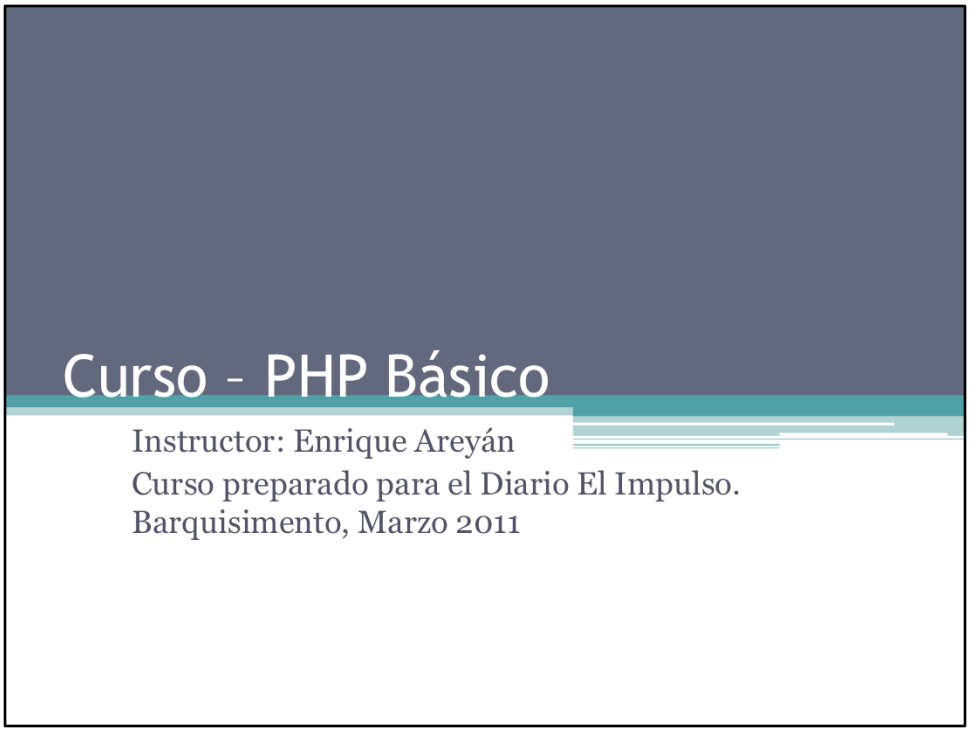


Este curso forma parte de una serie de cursos cuyo objetivo principal es el de enseñar el lenguaje de programación PHP a personas con conocimientos básicos de programación, y prepararlos para llevar a cabo proyectos de gran envergadura. Para ello se enseñará PHP básico (este curso), PHP orientado a Objetos, las mejores y más novedosas practicas en PHP, Patrones de Diseño y el Zend Framework.

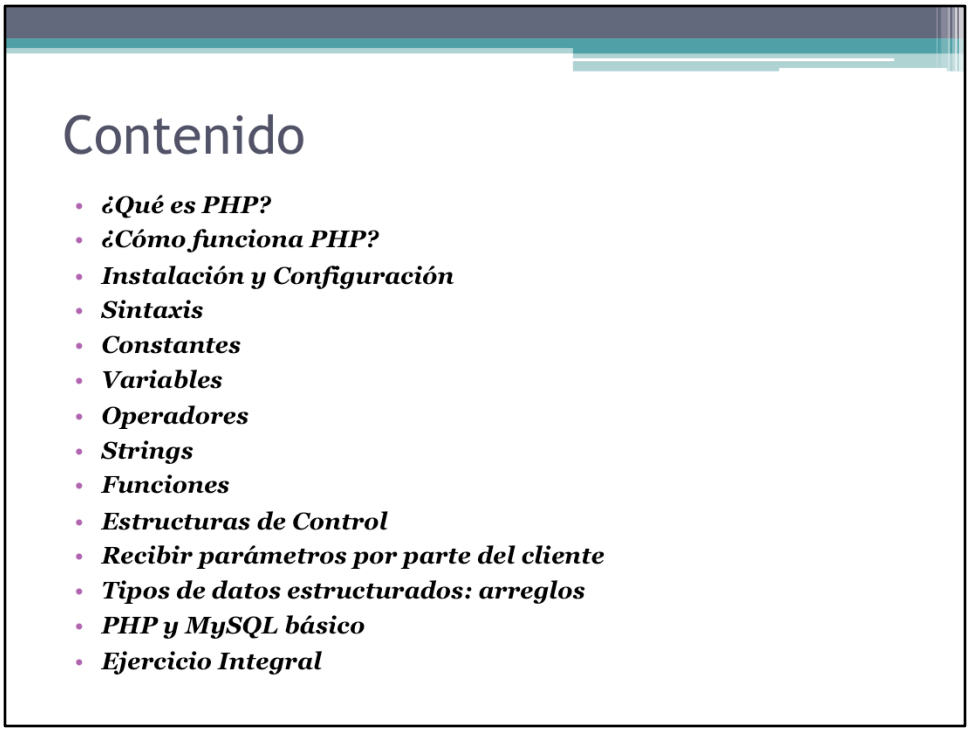
Acerca del Instructor

- Enrique Areyán
- Lic. Computación UCV
- Experiencia de aprox. 10 años (desde 2001) trabajando en aplicaciones web
- Certificado “Building PHP Applications using Zend Framework”
- Co-fundador www.estoesweb.com

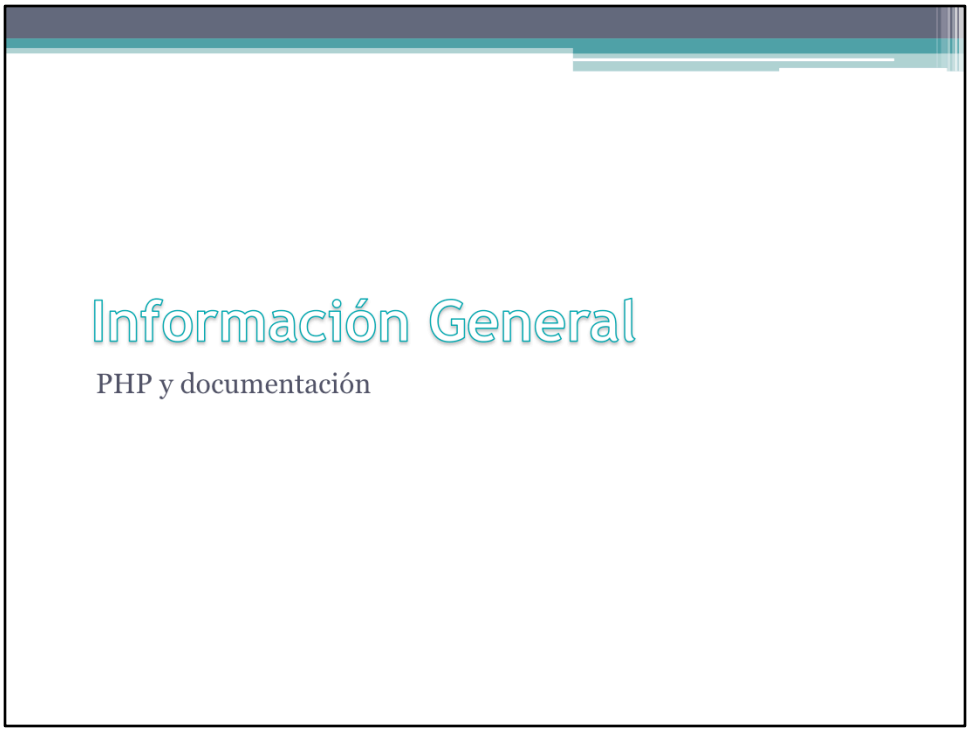
Pueden contactarme a través del correo electrónico enrique3@gmail.com

Antes de Comenzar

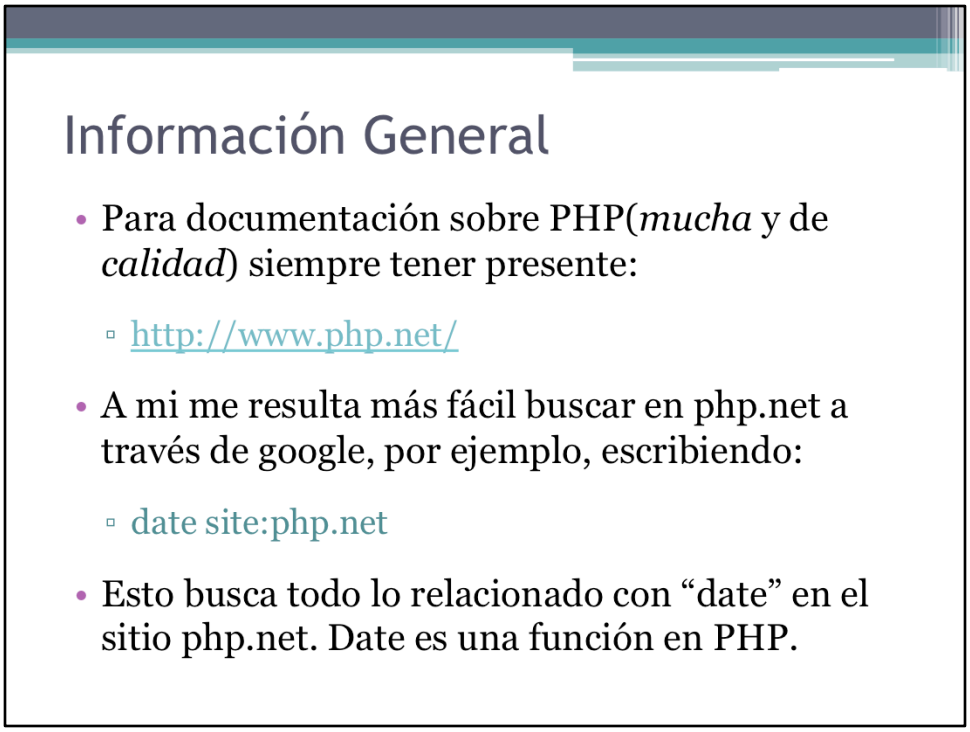
- Bajar (o usar pentdrive) WAMP
- Instalarlo
- Mover las siguientes carpeta del pentdrive a c:\wamp\www
  - Ejemplos
  - Ejercicios
  - Ejercicio Integral



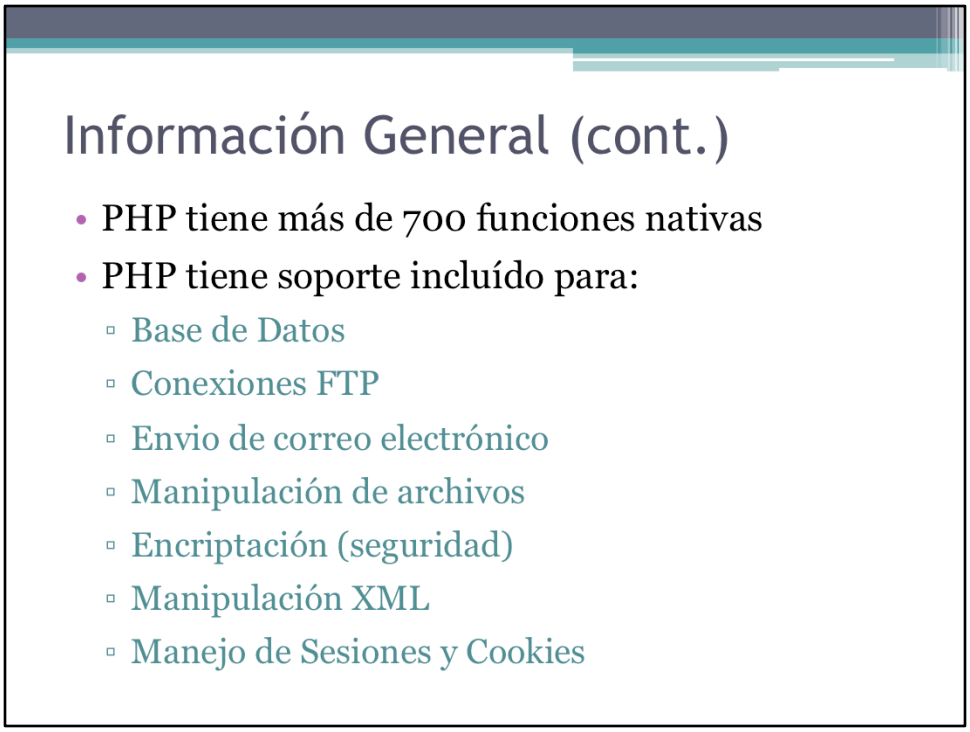
Este curso básico de PHP pretender enseñar las herramientas básicas para comenzar a desarrollar en este lenguaje. Aquí se cubrirán los aspectos básicos más importantes tanto desde el punto de vista teórico como práctico.
Se asume que el participante tiene conocimientos básicos de programación, no necesariamente para la Web.



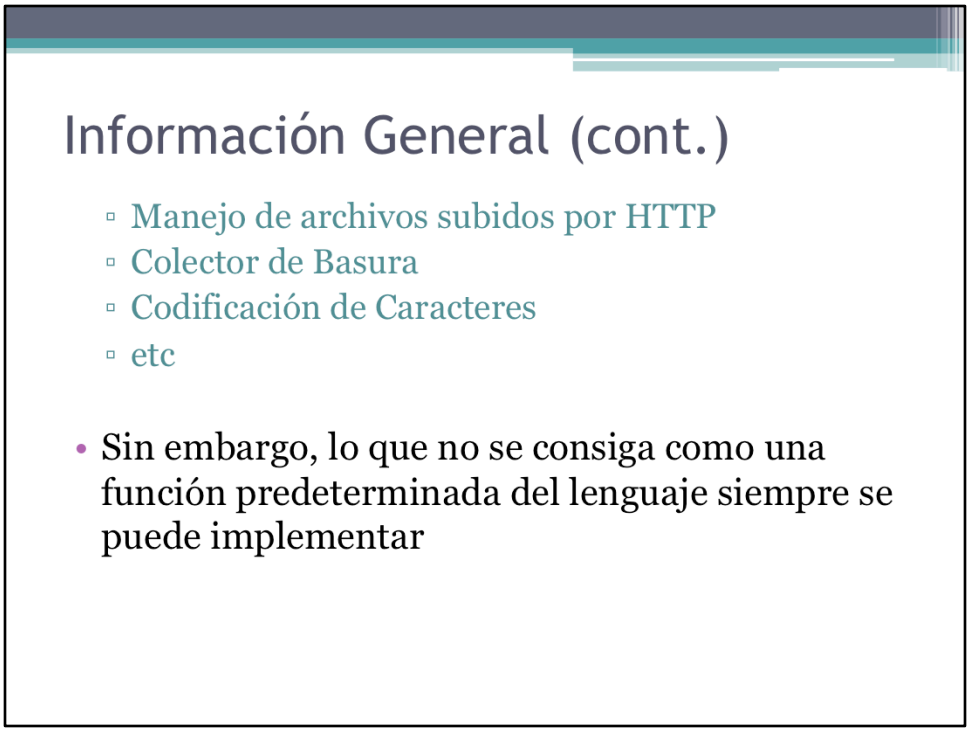
Una revisión general.



La documentación en línea sobre PHP es vasta y de calidad. Si algo no se consigue en el sitio de php, www.php.net, utilizando google se tienen altas probabilidades de solventar esta deficiencia.



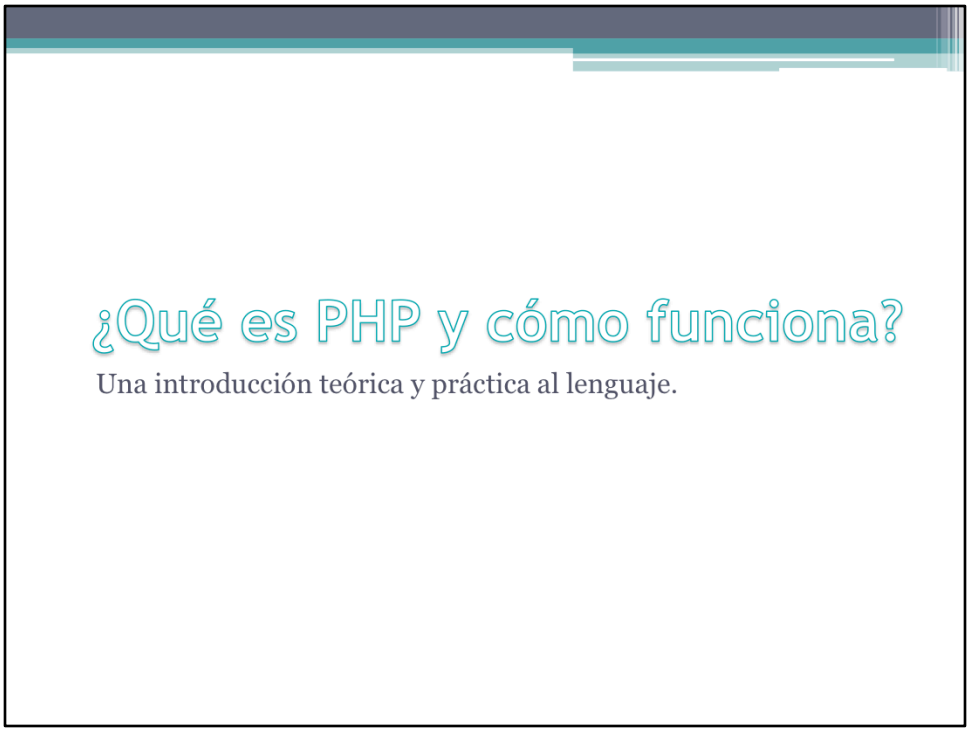
Una función nativa es aquélla que ya viene incluída como parte de las librerías internas del lenguaje y por lo tanto no debe ser implementada por el programador. Esto nos ahorra tiempo.



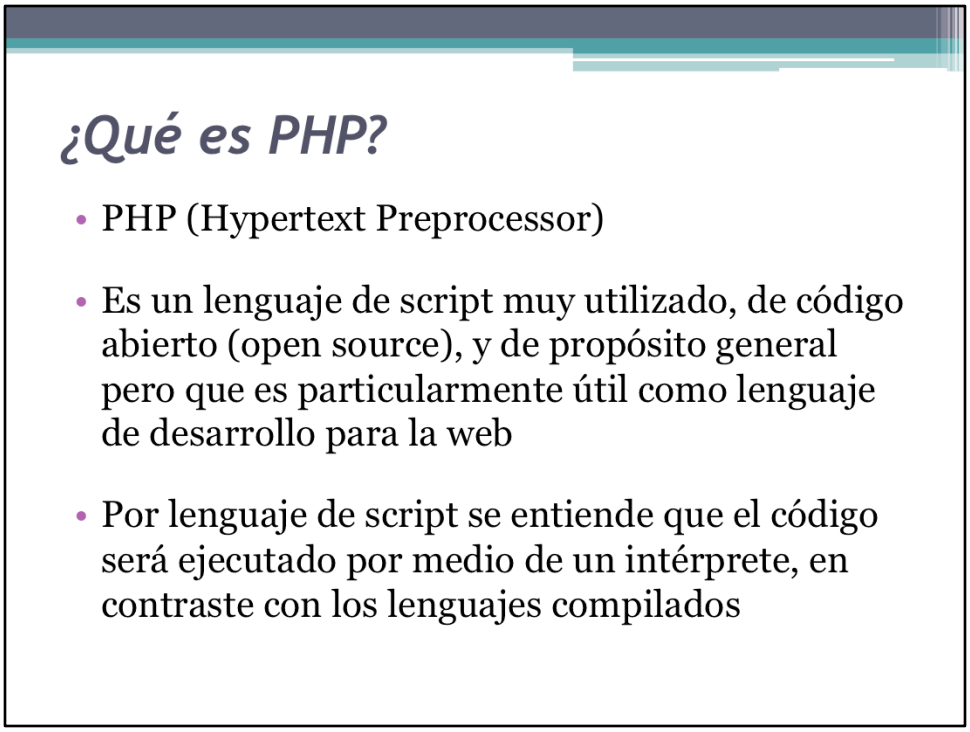
PHP se caracteriza por ser un lenguaje flexible. Esta característica tiene sus ventajas y desventajas, como veremos más adelante.

Glosario

- Usuario
  - Tu, el programador
- Cliente
  - La persona que visita la página web
- Interpretador o intérprete
  - El programa de PHP que se encarga de interpretar el código escrito en este lenguaje



En este apartado se tocará el tema de PHP desde un punto de vista conceptual. También se hablará en concreto cómo es que funciona y sus características principales.



Fuente: http://www.php.net/manual/en/intro-whatis.php

Esto implica que un archivo PHP no es más que un archivo de texto plano alojado en el servidor. El intérprete PHP recibe como entrada ese archivo y lo "interpreta" según sus reglas particulares.

En contraste, un lenguaje compilado se implementa mediante un compilador. Esto implica que una vez escrito el programa, éste se traduce a partir de su código fuente por medio de un compilador en un archivo ejecutable para una determinada plataforma.

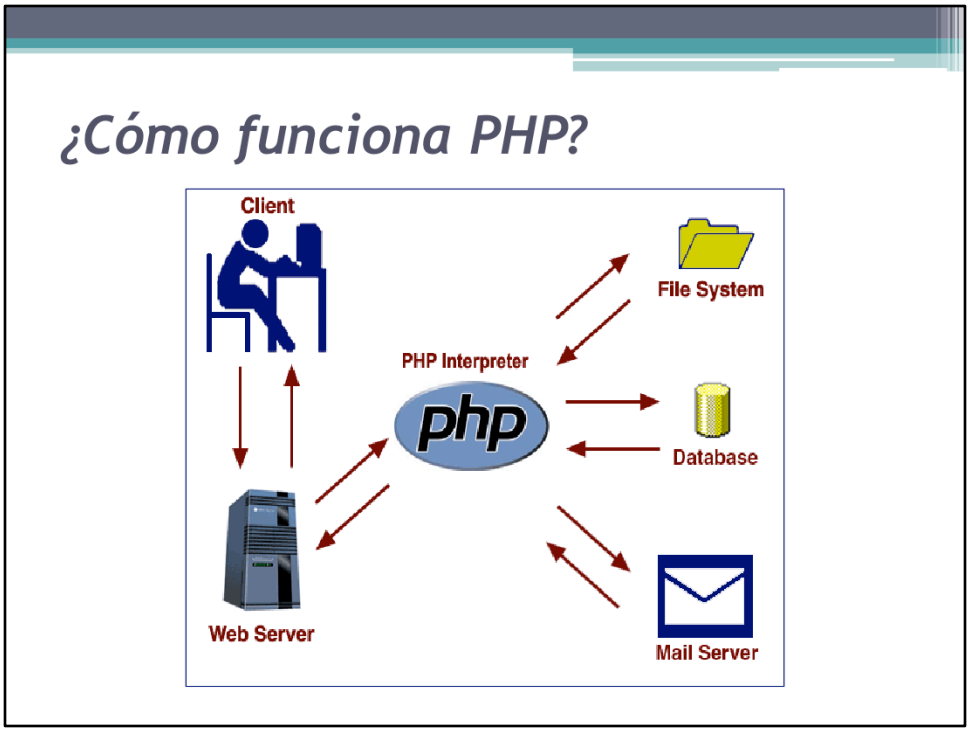
Los lenguajes compilados son lenguajes de alto nivel en los que las instrucciones se traducen del lenguaje utilizado a código máquina para su ejecución. Por el contrario un lenguaje interpretado es aquel en el que las instrucciones se traducen o interpretan una a una.

