

Tecnológico nacional de mexico

Practicas y ejercicios de php

Daniela samantha Orozco urbano

Programacion web

Victor villavicencio

Ejercicio 1 - 01_info.php

NOTA: Primer programa en php, para mostrar cómo se escribe el código dentro de una pagina

```
<html>
<head>
<title>PHP-Info</title>
</head>

<body>

<?php phpinfo();
?>

</body>
</html>
```

Responde a lo siguiente:

PREGUNTA	RESPUESTA
¿Qué es lo que hace la funcion phpinfo()?	Esta función nos mostrara información sobre la configuración que tiene nuestro php, incluyendo la versión que hayamos instalado, extensiones que pudimos incluir, variables agregadas al entorno, así como también las directivas de nuestra configuración.

Ejercicio 2 - 02_hola.php

NOTA: El famoso programa HOLA MUNDO en php

```
<html>

<head>
<title>Hola Mundo</title>
</head>

<body>
<h1> El famoso script Hello World</h1>
<p>
<?php

echo "Hola Mundo!";

?>
</p>
</body>

</html>
```

Responde a lo siguiente:

PREGUNTA	RESPUESTA
¿Cómo comenzamos un bloque de php?	Con <?php y lo cerramos con ?> si este es necesario.
¿Cómo imprimimos un mensaje en pantalla?	Usando echo "mensaje"; o print "mensaje";.

Ejercicio 3 - 03_hola2.php

NOTA: Variacion del programa anterior, usando ademas etiquetas HTML para darle enfasis

```
<html>

<head>
<title>Hola Mundo</title>
</head>

<body>
<h1> El famoso script Hello World</h1>
<p>
```

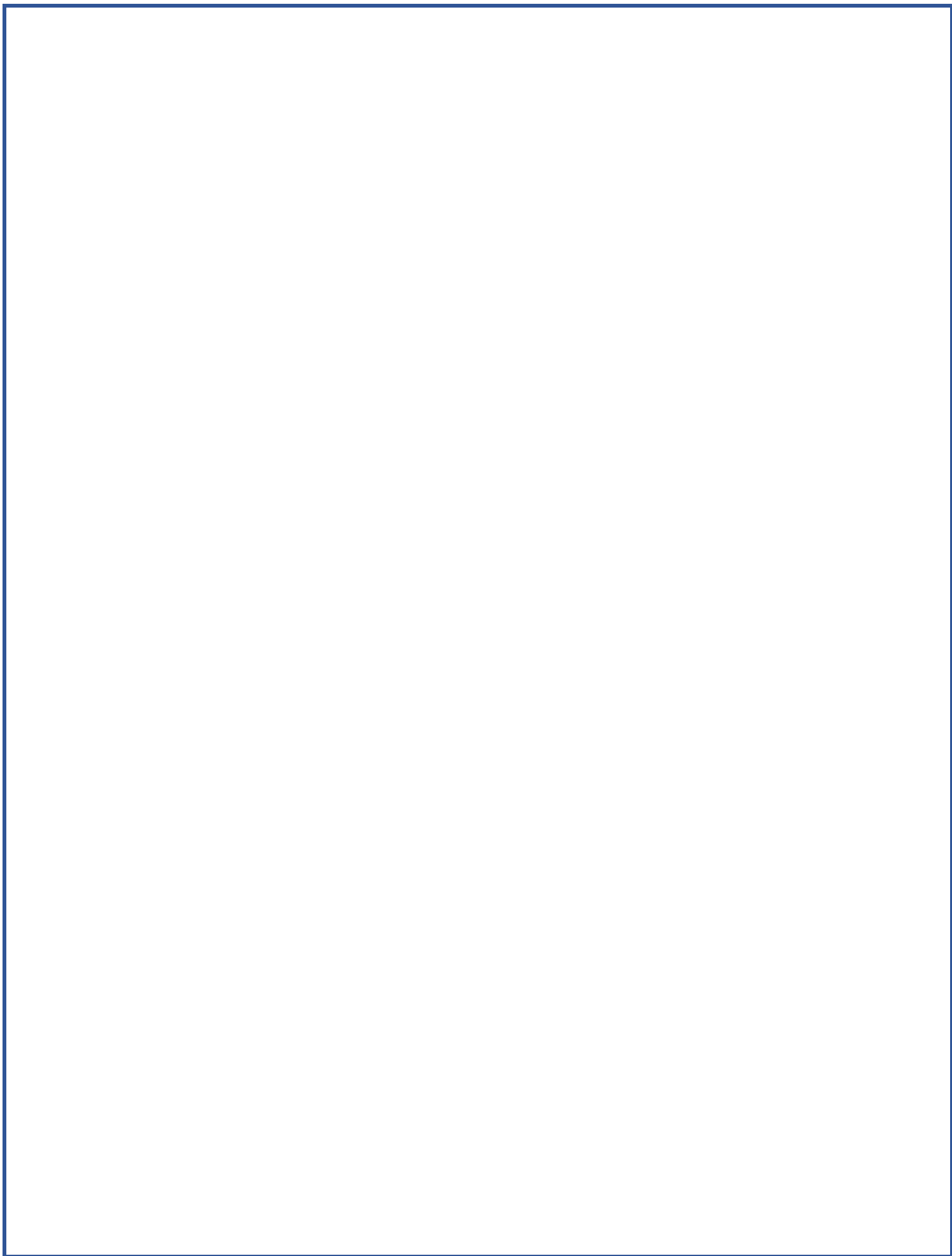
```
<?php

echo "<b>Hola</b> Mundo!";

?>
</p>
</body>

</html>
```

PREGUNTA	RESPUESTA
¿Cuál es la diferencia con el ejercicio 2?	En este código, la palabra " Hola " está en negrita porque se usó la etiqueta HTML dentro de echo, mientras que en el ejercicio anterior, el texto "Hola Mundo!" se imprimía sin formato especial.



Ejercicio 4 - 04_variable1.php

NOTA: Uso de variables en php y como se declaran

```
<html>

<head>
<title></title>
</head>

<body>
<h1>Primer ejemplo de Variables </h1>
<p>
<?php
$Name = "Miguel";
echo "Hola <b>$Name</b>, encantado de conocerte";

?>
</p>
</body>

</html>
```

Responde a lo siguiente:

PREGUNTA	RESPUESTA
¿Cómo declaramos una variable?	En PHP, una variable se declara con el símbolo \$, seguido del nombre de la variable, como lo ya mostrado en el código visto de este ejercicio \$Name = "Miguel";.
¿Importan las mayúsculas y minúsculas en la declaracion de las variables?	En php esto importa mucho al ser variables muy sensibles a mayúsculas y minúsculas.

Ejercicio 5 - 05_variable2.php

NOTA: Uso de variables en variacion al programa anterior, usando concatenacion de variables

```
<html>

<head>
<title></title>
</head>

<body>
<h1>Segundo ejemplo de Variables usando el operador concatenacion .</h1>
<p>
<?php
$Name = "Miguel";
echo "Hola <b>" . $Name . "</b>, encantado de conocerte";

?>
</p>
</body>

</html>
```

Responde a lo siguiente:

PREGUNTA	RESPUESTA
¿Qué es la concatenacion?	Básicamente una concatenación para unir dos o mas cadenas de texto en una sola.
¿Cómo hacemos la concatenacion en php?	En PHP, se usa el operador <code>.</code> para concatenar nuestras cadenas.

Ejercicio 6 - 06_break.php

NOTA: Ejemplo para ver como se formatea el código para el usuario

```
<html>

<head>
<title></title>
</head>

<body>
<h1>Break en la ventana del navegador </h1>
<p>
<?php
$Name = "Miguel";
echo "Hola <b>$Name</b>, encantado de conocerte<br>";
echo "Gracias por venir!";
?>
</p>
</body>

</html>
```

Responde a lo siguiente:

PREGUNTA	RESPUESTA
¿Qué es lo que hace el script?	El script imprime en la página web el mensaje colocando " Miguel " en negrita, seguido de un salto de línea. Luego, en la siguiente línea, muestra " Gracias por venir! ".

Ejercicio 7 - 07_break2.php

NOTA: Ejemplo de variacion del script anterior, para ver el formato dentro de php en avance de linea.

