

PRÁCTICA 8. Gráficos y sesiones

Realizar los siguientes ejercicios ayudándose del código y las explicaciones añadidas al enunciado

Escriba un programa de dos páginas que muestre un valor numérico y permita subirlo o bajarlo mediante dos botones.

- La primera página contiene un formulario con tres botones de tipo `submit` con el mismo `name`.
- La segunda página recibe el dato, modifica el número y redirige a la primera página.
- El número se guarda en una variable de sesión. Si la variable no está definida, se le dará el valor 0.

SUBIR Y BAJAR NÚMERO

Haga clic en los botones para modificar el valor:

-

8

+

Poner a cero

0. Puesto que la página 2 redirige a la página 1, la página 2 no debe tener nada de HTML, ni ningún `print`.
1. Primero, la página 2 debe recoger el dato enviado por el formulario de la página 1. En la página 1 hay 3 botones, pero los tres tienen el mismo `name`, por lo que sólo hay que recoger un dato.
Como en el ejercicio anterior: función `recoge()`, variable para recoger el dato, variable auxiliar que indica la corrección del dato, comprobación y reenvío al formulario si no es uno de los valores válidos ...
2. El valor que muestra la página 1 se modifica en la página 2, dependiendo del dato recibido (se aumenta, se reduce o se pone a cero).
3. Para que la página 2 pueda guardar datos en `$_SESSION`, debemos unir la página a la sesión. Conviene dar nombre a la sesión para que cada ejercicio utilice su propia matriz `$_SESSION` distinta del resto.
4. La variable de sesión se debe inicializar antes de modificarla (la manera más simple de hacerlo es darle valor si no existe).
5. Finalmente, la página 2 redirige a la página 1.
6. La página 1 muestra el número guardado en `$_SESSION`.
7. Para que la página 1 pueda usar datos de `$_SESSION`, debemos unir la página a la sesión. El nombre de la sesión debe ser el mismo en ambas páginas.
8. Pero la primera vez que se abre la página, `$_SESSION` no contiene ningún dato e intentar imprimirlo provocaría un error, por lo que damos valor a la variable si no existe.

```
<?php
// Se accede a la sesión
session_name("sesiones-1-11");
session_start();

// Si el número no está guardado en la sesión, lo pone a cero
if (!isset($_SESSION["numero"])) {
    $_SESSION["numero"] = 0;
}
?>

<!DOCTYPE html>
...
<p>
    <button type="submit" name="accion" value="bajar" style="font-size: 4rem">-</button>

<?php
// Muestra el número, guardado en la sesión
print "    <span style=\"font-size: 4rem\">$_SESSION[numero]</span>\n";
?>

    <button type="submit" name="accion" value="subir" style="font-size: 4rem">+</button>
</p>
...
```



```
<?php
// Se accede a la sesión
session_name("sesiones-1-11");
session_start();

// Si el número no está guardado en la sesión, vuelve al formulario
if (!isset($_SESSION["numero"])) {
    $_SESSION["numero"] = 0;
}

// Funciones auxiliares
function recoge($var)
{
    $tmp = (isset($_REQUEST[$var]))
        ? trim(htmlspecialchars($_REQUEST[$var], ENT_QUOTES, "UTF-8"))
        : "";
    return $tmp;
}

// Recogida de accion
$accion = recoge("accion");
$accionOk = false;

// Comprobación de accion
if ($accion != "cero" && $accion != "subir" && $accion != "bajar") {
    // Si no es una de las tres posibles acciones, se vuelve al formulario
    header("Location:sesiones-1-11-1.php");
    exit;
} else {
    $accionOk = true;
}

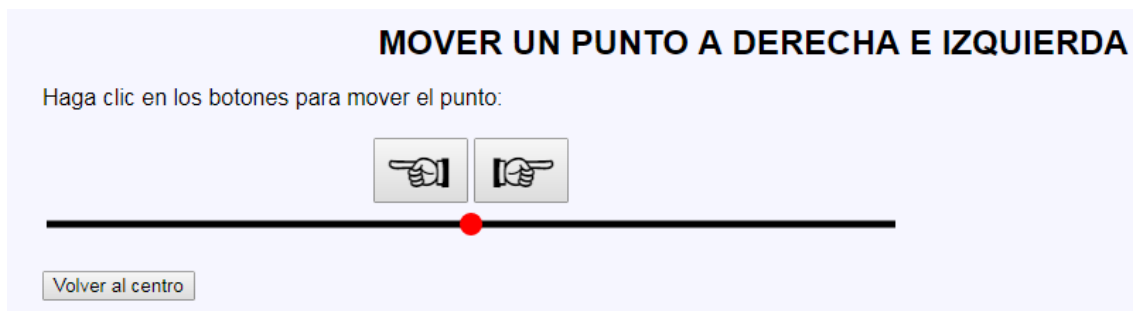
// Si la accion recibida es válida ...
if ($accionOk) {
    // Cambia el valor del número
    if ($accion == "cero") {
        $_SESSION["numero"] = 0;
    } elseif ($accion == "subir") {
        $_SESSION["numero"] ++;
    } elseif ($accion == "bajar") {
        $_SESSION["numero"] --;
    }

    // y vuelve al formulario
    header("Location:sesiones-1-11-1.php");
    exit;
}
?>
```