¿Qué es PHP? (cont.)

- En un principio PHP se utiliza como un lenguaje "encrustado" dentro del HTML
- El objetivo es crear páginas Web dinámicas, es decir, páginas que se generan al momento de la petición por parte del usuario
- En este caso, el encargado de generar el contenido de estas páginas es el interpretador de PHP que se encuentra en el servidor Web

Según Wikipedia: "Las páginas dinámicas que se generan al momento de la visualización se hacen a través de lenguajes interpretados, generalmente JavaScript, y la aplicación encargada de visualizar el contenido es la que debe generarlo. La páginas dinámicas que se generan al ser solicitadas son creadas por una aplicación en el servidor Web que alberga las mismas."

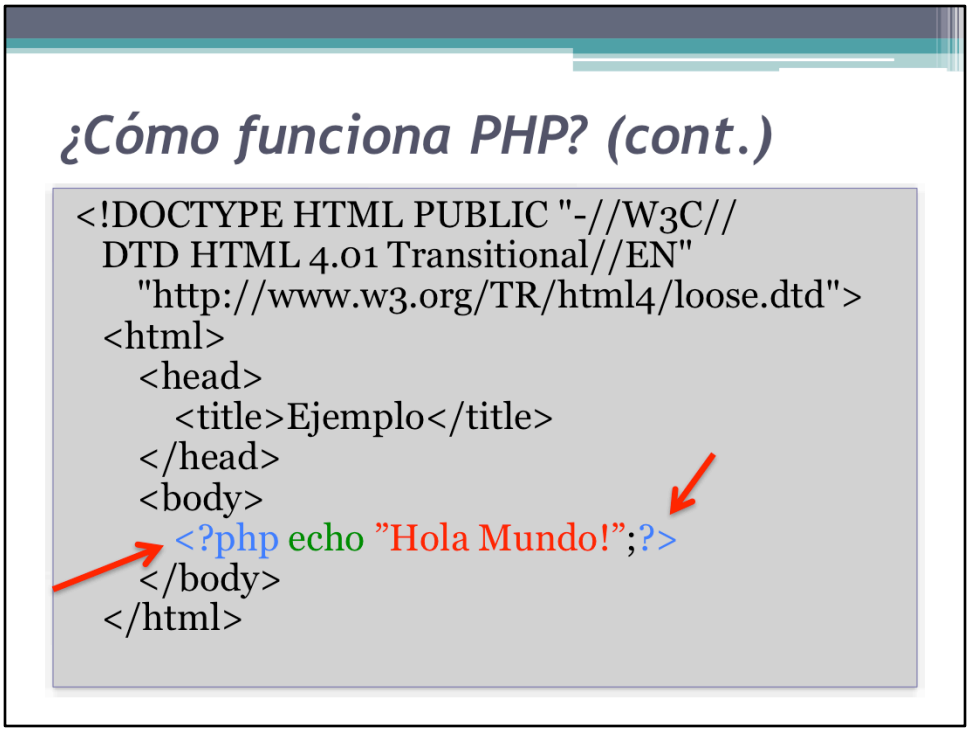
Fuente: http://es.wikipedia.org/wiki/P%C3%A1gina_web



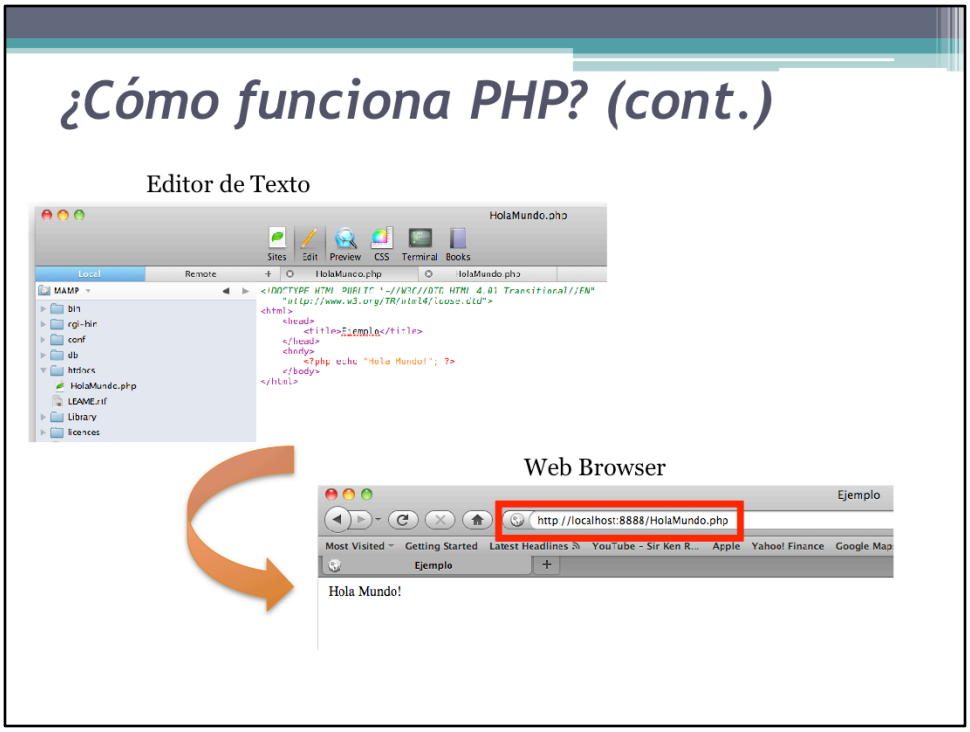
Cuando un usuario navega la Internet a una página que termine en la extensión .php, la petición (request) se envía a un servidor Web el cual redirecciona dicha petición al interpretador de PHP.

Como se muestra en el diagrama, el interpretador de PHP procesa la página que llegó como petición. Para este fin, el intérprete se comunicará con el sistema de archivos, base de datos, servidor de correo o cualquier otro componente del sistema del cual se requiera algún servicio para completar la petición.

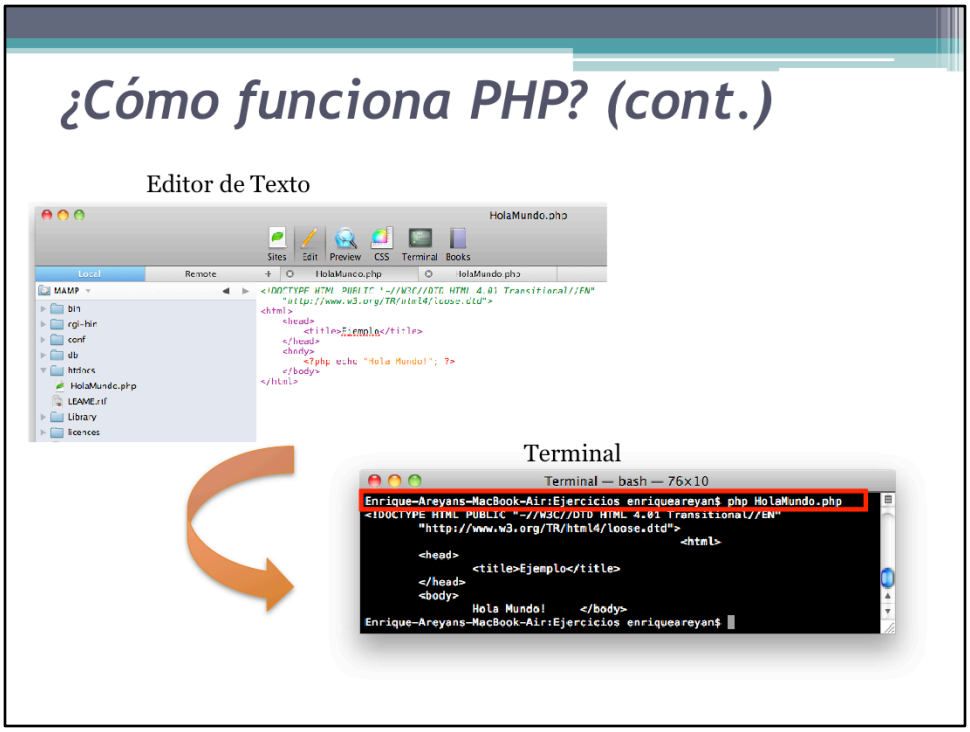
Fuente: http://www.learnphp-tutorial.com/PHPBasics.cfm



La función echo en PHP sirve para imprimir expresiones.

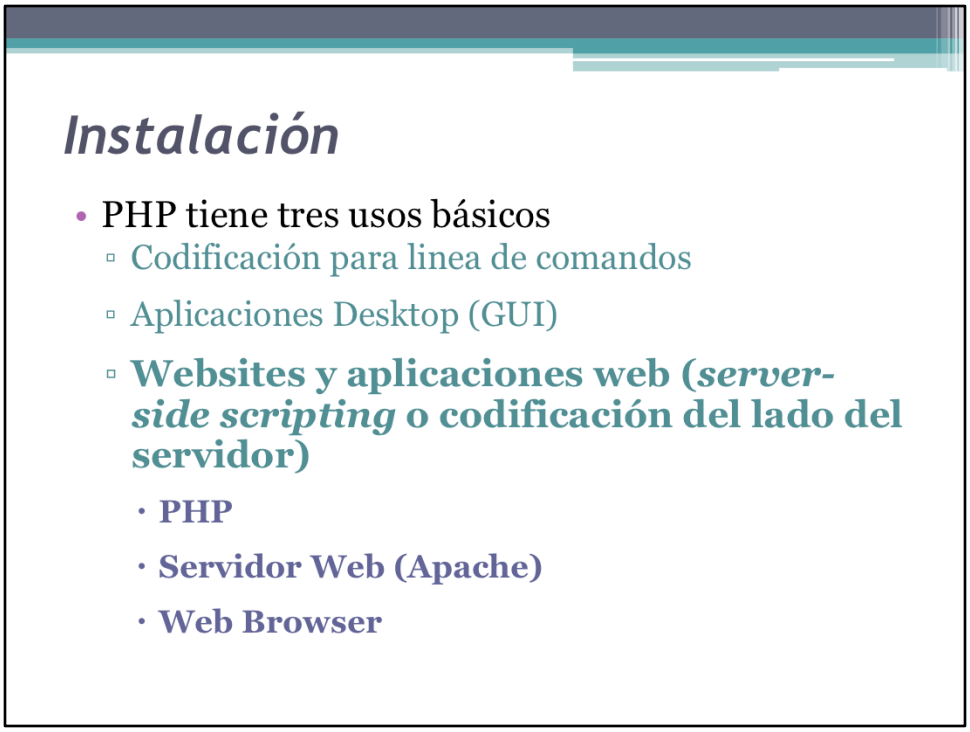


Ejercicio: Subir el archivo de este ejemplo al servidor, correrlo desde el browser y ver el código fuente generado (clic derecho, ver código fuente)



PHP también puede correr desde la línea de comandos o terminal. Esto es útil en ciertas ocasiones como por ejemplo para hacer “debugging”.

Una breve introducción a la instalación y configuración (tunning) del lenguaje.



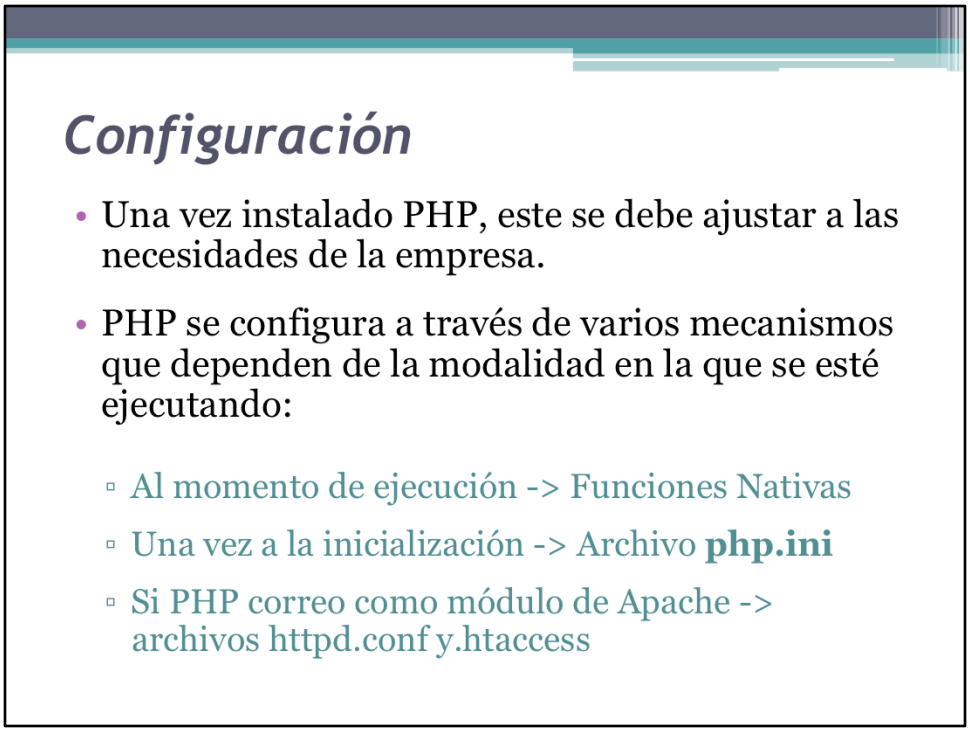
PHP tiene tres usos básicos: (1) para generar scripts a nivel de línea de comandos, (2) para crear aplicaciones desktop (posiblemente PHP no sea la mejor herramienta para esto) y)(3) para páginas y aplicaciones web, que es el uso más común. Aquí nos enfocamos en PHP para la web.

Por tanto, para instalar PHP para la web se necesitan tres cosas: (1) PHP, (2) un servidor web, el cual es típicamente Apache para sistema tipo *nix o IIS para Windows, y (3) un web browser (IE o Fireforx usualmente)

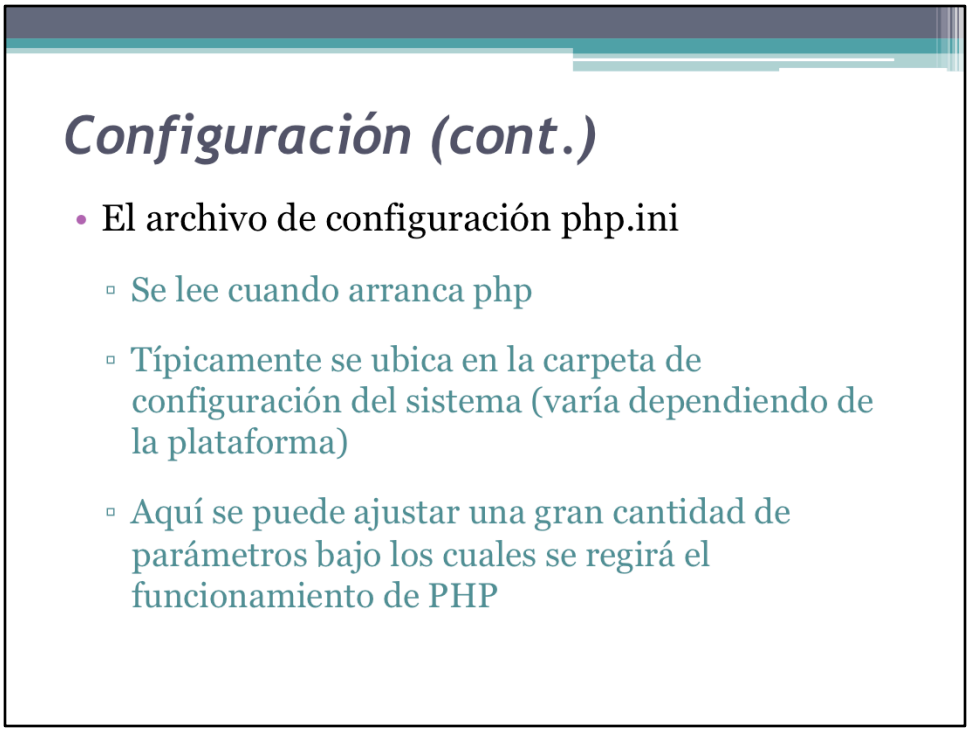
Instalación (cont.)

- PHP viene preinstalado en la mayoría de los servidores hoy en día
- En este curso no vamos a hacer mucho énfasis en la instalación de PHP.
- Sin embargo, se debe mencionar que en la mayoría de los casos PHP se instala como un plugin o módulo al servidor web, quién después se encarga de llamar a PHP para ejecutar una tarea

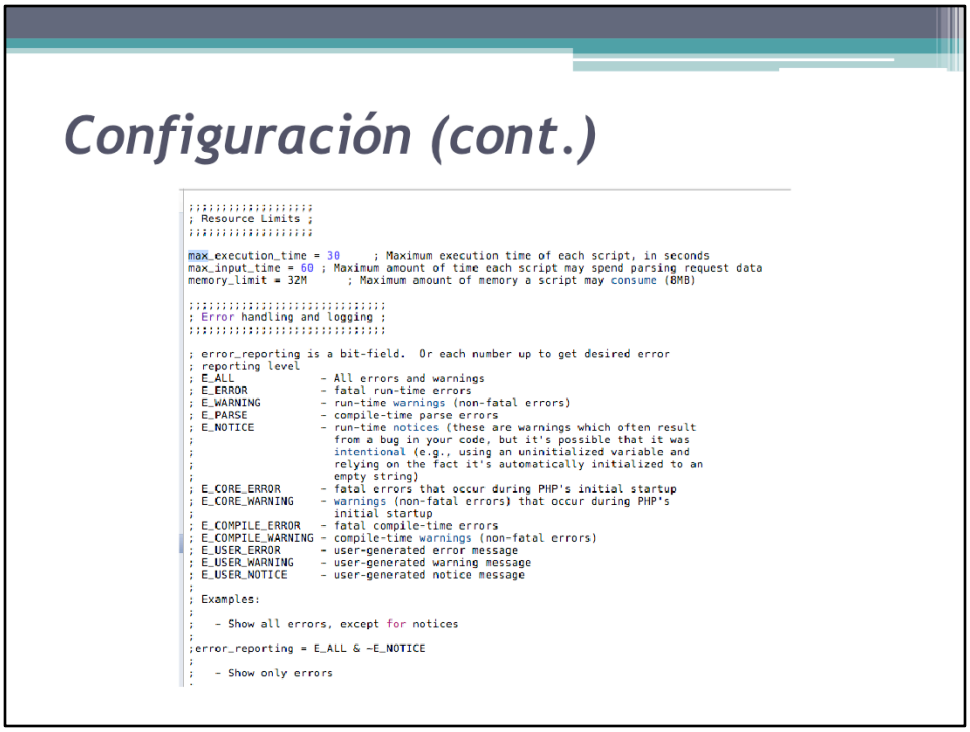
Para más información sobre la instalación de PHP, visite: http://www.php.net/manual/en/install.general.php



La configuración del lenguaje se puede parametrizar durante la ejecución del mismo, por cada script, o globalmente al momento de iniciar PHP.



Cualquier cambio que se quiera que tenga efecto sobre todo el ambiente de desarrollo se debe hacer en el archivo de configuración php.ini
Este archivo es leído una vez al momento de arrancar el sistema y aplica para todos los scripts que vayan a ejecutarse.
El archivo php.ini es muy amplio y otorga una gran flexibilidad al usuario.

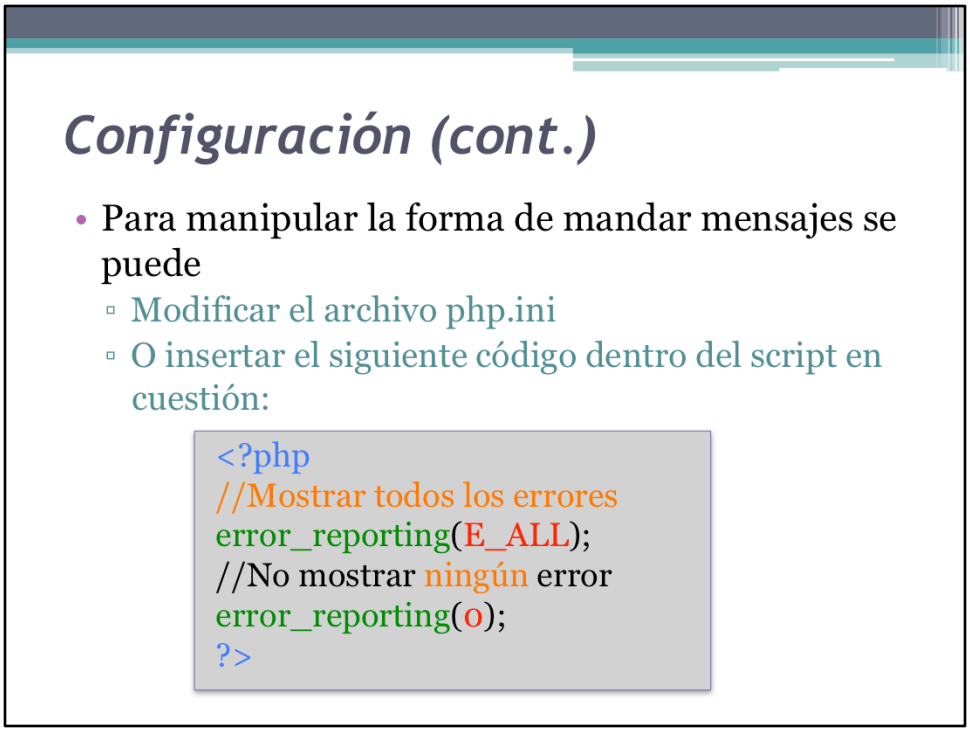


Un ejemplo de una pequeña porción del archivo php.ini. Se muestran varios de los parámetros más importantes del lenguaje.