Notar las diferencias cuando se haga la ejecucion via VER CODIGO FUENTE del navegador.

```
<html>

<head>
<title></title>
</head>

<body>
<h1>Break en la ventana del navegador con signo \ </h1>
<p>
<?php
$Name = "Miguel";
echo "Hola <b>$Name</b>, encantado de conocerte<br>\n";
echo "Gracias por venir!\n";
?>
</p>
</body>

</html>
```

Responde a lo siguiente:

PREGUNTA	RESPUESTA
¿Cuál es la diferencia con el ejercicio 6?	En este ejercicio, se usa \n dentro del echo, lo cual agrega un salto de línea en el código fuente HTML, pero no afecta la visualización en el navegador, en cambio, en el ejercicio 6, solo se usó , que sí genera un salto de línea visible en la página web.

Ejercicio 8 - 08_weekdays1.php

NOTA: Aunque los arrays se ven mas adelante según el temario del curso, en el cuaderno de trabajo vienen antes y no son difíciles de asimilar. Queda a discrecion ponerlos en orden o hacerlos despues del no. 21 que son los ciclos y antes de entrar a los formularios.

```
<html>

<head>
<title>Introduccion a los arrays, metodo largo</title>
</head>

<body>
<h1> Introduccion a los arrays, metodo largo </h1>
<p> A continuacion escribiremos los arrays de acuerdo al metodo largo </p>
<p>
<?php
/*
Este es el metodo largo para la creacion de arrays, donde cada uno de los
arreglos esta en una linea diferente, junto con su valor indice correspondiente
*/
$dia[0] = "domingo";
$dia[1] = "lunes";
$dia[2] = "martes";
$dia[3] = "miercoles";
$dia[4] = "jueves";
$dia[5] = "viernes";
$dia[6] = "sabado";
//mostrar el miercoles

echo $dia[3];

?>
</p>
</body>

</html>
```

Responde a lo siguiente:

PREGUNTA	RESPUESTA
¿Qué es un array?	es una estructura de datos que almacena múltiples valores bajo un mismo nombre, donde cada valor tiene un índice o clave que lo identifica.
¿Cómo declaramos un array?	En el método largo , declaramos un array asignando

	valores a sus índices, como <code>\$dia[0] = "domingo";</code> , o podemos usar el método corto con <code>array()</code> , como <code>\$dia = array("domingo", "lunes", "martes");</code> .
--	---

Ejercicio 9 - 09_weekdays2.php

NOTA: Es el mismo ejercicio de arreglos anterior, pero aquí se ven en una sola línea para simplificar el código. También, hay que hacer notar el uso de comentarios y como se ponen en php.

```
<html>

<head>
<title>Introduccion a los arrays, metodo corto</title>
</head>

<body>
<h1> Introduccion a los arrays, metodo corto </h1>
<p> A continuacion escribiremos los arrays de acuerdo al metodo corto </p>
<p>
<?php
/*
Este es el metodo corto para la creacion de arrays, donde cada todo el array
puede estar en una sola linea de codigo.
*/
$dia = array( "domingo", "lunes", "martes", "miercoles", "jueves", "viernes", "sabado");
//mostrar el miercoles

echo $dia[0];

?>
</p>
</body>

</html>
```

Responde a lo siguiente:

PREGUNTA	RESPUESTA
¿Qué diferencia hay con el script 7?	En el script 7, se utiliza el método largo para declarar el array, asignando los valores a los índices de manera individual, en el script actual, se utiliza el método corto, donde todo el array se declara en una sola línea usando la función array().
¿Cómo se ponen los comentarios?	Los comentarios en PHP se ponen de dos maneras: una sola línea y dos líneas //, /*, */.

Ejercicio 10 - 10_Operaciones.php

NOTA: Uno de los temas es ver como se realizan las operaciones en php, en este ejemplo se ven cadenas y numeros y como se hacen las operaciones aritmeticas basicas, asi como los incrementos y decrementos a las variables. Tambien vemos como se crea una variable en php.

```
<html>

<head>
<title>Ejemplo de operaciones</title>
</head>

<body>
<h1>Ejemplo de operaciones aritmeticas en PHP</h1>
<?php

$a = 8;
$b = 3;
echo $a + $b, "<br>";
echo $a - $b, "<br>";
echo $a * $b, "<br>";
echo $a / $b, "<br>";

$a++ ;
echo $a,"<br>";
$b--;
echo $b,"<br>";

?>

</body>

</html>
```

Responde a lo siguiente:

PREGUNTA	RESPUESTA
¿Cómo se declara una variable numerica en php?	Se declara una variable numérica simplemente asignando un valor numérico a una variable
¿Cómo se realiza una operación con variables numericas?	Las operaciones se realizan usando operadores aritméticos como +, -, *, /, etc., por ejemplo: \$a + \$b, \$a * \$b.
¿Es posible mostrar en pantalla en una misma linea cadenas y numeros?	Sí, se puede mostrar tanto cadenas como números en la misma línea usando echo en el programa que se va a hacer.

¿Cómo incrementamos una variable?	Se puede incrementar una variable utilizando el operador ++, por ejemplo: \$a++;.
-----------------------------------	---

Ejercicio 11 - 11_Comparacion.php

NOTA: Sobre el mismo punto de operaciones, pero ahora de comparacion con variables numericas y observar el resultado en tiempo de ejecucion.

```
<html>

<head>
<title>Ejemplo de operadores de Comparacion</title>
</head>

<body>
<h1>Ejemplo de operaciones comparacion en PHP</h1>
<?php

$a = 8;
$b = 3;
$c = 3;
echo $a == $b, "<br>";
echo $a != $b, "<br>";
echo $a < $b, "<br>"; echo
$a > $b, "<br>"; echo $a
>= $c, "<br>"; echo $a <=
$c, "<br>";

?>

</body>

</html>
```

Anota el significado de las operaciones de comparacion:

PREGUNTA	RESPUESTA
==	Compara si dos valores son iguales.
!=	Hace la comparación si dos valores son diferentes.
<	Compara si el valor de la izquierda es menor que el de la derecha.
>	Muestra si el valor de la izquierda es mayor que el de la derecha.
>=	Nos muestra si el valor de la izquierda es mayor o igual al de la derecha.
<=	Muestra si el valor de la izquierda es menor o igual al de la derecha.

Ejercicio 12 - 12_Logicos.php

NOTA: Por ultimo, las operaciones logicas se manejan según este ejemplo en php.

```
/* 12_logicos.php
Programa de ejemplo de operadores logicos
Por: Lic. Sergio Hugo Sanchez O.
Para: Universidad Matamoros
17, Mayo, 2009 */

<html>

<head>
  <title>Ejemplo de operadores Logicos</title>
</head>

<body>
<h1>Ejemplo de operaciones logicas en PHP</h1>
<?php

$a = 8;
$b = 3;
$c = 3;
echo ($a == $b) && ($c > $b), "<br>";
echo ($a == $b) || ($b == $c), "<br>";
echo !($b <= $c)b, "<br>";

?>

</body>

</html>
```

Anota el significado de las operaciones de comparacion:

PREGUNTA	RESPUESTA
&&	Operador AND (Y): Devuelve true si ambas condiciones son verdaderas.
	es el operador OR (O lógico) en PHP que devuelve true si al menos una de las condiciones es verdadera.