Ejercicio2

Escriba un programa de dos páginas que muestre un punto sobre una línea y permita moverlo a derecha o izquierda mediante dos botones.

- La primera página contiene un formulario con tres botones de tipo `submit` con el mismo atributo `name`.
- La segunda página recibe el dato, modifica la variable que contiene la posición y redirige a la primera página.
- El número se guarda en una variable de sesión. Si la variable no está definida, se le dará el valor 0.
- El ancho de la línea son 600px y las coordenadas van de -300 a 300.
- El punto avanza o retrocede de 20px en 20px.
- Cuando el punto sale del dibujo por un lado, aparece en el lado opuesto.



Este ejercicio es muy similar al anterior: mediante dos botones se aumenta o disminuye un valor. la diferencia es que en este ejercicio el valor no se muestra, pero se utiliza para situar un punto en una línea.

- Puesto que la página 2 redirige a la página 1, la página 2 no debe tener nada de HTML, ni ningún `print`.
- Primero, la página 2 debe recoger el dato enviado por el formulario de la página 1. En la página 1 hay 3 botones, pero los tres tienen el mismo `name`, por lo que sólo hay que recoger un dato.
- Como en el ejercicio anterior: función `recoge()`, variable para recoger el dato, variable auxiliar que indica la corrección del dato, comprobación y reenvío al formulario si no es uno de los valores válidos ...
- El valor que se utiliza en la página 1 se modifica en la página 2, dependiendo del dato recibido (se aumenta, se reduce o se pone a cero).
- Para que la página 2 pueda guardar datos en `$_SESSION`, debemos unir la página a la sesión. Conviene dar nombre a la sesión para que cada ejercicio utilice su propia matriz `$_SESSION` distinta del resto.
- La variable de sesión se debe inicializar antes de modificarla (la manera más simple de hacerlo es darle valor si no existe).
- Finalmente, la página 2 redirige a la página 1.
- La página 1 dibuja el punto rojo en la imagen SVG utilizando el número guardado en `$_SESSION`.
- Para que la página 1 pueda usar datos de `$_SESSION`, debemos unir la página a la sesión. El nombre de la sesión debe ser el mismo en ambas páginas.
- Pero la primera vez que se abre la página, `$_SESSION` no contiene ningún dato e intentar imprimirlo provocaría un error, por lo que damos valor a la variable si no existe.
- Para que cuando el punto salga por un lado del dibujo entre por el otro, se debe comprobar si el valor supera los extremos y, en su caso, cambiarlo por el otro valor extremo.

```
<?php
// Se accede a la sesión
session_name("sesiones-1-12");
session_start();

