    echo "mail enviado correctamente";  
};
```

FUNCION MAIL() PHP NOTAS

- Si indicamos parámetros de cabecera éstos deben acabar siempre con “\r\n” (retorno de línea)
- Si queremos que el texto de mensaje esté en formato html debemos indicar.

'Content-Type: text/html; charset=UTF-8';

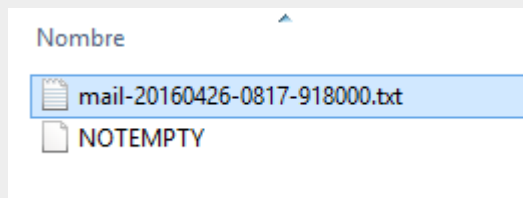
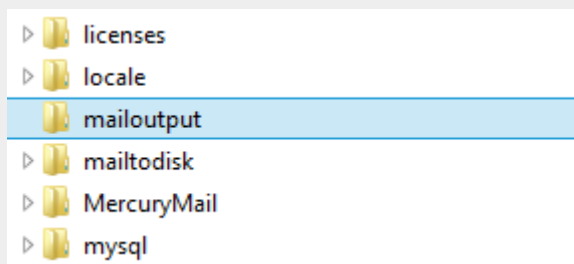
\$mensaje="<h2>imagen</h2>

";

VERIFICAR ENVIO

Si estamos trabajando en localhost el mail no se llegará a enviar a su destinatario ya que no disponemos del gestor de correo instalado.

Podemos ver el resultado del envío en la carpeta *mailoutput* de *htdocs*



EJERCICIO

Confeccionar un formulario para recoger la siguiente info:

- Nombre y apellidos
- Email
- Mensaje

Al pulsar el botón de 'enviar' se confeccionará un email para enviar a la dirección de correo electrónico especificada

SUBIR ARCHIVOS

SUBIR ARCHIVOS AL SERVIDOR

Para subir un archivo al servidor utilizaremos el mismo procedimiento que el utilizado para enviar un archivo en un correo electrónico:

- Formulario con `enctype`
- Input `type='file'`

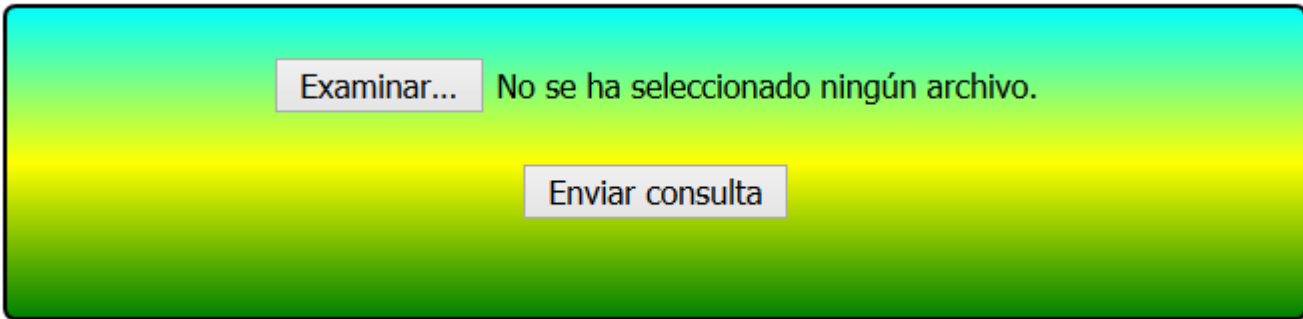
Pero ahora utilizaremos la siguiente función php:

```
move_uploaded_file($tmpNombre, $destinoServidor);
```

Si el archivo ubicado en `tmpNombre` es valido para la subida, sera movido a la ubicación `destinoServidor`. Esta función devuelve true o false. Si no ponemos ruta se alojara en la misma carpeta que la página php

EJERCICIO

Confeccionar un formulario con un único campo de **input** que nos servirá para seleccionar una imagen y subirla a una carpeta específica dentro de localhost



The image shows a web form with a light blue gradient background. At the top, there is a text input field with the placeholder text "Examinar...". To the right of the input field, the text "No se ha seleccionado ningún archivo." is displayed. Below the input field, there is a button labeled "Enviar consulta".