Ejercicio 13 - 13_Ejer01.php

RESULTADO:

```
<html>
<head>
  <title>completo</title>
</head>
<body>
  <h1>completo</h1>

  <?php
  // Declaración de variables
  $nombre = "Miguel";
  $a = 8;
  $b = 3;
  $c = 3;

  // Saludo con concatenación
  echo "<p>Hola <b>" . $nombre . "</b>, bienvenido al ejercicio completo.</p>";

  // Operaciones aritméticas
  echo "<h2>Operaciones Aritméticas</h2>";
  echo "Suma: " . ($a + $b) . "<br>";
  echo "Resta: " . ($a - $b) . "<br>";
  echo "Multiplicación: " . ($a * $b) . "<br>";
  echo "División: " . ($a / $b) . "<br>";

  // Incremento y decremento
  $a++;
  $b--;
  echo "Incremento de A: $a <br>";
  echo "Decremento de B: $b <br>";

  // Operadores de comparación
  echo "<h2>Operadores de Comparación</h2>";
  echo "¿A es igual a B? " . ($a == $b ? "Sí" : "No") . "<br>";
  echo "¿A es diferente de B? " . ($a != $b ? "Sí" : "No") . "<br>";
  echo "¿A es mayor que B? " . ($a > $b ? "Sí" : "No") . "<br>";
  echo "¿A es menor o igual a C? " . ($a <= $c ? "Sí" : "No") . "<br>";

  // Operadores lógicos
  echo "<h2>Operadores Lógicos</h2>";
  echo "¿A es mayor que B y C es igual a B? " . (($a > $b && $c == $b) ? "Verdadero" : "Falso") .
"<br>";
  echo "¿A es igual a B o C es igual a B? " . (($a == $b || $c == $b) ? "Verdadero" : "Falso") . "<br>";
```

```
echo "Negación de (B menor o igual a C): " . (!($b <= $c) ? "Verdadero" : "Falso") . "<br>";
```

```
// Arrays (método corto)
```

```
echo "<h2>Arrays</h2>";
```

```
$dias = array("Domingo", "Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado");
```

```
echo "El tercer día de la semana es: <b>$dias[2]</b><br>";
```

```
?>
```

```
</body>
```

```
</html>
```

Ejercicio 14 - 14_Redondeo.php

NOTA: Ejemplo para calculos en php usando variables, formateando el codigo del script y usando algunos comandos nuevos de php (similares a c++)

```
<!Ejemplo de redondeo>
<html>

<head>
  <title>Calculos </title>
</head>

<body>
<h1>Calculos, redondeo y formato. </h1>
<?php
  /* Primero declaramos las variables */
  $precioneto = 101.98;
  $iva = 0.196;
  $resultado = $precioneto * $iva;
  echo "El precio es de ";
  echo $precioneto;
  echo " y el IVA el ";
  echo $iva;
  echo "%<br>";
  echo "Resultado: ";
  echo round($resultado,2);
  echo " con ROUND() <br>";
  echo $resultado;
  echo " normal \n";
  echo "<br><br>";
  $resultado2 = sprintf("%01.2f", $resultado);
  echo "Usando la funcion SPRINTF se ve asi: ";
  echo $resultado2
?>

</body>

</html>
```

Responde a lo siguiente:

PREGUNTA	RESPUESTA
¿Qué funciones de PHP usamos en el script?	Se usan las funciones round() para redondear el resultado a 2 decimales y sprintf() para formatear el número con 2 decimales.

Ejercicio 15 - 15_if.php

NOTA: Ejemplo del uso de la sentencia IF en la comparativa de variables

```
<html>

<head>
  <title>Condicion IF</title>
</head>

<body>
<h1>Condicional IF</h1>
<?php
  $a = 8;
  $b = 3;
  echo "Valor de a: ",$a," y el valor de b: ",$b,"<br>";
  if ($a<$b)
  {
    echo "a es menor que b";
  }
  else
  {
    echo " a no es menor que b";
  }

?>

</body>

</html>
```

Responde a lo siguiente:

PREGUNTA	RESPUESTA
¿Qué hace la sentencia IF...ELSE?	La sentencia if...else evalúa una condición, si la condición resulta verdadera, ejecuta el bloque de código dentro de if, si esta es falsa, ejecuta el bloque dentro de else.

Ejercicio 16 – IF..Else

NOTA: Uso de la sentencia If..else y una funcion de fecha.

```
<html>
<head></head>
<body>

<?php
$dia=date("d");
if ($dia<=10)
{
    echo "sitio activo";
}
else
{
    echo "sitio fuera de servicio";
}
?>

</body>
</html>
```

Ejercicio 17 - 16_switch.php

NOTA: Uso de la sentencia SWITCH según el valor de una condicion de una variable, uso de comandos para salirse del codigo.

```
<html>

<head>
  <title>Condicional Switch</title>
</head>

<body>
<h1>Condicional Switch. Ejemplo</h1>
<?php
  /*Declaramos una variable con un valor de muestra */
  $posicion = "arriba";
  echo "La variable posicion es ",$posicion;
  echo "<br>";

  switch($posicion){
    case "arriba":          // Primer condicion si es arriba
      echo "La variable contiene el valor de arriba";
```

```
      break;
    case "abajo":          //Segunda condicion del supuesto echo
      "La variable contiene el valor de abajo";
      break;
    default:              //Condicion por default o si no es ninguna echo "La
      variable contiene otro valor distinto arriba y abajo";
  }
?>

</body>

</html>
```

PREGUNTA

¿Cuál es la diferencia de la instrucción SWITCH con la instrucción IF?

RESPUESTA

La instrucción **if** se usa para evaluar **condiciones lógicas** y permite comparar valores con operadores (>, <, ==, etc.). En cambio, **switch** es más eficiente cuando se comparan **valores específicos** contra múltiples casos posibles, ya que evalúa una sola variable contra diferentes opciones (case).

Ejercicio 18 - 17_Tabla1.php

NOTA: Ejemplo del uso de ciclos for, funciones, comandos html, variables y tablas para combinar php y html.

```
<html>

<head>
  <title>Tabla condicional</title>
</head>

<body>
<h1>Tabla condicional 1</h1>
<?php
  /* Crearemos una tabla de valores de seno y coseno de 0 a 2
  en incrementos de 0.01. Los valores negativos que resulten los queremos
  mostrar en rojo, y los valores positivos en azul */

  /* En primer lugar vamos a crear una funcion con las condicionales
```

Aqui estamos usando la sentencion de FUNCTION al que le pasamos un valor dependiendo de este valor, ejecutamos una condicion IF..ELSE... y asignamos un color al tipo de letra para generar la tabla

```
*/  
function muestra($valor) {  
  if ($valor < 0.5)  
    $color = "red";  
  else  
    $color = "blue";  
  echo "<td><font color='$color'>$valor</font></td>\n";  
}
```

```
?>  
<table border="1">  
<?  
  for ($x=0; $x<=2; $x+=0.01){  
    echo "<tr>";  
    muestra($x);  
    muestra(sin($x));  
    muestra(cos($x));  
    echo "</tr>";  
  }  
?>  
</body>  
</html>
```

Responde a lo siguiente:

PREGUNTA	RESPUESTA
Explica que es lo que hace el script	El script genera una tabla HTML con valores de seno y coseno desde 0 hasta 2 en incrementos de 0.01, para cada una de las filas, muestra el valor de x, sin(x) y cos(x), aplicando colores condicionales: si el valor es menor que 0.5, el texto se muestra en rojo; si es 0.5 o mayor, se muestra en azul. Esto se logra mediante la función muestra(\$valor), que asigna el color y muestra el dato dentro de una celda (<td>).

Ejercicio 19 - 18_tabla2.php

NOTA: Variacion del ejercicio anterior, dando color a cada region par e impar, mediante nuevas instrucciones.

```
<html>
<head>
  <title>Tabla condicional 2</title>
</head>
<body>
<h1>Tabla condicional 2</h1>
<?php
  /* Crearemos una tabla de valores de seno y coseno de 0 a 2
  en incrementos de 0.01. Los valores negativos que resulten los queremos
  mostrar en rojo, y los valores positivos en azul */

  /* Variacion. Un color diferente cada fila que se imprima */
  function muestra($valor) {
    if ($renglon % 2)
      $fondo = "#eeeeee";
    else
      $fondo = "#dddddd";

    if ($valor < 0.5)
      $color = "red";
    else
      $color = "blue";
    echo "<td bgcolor='$fondo'><font color='$color'>$valor</font></td>\n";
  }

  ?>
<table border="1">
<?
  $nrenglon = 0;
  for ($x=0; $x<=2; $x+=0.01){
    echo "<tr>";
    muestra($x);
    muestra(sin($x));
    muestra(cos($x));
    echo "</tr>";
  }
  ?>
</body>
</html>
```

Responde a lo siguiente:

PREGUNTA	RESPUESTA
¿Cuál es la diferencia con el script anterior?	La diferencia principal es que en este script cambia el color de fondo de cada fila alternando entre dos tonos (#eeeeee y #dddddd) para mejorar la legibilidad, seguido de eso, hay un error en el código: la variable \$renglon no está definida ni incrementada en la función muestra(), por lo que la alternancia de colores no funcionará correctamente. Para corregirlo, la variable \$nrenglon debería pasarse correctamente dentro del for.

