PHP

Roberto Gómez Cárdenas
rogomez@itesm.mx
http://homepage.cem.itesm.mx/rogomez

Lámina 1 Roberto Gómez C.

¿Qué es PHP?

- PHP: PHP: Hypertext Preprocessor
- Procesador de Hipertexto
- Lenguaje interpretado de alto nivel que permite desarrollar fácilmente páginas dinámicas
  - Similar a los lenguajes C y Perl
- El código va embebido en HTML mediante etiquetas especiales de comienzo y final
- El código PHP se ejecuta en el servidor.
  - El cliente nunca vera el código del programa PHP, sólo llegarán las páginas HTML que genere el programa.

Lámina 2 Roberto Gómez C.

Esquema funcionamiento

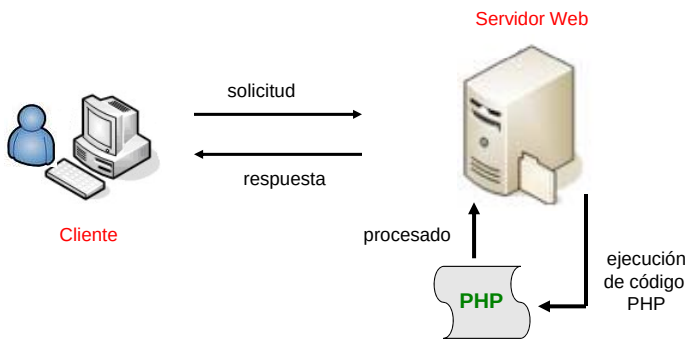


Lámina 3 Roberto Gómez C.

¿Para qué sirve?

- Uso primario
  - Escribir scripts de servidor
    - Procesar la información de formularios, generar páginas con contenidos dinámicos, o mandar y recibir cookies.
  - Otros
    - Scripts de línea de comandos
    - Aplicaciones gráficas clientes

Lámina 4 Roberto Gómez C.

¿Por qué PHP?

- Razones para usar este lenguaje
  - Fácil de aprender
  - Permite hacer varias cosas
    - Acceso a bases de datos
    - Creación de documentos PDF, imágenes y películas flash.
    - Soporte de comunicación con otros servicios.
    - Procesamiento de texto.
  - Esta ampliamente probado como herramienta y se puede usar en cualquier plataforma.
  - Existen numerosos recursos en la web que pueden facilitar el desarrollo de las aplicaciones (bibliotecas de clases).

Lámina 5 Roberto Gómez C.

Sintaxis básica

- Saliendo de HTML
  - PHP solo interpreta el texto entre etiquetas especiales.
    - <?php CODIGO PHP ?>
    - <? CODIGO PHP ?>
    - <script language="php"> CODIGO PHP </script>
    - <% CODIGO PHP %>
  - Separación de instrucciones por el carácter: ;
  - Comentarios
    - Tipo C, C++

Lámina 6 Roberto Gómez C.

Tipos

- Soporte de tipos PHP
  - array
  - punto flotante
  - entero
  - objeto
  - cadena de caracteres
- La ligadura del tipo es dinámica, aunque se puede forzar la conversión a un tipo.
  - Funciones gettype( ), settype( )

Lámina 7 Roberto Gómez C.

Variables

- Las variables se representan como un signo de dólar seguido por el nombre de la variables.
- El nombre de la variables es sensible a minúsculas y mayúsculas.
  - Asignación por valor

```
$i = 10;
$nombre = “beto”;
```

  - Asignación por referencia

```
$ref = &$nombre
```

  - Variables de tipo arreglo (array)

```
$a = array (“uno”=>1, “dos”=>2, “tres”=>3);
```

Lámina 8 Roberto Gómez C.

Clasificación variables

- Variables variables
  - Son nombres de variables que se pueden establecer y usar dinámicamente.

```
$a = “hola”;
$$a = “mundo”;
echo ${$a};
Echo $mundo;
```

- Constantes
  - Se pueden definir constantes de valores escalares.
  - Por ejemplo, considerando la función define( ).

```
define(PI, 3.141592);
```

Lámina 9 Roberto Gómez C.

Tipos operadores

- Operadores aritméticos.
- Operadores de asignación.
- Operadores de manipulación de bits
- Operadores de comparación.
- Operadores de incremento y decremento.
- Operador de ejecución (“ “).
- Operadores lógicos.
- Operadores de cadenas.

Lámina 10 Roberto Gómez C.

Ámbito de variables

- Es el contexto dentro del que la variable esta definida (abarca archivos incluidos y requeridos), habitualmente el documento.
- En las funciones definidas por el usuario el ámbito, por defecto, es local a la función.
  - Se pueden definir variables globales en una función con la palabra reservada global.
  - También se puede utilizar el array $GLOBALS.
  - Las variables estáticas son variables globales a la función pero que conservan el valor en llamadas sucesivas.
    - Se definen con static

Lámina 11 Roberto Gómez C.

Instalación y uso PHP

- Instalar PHP
  - Linux:
    - **PHP**: www.php.net
    - php-3.0.x.tar-gz
  - Windows
    - EasyPHP
- Comprobando que funciona
  - Crear un archivo con la siguiente línea

```
<?php phpinfo() ?>
```

  - Nombre archivo: p1.php
  - Guardar el archivo en en el directorio de documentos de Apache y llamarlo desde el navegador.
    - Si todo se ha echo bien saldrá una página con todas las variables de PHP.

Lámina 12 Roberto Gómez C.

Si todo sale bien…

PHP Version 5.2.3-1ubuntu6.3

System	Linux grenadine 2.6.18-xenU #3 SMP Thu Jan 10 15:56:11 CET 2008 i686
Build Date	Jan 10 2008 09:24:13
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
additional .ini files parsed	/etc/php5/apache2/conf.d/curl.ini, /etc/php5/apache2/conf.d/gd.ini, /etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/mysqli.ini, /etc/php5/apache2/conf.d/pdo.ini, /etc/php5/apache2/conf.d/pdo_mysql.ini, /etc/php5/apache2/conf.d/pspell.ini, /etc/php5/apache2/conf.d/tidy.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled
IPv6 Support	enabled

Lámina 13 Roberto Gómez C.