// Si la posición no está guardada en la sesión, la pone a cero
if (!isset($_SESSION["posicion"])) {
    $_SESSION["posicion"] = 0;
}
?>

<!DOCTYPE html>
...
<line x1="-300" y1="10" x2="300" y2="10" stroke="black" stroke-width="5" />

<?php
// Dibuja el círculo en su posición
print "    <circle cx=\"$_SESSION[posicion]\" cy="10" r="8" fill="red" />\n";
?>

</svg>
...
```



```
<?php
// Se accede a la sesión
session_name("sesiones-1-12");
session_start();

// Si la posición no está guardada en la sesión, la pone a cero
if (!isset($_SESSION["posicion"])) {
    $_SESSION["posicion"] = 0;
}

// Funciones auxiliares
function recoge($var)
{
    $tmp = (isset($_REQUEST[$var]))
        ? trim(htmlspecialchars($_REQUEST[$var], ENT_QUOTES, "UTF-8"))
        : "";
    return $tmp;
}

// Recogida de acción
$action = recoge("accion");
$actionOk = true;

// Comprobación de acción
if ($action != "centro" && $action != "izquierda" && $action != "derecha") {
    // Si no es una de las tres posibles acciones, se vuelve al formulario
    header("Location:sesiones-1-12-1.php");
    exit;
} else {
    $actionOk = true;
}

// Si la acción recibida es válida ...
if ($actionOk) {
    //se mueve el punto
    if ($action == "centro") {
        $_SESSION["posicion"] = 0;
    } elseif ($action == "izquierda") {
        $_SESSION["posicion"] -= 20;
    } elseif ($action == "derecha") {
        $_SESSION["posicion"] += 20;
    }

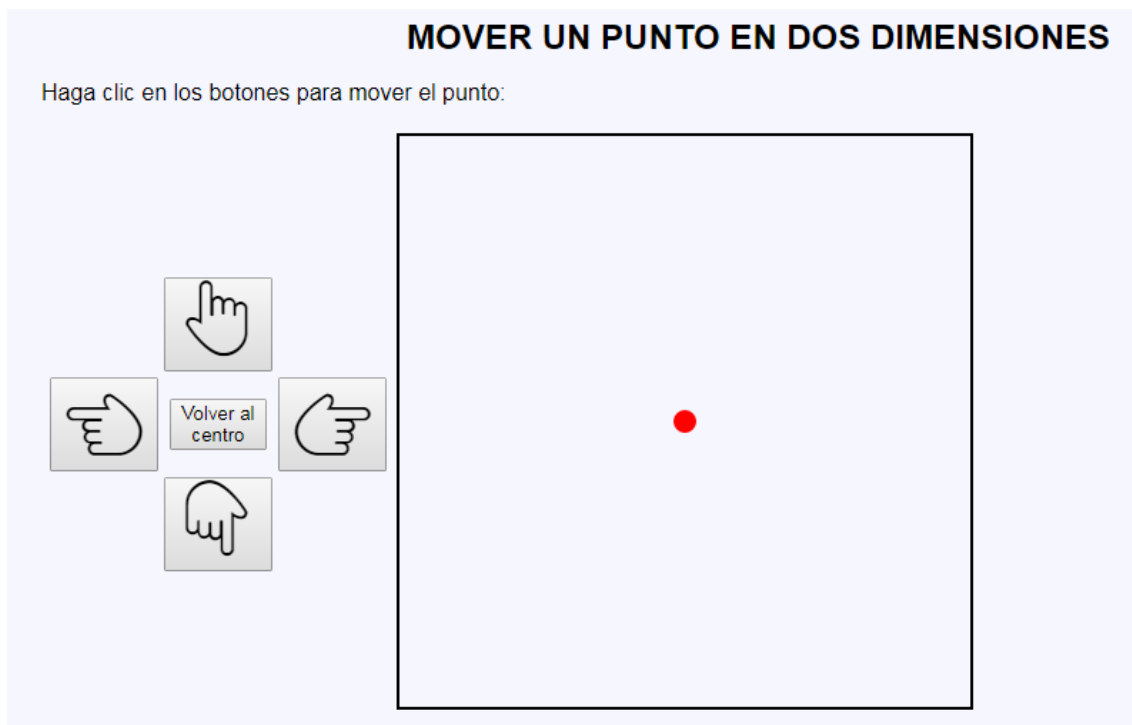
    // si sale por un lado, entra por el otro
    if ($_SESSION["posicion"] > 300) {
        $_SESSION["posicion"] = -300;
    } elseif ($_SESSION["posicion"] < -300) {
        $_SESSION["posicion"] = 300;
    }

    // y vuelve al formulario
    header("Location:sesiones-1-12-1.php");
    exit;
}
?>
```

Ejercicio 3

Escriba un programa de dos páginas que muestre un punto en un cuadrado y permita moverlo en vertical u horizontal mediante cuatro botones.

- La primera página contiene un formulario con cinco botones de tipo `submit` con el mismo atributo `name`.
- La segunda página recibe el dato, modifica las variables que contienen la posición (X e Y) y redirige a la primera página.
- Los dos números se guardan en dos variables de sesión. Si las variables de sesión no están definidas, se les dará el valor 0.
- El plano tiene 400px de lado y las coordenadas van de -200 a 200.
- El punto avanza o retrocede de 20px en 20px.
- Cuando el punto sale del dibujo por un lado, aparece en el lado opuesto.



Este ejercicio es muy similar al anterior. La diferencia es que en este ejercicio se tendrán que guardar dos valores (coordenada X y coordenada Y) para situar el punto en el plano.