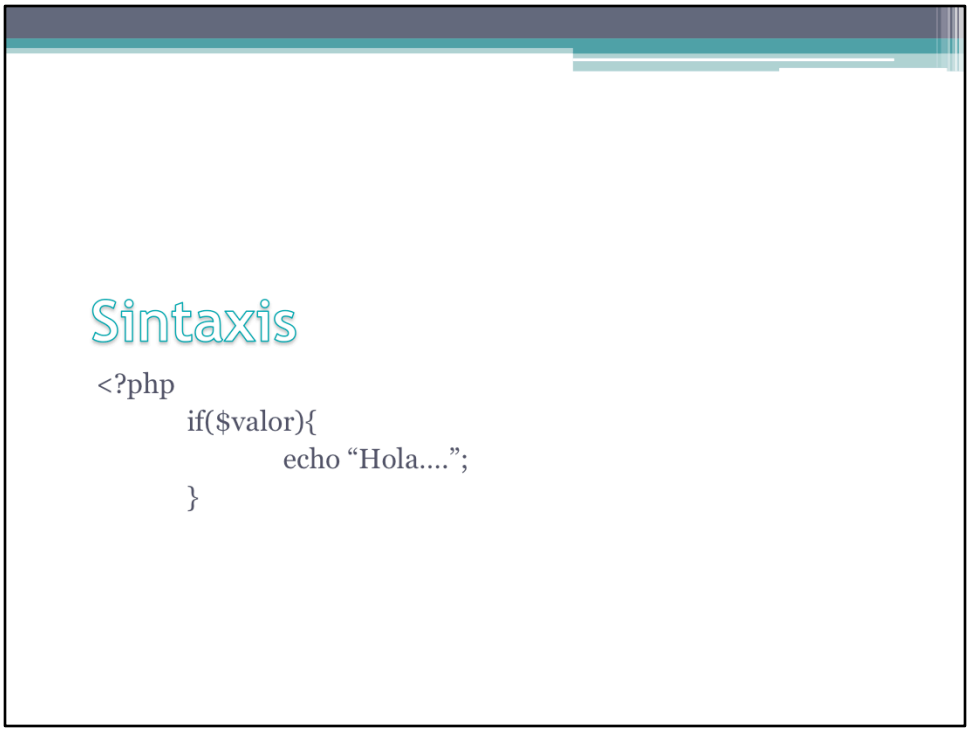
Configuración (cont.)

- Un error en PHP es aquélla condición bajo la cuál el script no puede seguir ejecutándose
- Una notificación en PHP es un "error" más suave, en donde el interpretador genera un mensaje pero la ejecución continúa
  - En el ambiente de desarrollo se desea ver todo (errores o notificaciones)
  - En el ambiente de producción no se desea que el cliente vea ningún mensaje de error. Mejor es guardarlo en un archivo para su posterior "autopsia"

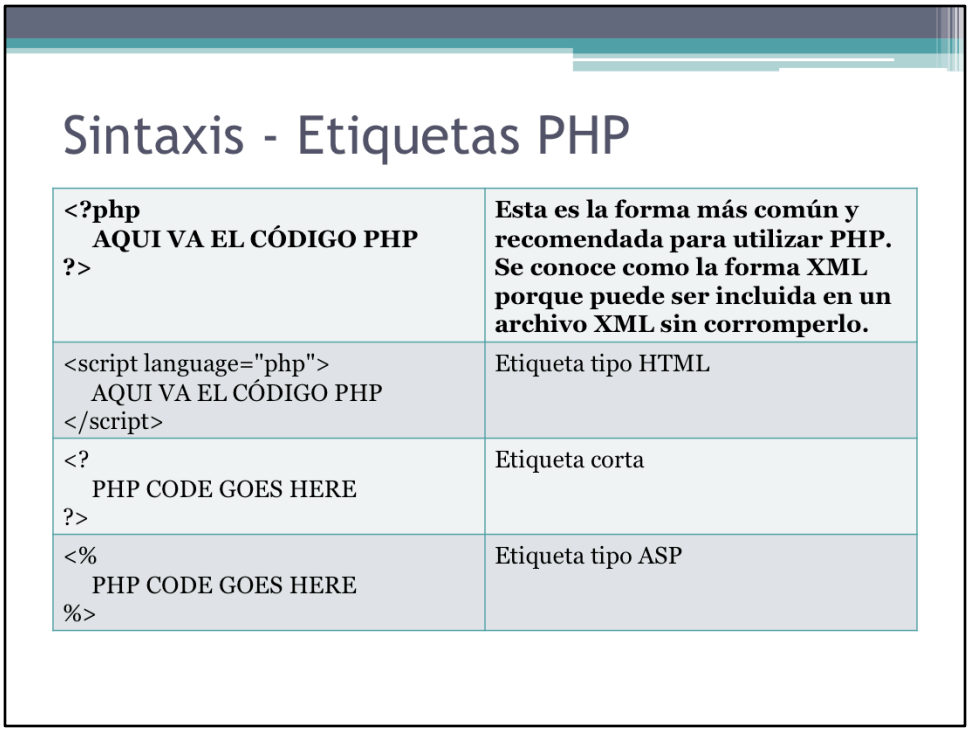
Desde el punto de vista del cliente, cualquier mensaje foráneo a la aplicación se considera como un error y levanta sospechas sobre el funcionamiento de la misma. Muchos mensajes de error puede llevar a que el usuario desconfíe plenamente de la aplicación y deje de utilizarla.



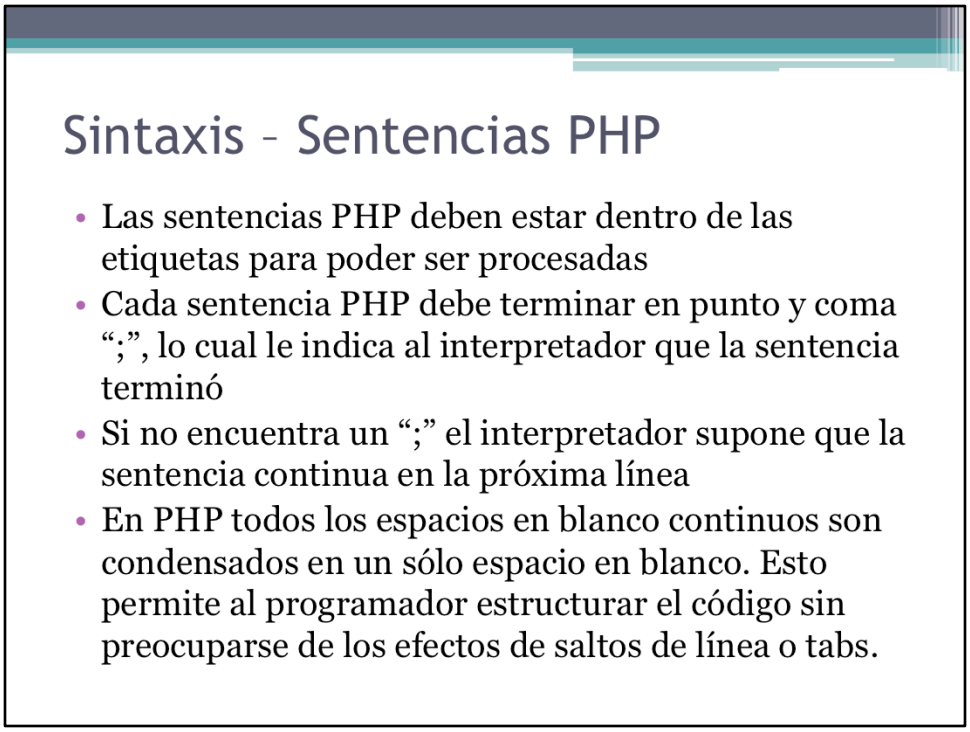
Se muestra como cambiar el comportamiento de PHP al momento de ejecución.



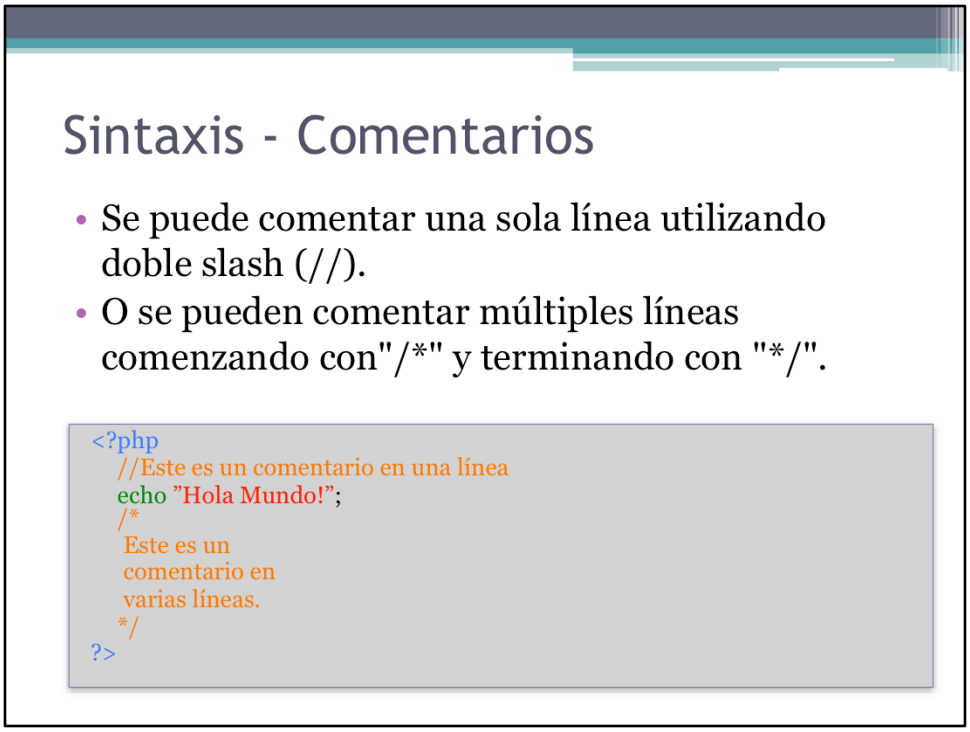
Para manejar cualquier lenguaje hay que empezar por el a,b,c; en este caso, la sintaxis. En general, la sintaxis de un lenguaje (hablado y escrito, o de computador) establece las reglas de cómo se deben formar expresiones válidas en el mismo.



A lo largo de este curso emplearemos la primera forma de las etiquetas para insertar código PHP.



PHP es un lenguaje muy sencillo que presenta pocas restricciones al momento de usarlo.

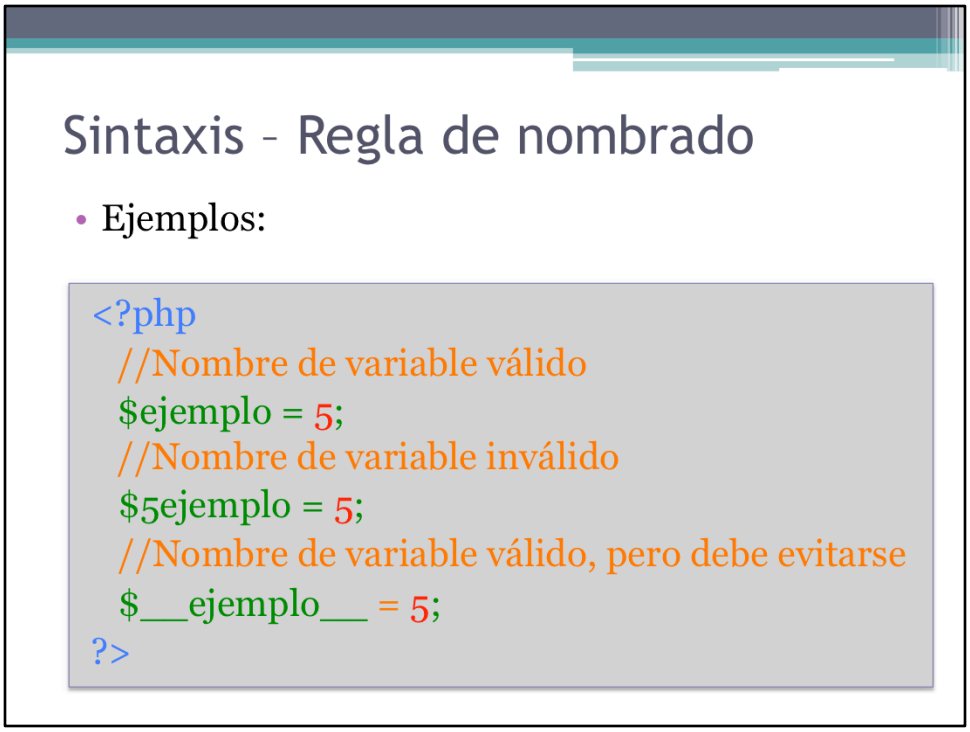


Una de las reglas fundamentales para llevar a cabo proyectos de gran envergadura, es la intradocumentación del código. En esta lámina se muestra como comentar el código del lenguaje PHP.

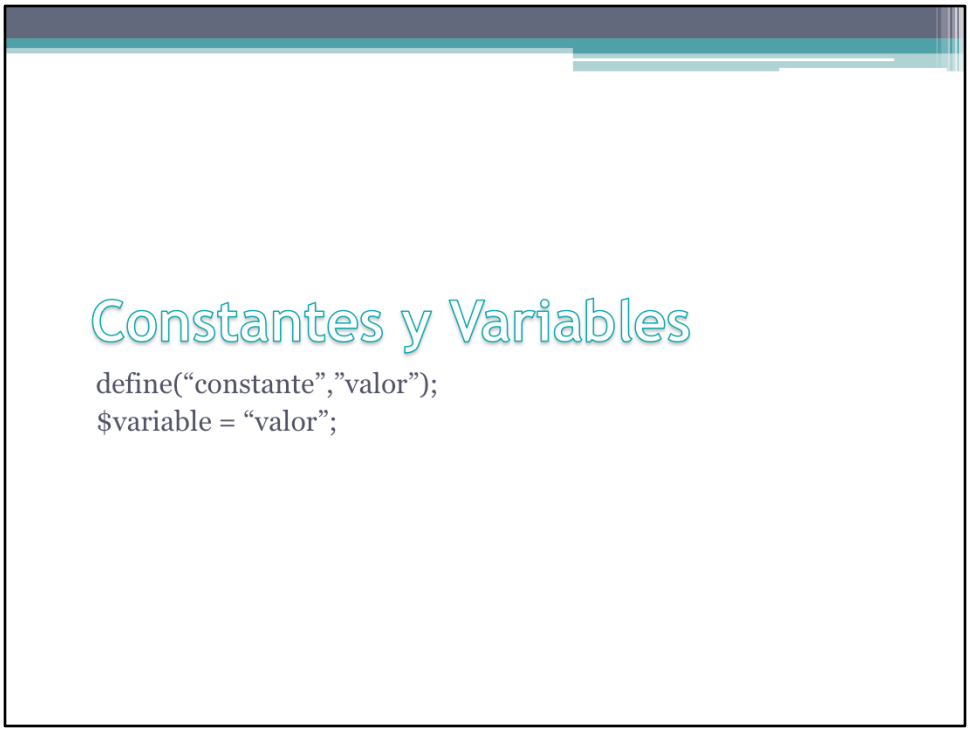
Sintaxis – Regla de nombrado

- Las siguientes reglas aplican para todas aquellas sentencias a las cuales el usuario (programador) necesite darles un nombre. Por ejemplo: constantes, variables, funciones,...
  - El nombre debe comenzar con una letra o "underscore" (_)
  - Seguido de cualquier cantidad de letras, números o "underscores"
  - Visto como expresión regular:
    *[a-zA-Z_\x7f-\xff][a-zA-Z0-9_\x7f-\xff]**

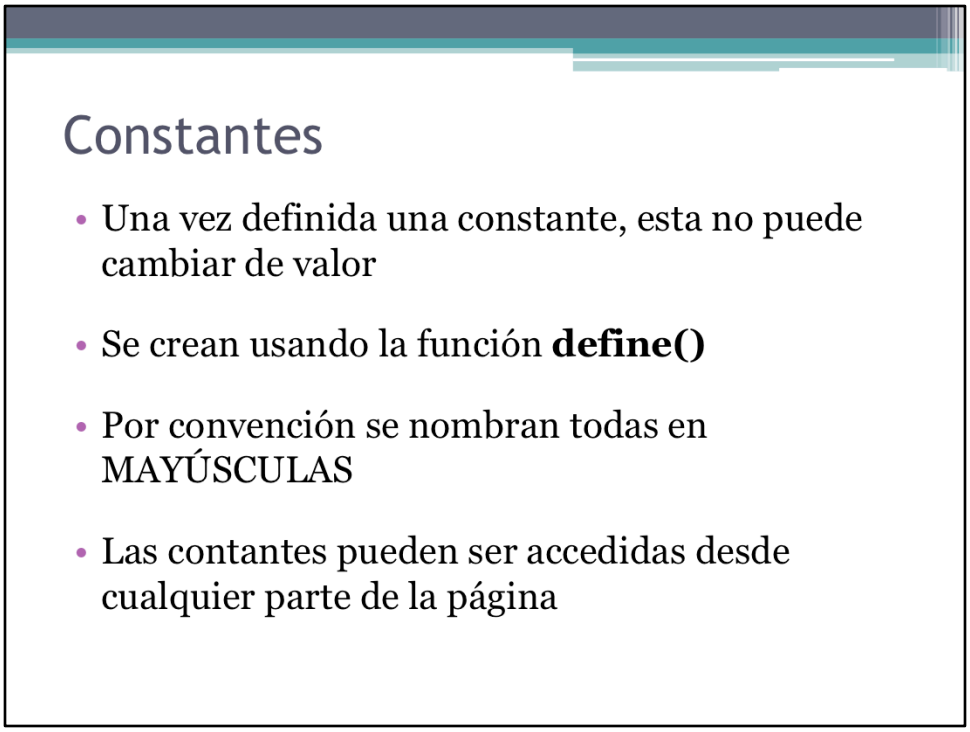
PHP reserva algunas palabras claves para su uso interno. Éstas comienzan con doble underscore. De resto hay plena libertad en el nombrado de objetos.



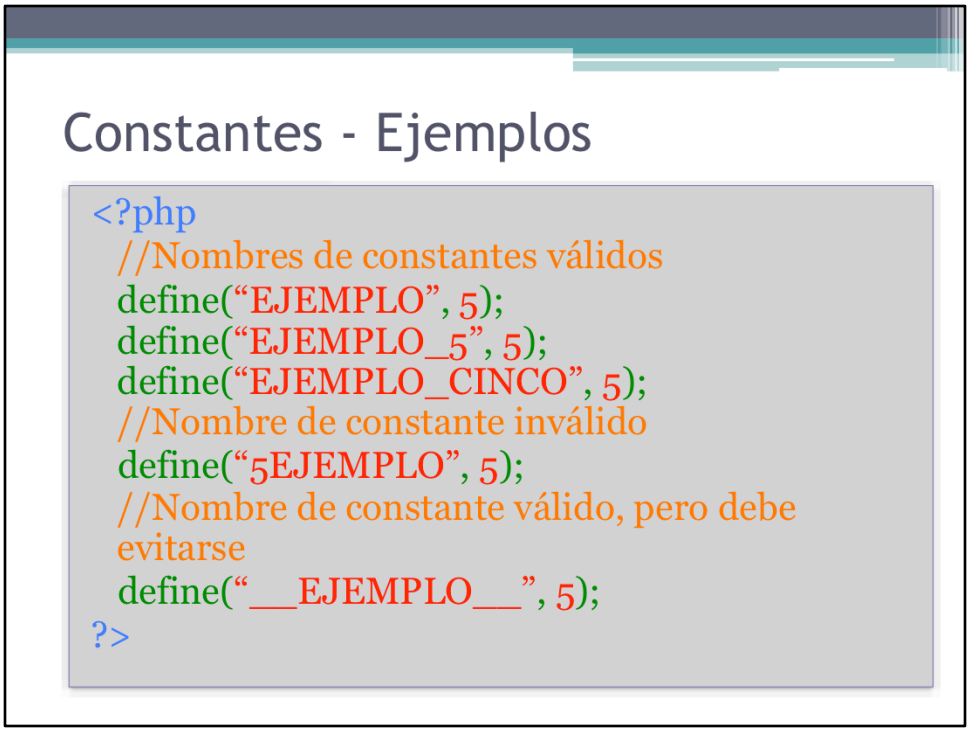
El último nombre de variable es válido y va a ejecutarse en su programa sin inconveniente técnico. No obstante, los nombre que comienzan con "underscores" son nombres utilizados por el equipo de desarrollo de PHP, y si por casualidad coincide el nombre que usted le está dando, por ejemplo, a una variable, con el nombre que ellos en el futuro utilicen para una variable u otro objeto, entonces el de ellos tendrá la prioridad.
Como consecuencia, su script dejará de funcionar o presentará errores que no presentaba con versiones anteriores las cuales no incorporaban de forma nativa el nombre de de dicha variable. En pocas palabras, habrá una colisión entre el nombre del objeto que ellos definen y el suyo.



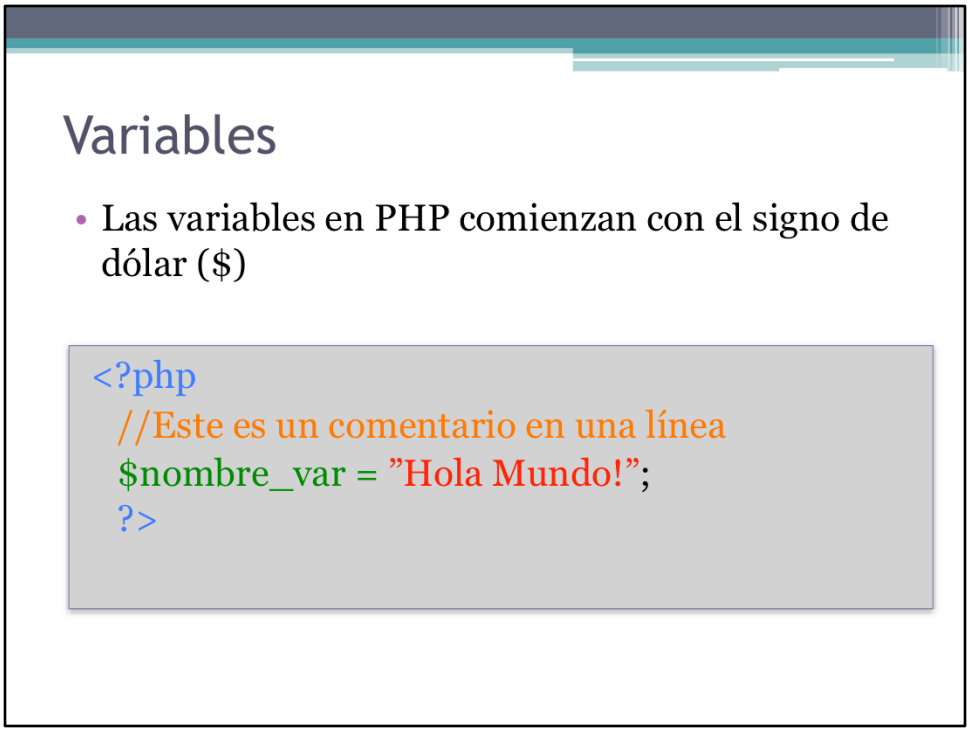
Los bloques fundamentales de cualquier lenguaje de programación son las variables. Sin ellas no se alcanzaría dinamismo y la utilidad de los programas sería escasa. Por su parte, las constates ayudan a establecer reglas uniformes al momento de llevar a cabo proyectos. En este apartado vemos en detalles como funionan constantes y variables en PHP.



Reglas para la creación y lectura de constantes. Importante: el valor de las constantes no puede cambiar en el momento de ejecución de un programa.



El nombre de constante inválido no mostrará error hasta que se use la constante en otra sentencia. Por ejemplo, si deseas imprimir el valor de la constante: echo 5EJEMPLO;, el interpretador de PHP arrojará un error. Esto solo ocurre con constantes, ya que con otro tipo de objetos el error se arroja inmediatamente.