Ejercicio 20 - 19_tabla3.php

NOTA: Variacion del ejemplo anterior usando variables globales. Dejar que los alumnos investiguen en su cuaderno de trabajo o aulas virtuales como se realiza este ejemplo

```
<html>
<head>
  <title>Tabla condicional 3</title>
</head>
<body>
<h1>Tabla condicional 3</h1>
<?php
  /* Crearemos una tabla de valores de seno y coseno de 0 a 2
  en incrementos de 0.01. Los valores negativos que resulten los queremos
  mostrar en rojo, y los valores positivos en azul */

  /* Variacion. Un color diferente cada fila que se imprima
  Para que funcione bien, necesitamos que la variable renglon se tome
  "desde afuera" y no de manera local dentro de la funcion*/
  function muestra($valor) {
    global $nrenglon;
    if ($nrenglon % 2)
      $fondo = "#ffff00";
    else
      $fondo = "#ffffff";

    if ($valor < 0.5)
      $color = "red";
    else
      $color = "blue";
    echo "<td bgcolor='$fondo'><font color='$color'>$valor</font></td>\n";
  }

?>
```

```

<table border="1">
<?
    $nrenglon = 0;
    for ($x=0; $x<=2; $x+=0.05){
        echo "<tr>";
        muestra($x);
        muestra(sin($x));
        muestra(cos($x));
        echo "</tr>";
    }
?>
</body>
</html>

```

Ejercicio 21 - 20_while.php

NOTA: Uso de ciclo while para hacer una repeticion tantas veces quiera el usuario, para ello trabajamos con un formulario externo vinculado a php para hacerlo. Mostrar la conexión y variables externas, así como el uso del while.

```

<html>

<head>
    <title>Bucle While</title>
</head>

<body>
<h1>Bucle While</h1>
<?php
    /* Mostraremos el uso de la sentencia While y comenzamos a usar entrada
    del teclado mediante un formulario simple */
    if ( isset( $_POST['number'] ) ) {
        $number = $_POST['number'];
        $counter = 1;
        while ($counter <= $number) {
            echo "Los bucles son faciles!<br>\n";
            $counter++;
        }
        echo "Se acabo.\n";
    }
}

```

```

?>
</p>
</body>

</html>

```

20.html

```

<html>
<head>
  <title></title>
</head>
<body>
<form action="20_while.php" method="post">
¿Cuántas veces?
<input type="text" name="number">
<input type="submit" value="Enviar">
</form>
</body>
</html>

```

Responde a lo siguiente:

PREGUNTA	RESPUESTA
¿Por qué tenemos dos archivos en este ejercicio, uno php y el otro html?	El ejercicio usa dos archivos porque uno es responsable de mostrar el formulario HTML (20.html), que es donde el usuario ingresa un valor (el número de repeticiones del bucle), el otro archivo, PHP (20_while.php), recibe el dato enviado desde el formulario a través de una petición POST, procesa la información y ejecuta la lógica del bucle while.
¿Qué es un formulario?	Un formulario en HTML es un componente que permite a los usuarios ingresar datos, como texto, selecciones, etc, vistos en este programa.
Explica el ciclo While	El ciclo while en PHP (y otros lenguajes de programación) ejecuta un bloque de código repetidamente mientras una condición sea verdadera, en el ejemplo, el ciclo while se usa para imprimir el mensaje "Los bucles son fáciles!" un número de veces determinado por el valor que el usuario ingresa en el formulario. El ciclo continúa ejecutándose hasta que el valor de \$counter sea mayor que el número ingresado.
Que hace \$Counter++	La expresión \$counter++ incrementa el valor de la variable \$counter en 1. Esto se conoce como un operador de post-incremento.

Ejercicio 22 - 20_whileB.php

NOTA: Variacion sobre el script anterior, pero usando un solo archivo para poner juntos el formulario y el codigo php. Opcional si los alumnos realizan solos este ejercicio o lo explica el maestro según su disposicion y entendimiento.

```
<html>

<head>
  <title></title>
</head>

<body>
<h1>Bucle While 2</h1>
<form action="20_whileB.php" method="post">
¿Cuantas veces?
<input type="text" name="number">
<input type="submit" value="Enviar">
</form>
<p>
<?php
  /* Mostraremos el uso de la sentencia While y comenzamos a
  usar entrada del teclado mediante un formulario simple */
  if ( isset( $_POST['number'] ) ) {
    $number = $_POST['number'];
    $counter = 1;
    while ( $counter <= $number ) {
      echo "<b>$counter</b>.- Los bucles son faciles!<br>\n";
      $counter++;
    }
    echo "Se acabo.\n";
  }
?>
</p>
</body>

</html>
```

Responde a lo siguiente:

PREGUNTA	RESPUESTA
¿Cuál es la diferencia con el script anterior?	Este script tiene una estructura más sencilla y muestra el número de la iteración en cada ciclo, mientras que el anterior no hacía eso.

Ejercicio 23 - 21_libreria.phtml

NOTA: Uso de funciones en php y como se vinculan con otros archivos php. Notar la extension de la funcion en el primer archivo y como se incrustan estas librerias de funciones en otros scripts de php para ser usados.

```
<html>

<head>
<title>Uso de Librerias y Funciones</title>
</head>

<body>
<?php
function CabeceraPagina()
{
?>
<FONT SIZE="+1">Esta cabecera estará en todas sus páginas.</FONT><BR>
<hr>
<?
    }

function PiePagina()
{
?>
<hr>
<FONT SIZE="-1">Este es el pie de página.</FONT><BR>
Autor: Joaquin Gracia
<?
    }
?>

</body>
</html>
```

22_pag1.php

```
<html>

<head>
<title>Ejemplo Pagina 1 Libreria</title>
</head>

<body>
<?php include("21_libreria.phtml") ?>
<?php CabeceraPagina(); ?>

Página 1
<BR><BR><BR><BR><BR>

Contenido blalbl blalb alb<BR><BR> más
cosas...<BR><BR>

fin<BR><BR>

<?php PiePagina(); ?>

</body>

</html>
```

23_pag2.php

```
<html>

<head>
<title>Ejemplo Pagina 2 Librerias</title>
</head>

<body>
<?php include("21_libreria.phtml") ?>
<?php CabeceraPagina(); ?>

Esta es otra página<BR><BR>
completamente distinta<BR><BR>
pero comparte el pie y la cabecera con la otra.<BR><BR>

<?php PiePagina(); ?>
```



```
</body>
</html>
```

Responde a lo siguiente:

PREGUNTA	RESPUESTA
¿Que es lo que hace este script?	Este script define dos funciones personalizadas, CabeceraPagina() y PiePagina(), que se utilizan para incluir un encabezado y un pie de página en múltiples páginas, seguido en las páginas 22_pag1.php y 23_pag2.php, se incluyen estas funciones y el contenido se genera dinámicamente con un encabezado común y un pie de página común usando esas funciones.
¿Cómo creamos una función personalizada?	Para crear una función personalizada en PHP, utilizamos la palabra clave function seguida del nombre de la función y los parámetros entre paréntesis, si los hay, el bloque de código que la función ejecuta está contenido entre llaves {}.
¿Cómo insertamos una función?	Para insertar o llamar a una función, simplemente escribimos su nombre seguido de paréntesis, si la función tiene parámetros, los pasamos dentro de los paréntesis.
¿Cómo creamos un archivo de funciones o librería de funciones?	Para crear un archivo de funciones, basta con escribir las funciones dentro de un archivo .php o .phtml y luego usar include() o require() para incluir ese archivo en otros scripts.

Ejercicio 24 - 24_analisis.php

NOTA: Primer ejercicio con formularios, hacer notar como se pasa la variable del formulario al script php.

```
<html>

<head>
<title>EJERCICIO 3_1: ANALISIS DE FORMULARIO</title>
</head>

<body>
<h1>EJERCICIO 3_1: ANALISIS DE FORMULARIO</h1><br>
<br>
<h1>analisis de formularios (analisis.php)</h1>
<?php

echo "Hola <b>{$_POST['firstname']}</b>, encantado de saludarte.";
?>

</body>
```

```
</html>
```

```
24_analisis.htm
```

```
<html>
```

```
<head>
```

```
<title>EJERCICIO 3_1: ANALISIS DE FORMULARIO</title>
```

```
</head>
```

```
<body>
```

```
<h1>EJERCICIO 3_1: ANALISIS DE FORMULARIO</h1><br>
```

```
<br>
```

```
<h1>Rellena los campos (form.html)</h1>
```

```
<form action="24_analisis.php" method="post"> Tu nombre: <input type="text" name="firstname">
```

```
<input type="submit">
```

```
</form>
```

```
</body>
```

```
</html>
```

Responde a lo siguiente:

PREGUNTA	RESPUESTA
¿Qué hace la variable \$_POST?	La variable \$_POST en PHP es un array asociativo que recoge los datos enviados a través de un formulario HTML utilizando el método POST
¿Por qué usamos el metodo POST en los formularios?	Usamos el método POST en los formularios cuando queremos enviar datos de manera segura y ocultar la información que el usuario introduce.