Ejecutando programas php

- Los programas se pruebas a través de un browser que accede a un servidor donde se encuentran los programas.
- Editar archivos con el block de notas, con dreamweaver, eclipse, netbeans, o con el editor que se prefiera.
- Almacenar el archivo en el directorio donde se almacenan los archivos html por default del servidor web.
- Abrir el browser con la dirección del servidor y la ruta donde esta el programa.

Lámina 14 Roberto Gómez C.

Ejemplo programa PHP

```
<?php
echo "<center>
        <h1>
          Saludo desde PHP
        </h1>
      </center>"
?>
```

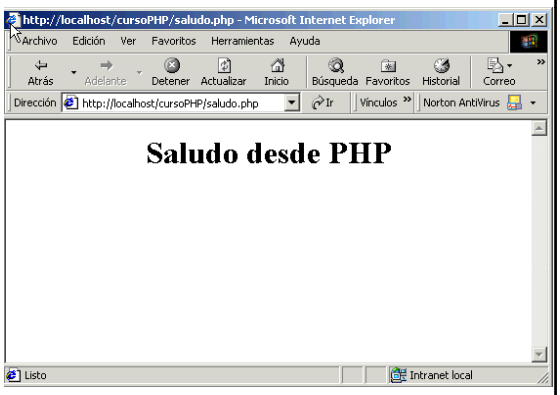


Lámina 15 Roberto Gómez C.

Ejecutando desde línea comandos

- Es el mismo código, solo que la primera línea debe indicar donde se encuentra el interprete de php.
- Ejemplo (nombre archivo p1.php):

```
#!/usr/bin/php
<?php
echo "<center>
        <h1>
          Saludo desde PHP
        </h1>
      </center>"
?>
```

```
$ ./p1.php
Saludo desde PHP
$
```

Lámina 16 Roberto Gómez C.

Un segundo ejemplo

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>
Parte de HTML normal.
<BR><BR>

<?php
  echo "Parte de PHP<br>";
   for($i=0;$i<10;$i++)
  {
    echo "Linea ".$i."<br>";
  }
?>

</body>
</html>
```

Lámina 17 Roberto Gómez C.

Variables

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>
Parte de HTML normal.
<BR><BR>

<?php
  $a = 1;
  $b = 3.34;
  $c = "Hola Mundo";
  echo $a,"<br>",$b,"<br>",$c;
?>

</body>
</html>
```

Lámina 18 Roberto Gómez C.

Operadores aritméticos

Operador	Nombre	Ejemplo	Descripción
+	Suma	5 + 6	Suma de dos números
-	Resta	7 – 9	Resta de dos números
*	Multiplicación	6 * 3	Multiplica dos números
/	División	4 / 9	Divide dos números
%	Módulo	7 % 2	Devuelve el residuo
++	Suma 1	$a ++	Suma 1 al contenido de la variable.
--	Resta 1	$a--	Resta 1 al contenido de una variable.

Lámina 19 Roberto Gómez C.

Operadores de asignación

Operador	Ejemplo	Es el mismo que…
=	x=y	x = y
+=	x+=y	x = x + y
-=	x -=y	x = x – y
*=	x*=y	x = x * y
/=	x/=y	x = x / y
.=	x.=y	x = x . Y
%=	x%=y	x = x % y

Lámina 20 Roberto Gómez C.

Ejemplo operadores aritméticos

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>

<?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>
```

Lámina 21 Roberto Gómez C.

Operadores de comparación

Operador	Nombre	Ejemplo	Descripción
==	Igual	$a == $b	$a es igual $b
!=	Diferente	$a != $b	$a es diferente $b
<	Menor que	$a < $b	$a es menor que $b
>	Mayor que	$a > $b	$a es mayor que $b
<=	Menor o igual	$a <= $b	$a es menor o igual que $b
>=	Mayor o igual	$a >= $b	$a es mayor o igual que $b

Lámina 22 Roberto Gómez C.

Ejemplo operadores comparación

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>

<?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 $b <= $c,"<br>";
?>

</body>
</html>
```

Lámina 23 Roberto Gómez C.

Operadores lógicos

Operador	Nombre	Ejemplo	Descripción
&&	Y	(7 > 2) && (2 < 4)	Regresa verdadero cuando ambas condiciones son verdaderas.
and	Y	(7 > 2) and (2 < 4)	Regresa verdadero cuando ambas condiciones son verdaderas.
||	O	(7 > 2) || (2 < 4)	Regresa verdadero cuando al menos una de las dos es verdadera.
or	O	(7 > 2) or (2 < 4)	Regresa verdadero cuando al menos una de las dos es verdadera.
!	No	! (7 > 2)	Niega el valor de la expresión.

Lámina 24 Roberto Gómez C.

Ejemplo operadores lógicos

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>

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

</body>
</html>
```

Lámina 25 Roberto Gómez C.

Sintaxis operador condicional

```
<?php
   if (condición)
   {
     Instrucciones a ejecutar
     cuando la
     condición es cierta.
   }
   else
   {
     Instrucciones a ejecutar
     cuando la
     condición es falsa.
   }
?>
```

Lámina 26 Roberto Gómez C.

Primer ejemplo operador condicional

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>

<?php
  $a = 8;
  $b = 3;
  if ($a < $b)
  {
    echo "a es menor que b";
  }
  else
  {
    echo "a no es menor que b";
  }
?>

</body>
</html>
```

Lámina 27 Roberto Gómez C.