0. Puesto que la página 2 redirige a la página 1, la página 2 no debe tener nada de HTML, ni ningún `print`.
1. Primero, la página 2 debe recoger el dato enviado por el formulario de la página 1. En la página 1 hay 5 botones, pero los cinco tienen el mismo `name`, por lo que sólo hay que recoger un dato.
Como en el ejercicio anterior: función `recoge()`, variable para recoger el dato, variable auxiliar que indica la corrección del dato, comprobación y reenvío al formulario si no es uno de los valores válidos...
2. Los dos valores que se utilizan en la página 1 se modifican en la página 2, dependiendo del dato recibido (se aumenta o se reduce uno de ellos o se ponen los dos a cero).
3. Para que la página 2 pueda guardar datos en `$_SESSION`, debemos unir la página a la sesión. Conviene dar nombre a la sesión para que cada ejercicio utilice su propia matriz `$_SESSION` distinta del resto.
4. Las variables de sesión se deben inicializar antes de modificarlas (la manera más simple de hacerlo es darles valor si no existen).
5. Finalmente, la página 2 redirige a la página 1.
6. La página 1 constuye una imagen SVG utilizando los dos números guardados en `$_SESSION`.
7. Para que la página 1 pueda usar datos de `$_SESSION`, debemos unir la página a la sesión. El nombre de la sesión debe ser el mismo en ambas páginas.
8. Pero la primera vez que se abre la página, `$_SESSION` no contiene ningún dato e intentar imprimirlo provocaría un error, por lo que damos valor a las variables si no existen.
9. Para que cuando el punto salga por un lado del dibujo entre por el otro, se debe comprobar si el valor supera los extremos y, en su caso, cambiarlo por el otro valor extremo.



```
<?php
// Se accede a la sesión
session_name("sesiones-1-13");
session_start();

// Si alguna posición no está guardada en la sesión, las pone a cero
if (!isset($_SESSION["x"]) || !isset($_SESSION["y"])) {
    $_SESSION["x"] = $_SESSION["y"] = 0;
}
?>

<!DOCTYPE html>
...
    <td>
        <svg version="1.1" xmlns="http://www.w3.org/2000/svg"
            width="400" height="400" viewBox="-200 -200 400 400" style="border: black 2px solid">

<?php
// Dibuja el círculo en su posición
print "        <circle cx=\"$_SESSION[x]\" cy=\"$_SESSION[y]\" r=\"8\" fill=\"red\" />\n";
?>

    </svg>
</td>
...

```

```
<?php
// Se accede a la sesión
session_name("sesiones-1-13");
session_start();

// Si alguna posición no está guardada en la sesión, las pone a cero
if (!isset($_SESSION["x"]) || !isset($_SESSION["y"])) {
    $_SESSION["x"] = $_SESSION["y"] = 0;
}

// Funciones auxiliares
function recoge($var)
{
    $tmp = (isset($_REQUEST[$var]))
        ? trim(htmlspecialchars($_REQUEST[$var], ENT_QUOTES, "UTF-8"))
        : "";
    return $tmp;
}

// Recogida de acción
$action = recoge("accion");
$actionOk = true;

// Comprobación de acción
if ($action != "centro" && $action != "izquierda" && $action != "derecha"
    && $action != "arriba" && $action != "abajo") {
    // Si no es una de las tres posibles acciones, se vuelve al formulario
    header("Location:sesiones-1-13-1.php");
    exit;
} else {
    $actionOk = true;
}

// Si la acción recibida es válida ...
if ($actionOk) {
    //se mueve el punto
    if ($action == "centro") {
        $_SESSION["x"] = $_SESSION["y"] = 0;
    } elseif ($action == "izquierda") {
        $_SESSION["x"] -= 20;
    } elseif ($action == "derecha") {
        $_SESSION["x"] += 20;
    } elseif ($action == "arriba") {
        $_SESSION["y"] -= 20;
    } elseif ($action == "abajo") {
        $_SESSION["y"] += 20;
    }
}

```

```
// si sale por un lado, entra por el otro
if ($_SESSION["x"] > 200) {
    $_SESSION["x"] = -200;
} elseif ($_SESSION["x"] < -200) {
    $_SESSION["x"] = 200;
}

// si sale por un lado, entra por el otro
if ($_SESSION["y"] > 200) {
    $_SESSION["y"] = -200;
} elseif ($_SESSION["y"] < -200) {
    $_SESSION["y"] = 200;
}

// y vuelve al formulario
header("Location:sesiones-1-13-1.php");
exit;
}
?>
```



Ejercicio 4

Escriba un programa de dos páginas que muestre gráficamente los votos recogidos por dos opciones.