Ejercicio 25 - 25_analisis.php

NOTA: Variacion del formulario anterior con mas variables por pasar.

```
<html>
<head>
<title>EJERCICIO 3_1: ANALISIS DE FORMULARIO</title>
</head>
<body>
<h1>EJERCICIO 3_1: ANALISIS DE FORMULARIO</h1><br>
<br>
<h1>analisis de formularios (analisis.php)</h1>
<?php
if($_POST['gender'] == 0) { echo
"Hola Sr. ";
} else {
echo "Hola Sra. ";
}

echo "<b>{"$_POST['lastname']}</b>, encantado de saludarte.";
?>
</body>
</html>
```

25_analisis.html

```
<html>
<head>
<title>EJERCICIO 3_1: ANALISIS DE FORMULARIO Version 2</title>
</head>
<body>
<h1>EJERCICIO 3_1: ANALISIS DE FORMULARIO Version 2</h1><br>
<br>
<h1>Rellena los campos (form.html)</h1>
<form action="25_analisis.php" method="post">
<input type="radio" name="gender" value="0"> Sr.
<input type="radio" name="gender" value="1"> Sra.<br> Tu
apellido:<br>
<input type="text" name="lastname">
<input type="submit">
</form>
</body>
</html>
```

Ejercicio 26 - 26_analisis.php

NOTA: Variacion del formulario analisis poniendo el codigo html del formulario y el codigo del script php en un mismo archivo.

```
<html>

<head>
<title>EJERCICIO 3_2</title>
</head>

<body>
<h1>EJERCICIO 3_1: ANALISIS DE FORMULARIO Version 3 Todo en una pagina</h1><br>
<br>
<h1>Rellena los campos (form.html)</h1>
<form action="25_analisis.php" method="post">
<input type="radio" name="gender" value="0"> Sr.
<input type="radio" name="gender" value="1"> Sra.<br> Tu
apellido:<br>
<input type="text" name="lastname">
<input type="submit" name="submitbutton" value="Envialo!">
</form>
<?php
if (isset($_POST['gender']) && isset($_POST['lastname']) && $_POST['lastname'] != "") {
if($_POST['gender'] == 0) {
echo "Hola Sr. ";
} else {
echo "Hola Sra. ";
}
echo "<b>{$_POST['lastname']}</b>, encantado de saludarte.\n";
} else {
if (isset($_POST['submitbutton'])) {
echo "Por favor rellena todos los campos" ;
}
}

?>

</body>

</html>
```

Responde a lo siguiente:

PREGUNTA	RESPUESTA
¿Qué hace la funcion isset()?	La función isset() en PHP se utiliza para comprobar si una variable está definida y no es NULL, es usada comúnmente para verificar si un campo de formulario ha sido enviado
¿Cuáles son los operadores relacionales que estamos usando?	Los operadores relacionales utilizados en este código son: ==: Este operador compara si dos valores son iguales, se usa para comparar el valor de los campos gender y verificar si es igual a 0 (Sr.) o 1 (Sra.). !=: Este operador comprueba si dos valores son diferentes, en el código se usa en \$_POST['lastname'] != "" para verificar que el campo "lastname" no esté vacío.

Ejercicio 27 - 27_pass1.php

NOTA: Este programa 27 es para poner password a archivos html. El primer archivo lleva código php con el password puesto en el encabezado antes de iniciar el código html, para redirigir a los archivos correctos o incorrectos. Luego viene el formulario que pide el password, de acuerdo al password dado, se dirige a newpage1 o newpage2 según sea el caso. Si no se sabe el password correcto se va a un tercer archivo llamado sorry.

```
<?php

if (isset($_POST['pw'])) {
$pw = $_POST['pw']; if
($pw == "magic") {
header ("Location: newpage1.html");
} elseif ($pw == "abracadabra" ){ header
("Location: newpage2.html");
} else {
header ("Location: sorry.html");
}
}

?>

<html>

<head>
<title>Ejemplo de password y header</title>
</head>

<body>
<h1> Ejemplo de password y funcion header </h1>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
<input type="text" name="pw">
```

```
<input type="submit" value="Enviala">
</form>
```

```
</body>
```

```
</html>
```

Newpage1.html

```
<html>
<head>
<title>newpage1</title>
</head>
```

```
<body>
Esta es la pagina protegida con MAGIC
newpage1.html
</body>
</html>
```

Newpage2.html

```
<html>

<head>
<title>newpage2</title>
</head>
```

```
<body>
Esta es la pagina protegida con ABRACADABRA
newpage2
</body>

</html>
```

Sorry.html

```
<html>

<head>
<title>Sorry</title>
</head>
```

```
<body>
Lo sentimos, ese no es el password correcto. Intentalo otra vez
sorry.html
```



```
</body>
```

```
</html>
```

Ver las variaciones del Cuaderno de trabajo pass2 y pass3

Responde a lo siguiente:

PREGUNTA	RESPUESTA
¿Por qué se pone el código php antes del código html?	El código PHP generalmente se coloca antes del código HTML porque el PHP se ejecuta en el servidor y genera contenido dinámico antes de que se envíe al navegador.
¿Qué hace \$_SERVER?	\$_SERVER es un superglobal en PHP que contiene información sobre el servidor y el entorno de ejecución. En este caso, \$_SERVER['PHP_SELF'] devuelve el nombre del archivo que se está ejecutando actualmente.
¿Es recomendable poner las claves de esta forma en php?	No es recomendable poner claves sensibles (como contraseñas) directamente en el código PHP, ya que hacerlo puede comprometer la seguridad del sistema.

Ejercicio 28 - 28_feedback.php

NOTA: Pequeño script que muestra un formulario que se enviara por correo electronico al webmaster todo en un solo archivo. Indicar a los alumnos que es lo que hace cada linea para que la desglosen e indiquen lo aprendido.

```
<html>
<head>
<title> un pequeño mailer para recopilar la opinión </title>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1" >
</head>
<body>
<h1>Feedback-Mailer</h1>
<p>¡Enviame un e-mail!</p>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post"> Tu
dirección de e-mail: <br>
<input type="text" name="Mail" ><br>
Tu comentario: <br>
<textarea name="message" cols="50" rows="5">
```

```

</textarea><br>
<input type="submit" value="Enviar">
</form>
<?php
$receiverMail = "tudireccion@tudominio.es"; // escribe aqui tu direcciòn if
(isset($_POST['Mail']) && $_POST['Mail'] != "") {
if (mail ($receivermail, "¡Tienes correo nuevo!", $_POST['message'], "From:
$_POST[Mail]")) {
echo "<p>Gracias por enviarme tu opiniòn.</p>\n";
} else {
echo "<p>Lo siento, ha ocurrido un error.</p>\n";
}
}
?>
</body>
</html>

```

Ver la variacion del script en el Cuaderno de Trabajo con FORMMAILER UNIVERSAL.

Responde a lo siguiente:

PREGUNTA	RESPUESTA
¿Es importante poner la linea 4 del script en nuestro codigo?	Sí, la línea 4 es importante porque define el tipo de codificación de caracteres que el navegador debe usar para mostrar la página correctamente, meta http-equiv="content-type" content="text/html; charset=ISO-8859-1" establece que el documento HTML utiliza la codificación de caracteres ISO-8859-1 (también conocida como Latin-1), que es adecuada para muchos idiomas europeos.
¿Funciona bien este programa?	Algunos problemas menores pero aun así esto es un gran si, la función mail() depende de la configuración del servidor de correo, por lo que es necesario asegurarse de que el servidor donde se está ejecutando el script tenga configurado un servidor de correo funcional para enviar los emails.