Segundo ejemplo operador condicional: switch

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>

<?php
  $posicion = "arriba";

  switch($posicion) {
    case "arriba":   // Bloque 1
      echo "La variable contiene";
      echo " el valor arriba";
      break;
    case "abajo":   // Bloque 2
      echo "La variable contiene";
      echo " el valor abajo";
      break;
   default:   // Bloque 3
        echo "La variable
contiene otro valor";
        echo " distinto de arriba
y abajo";
    }
?>

</body>
</html>
```

Lámina 28 Roberto Gómez C.

Sintaxis de la primitiva while( )

```
<?php
  while (condición)
  {
    Instrucciones a ejecutar.
  }
?>
```

Lámina 29 Roberto Gómez C.

Ejemplo while( )

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>
Inicio<BR>

<?php
  $i=0;
  while ($i<10)
  {
    echo "El valor de i es ", $i,"<br>";
    $i++;
  }
?>

Final<BR>

</body>
</html>
```

Lámina 30 Roberto Gómez C.

Sintaxis for(; ;)

```
<?php
  for (inicial ; condición ; ejecutar en iteración)
  {
    Instrucciones a ejecutar.
  }
?>
```

Lámina 31 Roberto Gómez C.

Ejemplo usos for(; ;)

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>
Inicio<BR>

<?php
  for($i=0 ; $i<10 ; $i++)
  {
    echo "El valor de i es ", $i,"<br>";
  }
?>

Final<BR>
</body>
</html>
```

Lámina 32 Roberto Gómez C.

Primitivas de salida

- Sintaxis

```
<?php
  printf(cadena formato, variable1, variable2...);
?>
```

- Formateo de variables

Elementos	Tipo de variable
%s	Cadena de caracteres
%d	Numero sin decimales
%f	Numero con decimales
%c	Carácter ASCII

Lámina 33 Roberto Gómez C.

Ejemplo primitiva de salida

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>

<?php
  $var="texto";
  $n=3;
  printf("Posible intercalar <b>%s</b> con números <b>%d</b> <br>",$var,$n);
  printf("<TABLE BORDER=1 CELLPADDING=20>");
  for ($i=0;$i<10;$i++)
  {
    printf("<tr><td>%10.d</td></tr>",$i);
  }
  printf("</table>");
?>

</body>
</html>
```

Lámina 34 Roberto Gómez C.

Primitica printf( )

- Despliega una cadena de caracteres formateada.
- Sintaxis:

  printf(formato,arg1,arg2,arg++)

- La cadena que contiene las instrucciones de formato permite usar tanto patrones como caracteres normales.
  - De la misma forma que se hace en lenguaje C
- Los patrones van precedidos del signo %, y sus valores se incluiran en la cadena formateada, o indicarán como debe ser presentada.

Lámina 35 Roberto Gómez C.

Caracteres de formato

Carácter	Descripción: el argumento es interpretado/tratado como un:
b	número entero, y presentado como un número binario.
c	entero, y presentado como el carácter ASCII con dicho valor.
d	entero y presentado con notación decimal
e	entero y presentado con notación exponencial.
f	*double* y presentado como un número de coma flotante.
o	entero, y presentado como un número octal.
x	entero y presentado como un número hexadecimal (con minúsculas).
X	entero y presentado como un número hexadecimal (con mayusculas).
s	una cadena y es presentado como tal.

Lámina 36 Roberto Gómez C.

Ejemplo de uso

```
<?php
        // M_PI es una variable predefinida  con el valor de Pi
  printf ("Pi es: %d", M_PI);    // 3
  printf ("Pi es: %.2f", M_PI);  // 3.14
  printf ("Pi es: %.3f", M_PI);  // 3.142
  printf ("Pi es: %.4f", M_PI);  // 3.1416

  $numero = 65;

  printf("%c",$numero);     // imprime   "A"
  for($count=97; $count<=122; $count++)        // imprime caracteres ascii desde
    {                                          // la posicion 97 a la 122:
    printf ("%c" , $count);                    // abcdefghijklmnopqrstuvwxyz
    }

   printf("uno vale %d; otro vale %c", $numero,$numero);
   // imprime "uno vale 65; otro vale A"
                                          :
                                          :
```

Ejemplo tomado de: http://www.ignside.net/man/php/printf.php

Ejemplo de uso

```
:
$number= 255;

printf("En formato decimal: %d",$number);          //"En formato decimal: 255"
printf("En formato hexadecimal: %x",$number);  //"En formato hexadecimal: ff"
printf("En formato hexadecimal: %X",$number); //"En formato hexadecimal: FF"
printf("En formato octal: %o",$number);             //"En formato octal: 377"
printf("Decimal exponencial: %e",$number);      //"Decimal exponencial: 2.55000e+2"
printf("En binario: %b",$number);                   //"En binario: 11111111"
printf("%.2s\n", "patata");     // pa
printf("%.4s\n", "patata");     // pata
printf("*%9.4s\n", "patata");   // pata Total 9 caracteres, resto espacios
printf("*%9s\n", "patata");     // patata. El resto hasta 9 caracteres lo rellena con espacios
printf("%-9s*\n", "patata");    // patata. El resto hasta 9 caracteres  lo rellena con espacios
                                // a la izquierda
printf("%09s\n", "patata");     // patata. El resto hasta 9 caracteres lo rellena con ceros
printf("%'x9s\n", "patata");    // patata. El resto hasta 9 caracteres lo rellena con x
                                // (fijarse en la comilla simple)
?>
```

Ejemplo tomado de: http://www.ignside.net/man/php/printf.php

Nota sobre printf() y html

- Espacios en blanco en HTML no son considerados.
- Por ejemplo:

```
<?php
    $frase = “America      sera      campeon”;
    printf(“%s”,frase);
?>
```

- La salida es:

```
America    sera    campeon”
```

- Posible usar la clausula <pre>

```
<?php
    $frase = “America      sera      campeon”;
    print“<pre>\n”;
    printf(“%s”,frase);
    print“</pre>\n”;
?>
```

Lámina 39 Roberto Gómez C.