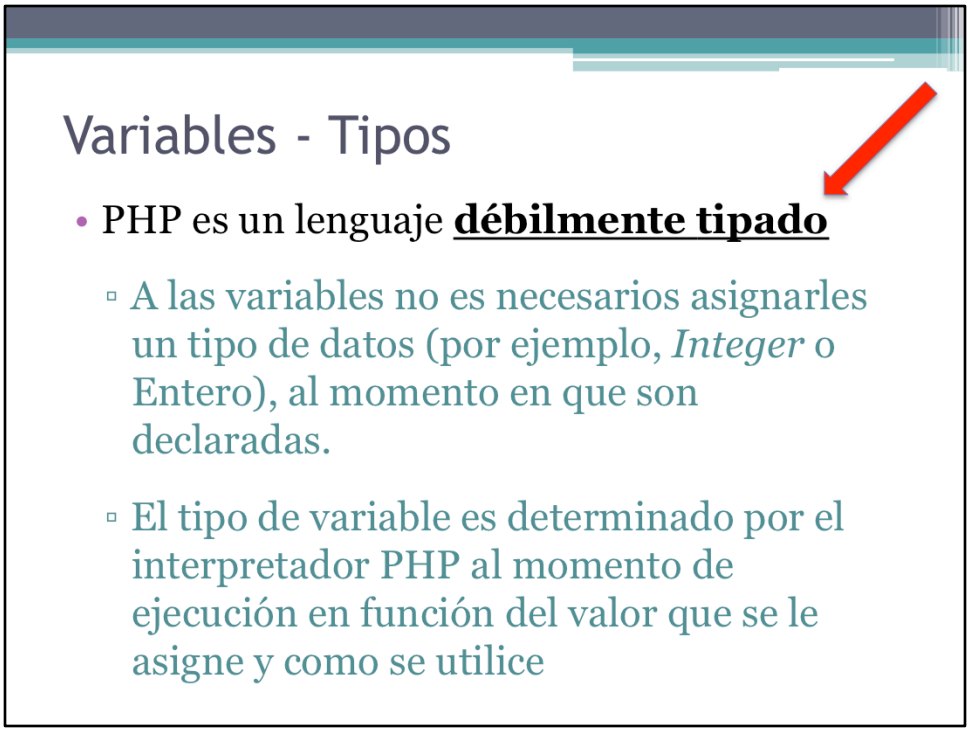


Las variables forman la estructura básica de cualquier lenguaje de programación. En PHP una variable se declara prefijando el símbolo de dólar ($) y siguiendo las reglas de nombrado anteriormente descritas.

Variables - Tipos

Tipo de Variable	Descripción
Integer (Entero)	Números enteros
Double (doble)	Números reales
String (cadena)	Cadena de caracteres (texto)
Boolean (variable lógicas)	True or false (verdadero o falso)
Array (arreglo)	Lista de datos
Objetc (objeto)	Instancia de una clase

Más sobre objetos en el próximo curso.

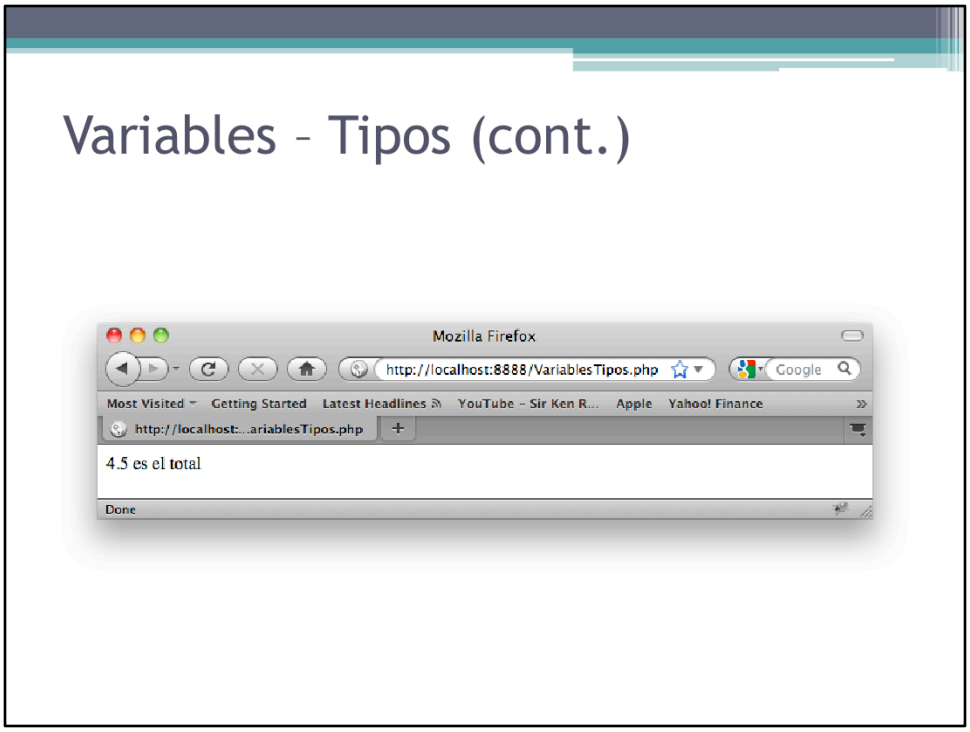


Es próximos lecciones se verá claramente porque el hecho de que PHP sea débilmente tipado es una de las fuentes de errores más comunes al momento de llevar a cabo un proyecto de grandes dimensiones.
Aunque es cierto que esta característica hace de PHP un lenguaje sencillo de aprender y muy flexible, también es cierto que introduce desorden desde el inicio del proyecto. Dicho desorden se acumula hasta el punto en que resulta muy costoso mantener el código fuente.

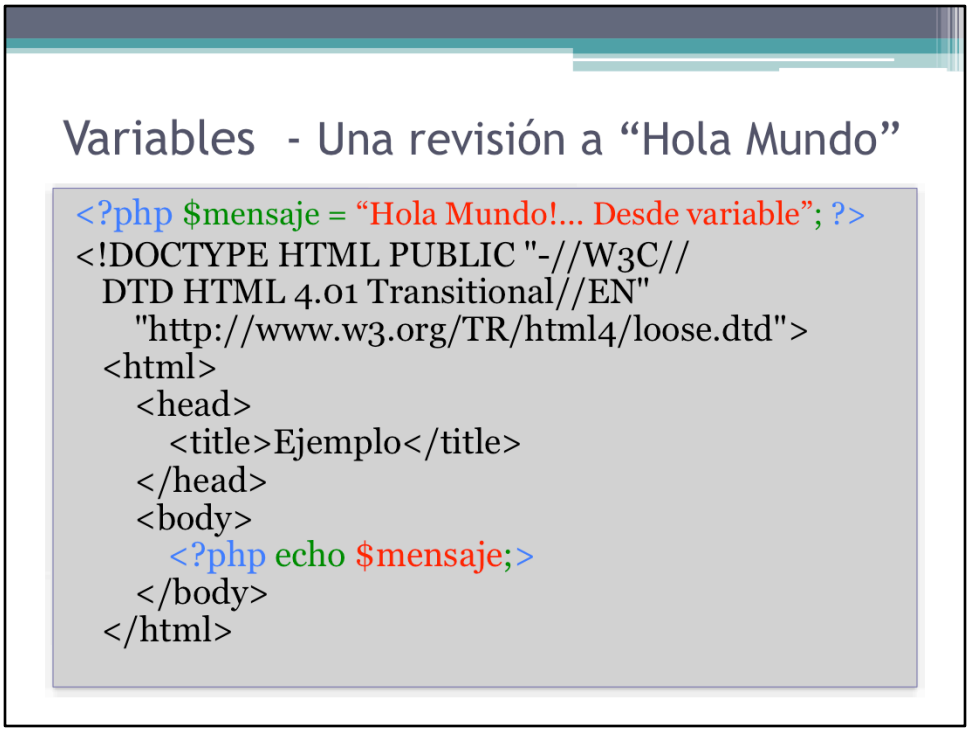
Variables – Tipos (cont.)

```php
<?php
//variable tipo string
$var = "0";
//var ahora cambia a tipo double
$var += 2.5;
//var es ahora entero
$var += 2;
//var es ahora un string
$var .= " es el total";
echo $var;
?>
```

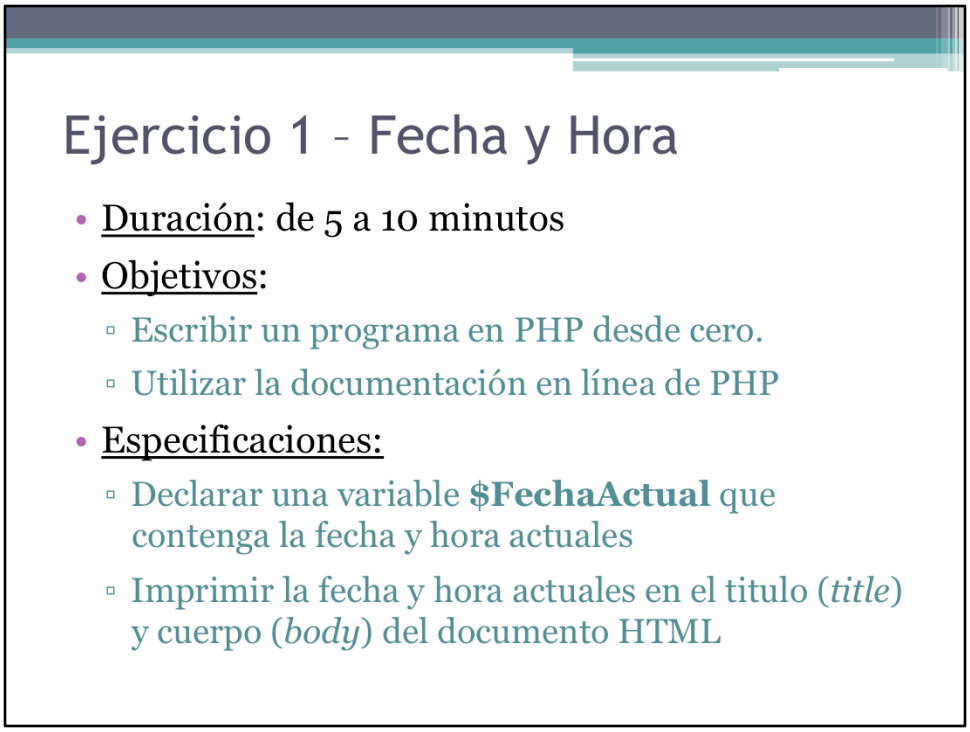
Este ejemplo demuestra como los tipos de variables en PHP cambian sin hacer referencia explícita a este cambio.



Resultado de ejecutar el código del ejemplo anterior. El restulado es un string.



Una nueva versión del clásico “Hola Mundo!” utilizando variables.

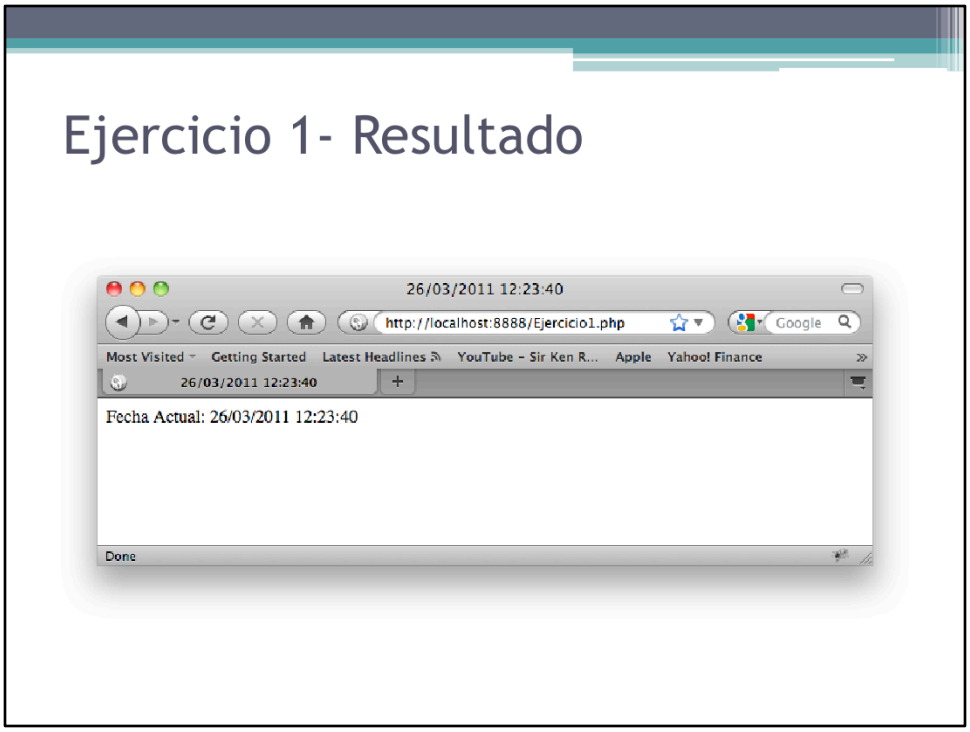


Este es un primer ejercicio para familializarse con la sintaxis de PHP, el uso de variables y la documentación en línea.

Ejercicio 1 – Fecha y Hora (cont.)

- Instrucciones:
  - Crear un nuevo archivo **Ejercicio1.php**
  - Salvarlo en la carpeta */htdocs/CursoPHPBasico/Ejercicios*
  - Buscar en la documentación de PHP (php.net) la función ***date***
  - Asignar a la variable $FechaActual el resultado de la función ***date*** con los parámetros adecuados
  - Imprimir $FechaActual en el *title* y en el *body* del documento HTML

Seguir las instrucciones.

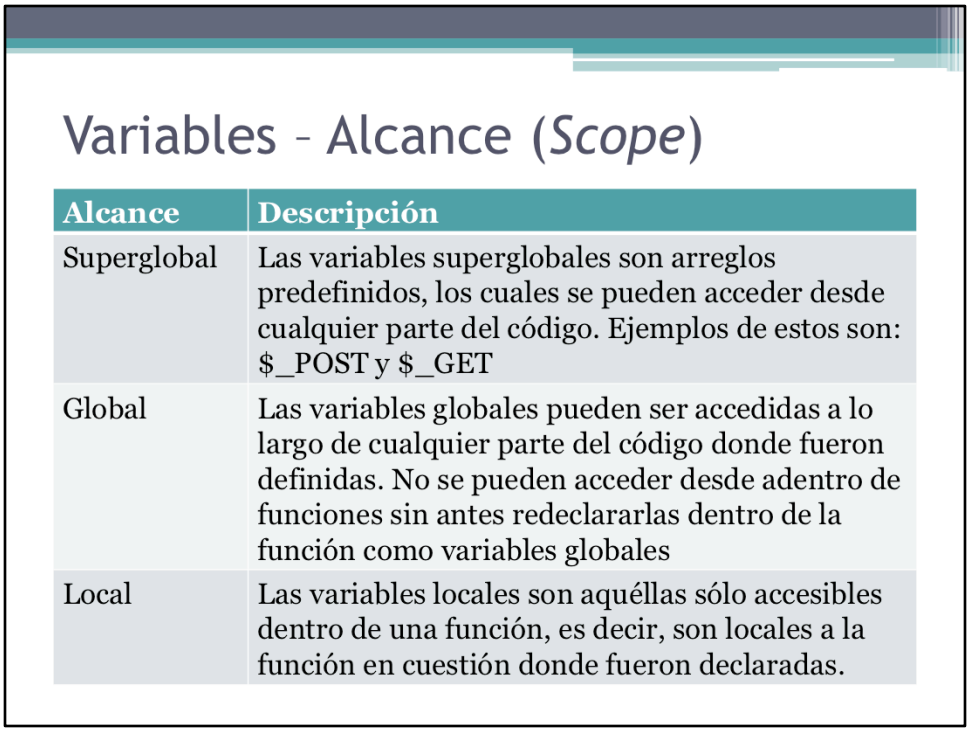


Resultado del ejercicio.

Ejercicio 1- Solución

```
<?php $FechaActual = date("d/m/Y h:i:s");?>
<html>
  <head>
      <title><?=$FechaActual?></title>
  </head>
  <body>
      Fecha Actual: <?=$FechaActual?>
  </body>
</html>
```

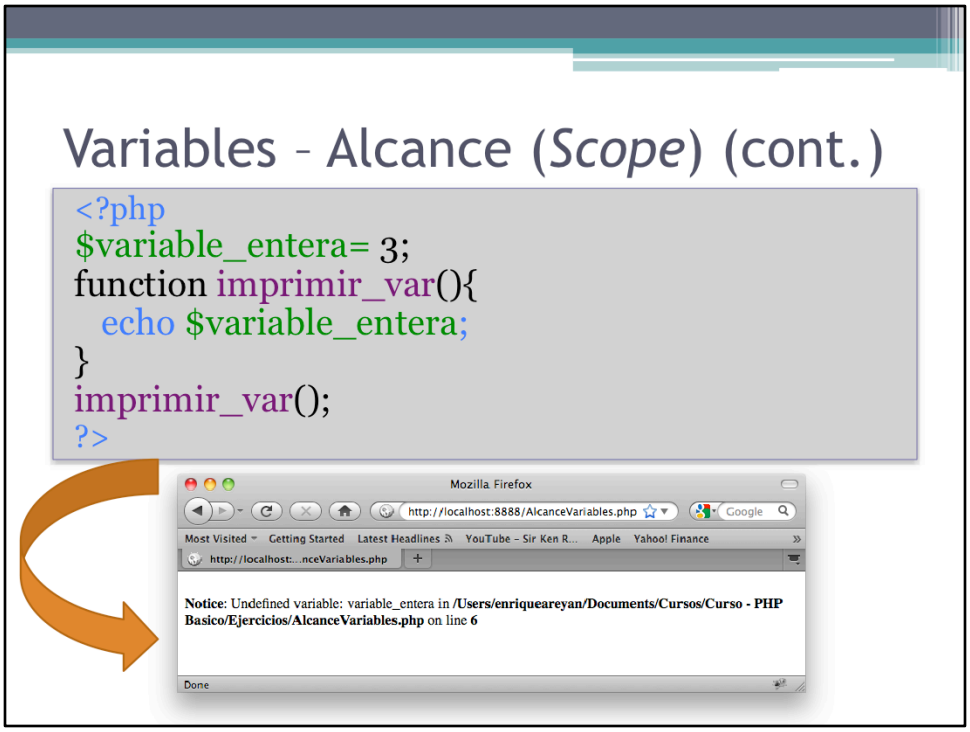
El parámetro "d/m/Y h:i:s" dentro de la función date, resulta en que la fecha se devuelva con el formato dia/mes/año hora:minutos:segundos.



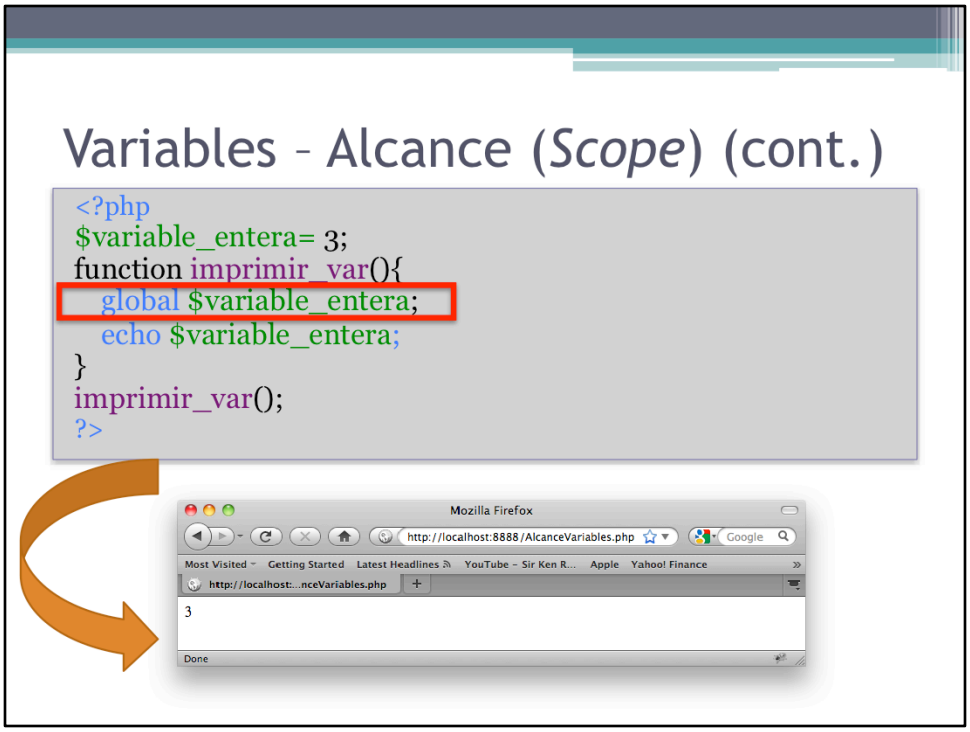
Variables – Alcance (*Scope*)

Alcance	Descripción
Superglobal	Las variables superglobales son arreglos predefinidos, los cuales se pueden acceder desde cualquier parte del código. Ejemplos de estos son: $_POST y $_GET
Global	Las variables globales pueden ser accedidas a lo largo de cualquier parte del código donde fueron definidas. No se pueden acceder desde adentro de funciones sin antes redeclararlas dentro de la función como variables globales
Local	Las variables locales son aquéllas sólo accesibles dentro de una función, es decir, son locales a la función en cuestión donde fueron declaradas.

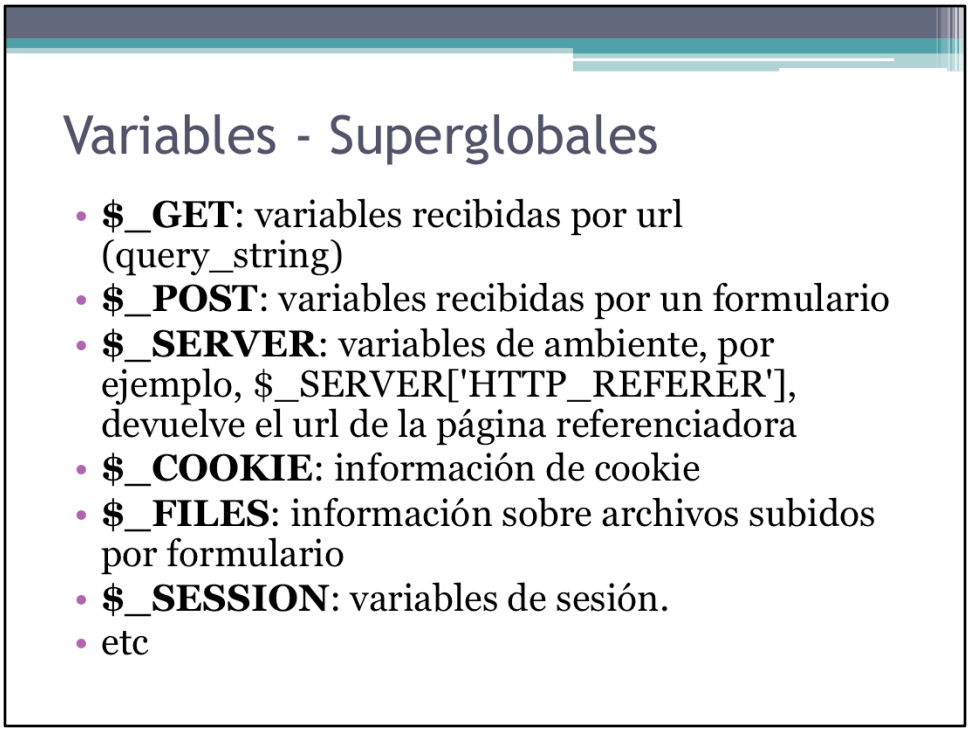
El alcance (scope) de una variable determina desde donde esta puede ser accedida.