Ejercicio 29 - 29_counter.php

NOTA: Un contador de visitas para cualquier pagina web. Es necesario crear un archivo de texto en blanco llamado counter.txt en la misma ubicación donde se ejecuta el script.

```
<html>
<head>
<title>Contador Sencillo</title>
</head>

<body>
<h1>Contador Sencillo</h1>
<p>Cantidad de visitas:
<b>
<?php
// Para el contador necesitamos un archivo de texto externo donde almacenamos
// las visitas
// Aqui se verán las funciones para trabajar con archivos, en este caso
// abrir, leer, grabar y cerrar. El contador simplemente se va sumando.

$fp = fopen("counter.txt", "r+");
$counter = fgets($fp, 7);
echo $counter;
$counter ++;
rewind($fp); fputs($fp,
$counter); fclose($fp);

?>
</b></p>
</body>

</html>
```

Responde a lo siguiente:

PREGUNTA	RESPUESTA
¿Qué hace la función fopen()?	La función fopen() se utiliza para abrir un archivo en modo de lectura o escritura.
¿Qué hace fgets()?	La función fgets() lee una línea de un archivo, en nuestro caso, lea hasta 7 caracteres del archivo "counter.txt"
¿Qué hace rewind()?	La función rewind() restablece el puntero del archivo al principio.
¿Qué hace fputs()?	La función fputs() escribe contenido en un archivo, se utiliza para escribir el

	valor actualizado del contador en el archivo "counter.txt".
--	---

¿Qué hace fclose()?	La función fclose() cierra el archivo después de haber terminado con las operaciones de lectura y escritura.
¿Por qué necesitamos counter.txt?	El archivo counter.txt se utiliza para almacenar el contador de visitas.

Ejercicio 30 - 30_guestbook.php

NOTA: Libro de visitas basado en php. Es necesario crear un archivo de texto llamado guestbook.txt para almacenar los comentarios de los visitantes.

El código es bastante largo y contiene varios comentarios para facilitar la lectura del mismo.

```
<html>
<head>
<title>Un libro de visitas muy sencillo</title>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1">
</head>
<body>
<h1>Libro de visitas</h1>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post" Tu
comentario:<br>
<textarea cols="55" rows="4" name="comment"></textarea><br> Tu
nombre:<br>
<input type="text" name="name"><br>
Tu e-mail:<br>
<input type="text" name="email"><br>
<input type="submit" value="publica:">
</form>
<h3>Mostrar todos los comentarios</h3>
<?php
//Guarda el nombre del archivo en la variable
$file = "guestbook.txt";
//¿variable comment definida? ¿Nombre e e-mail no estan vacios?
//if isset($_POST['comment']) && $_POST['name'] != "" && $_POST['email'] != "" {
if (isset($_POST['comment']) && isset($_POST['name']) != "" && $_POST['email'] != "") {

$comment = $_POST['comment'];
$name = $_POST['name'];
$email = $_POST['email'];

//El archivo se abre para escritura-lectura
$fp = fopen($file, "r+");
//Leer todos los datos y almacenar en $old
```

```

Sold = fread ($fp, filesize($file));
//Se crea el vinculo de e-mail
$email = "<a href=\mailto:$email\>$email</a>";
//Se incluye la fecha y se le da formato
$dateOfEntry = date ("y-n-j");
//Ocultar caracteres html, eliminar slashes, mantener saltos de linea
$comment = htmlspecialchars($comment);
//      $comment = stripslashes(n12br($comment));
      $comment = stripslashes($comment);
//"Montar la entrada (entry) del libro de visitas
$entry="<p><b>$name</b> ($email) wrote on <i>$dateOfentry</i>;<br>$comment</p>\n";
//El cursor invisible salta al principio
rewind($fp);
//Escribir en la nueva entrada antes de las antiguas en el archivo: fputs($fp,
"$entry \n $old");
//cerrar el archivo fclose($fp);
}
//Mostrar el archivo completo
readfile($file);
?>
</body>
</html>

```

Ver la variacion con proteccion contra RELOAD en guestbook2.php del Cuaderno de trabajo pag53, ademas de la prevencion esta la averiguacion del IP del cliente y las funciones DATE

Responde a lo siguiente:

PREGUNTA	RESPUESTA
¿Qué hace htmlspecialchars()?	La función htmlspecialchars() convierte caracteres especiales a sus representaciones HTML, significando que convierte caracteres como <, >, &, y " en sus equivalentes HTML (<, >, &, ") para evitar que el navegador interprete el texto como código HTML o JavaScript.
¿Qué hace stripslashes()?	La función stripslashes() elimina la barra invertida (\) que se coloca antes de ciertos caracteres (como comillas o comillas dobles) para escaparlos.
¿Qué hace readfile()?	La función readfile() lee un archivo y lo envía directamente al navegador, se utiliza para mostrar el contenido del archivo "guestbook.txt".

Ejercicio 31 - 31_encuesta.php

NOTA: Hacer un script para un POLL o ENCUESTA que según la decision de los usuarios seleccionen de un tema. Se necesita un archivo de texto llamado RESULTS.TXT para almacenar lo que responden los usuarios. Codigo largo para hacer.

```
<?php
setcookie("check", 1);
if (isset($_POST['submit'])) {
setcookie("vote", 1);
}
?>
<html>
<head>
<title>Encuesta de opinion</title>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1">
</head>
<body>
<h1>Encuesta</h1>
<h3>¿Que opinas de este curso de php?</h3>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
<input type="radio" name="reply" value="0">
Excelente, he aprendido mucho.<br>
<input type="radio" name="reply" value="1">
Mas o menos, es muy complicado.<br>
<input type="radio" name="reply" value="2">
¡Bah! para que quiero aprender php
<br> <br>
<?php
if (empty($_POST['submit']) && empty($_COOKIE['voted'])) {
//Mostrar el botón submit solo si el formulario todavia
// no se ha enviado y el usuario no ha votado.
?>
<input name="submit" type="submit" value="vota!">
<?php
} else {
echo "<p>Gracias por tu voto.</p>\n";
//¿Formulario enviado? ¿cookies activas? ¿pero todavia no se ha votado?
if (isset($_POST['reply']) && isset($_COOKIE['check']) && empty($_COOKIE['voted'])) {
//Guardar nombre de archivo en la variable
$file="results.txt";
$fp=fopen($file, "r+");
```



```
$vote=fread($fp, filesize($file));  
//Descomponer la string del archivo en array con coma como separador  
$arr_vote = explode(",", $vote); //explode convierte la string en array  
//¿que valor se ha seleccionado en el formulario?  
//¡El recuento aumenta en 1!  
$reply = $_POST['reply'];  
$arr_vote[$reply]++;  
//volver a montar la string  
$vote = implode(",", $arr_vote); // implode vincula elementos de la array a string rewind($fp);  
//Escribir nueva string en el archivo fputs($fp, $vote);  
fclose($fp);  
}  
}  
?>  
</form>  
<p>  
[ <a href="results.php" target="_blank">ver resultados de la encuesta</a>]  
</p>  
</body>  
</html>
```

Ejercicio 33 - 33_seguridad

NOTA: Este es el primer ejercicio del segundo cuaderno o manual del Curso: PHP y MySQL (PC Cuadernos No.30). A partir de aquí se manejarán los ejercicios de este cuaderno. Aquí se muestra una falla en la seguridad de PHP cuando tenemos el `register_globals = ON` en la configuración de `php.ini`.

Para ver que es fácil entrar a la página protegida con password, basta teclear en el navegador:

http://localhost/33_seguridad.php?login=1

Y entraremos en el área “protegida” por el programa. Así cualquiera puede entrar.

```
<html>
<head>
  <title>El gran agujero de seguridad</title>
</head>
<body>
  <h2>Agujero de seguridad en register_globals = On</h2>
  <form action="<?php echo "33_seguridad.php"; ?>" method="post">
    Contraseña: <input type="password" name="pass">
    <input type="submit" value="Enviar">
  </form>
  <?php
    if ($pass == "abc_xyz_123") {
      $login = true;
    }
    if ($login) {
      echo "<p>Aquí empieza el arma secreta.</p>";
    }
  ?>
</body>
</html>
```

Ejercicio 34 - 34_album.php

NOTA: Este ejemplo muestra como crear un album de fotos, asi como algunas funciones mas avanzadas de php que pueden ser consultadas en el manual de php oficial. El programa subira a un directorio especificado una imagen seleccionada por el usuario siempre que sea GIF o JPG y no pase de 200kb, a este directorio y luego mostrara todas las imágenes que se hayan subido.

Contiene varias sentencias de condicion, funciones de php para manejo de archivos y directorios y un formulario, asi como variables internas y externas.