Funciones formateo de cadenas

- strlen(cadena)
  - Número de caracteres de una cadena
- split(separador, cadena)
  - Divide cadena en varias, usando separador
- sprintf(cadena formato, var1, var2, …)
  - Igual que printf() pero el resultado se regresa en forma de cadena
- substr(cadena, inicio, longitud)
  - Extrae una subcadena de otra, empezando por inicio y de longitud

Lámina 40 Roberto Gómez C.

Funciones formateo de cadenas

- chop(cadena)
  - Elimina los saltos de línea y los espacios finales de una cadena.
- strpos(cadena1, cadena2)
  - Busca la cadena2 dentro de cadena1, regresando la posición en que se encuentra.
- str_replace(cadena1, cadena2, texto)
  - Reemplaza la cadena1 por la cadena2 en el texto.

Lámina 41 Roberto Gómez C.

Ejemplo uso de cadenas

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>

<?php
  echo strlen("12345"),"<br>";

  $palabras=split(" ","Esto es una prueba");
  for($i=0;$palabras[$i];$i++)
    echo $palabras[$i],"<br>";

  $resultado=sprintf("8x5 = %d <br>",8*5);
  echo $resultado,"<br>";

  echo substr("Devuelve una subcadena de otra",9,3),"<br><br>";
```

Lámina 42 Roberto Gómez C.

Ejemplo uso de cadenas

```
if (chop("Cadena \n\n ") == "Cadena")
    echo "Iguales<br><br>";

  echo strpos("Busca la palabra dentro de la frase", "palabra"),"<br><br>";

  echo str_replace("verde","rojo","Un pez de color verde, como verde es la
hierba."),"<br>";

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

Lámina 43 Roberto Gómez C.

Funciones

```
<?php
  function Nombre(parametro1, parametro2...)
  {
    Instrucción1;
    Instrucción2;
    Instrucción3;
    Instrucción4;

    return valor_de_regreso;
  }
?>
```

Lámina 44 Roberto Gómez C.

Ejemplo función

```
<?php

  function media_aritmetica($a, $b)
  {
    $media=($a+$b)/2;
    return $media;
  }

  echo media_aritmetica(4,6),"<br>";
  echo media_aritmetica(3242,524543),"<br>";

?>
```

Lámina 45 Roberto Gómez C.

Librerías en PHP

- Permiten agrupar varias funciones y variables en un mismo archivo.
- Luego se puede incluir esta librería en distintas páginas y disponer de estas funciones fácilmente.
- La instrucción para incluir una librería se utiliza la instrucción

include (“nombre de la librería”)

Lámina 46 Roberto Gómez C.

Ejemplo definición librería (1)

- Archivo: libpagina.html

```
<?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: Cachafas
<? }
?>
```

Lámina 47 Roberto Gómez C.

Primer ejemplo uso librerías

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>
<?php include("libpagina.phtml") ?>
<?php CabeceraPagina(); ?>
<TABLE>
<TR>
```

```
<TD>
      Esta es otra página<BR><BR>
      completamente distinta<BR><BR>
      pero comparte el pie y la cabecera con la
otra.<BR><BR>
  </TD>
</TR>
</TABLE>
<?php PiePagina(); ?>

</body>
</html>
```

Lámina 48 Roberto Gómez C.

Segundo ejemplo uso librerías

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>
<?php include("libpagina.phtml") ?>
<?php CabeceraPagina(); ?>
<TABLE>
<TR>

<TD>
Página 1
<BR><BR><BR><BR><BR>

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

```
fin<BR><BR>
  </TD>
</TR>
</TABLE>
<?php PiePagina(); ?>
</body>
</html>
```

Lámina 49 Roberto Gómez C.

Funciones require y require_once

- La función require() es identica a include( ) excepto que en presencia de una falla producira un error E_ERROR.
  - Detendra el script, mientras que include( ) emitira una advertencia (E_WARNING) que permitira que el script continue.
- El enunciado requiere_once() es identico a require excepto que PHP verificará si el archivo ha sido incluido previamente, si es el caso, no lo incluirá de nuevo.

Lámina 50 Roberto Gómez C.

Formularios: envío y recepción datos

- PHP proporciona una forma sencilla de manejar formularios.
- Los formularios permiten procesar la información que el usuario ha introducido.
- Al diseñar un formulario debemos indicar la página PHP que procesará el formulario, así como en método por el que se le pasará la información a la página.

Lámina 51 Roberto Gómez C.

Formularios en HTML

- Permiten desde una presentación web, solicitar información.
- Están compuestos por tantos campos como información se vaya a capturar.
- Una vez capturados se envían a otro lugar para su procesamiento.
- La declaración se hace usando las etiquetas:

```
<form>
   ..........
   ..........
   ..........
</form>
```

Lámina 52 Roberto Gómez C.

Atributos formularios

- `action=""`
  - Entre comillas se indica el programa que va a tratar las variables enviadas con el formulario, puede tratarse de un CGI, un script PHP o la URL mailto.
- `method=""`
  - Indica el método de transferencia de las variables.
    - Post, si se envía a través del STDIO.
    - Get, si se envía a través de la URL.
- Ejemplo

```
<form  action="procesa.php"   method="post" >
          ::::::::::::
          ::::::::::::
</form>
```

Lámina 53 Roberto Gómez C.

Botones

- Utilizados para enviar la información introducida o limpiar los campos donde se va a introducir la información.
- Se definen mediante la etiqueta <input> a la que le acompañan los atributos:
  - `type=""`
    - Entre comillas pueden ir los valores de:
      - submit para enviar los datos del formulario
      - reset para borrar los datos que se han introducido.
  - `value=""`
    - Indica el texto que incorporaran los botones.
    - Normalmente, enviar y borrar.

Lámina 54 Roberto Gómez C.

Ejemplos botones