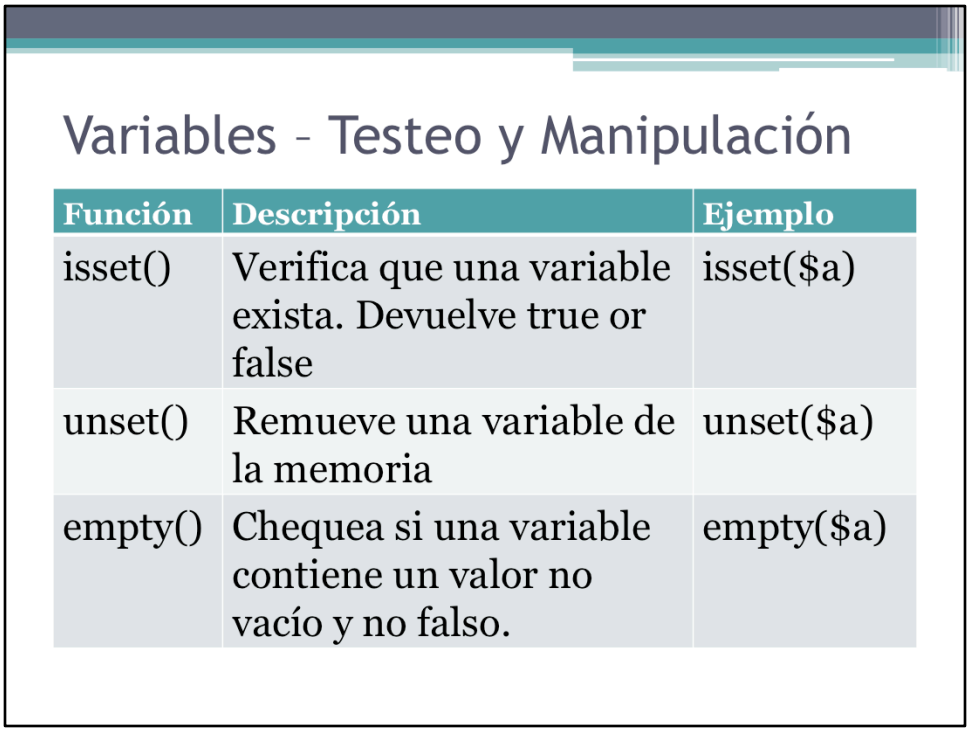
La notificación se genera porque la variable $variable_entera no existe dentro del ambiente de la función.



Una vez declarada como global la variable $variable_entera, esta pasa al ambiente de la función y su valor se puede imprimir.



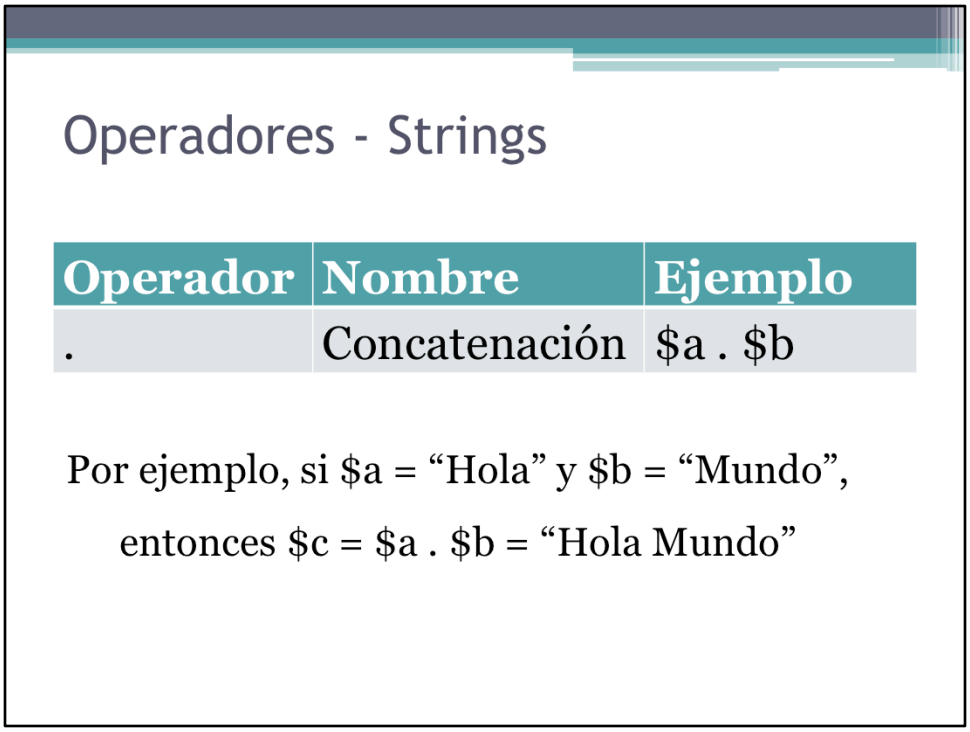
La lista completa de variables superglobales se encuentra en php.net



Funciones útiles al momento de manipular variables y verificar su existencia.

Una vez que se tienen las variables, estas deben poder ser operadas para llevar a cabo tareas fundamentales. Este apartado explora el uso de los operadores en PHP.

Los operadores matemáticos trabajan sobre datos de tipo entero o real y producen un nuevo dato de tipo entero o real según sea la definición de la operación y los valores de entrada. Los operadores matemáticos son binarios, es decir, actúan sobre dos variables para producir un nuevo valor.

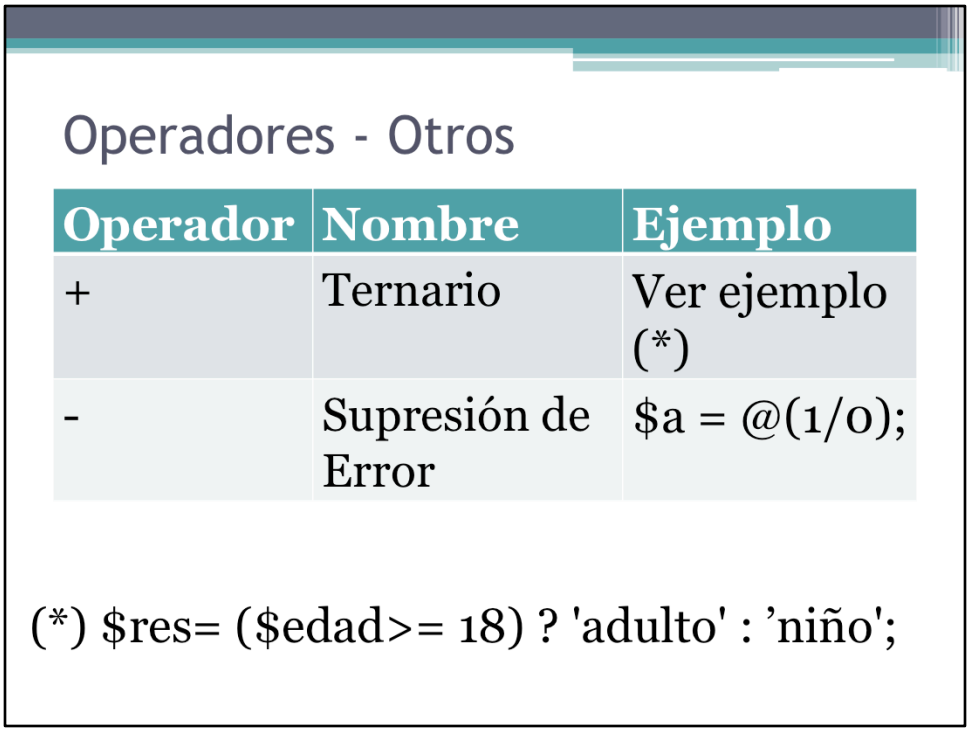


La concatenación de string es una operación fundamental de PHP y será ampliamente utilizada a lo largo de este curso.

Operadores - Asignación

Operador	Nombre	Ejemplo
=	Asignación	$a = 1; $a = “Cualquier texto”;
+= -= *= /= %= .=	Combinación de asignaciones	$a += 1; ($a = $a+1;) $a -= 1; $a *= 2; $a /= 2; $a %= 2; $a .= “texto”;
++	Incremento en uno	$a++; ($a = $a+1;)
--	Decremento en uno	$a--; ($a = $a-1;)

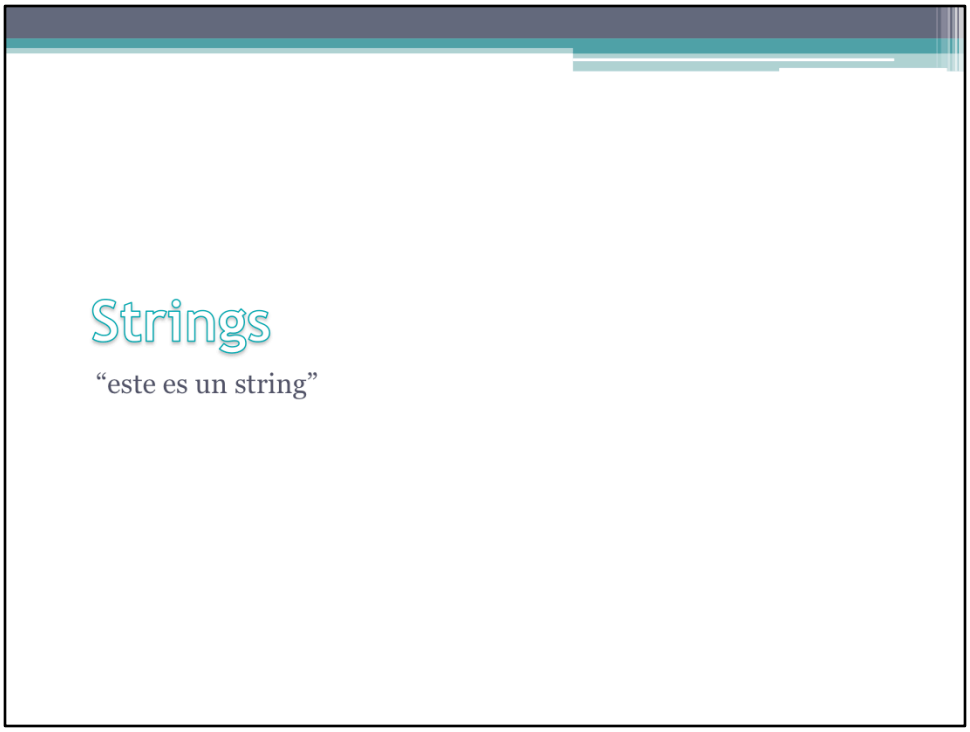
Al igual que los operadores matemáticos, los operadores de asignación son operadores binarios. En este caso se tiene, por una parte, una variable a la que se le va a asignar un valor y, por otra parte, una expresión. El resultado de la operación es asignar el resultado de la expresión a la variable en cuestión.



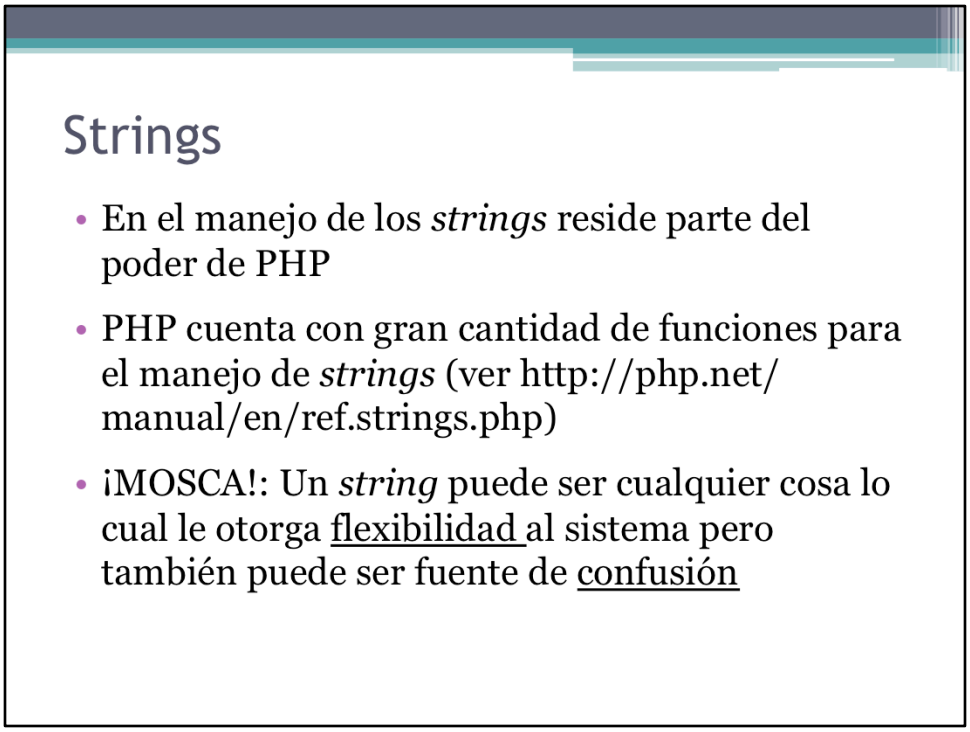
Operadores especiales. No son muy comúnmente utilizados pero pueden resultar útiles en ciertas ocasiones.

Operadores - Ejemplos

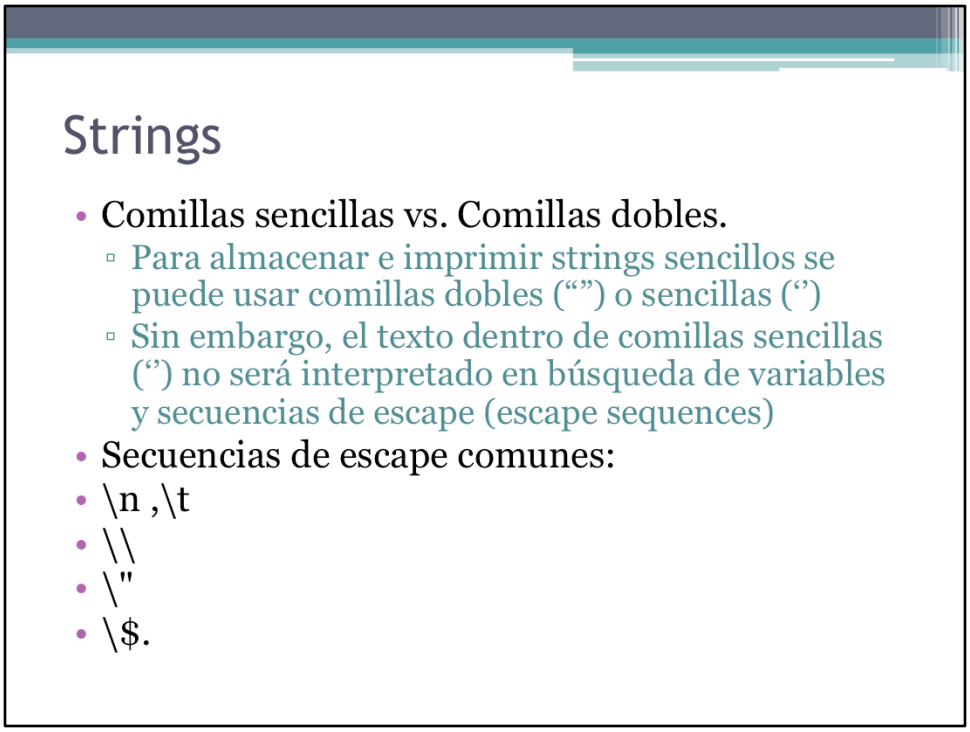
- Ver ejemplos en el archivo Operadores.php



PHP es un lenguaje flexible. Esta flexibilidad se pone de manifiesto en la facilidad con la que se pueden manipular strings o cadenas de caracteres. En este apartado exploramos los strings en el contexto de un script en PHP.



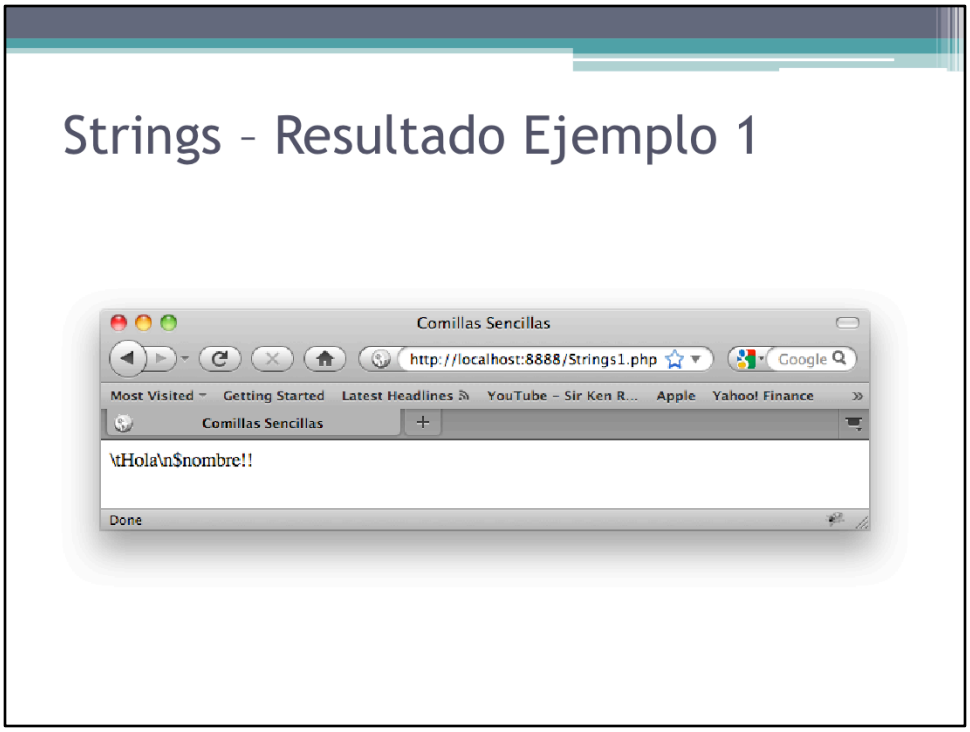
Algunas notas sobre strings. Recordar que con mayor flexibilidad también aumenta la posible confusión que se pueda presentar en el código.



La secuencia de escape es utilizada en caracteres para los cuales existe ambigüedad al momento de imprimirlos. La secuencia de escape precede dicho caracter por un backslash (\). Algunas secuencias de escape comúnmente utilizadas son: \n para un salto de línea, \t para tab, \\ para backaslash, \" para comillas dobles, y \$ para signo de dólar.

Strings – Ejemplo 1

```
<html>
  <head>
      <title>Comillas Sencilla</title>
  </head>
  <body>
      <?php
            $nombre = “Juan”;
            echo '\tHola\n$nombre !!';
      ?>
  </body>
</html>
```

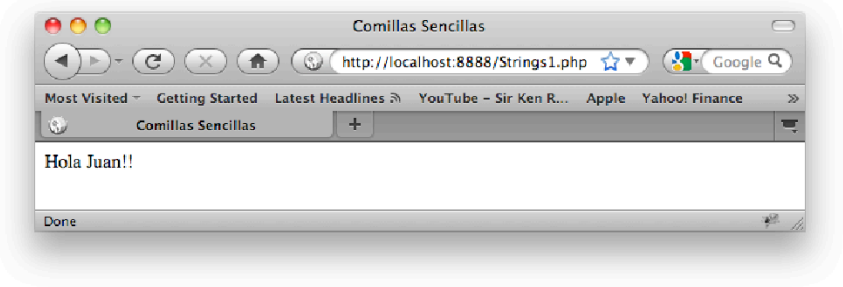


Se imprime el string textualmente

Strings – Ejemplo 2

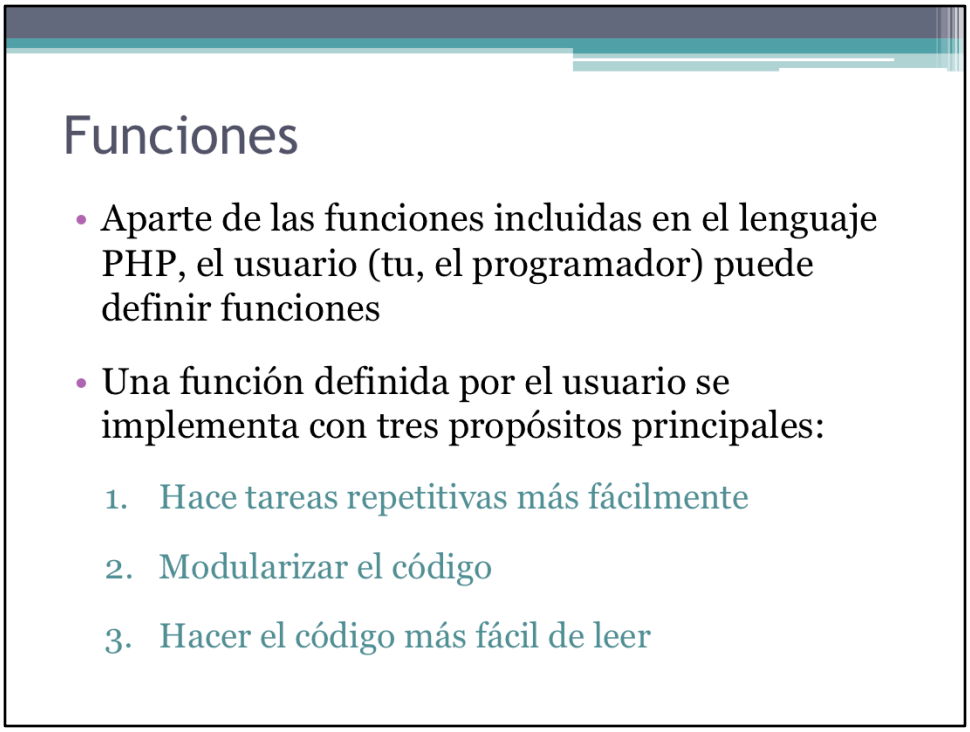
```
<html>
  <head>
      <title>Comillas Sencilla</title>
  </head>
  <body>
      <?php
            $nombre = “Juan”;
            echo “\tHola\n$nombre !!”;
      ?>
  </body>
</html>
```

Strings – Resultado Ejemplo 2

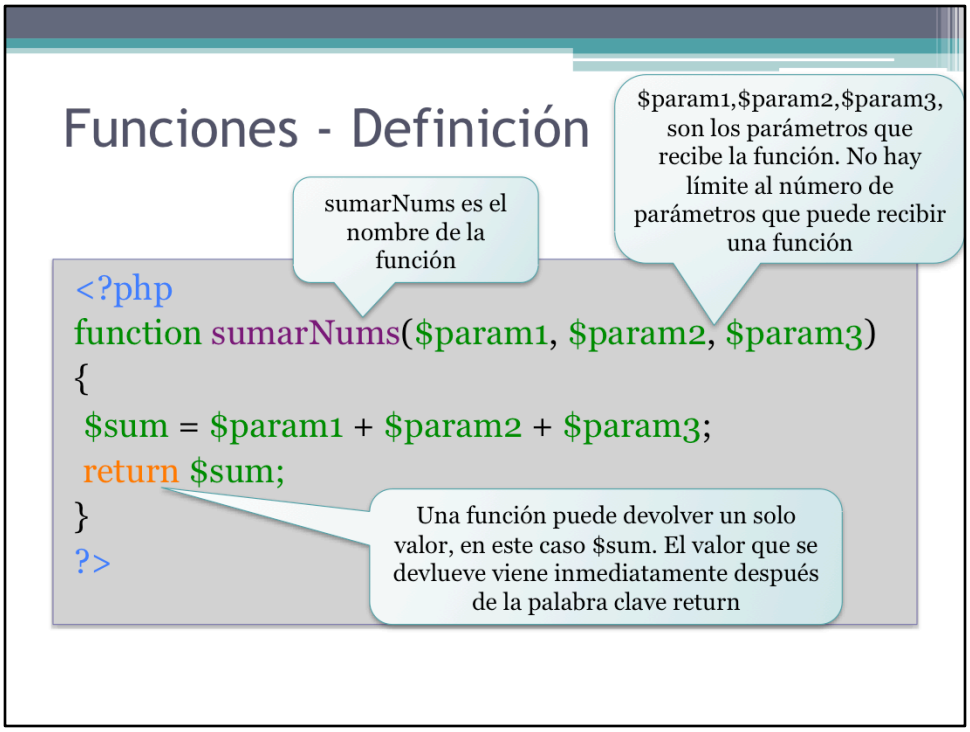


Se interpreta el contenido del string antes de imprimirlo

Las funciones nos proveen de herramientas para customizar nuestro código y hacerlo más legible y comprensible. En este apartado trabajamos con funciones definidas por el usuario en PHP. Se enseña como crear una función y se da un ejemplo práctico de su uso.