Mas adelante viene una extension de este programa como prueba final.

```
<html>
<head>
  <title>Mi album de fotografias</title>
</head>

<body>
<h1> Mi album de fotos en linea </h1>
<h3> Cargar archivo </h3>

<form action='<?php echo $_SERVER['PHP_SELF'] ?>' method="post" enctype="multipart/form-data">
<input type="file" name="archivo">
<input type="submit" name="submit" value="Cargar archivo">
</form>

<?php
$ruta="img/"; // Indicar ruta
if (isset($_FILES['archivo']) && $_FILES['archivo']['size'] > 0) {
  $tamanyomax = 200000; // Indicar tamaño en bytes
  $nombretemp = $_FILES['archivo']['tmp_name'];
  $nombrearchivo = $_FILES['archivo']['name'];
  $tamanyoarchivo = $_FILES['archivo']['size'];
  $tipoarchivo = GetImageSize( $nombretemp );
  if ($tipoarchivo[2] == 1 || $tipoarchivo[2] == 2) { //Gif o Jpg?
    if (tamanyoarchivo <= $tamanyomax) { // Archivo demasiado grande?
      if (move_uploaded_file($nombretemp, $ruta.$nombrearchivo)) {
        echo "<p>El archivo se ha cargado <b>con éxito</b>."
        Tamaño de archivo: <b>$tamanyoarchivo</b> bytes,
        Nombre de imagen: <b>$nombrearchivo</b><br></p>";
      } else {
        echo "<p>No se ha podido cargar el archivo.</p>";
      }
    }
  }
}
```

```

    }
} else {
    echo "<p>El archivo tiene mas de <b>$tamanyomax bytes</b> y es demasiado grande.</p>";
}
} else {
    echo "<p>No es un archivo GIF o JPG valido.</p>";
}
echo "<form action='{$_SERVER['PHP_SELF']}' method='post'>
    <input type='submit' value='OK'></form>";
}
$handle = opendir($ruta); // Abrir archivos
while ($file=readdir($handle)) {
    if ($file != "." && $file != "..") {
        $tamanyo = GetImageSize($ruta.$file);
        echo "<p><img src='$ruta$file' $tamanyo[3]><br></p>\n";
    }
}
closedir($handle); // Fin lectura de archivos
?>

</body>
</html>

```

PRACTICA 1

Para la leccion 1.1 deberas leer las paginas de la 4 a la 10 del Cuaderno de Trabajo. A continuacion responde lo que se te pide:

PREGUNTA	RESPUESTA
¿Quién invento PHP?	Rasmus Lerdorf creo php
¿Qué es PHP?	Un lenguaje de programación de código abierto que mayormente se usa para el desarrollo de aplicaciones y páginas web.
¿Qué significa PHP?	Su significado es el de Hypertext Preprocessor

¿Cuáles son las diferencias mas notables entre JavaScript y PHP?	Las diferencias que mas podemos ver son que php se puede ejecutar en un servidor, javascript por su parte se ejecuta en un navegador, otra cosa que podemos ver es que el aprendizaje en php es más rápido que lo que podemos ver en el javascript
¿Qué tipo de licencia tiene PHP?	PHP v3.01, la cual es una licencia de código abierto.
¿Dónde podemos encontrar ayuda para php?	En muchas partes de la web ya sea por su propia pagina abierta al publico o si quieres algo diferente puedes ir a youtube o cualquier red social y buscar tutoriales, videos que puedan ayudar en php.
¿Qué necesitamos para trabajar con PHP?	un servidor web, un analizador de PHP y un navegador web, es necesario que el servidor web tenga soporte para PHP.
¿Cómo accesamos a nuestro servidor local?	En este caso usaremos XAMPP solo tenemos que escribir en nuestro buscador http://localhost para acceder a nuestro servidor local.
¿Dónde se encuentra la pagina de inicio de XAMPP?	Es lo mismo de la anterior pregunta solo inicia el panel de XAMPP con lo que necesites como mysql, etc y solo usa el http localhost para ir a la pagina de inicio.

PRACTICA 2

Investiga sobre las diferentes versiones de php existentes y sus características

PHP 4 (2000)

- **Motor Zend:** Introdujo el motor Zend, que mejoró significativamente el rendimiento.
- **Soporte para sesiones:** Mejor manejo de sesiones.
- **Buffering de salida:** Permite manipular la salida antes de enviarla al navegador.

- **Soporte para más APIs:** Mejor integración con bases de datos y protocolos.

PHP 5 (2004)

- **Programación Orientada a Objetos (POO):** Introdujo un modelo de objetos más robusto con soporte para clases, métodos, propiedades, etc.
- **PDO (PHP Data Objects):** Una extensión para acceder a bases de datos de manera segura y consistente.
- **Mejoras en XML:** Soporte mejorado para manejo de XML.
- **Excepciones:** Introdujo el manejo de excepciones con try, catch, y throw.
- **Mejoras en rendimiento:** Optimizaciones en el motor Zend.

PHP 5.4 (2012)

- **Sintaxis corta para arrays:** [] en lugar de array().
- **Traits:** Permite reutilizar código en clases sin herencia múltiple.
- **Mejoras en el servidor web integrado:** php -S para desarrollo local.
- **Mejoras en rendimiento:** Optimizaciones adicionales.

PHP 5.6 (2014)

- **Operador de exponenciación:** **.
- **Argumentos variables con ...:** Permite pasar arrays como argumentos.
- **Constantes escalares:** Permite usar expresiones en constantes.
- **Mejoras en seguridad:** Mejor manejo de contraseñas con password_hash().

PHP 7.0 (2015)

- **Mejoras de rendimiento:** Hasta el doble de velocidad en comparación con PHP 5.6.
- **Tipos escalares:** Declaración de tipos para parámetros y valores de retorno (int, float, string, bool).
- **Operador de fusión null:** ?? para simplificar la verificación de valores nulos.
- **Declaraciones de tipo de retorno:** Especificar el tipo de retorno de una función.
- **Nuevo operador de nave espacial:** <=> para comparaciones de tres vías.
- **Mejoras en manejo de errores:** Excepciones para errores fatales.

PHP 7.1 (2016)

- **Tipos nullable:** Permite especificar que un tipo puede ser null (por ejemplo, ?string).
- **Void return type:** Especificar que una función no devuelve ningún valor.
- **Iterables:** Tipo para arrays y objetos iterables.
- **Mejoras en seguridad:** Mejor manejo de claves en mcrypt.

PHP 7.3 (2018)

- **Flexibilidad en Heredoc/Nowdoc:** Mejoras en la sintaxis.
- **Funciones de array:** array_key_first() y array_key_last().
- **Mejoras en JSON:** Soporte para manejo de errores en json_encode().

PHP 7.4 (2019)

- **Propiedades tipadas:** Declaración de tipos para propiedades de clases.
- **Arrow functions:** Sintaxis más corta para funciones anónimas.
- **Precarga de clases:** Mejora el rendimiento al cargar clases en memoria.
- **Operador de asignación null coalescing:** ??=.

PHP 8.0 (2020)

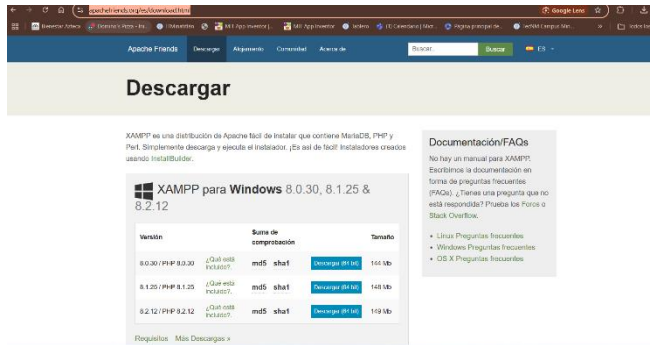
- **JIT (Just-In-Time Compilation):** Mejora significativa en el rendimiento.
- **Atributos:** Sintaxis para metadatos en clases y funciones.
- **Union types:** Permite especificar múltiples tipos para parámetros y retornos.
- **Constructor Property Promotion:** Simplifica la declaración de propiedades en constructores.
- **Mejoras en manejo de errores:** Excepciones para advertencias.

PHP 8.2 (2022)

- **Readonly classes:** Clases completas de solo lectura.
- **Mejoras en tipos:** Tipos null, false, y true como tipos independientes.
- **Sensitive Parameter:** Marca parámetros sensibles para evitar que se muestren en errores.
- **Nuevas funciones:** Como `str_increment()` y `str_decrement()`.