```
<form action="procesa.php" method="post" >
          ::::::::::::
<INPUT TYPE="submit"><INPUT TYPE="Reset">
</form>
```


```
<form action="procesa.php" method="post" >
          ::::::::::::
<INPUT TYPE="submit" value="Enviar">
<INPUT TYPE="Reset" value="Borrar">
</form>
```

Lámina 55 Roberto Gómez C.

Campos entrada de formularios

- Necesario usar la etiqueta <input>
- Define la introducción de variables.
- Se cuenta con los siguientes atributos:
  - `type=""` :Indica el tipo de variable a introducir
  - `method=""` :Indica el nombre de la "variable" donde se va a almacenar la información introducida

Lámina 56 Roberto Gómez C.

Variable de tipo texto (text)

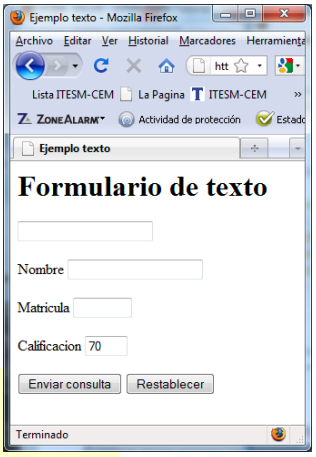
- Si el atributo type toma el valor de text

  `<input type="text"……`

  – Indica que el campo a introducir será un texto

  – Sus atributos:

  - maxlenght=“ ”: limitará el número máximo de caracteres a introducir
  - size=“ ” :limita el número de caracteres a mostrar en pantalla
  - value=“ ” :indica que no hay valor inicial del campo.



- Ejemplo:

```
<form  action=“procesa.php"   method=“post“ >

<input type="text" name="algo"><br>
Nombre <input type="text" name ="nombre"> <br>
Matricula <input type="text" name="mat" size="6"><br>
Calificacion <input type="text" name="cal" size="3" value="70">
<br>

<input type="submit"><input type="Reset">
</form>
```


Password y Checkbox

- Si el atributo type toma el valor de password

  `<input type=“password“……`

  – Indica que el campo recibirá una contraseña.

  – Se desplegarán asteriscos en lugar de caracteres.

  – Sus atributos serán los mismos que para text.

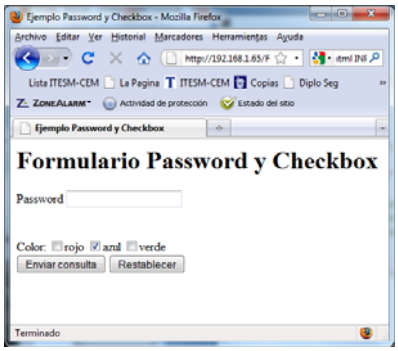
- Si el atributo type toma el valor de checkbox

  `<input type=“checkbox“……`

  – El campo se elegirá marcando una casilla cuadrada de entre varias opciones

  – Atributos:

  - value=“ ” entre comillas se indicará el valor de la casilla
  - checked la casilla aparecerá marcada por defecto



- Ejemplo:

```
<form  action="procesa.php"   method="post" >
Password <input type="password" name ="passwd"> <br>
Color:  <input type="checkbox" name="color" value="rojo">rojo
        <input type="checkbox" name="color" value="azul" checked>azul
        <input type="checkbox" name="color" value="verde">verde  <br>
<input type="submit"><input type="Reset">
</form>
```

Radio

- Si el atributo type toma el valor de radio

  <input type="radio"……

  – El campo se elegirá marcando una casilla circular de entre varias opciones

  – Atributos:

  - value=" " entre comillas se indicará el valor de la casilla
  - checked la casilla aparecerá marcada por defecto

- Ejemplo:


```
<form  action="procesa.php"   method="post" >
Color:  <input type="radio" name="color" value="rojo">rojo
          <input type="radio" name="color" value="azul" checked>azul
          <input type="radio" name="color" value="verde">verde  <br>
<input type="submit"><input type="Reset">
</form>
```

Lámina 59 Roberto Gómez C.

Image

- Si el atributo type toma el valor de image

  <input type="image"……

  – Crea una imagen que es un botón para enviar la información capturada.

  – La mayor parte de los atributos asociados con <img …> son válidos

  – Atributos:

  - alt=" " texto desplegado en caso figura no se encuentre
  - Src=" "ubicación imagen a usar

- Ejemplo:


```
<form  action="procesa.php"   method="post" >
Nombre: <input name="nombre" type="text">
<input  type=image
    src="./america.jpg"
    alt="PULSE AQUI"
    height=55 width=80
    align=absmiddle>
<input type="submit"><input type="Reset">
</form>
```

Lámina 60 Roberto Gómez C.

Hidden

- Si el atributo type toma el valor de hidden

  <input type="hidden"……

  – El usuario no puede modificar su valor ya que no está visible.

  – Se manda siempre junto al atributo value= seguido de su valor entre comillas.

- Ejemplo:

```
<form  action="procesa.php"   method="post" >
<input type="hidden" name="nombre"
                                value="Pablo">
<br>
<input type="submit" value="Enviar dato">
</form>
```

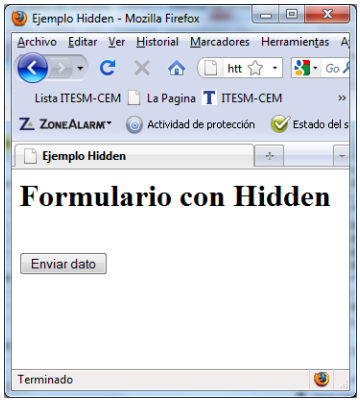


Lámina 61 Roberto Gómez C.

Ejemplo varias opciones

```
<H1>Formularios</H1>

<FORM ACTION="proceso.php" METHOD="POST">

Texto: <INPUT TYPE="text" NAME="nombre"><BR>

Password: <INPUT TYPE="password" NAME="contra"><BR>

Sexo:<INPUT TYPE="radio"  NAME="boton1"  VALUE="1"> Hombre
      <INPUT TYPE="radio" NAME="boton1" VALUE="2">Mujer<BR>