Algunas de las características de las funciones en general que aplican para PHP.



Definición de una función muy sencilla. El concepto de función se inspira en el de una función matemática en cuanto que se pueden tener múltiples variables de entrada pero sólo una salida. Sin embargo, visto desde un mayor nivel de abstracción, esta salida o valor de retorno pudiera ser una variable de tipo estructurado que da cabida a devolver más de un valor.

Funciones (cont.)

- Al igual que en el caso de las variables, las funciones son débilmente tipadas
  - No se requiere definir tipos de parámetros
  - No se requiere definir tipo de dato del valor de retorno
- A una función se le puede pasar cualquier cosa y devuelve cualquier cosa
  - Puede hacer el código confuso
  - Queda por parte del usuario verificar la validez de los datos de entrada y salida

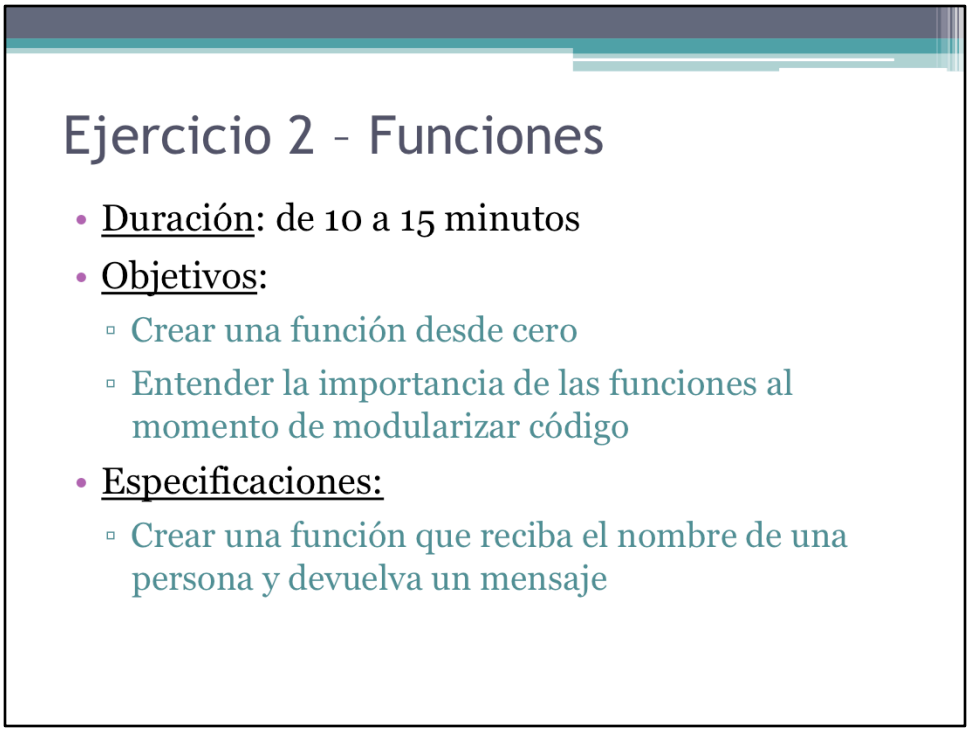
Funciones - Ejemplo

```php
<?php
function sumarNums($param1, $param2, $param3)
{
 $sum = $param1 + $param2 + $param3;
 return $sum;
}
echo sumarNums(1,2,3);          => 6
echo sumarNums(1,2,"Hola"); =>3
echo sumarNums(true,2,'a');   =>3
?>
```

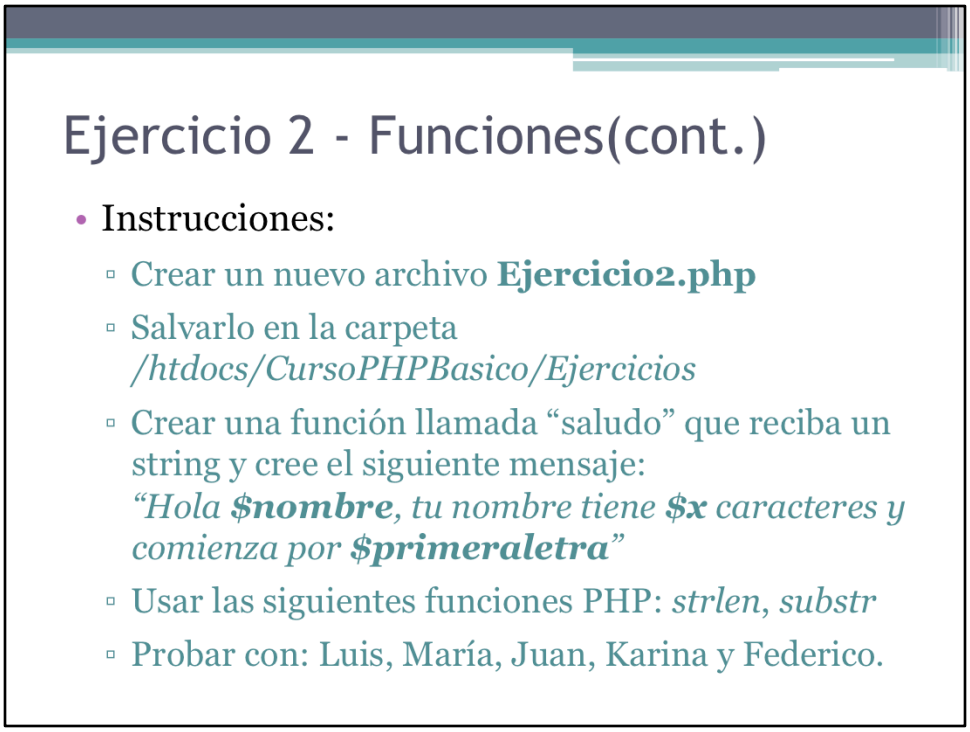
PHP no arroja ninguna notificación o mensaje de error. ¿Es este comportamiento lógico?

Funciones – Ejemplo (cont.)

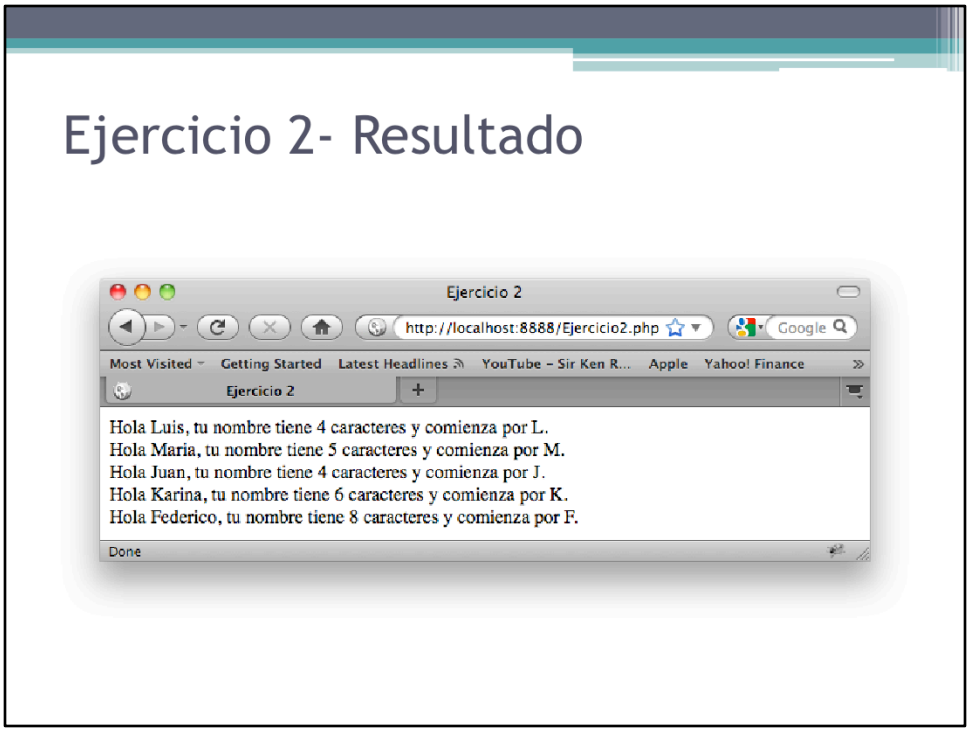
```php
<?php
function sumarNums($param1, $param2, $param3)
{
 if(!is_numeric($param1) || !is_numeric($param2)|| !
   is_numeric($param3)){
   die("Algun parametro no es de tipo numerico");
 }
 $sum = $param1 + $param2 + $param3;
 return $sum;
}
?>
```



Este segundo ejercicio añade complejidad a una página web sencilla.



Seguir las instrucciones.

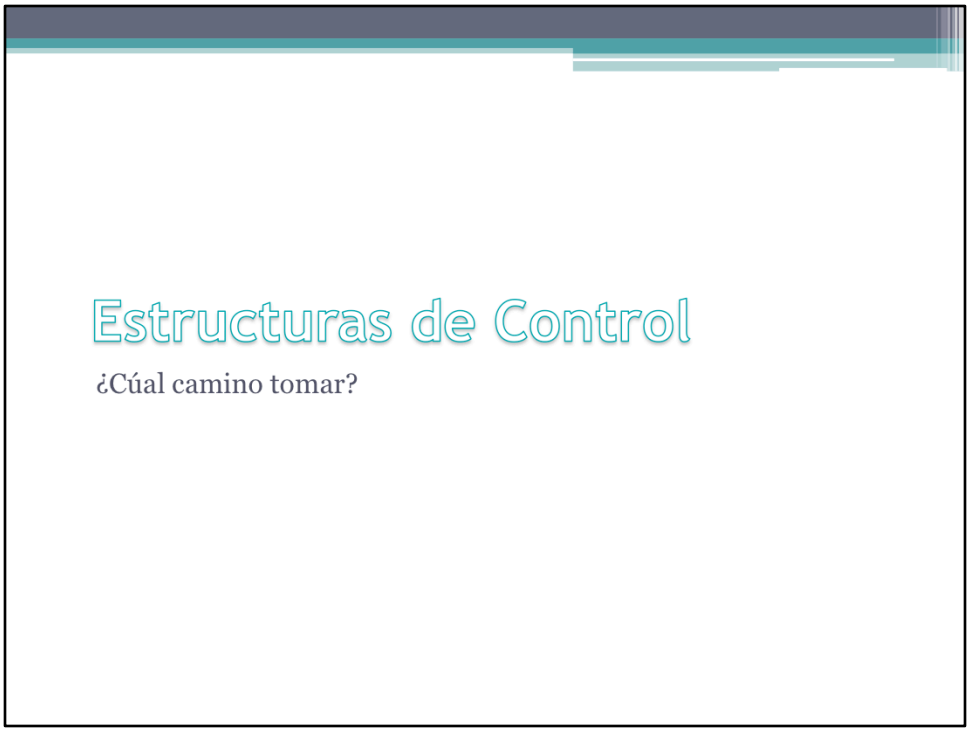


El trabajo que se podría tornar tedioso se hace sencillo con el uso de funciones.

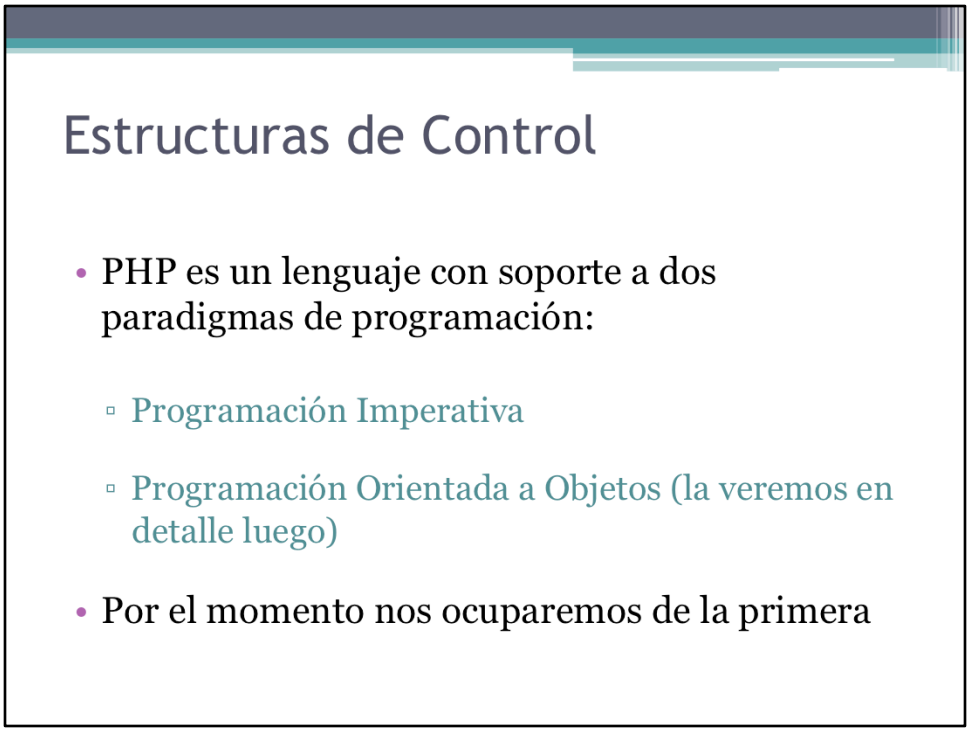
Ejercicio 2- Solución

```
<?php
    function saludo($nombre){
            $x = strlen($nombre);
            $primera_letra = substr($nombre,0,1);
            $mensaje = "Hola $nombre, tu nombre tiene $x caracteres y comienza por $primera_letra. <br/>";
            return $mensaje;
    }
?>
<html>
    <head>
            <title>Ejercicio 2</title>
    </head>
    <body>
            <?=saludo("Luis")?>
            <?=saludo("Maria")?>
            <?=saludo("Juan")?>
            <?=saludo("Karina")?>
            <?=saludo("Federico")?>
    </body>
</html>
```

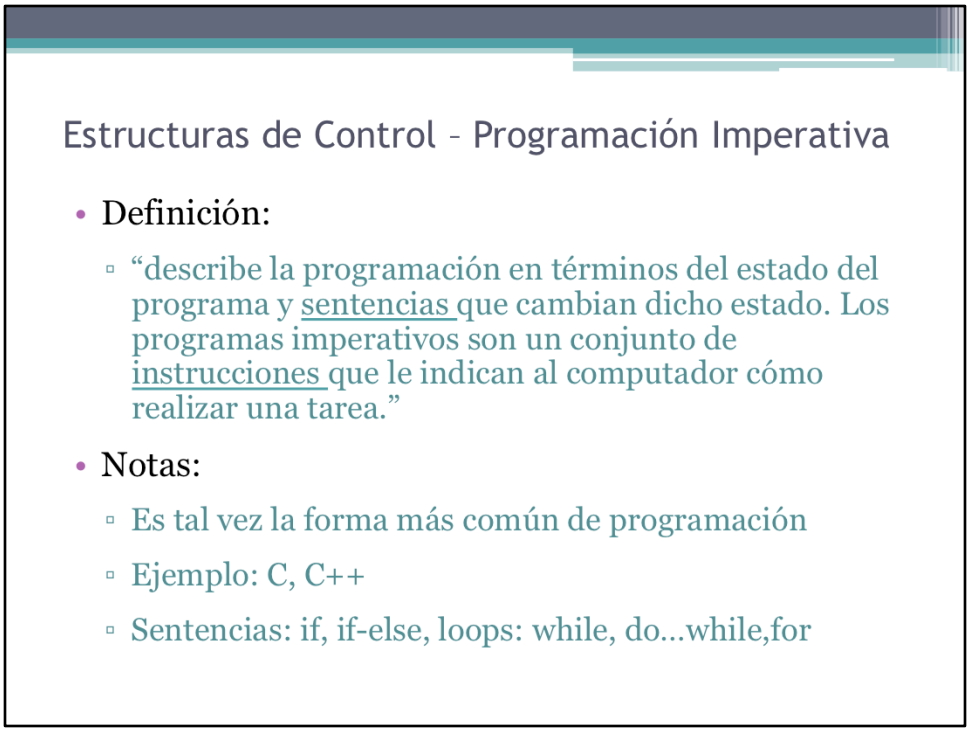
El código resulta mucho más sencillo de enteder y por ende de actualizar.



En los lenguajes imperativos, las estructuras de control, junto con las variables, forman los bloques fundamentales para la construcción de cualquier programa.

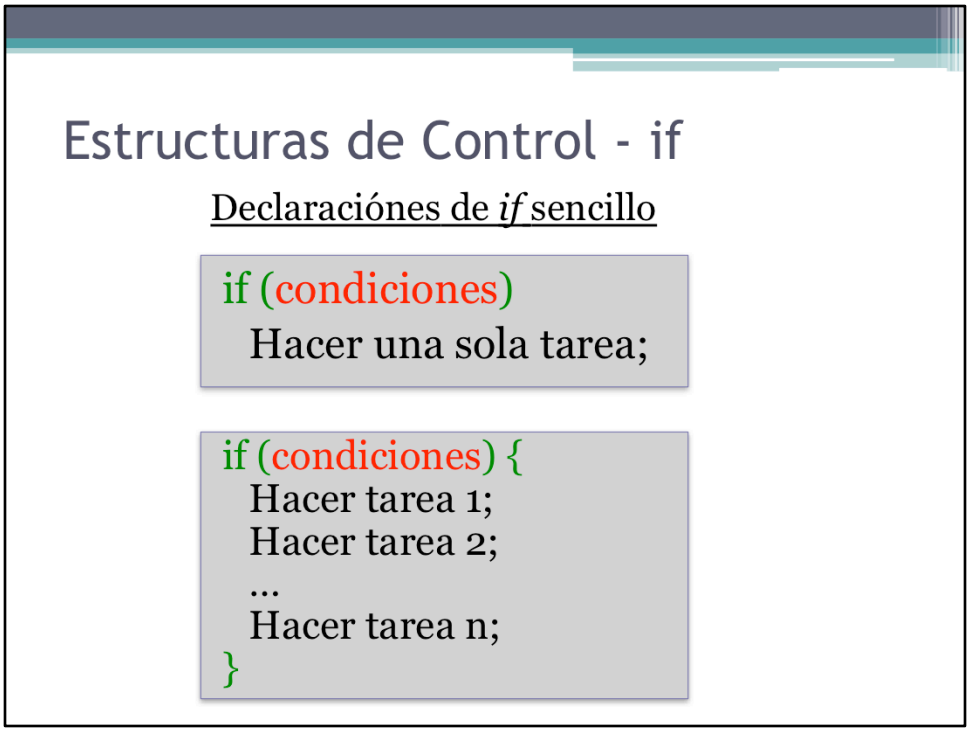


PHP orientado a objetos es el curso que le sigue a este.

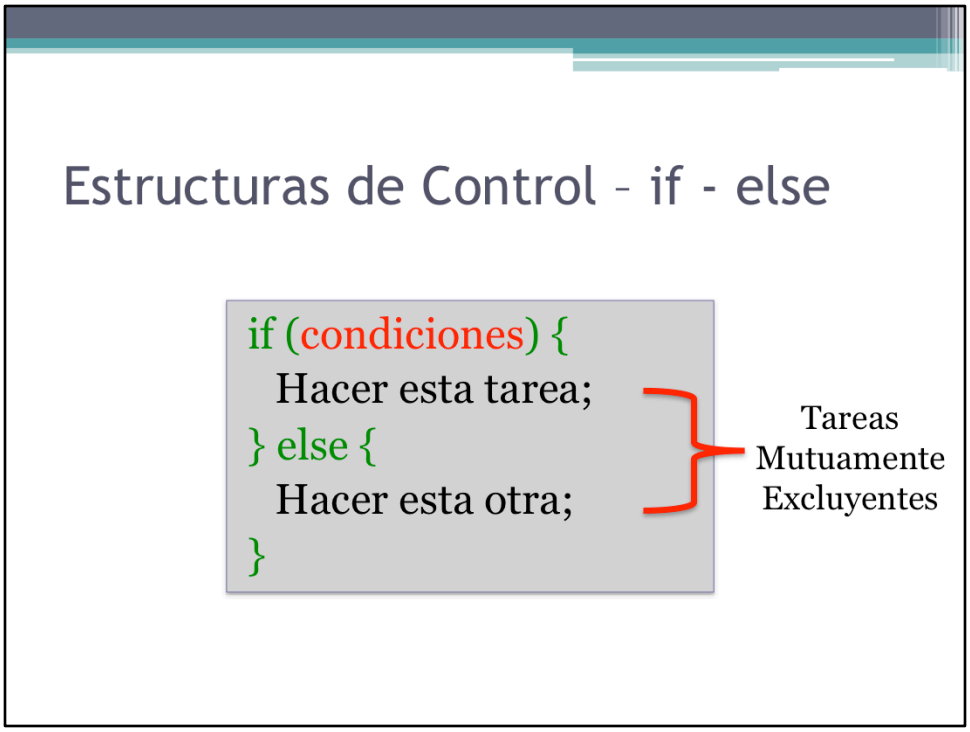


Definición tomada de Wikipedia.