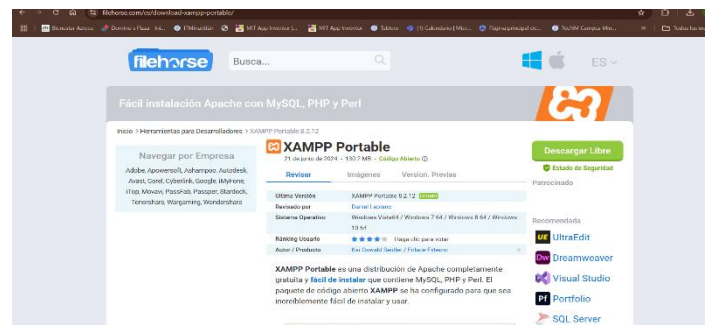
PHP 8.3 (2023)

- **Mejoras en JIT:** Optimizaciones adicionales.
- **Nuevas funciones:** Como `json_validate()` para validar JSON sin decodificarlo.
- **Mejoras en seguridad:** Refuerzos en el manejo de contraseñas y criptografía.



PRACTICA 3

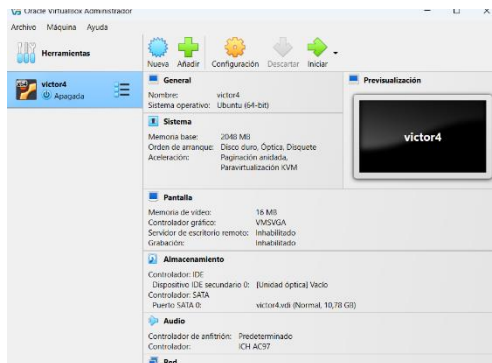
Realiza la practica de Instalacion XAMPP de tu Cuaderno de Trabajo (Busca en internet y descarga el XAMPP. Tambien deberas instalarlo en un USB personal)



PREGUNTA	RESPUESTA
¿De que sitio obtuviste el XAMPP?	De su página principal la oficial
¿De que sitio obtuviste el XAMPP Portable?	En una pagina conocida como filehorse.
¿Qué servicios instalaste del XAMPP?	Todos los servicios, apache, filezilla, mysql, mercury, tomcat.

PRACTICA 4

Realiza la práctica de virtualización de tu Cuaderno de Ejercicios.



en este caso yo use la de virtualbox y Ubuntu server.

PRACTICA 5

Confeccione un programa que muestre una serie de mensajes en la pagina empleando el comando ECHO. Tenga en cuenta que cuando utiliza el comando ECHO el mensaje se debe encerrar entre comillas dobles.

PRACTICA 6

Sabiendo que la funcion RAND nos retorna un valor aleatorio entre un rango de dos enteros:

```
$num = rand(1,100);
```

En la variable \$num se almacena un valor entero que la computadora genera en forma aleatoria entre 1 y 100. Hacer un programa que lo muestre por pantalla al valor generado. Mostrar ademas si es menor o igual a 50 o si es mayor. Para imprimir el contenido de una variable tambien utilizamos el comando ECHO

PRACTICA 7

Definir una variable de cada ipo: integer, double, string y boolean. Luego imprimirlas en la pagina, una por linea.

PRACTICA 8

Definir tres variables enteras. Luego definir un string que incorpore dichas variables y las sustituya en tiempo de ejecucion. Recordar que una variable se sustituye cuando el string esta encerrado por

comillas dobles:

```
$precio = 90;
```

```
Echo "La computadora tiene un precio de $precio";
```

PRACTICA 9

Generar un valor aleatorio entre 1 y 5. Luego imprimir en castellano el numero (Ej. Si se genera el 3 luego mostrar en la pagina el string "tres"). Para ver si una variable es igual a cierto valor debemos plantear una condicion similar a:

```
If ($valor==3)
{
    //algoritmo
}
```

PRACTICA 10

Mostrar la tabla de multiplicar del 2. Emplear el for, luego el while y por ultimo el do/while. La estructura FOR permite incrementar una variable de 2 en 2:

```
For ($f=2; $f<=20; $f=$f+2)
```

PRACTICA 10 – Formulario (controles text y submit)

Confeccionar un formulario que solicite la carga de un nombre de persona y su edad, luego mostrar en otra pagina si es mayor de edad (si la edad es mayor o igual a 18).

PRACTICA 11 – Formulario (control radio)

Solicitar que se ingrese por teclado el nombre de una persona y disponer tres controles de tipo radio que nos permitan seleccionar si la persona:

- 1) no tiene estudios
- 2) estudios primarios
- 3) estudios secundarios

En la pagina que procesa el formulario mostrar el nombre de la persona y un mensaje indicando el tipo de estudios que posee.

PRACTICA 12 - Formulario (control checkbox)

Confeccionar un formulario que solicite la carga del nombre de una persona y que permita seleccionar una serie de deportes que practica (futbol, basket, tennis, voley).

Mostrar en la pagina que procesa el formulario la cantidad de deportes que practica

PRACTICA 13 – Formulario (control select)

Confeccionar un formulario que solicite el ingreso del nombre de una persona y un combo de selección (en este ultimo permitir la selección de los ingresos anuales de la persona: 1-1000, 1001-3000, >3000).

En la pagina que procesa el formulario mostrar un mensaje si debe pagar impuestos a las ganancias (si supera 3000).

PRACTICA 14 – Formulario (control textarea)

Confeccionar una pagina que muestre un contrato dentro de un textarea, disponer puntos suspensivos donde el operador debe ingresar un texto. La pagina que procesa el formulario solo debe mostrar el contrato con las modificaciones que hizo el operador.

Ej. De un contrato puede ser:

En la ciudad de [.....], se acuerda entre la Empresa [.....] representada por el Sr. [.....] en su carácter de Apoderado, con domicilio en la calle [.....] y el Sr. [.....], futuro empleado con domicilio en [.....], celebrar el presente contrato a Plazo Fijo, de acuerdo a la normativa vigente de los articulos 90, 92, 93, 94, 95 y concordantes de la Ley de Contrato de Trabajo No. 20744.

PRACTICA 15 – Vectores (tradicionales)

Definir un vector con los nombres de los dias de la semana. Luego imprimir el primero y el ultimo elemento del vector.

PRACTICA 16 – Creacion de un archivo de texto

Confeccionar un programita en PHP que permita hacer el pedido de pizzas via internet.
El formulario debe ser:

Nombre: [.....]
Direccion: [.....]
Jamon y queso [x]
Cantidad [....]

Napolitana [x]
Cantidad [....]

Mozzarella [x]
Cantidad [....]

[Confirmar]

Para el ingreso del nombre, direccion y cantidad de pizzas de cada tipo disponer objetos de la clase "text". Disponer tres objetos de tipo "check" para seleccionar los tipos de pizzas.
Por ultimo disponer un boton para el envio de datos: "submit".

Grabar en un archivo de texto cada pedido, separados por una línea de puntos (obligatoriamente dar el nombre de texto como "datos.txt", esto es para que no se llene el disco duro de archivos.

PRACTICA 17 – Lectura de un archivo de texto

Confeccionar un programa que muestre el archivo de pedido de pizzas via internet anteriormente creado. (Obligatoriamente dar el nombre de archivo de texto como "datos.txt").

PRACTICA 18 – Vectores (asociativos)

Crear un vector asociativo que almacena las claves de acceso de 5 usuarios de un sistema. Acceder a cada componente por su nombre. Imprimir un componente por cada vector.

PRACTICA 19 – Funciones

Confeccionar un formulario que solicite la carga del nombre de usuario y su clave en dos oportunidades. En la página que se procesan los datos del formulario implementar una función que imprima un mensaje si las dos claves ingresadas son distintas.

PRACTICA 22 – Consulta (selección de registros de una tabla)

Confeccionar un programa que permita ingresar el nombre de un alumno en un formulario, luego mostrar los datos del mismo (tener en cuenta que puede haber más de un alumno con el mismo nombre)

PRACTICA 23 – Delete (Baja de un registro en una tabla)

Confeccionar un programa que permita ingresar el nombre de un curso por teclado y

posteriormente efectue el borrado de dicho registro en la tabla CURSOS. Mostrar un mensaje si no existe el curso.