Vehiculo:<INPUT TYPE="checkbox" NAME="Moto" VALUE="Si">Moto
         <INPUT TYPE="checkbox" NAME="Coche" VALUE="" CHECKED>Coche

<BR><BR>
<INPUT TYPE="submit"><INPUT TYPE="Reset">
</FORM>
```

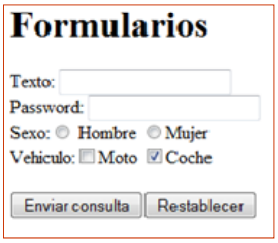


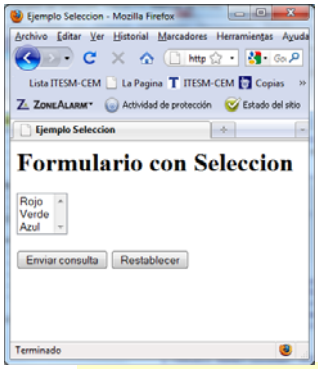
Lámina 62 Roberto Gómez C.

Campos de selección

- Uso de etiquetas <select> … </select>
- Encierran valores que se pueden elegir a partir de una lista predeterminada.
- Los atributos que acompañan a la etiqueta de apertura son:
  - Name=“” :Indica el nombre del campo de selección
  - Size=“”:Indicará el número de opciones visibles.
    - Si le asignamos 1, la selección se presentará como un menú desplegable.
    - Si le asignamos un valor mayor se presentará como una lista con barra de desplazamiento.
  - Multiple: Indica si se pueden llevar a cabo múltiples selecciones
- Las diferentes opciones de la lista se indican mediante la etiqueta <option> que puede acompañarse del atributo selected para indicar cual es la opción que aparecerá por defecto.
  - Si no se especifica, siempre será la primera de la lista.

Lámina 63 Roberto Gómez C.

Primer ejemplo seleccion




```
<form  action="procesa.php"   method="post" >
<select name=Colores" multiple>
  <option value="r">Rojo </option>
  <option value="v">Verde </option>
  <option value="a">Azul </option>
</select>
<br>
<input type="submit"><input type="Reset">
</form>
```

Lámina 64 Roberto Gómez C.

Segundo ejemplo seleccion


```
<form  action="procesa.php"   method="post" >
<select name=Colores" size="1">
  <option value="r">Rojo </option>
  <option value="v">Verde </option>
  <option value="a">Azul </option>
</select>
<br>
<input type="submit"><input type="Reset">
</form>
```

Lámina 65 Roberto Gómez C.

Areas de texto

- Con las tags <textarea>;.....</textarea> definimos un texto de múltiples líneas para que el visitante pueda incluir un comentario junto a sus datos.
- Junto a la tag de apertura pueden aparecer los siguientes atributos:
  - name="" Nombre del campo.
  - Cols="" Numero de columnas de texto visible.
  - Rows="" Numero de filas de texto visible.

Lámina 66 Roberto Gómez C.

Ejemplo áreas texto

```
<form  action="procesa.php"   method="post" >

<textarea cols=20 rows=10 name="Texto">
</textarea>
<br> <br>

<input type="submit"><input type="Reset">
</form>
```

Lámina 67

Roberto Gómez C.

Ejemplo envío datos

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>
<H1>Ejemplo de procesado de formularios</H1>
Introduzca su nombre:
<FORM ACTION="procesa.php" METHOD="GET">
<INPUT TYPE="text" NAME="nombre"><BR>
<INPUT TYPE="submit" VALUE="Enviar">
</FORM>
</body>
</html>
```

Al pulsar el botón enviar el contenido del cuadro de texto es enviado a la página indicada en el atributo ACTION de la etiqueta form.

Lámina 68

Roberto Gómez C.

Ejemplo recepción datos: *procesa.php*

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>
<H1>Ejemplo de procesado de formularios</H1>
El nombre que ha introducido es: <?php echo $_GET['nombre']
?>
<br>
</body>
</html>
```

Se ha creado una entrada en el arreglo $_GET[] con el índice 'nombre' y con el valor que haya introducido en el browser.

Lámina 69 Roberto Gómez C.

Métodos GET y POST

- Los datos de un formulario se envían mediante el método indicado en el atributo METHOD de la etiqueta FORM, los dos métodos posibles son GET y POST.
- La diferencia entre estos dos métodos radica en la forma de enviar los datos a la página
  - el método GET envía los datos usando la URL,
  - el método POST los envía por la entrada estándar STDIO.

Lámina 70 Roberto Gómez C.

Ejemplo método GET

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>
<H1>Ejemplo de procesado de formularios</H1>

<FORM ACTION="procesa2.php" METHOD="GET">
Introduzca su nombre:<INPUT TYPE="text" NAME="nombre"><BR>
Introduzca sus apellidos:<INPUT TYPE="text" NAME="apellidos"><BR>
<INPUT TYPE="submit" VALUE="Enviar">
</FORM>
</body>
</html>
```

Lámina 71 Roberto Gómez C.