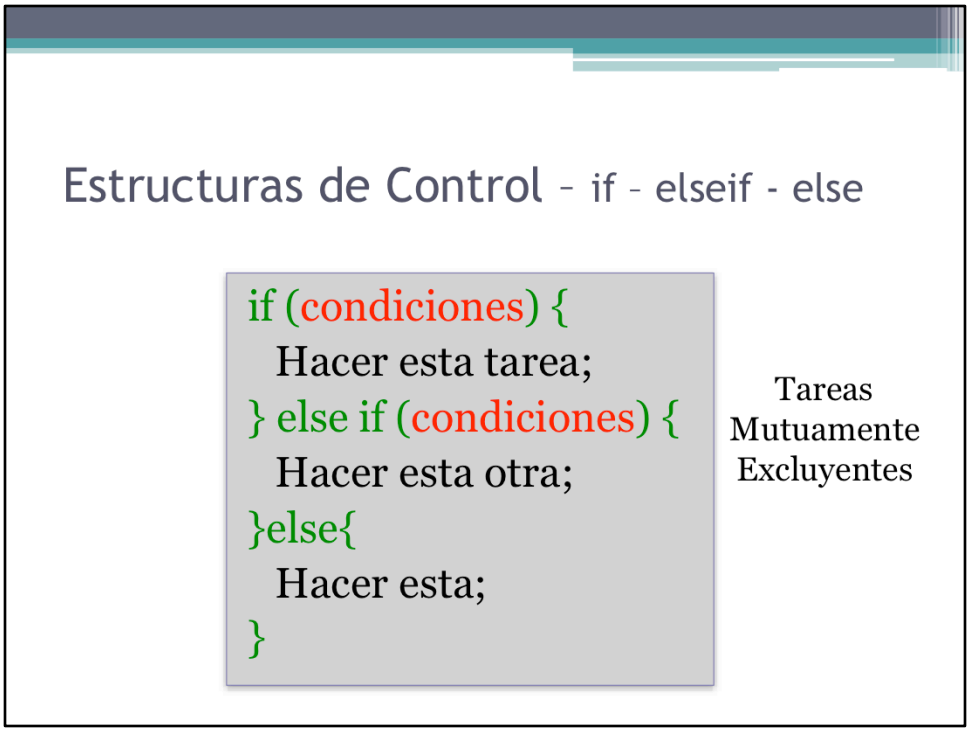
En esta sección se estudia los condicionales if, if-else, if-elseif-else, switch



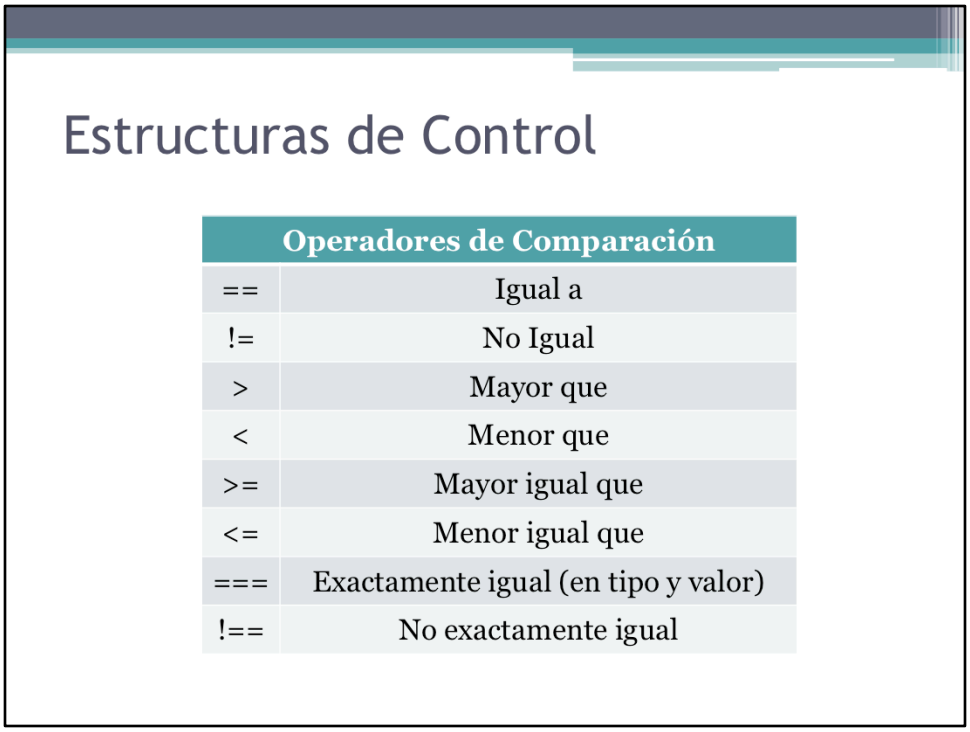
If sencillos. Este es el bloque básico de construcción en un lenguaje imperativa. La sintaxis en PHP es muy sencilla, tal como se muestra arriba.



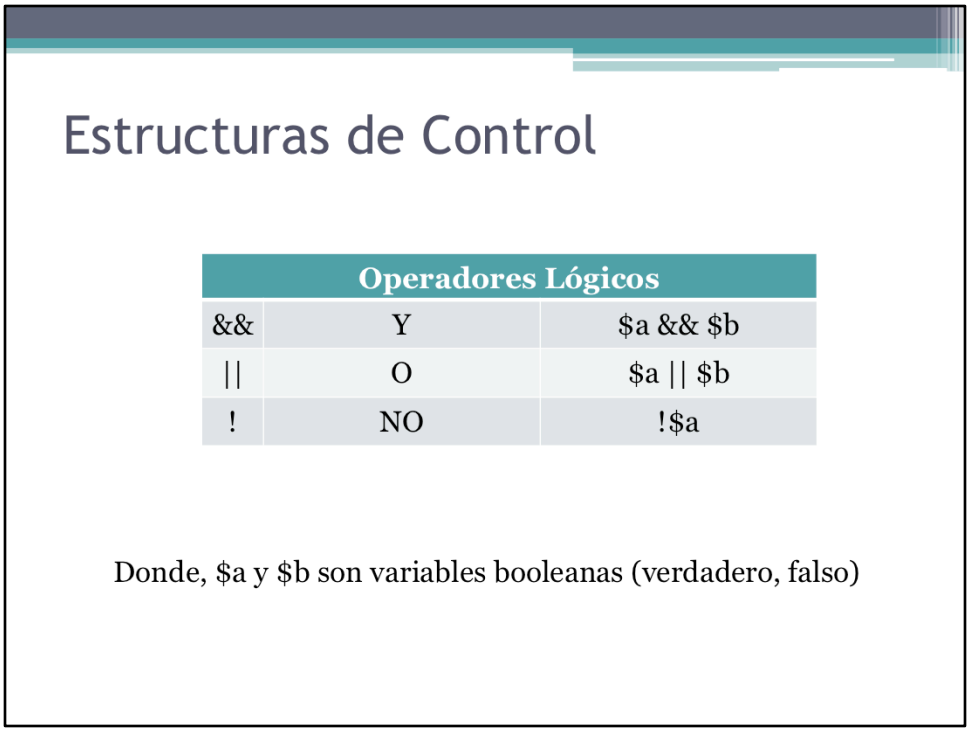
Las tareas son mutuamente excluyente, lo que significa que la realización de una de ellas impide la realización de la otra.



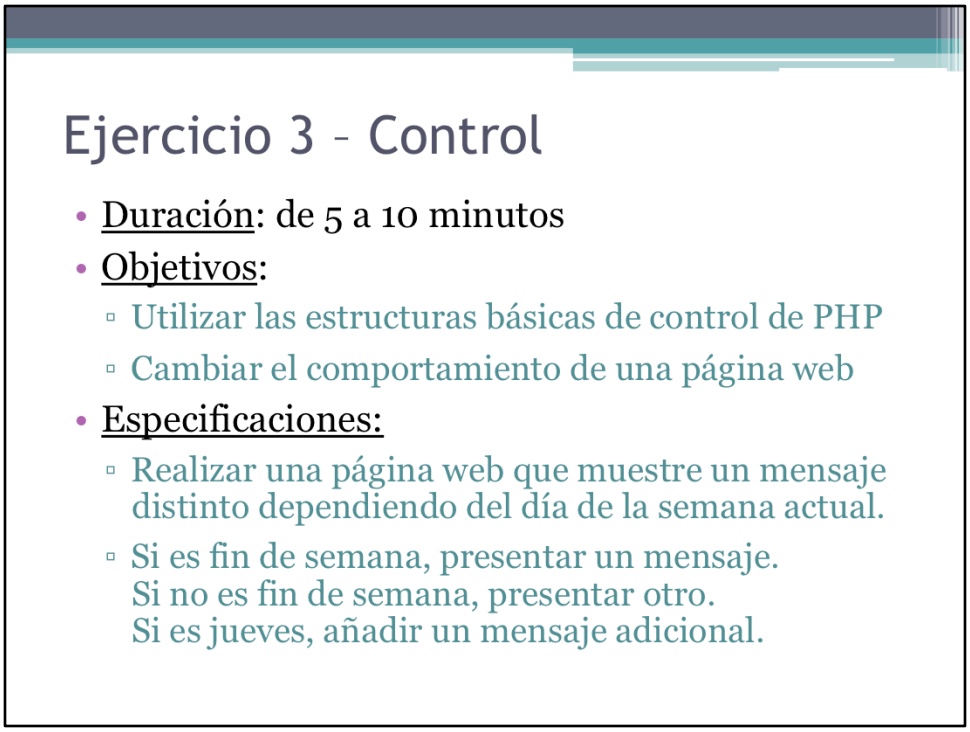
Se puede tener cualquier cantidad de bloques “else if”



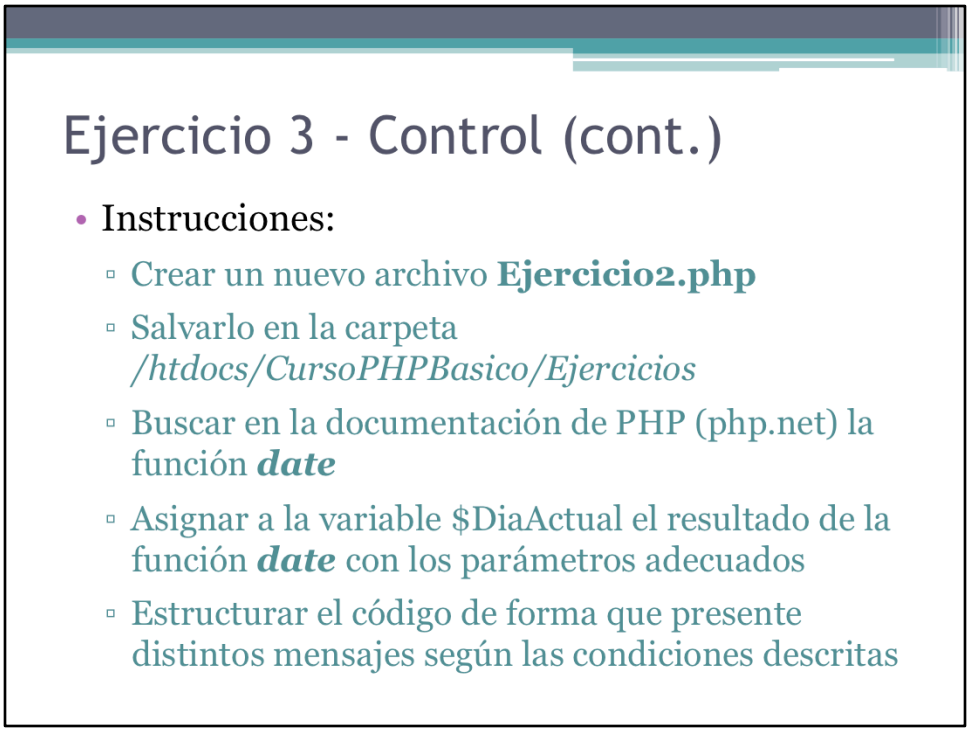
Como PHP es un lenguaje débilmente tipado, provee operaciones de comparación sobre valores y tipo de datos como una forma de controlar la ejecución de los scripts.



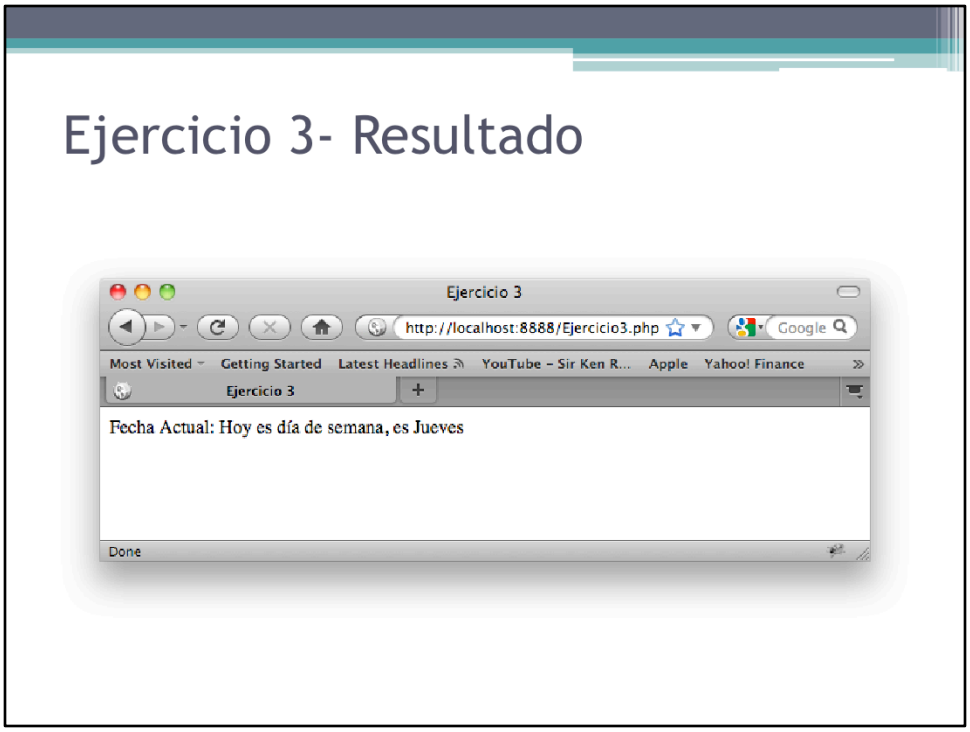
Operadores lógicos (también llamados booleanos) clásicos.



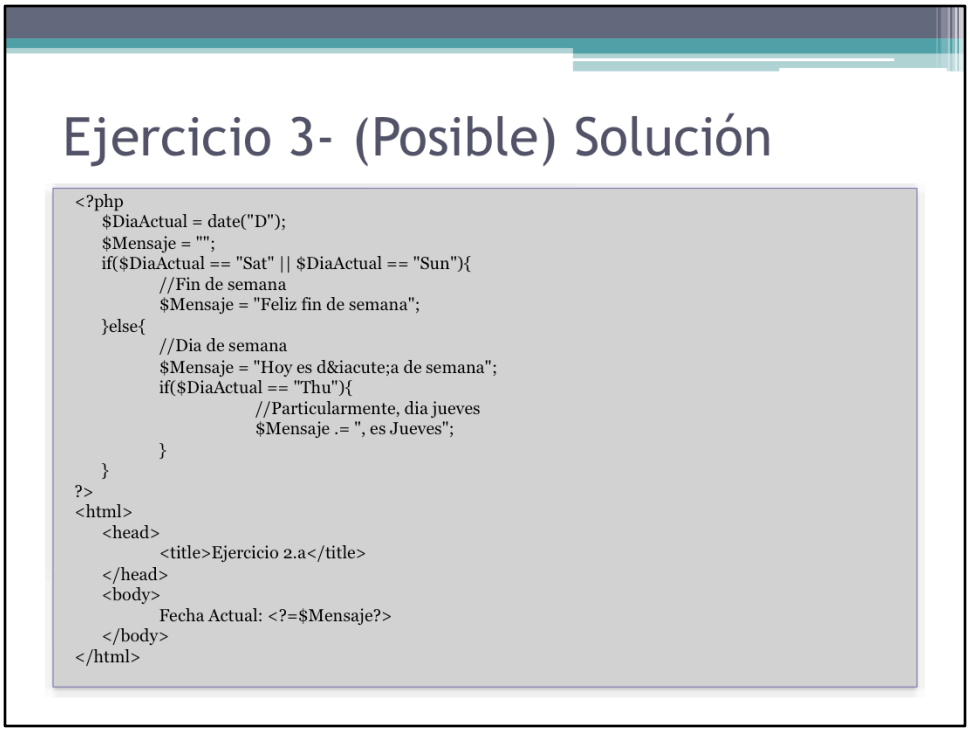
Un ejercicio para familiarizarse con las estructuras de control básica.



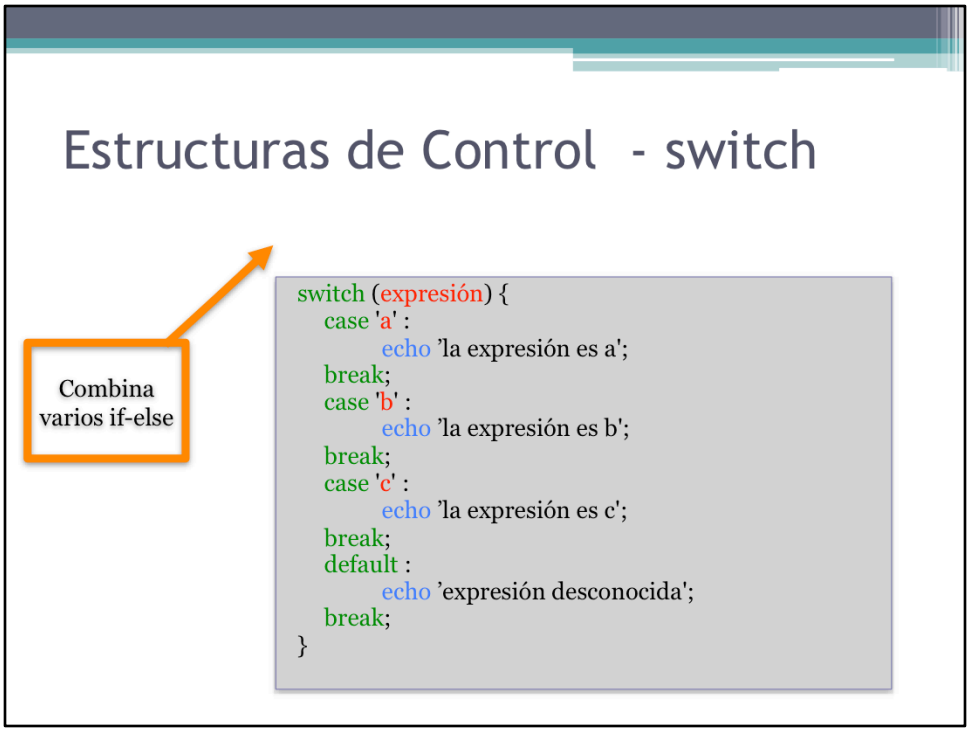
Seguir las instrucciones.



Se despliega el día de la semana actual, sin necesidad de cambiar el código por parte del usuario.



Código solución Ejercicio 3.



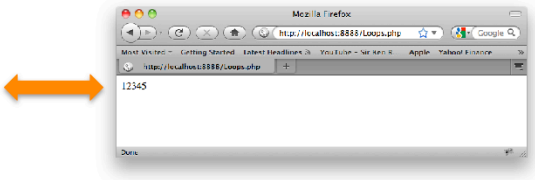
Switch clásico. Combina varios if en un solo bloque.

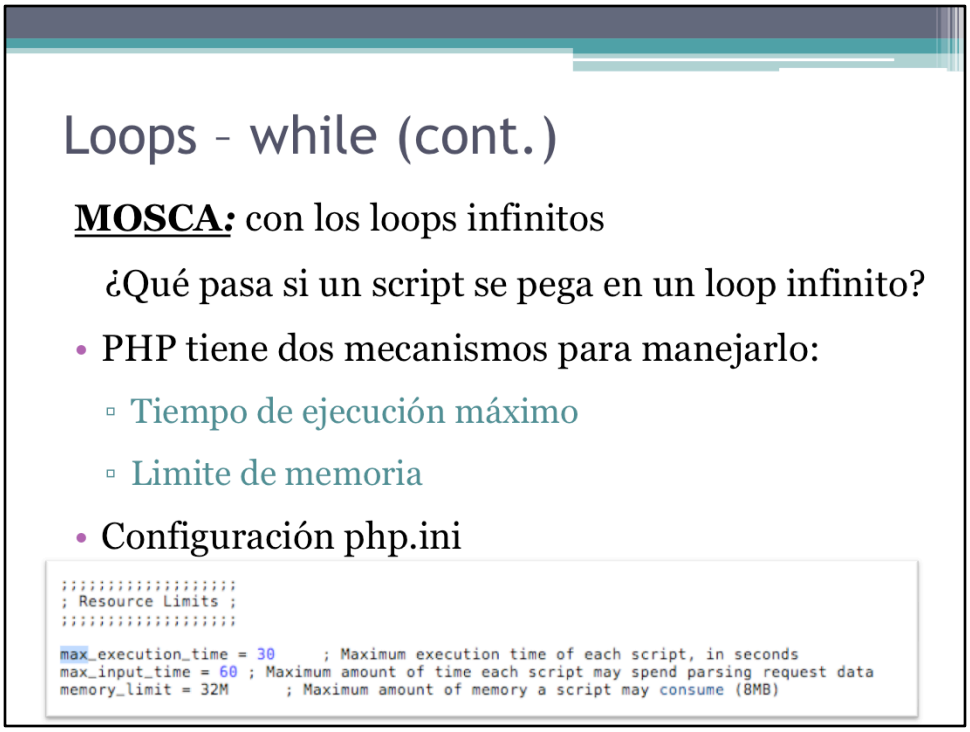
Los loops (en inglés) o bucles (traducción al español) son mecanismos de los lenguajes de programación modernos que le permiten al usuario realizar tareas repetitivas de manera sencilla. En esta sección estudiaremos estos mecanismos en el contexto de PHP y programación para la web.

Loops - while

- La directiva while se utiliza para ejecutar un bloque de código, repetidamente, mientras una o más condiciones sean verdad

```
$a = 1;
while($a < 6){
  echo $a;
  $a++;
}
```



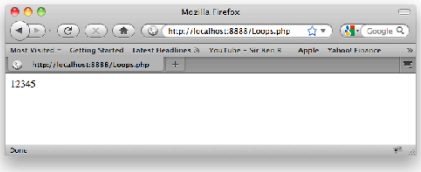


Si un script tarda más de 30 segundos en ejecutarse, php lo abortará. Esto también ocurre si el script consume más de 32MB de memoria. En el caso en que un script quede en un loop infinito, en 30 segundos el programa será abortado automáticamente.