- La primera página contiene un formulario con tres botones de tipo `submit` con el mismo atributo `name`.
 - Dos botones permiten votar a una u otra opción
 - El tercer botón pone a cero los contadores de votos
- La segunda página recibe el dato, modifica la variable de sesión que contiene el número de votos de la opción elegida y redirige a la primera página.
- Los dos números se guardan en dos variables de sesión. Si las variables de sesión no están definidas, se les dará el valor 0.
- Las franjas correspondientes a los votos se alargan de 10px en 10px y no tienen límite de tamaño.

VOTAR UNA OPCIÓN

Haga clic en los botones para votar por una opción:

Poner a cero

Este ejercicio es muy similar al anterior. La diferencia es que cada valor se utiliza en un dibujo distinto.

- Puesto que la página 2 redirige a la página 1, la página 2 no debe tener nada de HTML, ni ningún `print`.
- Primero, la página 2 debe recoger el dato enviado por el formulario de la página 1. En la página 1 hay 3 botones, pero los tres tienen el mismo `name`, por lo que sólo hay que recoger un dato.
No hace falta comprobar el dato porque el programa sólo hará caso de los valores correctos.
- Los dos valores que se utilizan en la página 1 se modifican en la página 2, dependiendo del dato recibido (se aumenta uno de ellos o se ponen los dos a cero).
- Para que la página 2 pueda guardar datos en `$_SESSION`, debemos unir la página a la sesión. Conviene dar nombre a la sesión para que cada ejercicio utilice su propia matriz `$_SESSION` distinta del resto.
- Las variables de sesión se deben inicializar antes de modificarlas (la manera más simple de hacerlo es darles valor si no existen).
- Finalmente, la página 2 redirige a la página 1.
- La página 1 constuye la primera imagen SVG utilizando uno de los números guardados en `$_SESSION`. El número se utiliza en dos sitios.
- La página 1 constuye la segunda imagen SVG utilizando el otro número guardado en `$_SESSION`. El número se utiliza en dos sitios.
- Para que la página 1 pueda usar datos de `$_SESSION`, debemos unir la página a la sesión. El nombre de la sesión debe ser el mismo en ambas páginas.
- Pero la primera vez que se abre la página, `$_SESSION` no contiene ningún dato e intentar imprimirlo provocaría un error, por lo que damos valor a las variables si no existen.



```
<?php
// Se accede a la sesión
session_name("sesiones-1-14");
session_start();

// Si algún contador no está guardado en la sesión, los pone a cero
if (!isset($_SESSION["a"]) || !isset($_SESSION["b"])) {
    $_SESSION["a"] = $_SESSION["b"] = 0;
}
?>

<!DOCTYPE html>
...
<tr>
    <td style="vertical-align: top;"><button type="submit" name="accion" value="a" style="font-size: 60px; line-height: 50px; color: hsl(200, 100%, 50%)">A</button></td>
    <td>
        <?php
        print "          <svg version='1.1' xmlns='http://www.w3.org/2000/svg' \n";
        print "            width='$_SESSION[a]' height='50'>\n";
        print "            <line x1='0' y1='25' x2='$_SESSION[a]' y2='25' stroke='hsl(200, 100%, 50%)' stroke-width='50' />\n";
        print "          </svg>\n";
        ?>
        </td>
    </tr>
    <tr>
        <td><button type="submit" name="accion" value="b" style="font-size: 60px; line-height: 50px; color: hsl(35, 100%, 50%)">B</button></td>
        <td>
            <?php
            print "          <svg version='1.1' xmlns='http://www.w3.org/2000/svg' \n";
            print "            width='$_SESSION[b]' height='50'>\n";
            print "            <line x1='0' y1='25' x2='$_SESSION[b]' y2='25' stroke='hsl(35, 100%, 50%)' stroke-width='50' />\n";
            print "          </svg>\n";
            ?>
            </td>
        </tr>
    </table>
    ...

```

```
<?php
session_name("sesiones-1-14");
session_start();

if (!isset($_SESSION["a"]) || !isset($_SESSION["b"])) {
    $_SESSION["a"] = $_SESSION["b"] = 0;
}

function recoge($var)
{
    $tmp = (isset($_REQUEST[$var]))
    ? trim(htmlspecialchars($_REQUEST[$var], ENT_QUOTES, "UTF-8"))
    : "";
    return $tmp;
}

$accion = recoge("accion");

if ($accion == "a") {
    $_SESSION["a"] += 10;
} elseif ($accion == "b") {
    $_SESSION["b"] += 10;
} elseif ($accion == "cero") {
    $_SESSION["a"] = $_SESSION["b"] = 0;
}

header("Location:sesiones-1-14-1.php");
?>

```

Medios disponibles:

Apuntes, libros, consulta de videos, tutoriales, etc. que el alumnado desee.