Ejemplo método POST

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>
<H1>Ejemplo de procesado de formularios</H1>
<FORM ACTION="procesa2.php" METHOD="POST">
Introduzca su nombre:<INPUT TYPE="text" NAME="nombre"><BR>
Introduzca sus apellidos:<INPUT TYPE="text" NAME="apellidos"><BR>
<INPUT TYPE="submit" VALUE="Enviar">
</FORM>
</body>
</html>
```

Lámina 72 Roberto Gómez C.

El archivo procesa2.php

```
<html>
<head>
  <title>Ejemplo de PHP</title>
</head>
<body>
<H1>Ejemplo de procesado de formularios</H1>
El nombre que ha introducido por GET es: <?php echo $_GET['nombre']," ",$_GET['apellidos'] ?><br>
El nombre que ha introducido por POST es: <?php echo $_POST['nombre']," ",$_POST['apellidos'] ?>
<br>
</body>
</html>
```

El resultado final es el mismo, solo que con el método GET podemos ver los parámetros pasados ya que están codificados en la URL.

Lámina 73 Roberto Gómez C.

MySQL

- Servidor de base de datos Open Source
  - Corre en diferentes plataformas (Unix y Windows)
- Servidor en red
  - No hay GUIs bonitos como MS
  - Se pueden encontrar clientes que proporcionen un GUI.
    - Por ejemplo: phpMyAdmin, y MyCC
- Buena base de datos para aplicaciones de pequeño y mediano tamaño.

Lámina 74 Roberto Gómez C.

Primeros pasos

- Hay que “levantar” el servidor

```
service mysql start;
```

- Comando para entrar a modo monitor

```
mysql –u xxxxx
```

- Para ejecutar un script SQL

```
mysql –u xxxxx < script.sql
```

Lámina 75 Roberto Gómez C.

Creando Tablas Base datos

- Creando base datos toto que mantendra la

```
create database toto;
```

- Accediendo a la base de datos toto

```
use  toto;
```

- Creación de una tabla denominada alumnos

```
create table alumnos ( nombre char(20), matricula char(20),
                       calificacion integer);
```

- Listado de todas las bases de datos

```
show databases;
```

- Descripción de una tabla

```
describe alumnos;
```

- Listado de todas las tablas de la base datos activo

```
show tables;
```

Lámina 76 Roberto Gómez C.

Los queries básicos de MySQL

- CREATE
  - Crear bases de datos y tablas.
- SELECT
  - Seleccionar renglones de tablas basadas en ciertas condiciones.
- DELETE
  - Borrar uno o más renglones de una tabla.
- INSERT
  - Insertar un nuevo renglón en una tabla.
- UPDATE
  - Actualizar renglones en una tabla.
- ALTER
  - Alterar la estructura de la tabla.

Lámina 77 Roberto Gómez C.

Query INSERT

- Sintaxis

```
INSERT [INTO] nombre_tabla [(nombre_columna,...)]
VALUES ((expresión | DEFAULT),...), (...),...
INSERT [INTO] nombre_tabla
SET nombre_columna=(expresión | DEFAULT), ...
```

- Ejemplos

```
insert into alumnos values ('Juan','456712',88);

insert into alumnos (nombre, matricula) values ('Pedro','112233');
```

Lámina 78 Roberto Gómez C.

Query SELECT

- Sintaxis

```
SELECT expresión FROM tabla
[WHERE condición]
[ORDER BY {unsigned_integer | col_name | formula} [ASC | DESC] ,...]
[LIMIT [offset,] row_count | row_count OFFSET offset]
```

- Ejemplo

```
select  * from alumnos;

select nombre, matricula from alumnos where calificacion < 70;
```

Lámina 79 Roberto Gómez C.

Query UPDATE

- Sintaxis

```
UPDATE nombre_tabla
SET nombre_columna1=expr1 [, nombre_columna2=expr2 ...]
[WHERE condición]
[ORDER BY ...]
[LIMIT row_count]
```

- Ejemplos

```
update alumnos set calificacion = 95 where nombre = 'Beto';

update alumnos set calificacion = 100 where calificacion > 94;

update alumnos set calificacion = 70 ;
```

Lámina 80 Roberto Gómez C.

Query DELETE

- Sintaxis

```
DELETE FROM nombre_tabla
[WHERE condición]
[ORDER BY ...]
[LIMIT row_count
```

- Ejemplos

```
delete from alumnos where nombre = 'Eve';
```

Lámina 81 Roberto Gómez C.

Accediendo a la bases datos MySQL desde PHP

- Los pasos para acceder desde PHP a una base de datos son los siguientes
  1. Conectar con el servidor de bases de datos.
  2. Seleccionar una base de datos.
  3. Enviar la instrucción SQL a la base de datos.
  4. Obtener u procesar los resultados.
  5. Cerrar la conexión con el servidor de bases de datos.

Lámina 82 Roberto Gómez C.

Funciones de acceso a MySQL

- Las funciones concretas de MySQL que realizan estas operaciones son:
  - mysql_connect()
    - Conectar con el servidor de bases de datos:
  - mysql_select_db()
    - Seleccionar una base de datos:
  - mysql_query()
    - Enviar la instrucción SQL a la base de datos:
  - mysql_num_rows() y mysql_fetch_array()
    - Obtener y procesar los resultados:
  - mysql_close()
    - Cerrar la conexión con el servidor de bases de datos:

Lámina 83 Roberto Gómez C.

Funcion mysql_connect()

- Conectar con el servidor de bases de datos.
- Devuelve un identificador de la conexión en caso de éxito y false en caso contrario.
- Sintaxis

```
$conexion = mysql_connect (servidor, username, password);
```

- Ejemplo

```
$conexion = mysql_connect (“localhost”, “cursophp”, “”)
  or die (“No se puede conectar con el servidor”);
$conexion = mysql_connect (“localhost”, “cursophp-ad”, “php.hph”)
  or die (“No se puede conectar con el servidor”);
```

Lámina 84 Roberto Gómez C.

Primer ejemplo

```
<html>
<head>
  <title> Ejemplo PHP y MySQL </title>
</head>
<body>
<H1> Ejemplo de uso de PHP para manejo de MySQL </H1>
<?php
printf("Intento de conexion al servidor localhost <br>");
$conex = mysql_connect("localhost", "root","")
  or die ("No se pudo conectar el servidor localhost ");

printf("Conexión exitosa <br>");
?>
</body>
</html>
```

Lámina 85 Roberto Gómez C.

¿Y si sale un error?

Fatal error: Call to undefined function mysql_connect()

- Verificar que la instalación de PHP ha sido compilada para soportar MySQL.
  - Página de prueba con <?php phpinfo(); exit(); ?>
  - Buscar MySQL en la página desplegada.
- Verificar que el archivo php.ini incluye la extensión de mysql
  - Buscar directiva: extension=mysql.so (linux)
  - Buscar directiva: extension=mysql.dll
  - Revisar la directiva: extension_dir
- Verificar si el paquete php-mysql se encuentra instalado, tecleando en la terminal:

  rpm –qa | grep php
  - Si no esta instalado, buscar el RPM de acuerdo a la versión de PHP que se tenga e instalarlo

Lámina 86 Roberto Gómez C.

Función mysql_select_db()

- Selecciona una base de datos
- Devuelve true en caso de éxito y false en caso contrario.
- Sintaxis

```
mysql_select_db (database);
```

- Ejemplo

```
mysql_select_db (“alumnos”)
  or die (“No se puede seleccionar la base de datos”);
```

Lámina 87 Roberto Gómez C.

Función msql_query()

- Envía un query SQL a la base de datos.
- Devuelve un identificador o true, dependiendo del query, si la instrucción se ejecuta correctamente y false en caso contrario.
- Sintaxis

```
$consulta = mysql_query (instrucción, $conexion);
```

- Ejemplo

```
$consulta = mysql_query (“select * from alumnos”, $conexion)
  or die (“Fallo en la consulta”);
```

Lámina 88 Roberto Gómez C.

Funciones mysql_num_rows() y mysql_fetch_array()

- Obtiene y procesa los resultados
- En el caso de que la instrucción enviada produzca unos resultados, mysql_query() devuelve las filas de la tabla afectadas por la instrucción.
- La función mysql_num_rows devuelve el número de filas afectadas
- Para obtener las distintas filas del resultado se utiliza la función mysql_fetch_array(), que obtiene una fila del resultado en un array asociativo cada vez que se invoca
  Sintaxis

```
$nfilas = mysql_num_rows ($consulta);
$fila = mysql_fetch_array ($consulta);
```

Lámina 89 Roberto Gómez C.

Acceso base datos MySQL

- Ejemplo

alumnos		
Nombre	**Matrícula**	**Calificación**
Beto	365321	90
Alicia	367813	78
Cachafas	364122	50
Paty	367819	89
Jaime	362129	60

- Query

```
select * alumnos where calificacion < 70
```

Lámina 90 Roberto Gómez C.

Acceso base datos MySQL

- Ejemplo

alumnos		
Nombre	**Matrícula**	**Calificación**
Beto	365321	90
Alicia	367813	78
Cachafas	364122	50
Paty	367819	89
Jaime	362129	60

$nfilas = 2

$consulta

- Query

```
$nfilas = mysql_num_rows ($consulta);
$fila = mysql_fetch_array ($consulta);
```

Lámina 91 Roberto Gómez C.

Obteniendo datos de las filas

```
$nfilas = mysql_num_rows ($consulta);
if ($nfilas > 0)
{
  for ($i=0; $i<$nfilas; $i++)
  {
    $fila = mysql_fetch_array ($consulta);
    procesar fila i-ésima de los resultados
  }
}
```

Lámina 92 Roberto Gómez C.

Accediendo a los datos de las filas

- Para acceder a un campo determinado de una fila se cuenta con dos opciones
  - Arreglo asociativo

    $fila[“nombre_del_campo”]
  - Indice del campo

    $fila[i]
- Ejemplo

```
for ($i=0; $i<$nfilas; $i++)
{
  $fila = mysql_fetch_array ($consulta);
  printf( “Nombre: %s <br> “, $fila[“nombre”]);
  printf( “Matricula: %s <br>“ , $fila[“matricula”]);
}
```

Lámina 93 Roberto Gómez C.

Función mysql_free_result()

- Libera memoria utilizada por un manejador de resultados.
- Regresa TRUE en caso de éxito y FALSE en caso de falla.
- Sintaxis

```
mysql_free_result(data)
```

Lámina 94 Roberto Gómez C.

Función msyql_close()

- Cierra la conexión con el servidor de bases de datos.
- Sintaxis

```
mysql_close ($conexion);
```

- Ejemplo

```
mysql_close ($conexion);
```

Lámina 95 Roberto Gómez C.

Ejemplo (1/2)

```
<html> <head>
  <title> Ejemplo PHP y MySQL </title>
</head>
<body>
<H1> Ejemplo de uso de PHP para manejo de MySQL </H1>
<?php

$conex = mysql_connect("localhost", "root","")
     or die ("No se pudo conectar con la base alumnos ");

    mysql_select_db("datos")
      or die ("No se puede seleccionar la base de datos");

 $desp = mysql_query("select nombre, matricula from alumnos where calificacion < 70",
                                                                          $conex)
      or die ("Fallo en la consulta");
 $nfilas = mysql_num_rows($desp);
          printf("Se encontraron %d registros <br>",$nfilas);
?>
```

Lámina 96 Roberto Gómez C.

Ejemplo (2/2)

```
<TABLE BORDER=1  CELLSPACING=1  CELLPADDING=1>
    <TR><TD>   Nombre </TD><TD>  Matricula   </TD> </TR>
  </TABLE>
<?php
        if ( $nfilas > 0) {
          for ($i=0; $i < $nfilas; $i++)
          {
            $fila = mysql_fetch_array($desp);
            printf("<tr><td> %s</td>
                  <td> %s </td>
                  </tr> <br>",$fila["nombre"],$fila["matricula"]);
          }
        }
        mysql_free_result($desp);
        mysql_close($conex);
?>

</body>
</html>
```

Lámina 97 Roberto Gómez C.

PHP

Roberto Gómez Cárdenas
rogomez@itesm.mx
http://homepage.cem.itesm.mx/rogomez

Lámina 98 Roberto Gómez C.