Loops – do...while

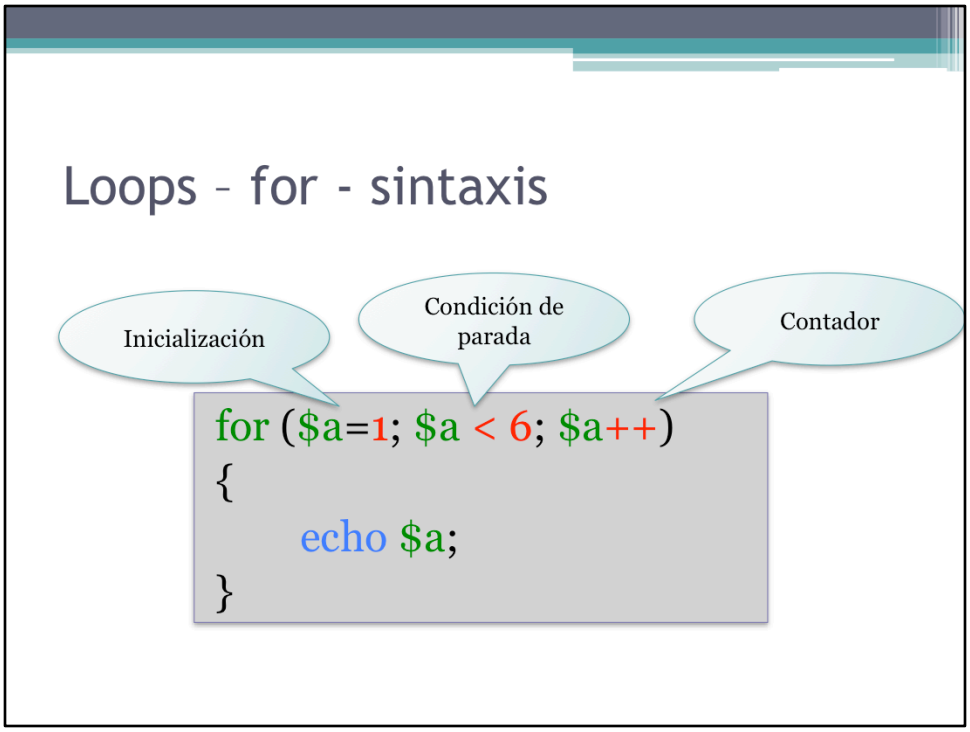
- En este caso la condición de parada se chequea luego de ejecutar el bloque de código una primera vez

```
$a=1;
do {
  echo $a;
  $a++;
} while ($a < 6);
```



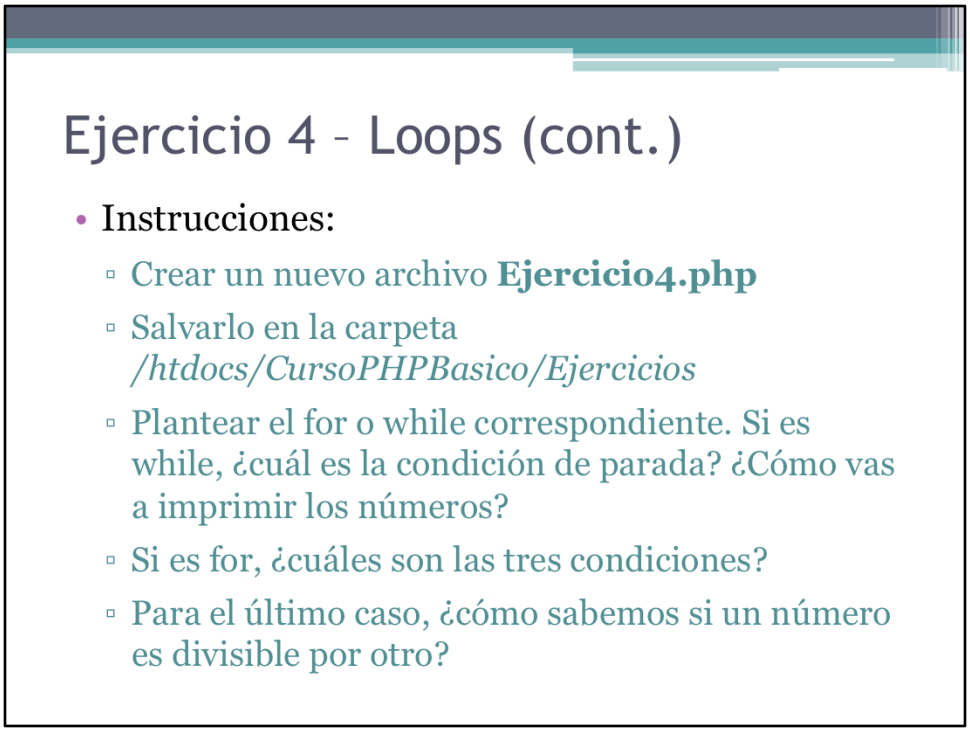
Loops - for

- Un *loop for* toma tres expresiones, separadas por punto y coma (;), y agrupadas en paréntesis, antes del bloque de código sobre el cual se va a iterar. Las reglas del for son:
  1. La primera expresión se ejecuta una sola vez antes de iniciar el loops. Es la condición de inicialización
  2. La segunda expresión se evalúa antes de cada iteración. Si es falsa, entonces termina el loop
  3. La tercera expresión se evalúa al final de cada iteración. Esta se usa como contador o cualquier otra expresión que calcule el final del loop


Loops – for - sintaxis
Inicialización
Condición de parada
Contador
for ($a=1; $a < 6; $a++)
{
echo $a;
}

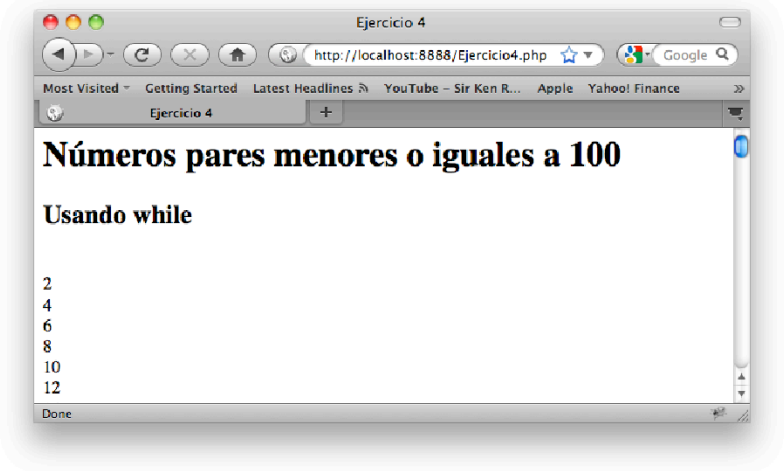
Ejercicio 4 – Loops

- Duración: de 10 a 15 minutos
- Objetivos:
  - Utilizar las estructuras iterativas de PHP
- Especificaciones:
  - Calcular e imprimir todos los números pares menores o iguales a 100, utilizando un **while**
  - Calcular e imprimir todos los números impares menores o iguales a 100, utilizando un **for**
  - Calcular e imprimir todos los números menores o iguales a 100 que sean divisibles por 2 y 3



Seguir las instrucciones.

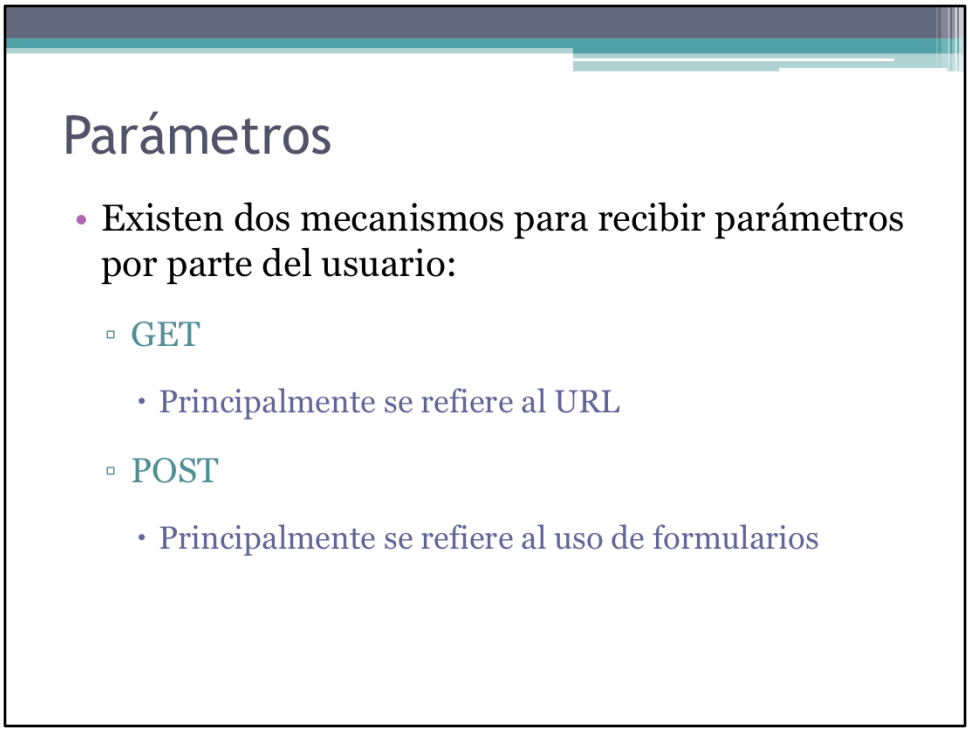
Ejercicio 4- Resultado



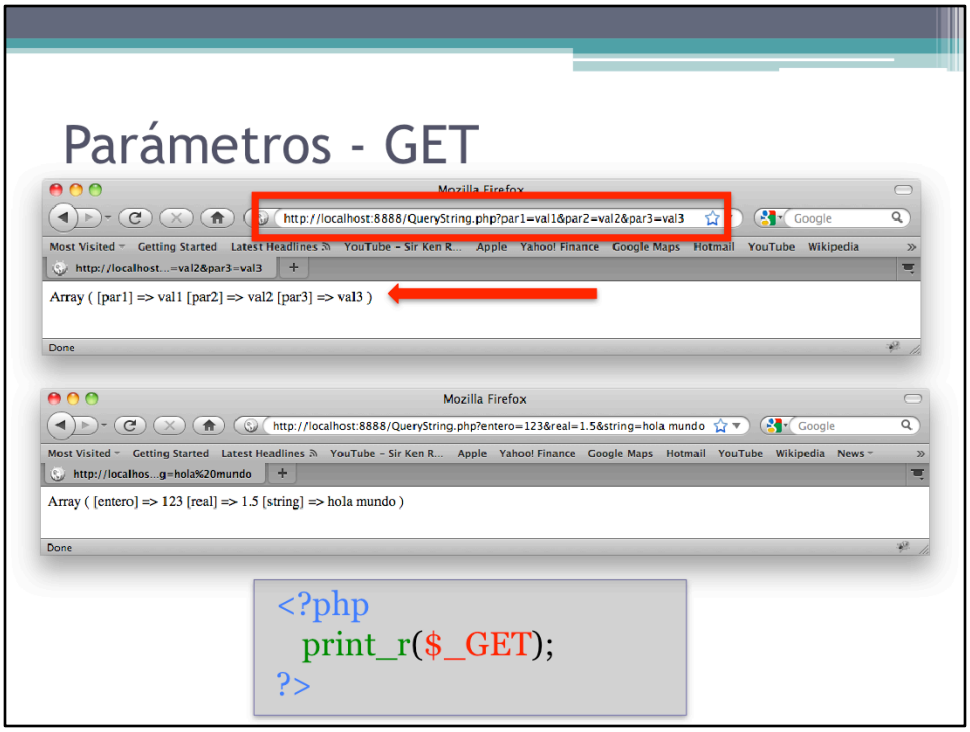
Ejercicio 4- (Posible) Solución

```
<html>
    <head>
        <title>Ejercicio 4</title>
    </head>
    <body>
        <h1>N&uacute;meros pares menores o iguales a 100</h1>
        <h2>Usando while</h2>
        <?php
            $numero = 1;
            while($numero<=50){
                echo "<br/>". ($numero * 2);
                $numero++;
            }
        ?>
        <h1>N&uacute;meros impares menores o iguales a 100</h1>
        <h2>Usando for</h2>
        <?php
            for($i=1;$i<100;$i=$i+2){
                echo "<br/>" . $i;
            }
        ?>
        <h1>N&uacute;meros menores o iguales a 100 divisibles por 2 y 3</h1>
        <?php
            for($i=1;$i<100;$i++){
                if($i%2 == 0 && $i%3==0){
                    echo "<br/>" . $i;
                }
            }
        ?>
    </body>
</html>
```

Un programa para la web no puede ser autocontenido o cerrado. Por el contrario, la única forma de que una página web sea dinámica y útil para el cliente es si éste es capaz de proveerle información. En esta sección veremos como es posible esa comunicación entre cliente y código PHP.

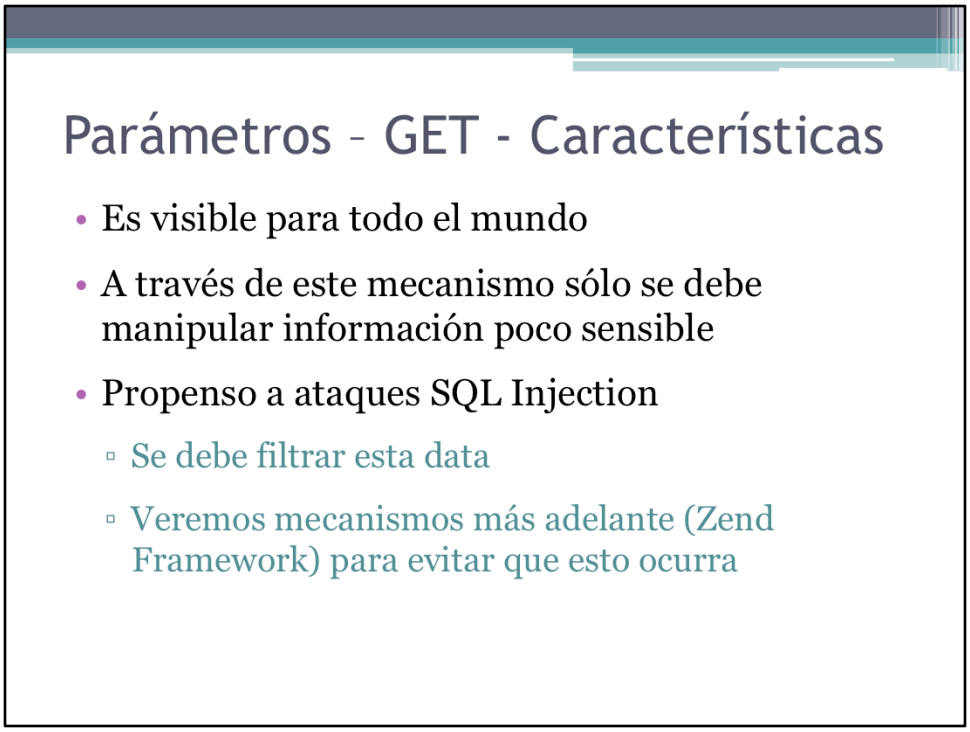


Existen dos formas básicas de recibir parámetros por parte del cliente, GET y POST. Cada una presenta sus desventajas y ventajas.

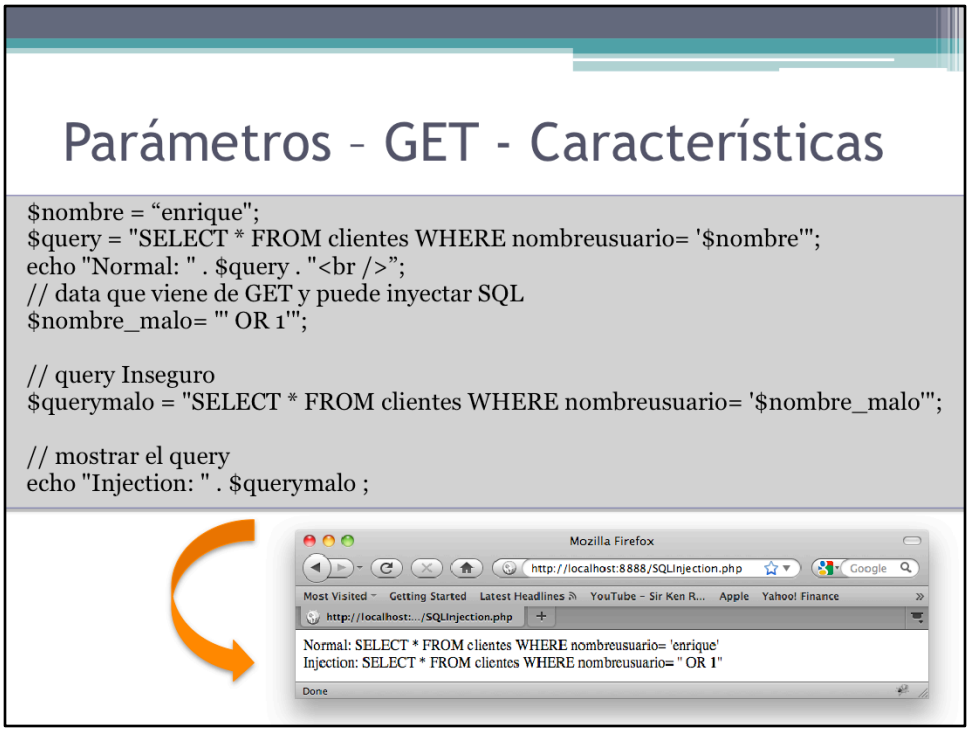


Esto sólo funciona para recibir parámetros via web browser. Para recibir parámetros a través del terminal existen otros mecanismos.

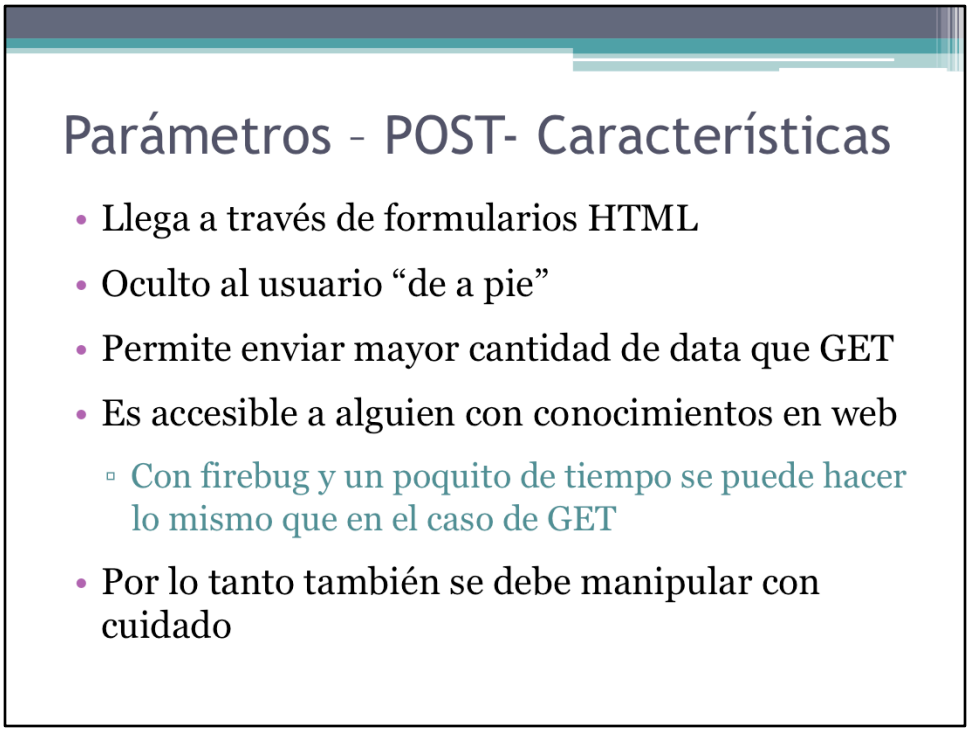
Ejemplo práctica del uso de los parámetros tipo GET para comunicar una página web con el cliente.



GET son los parámetros que se reciben por el url.



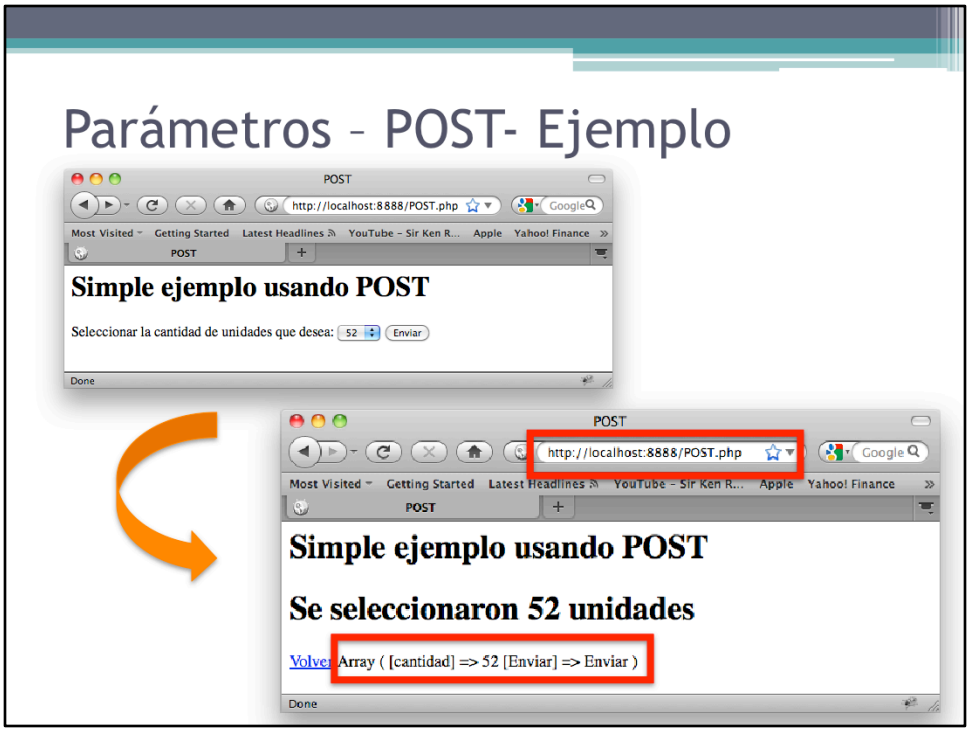
Ejemplo de SQL Injection. Utilizar los valores de los parámetros que se reciben directamente del usuario es fuente cómun de brechas en la seguridad del sistema. Estos parámetros se deben procesar, es decir, asegurarse que contienen información válida y no riesgosa, antes de utilizarlos.



La forma más común de pasar parámetros es a través del POST. Permite enviar una cantidad mucho mayor que GET y además esconde el mecanismo como se pasan los parámetros al cliente “de a pie”.

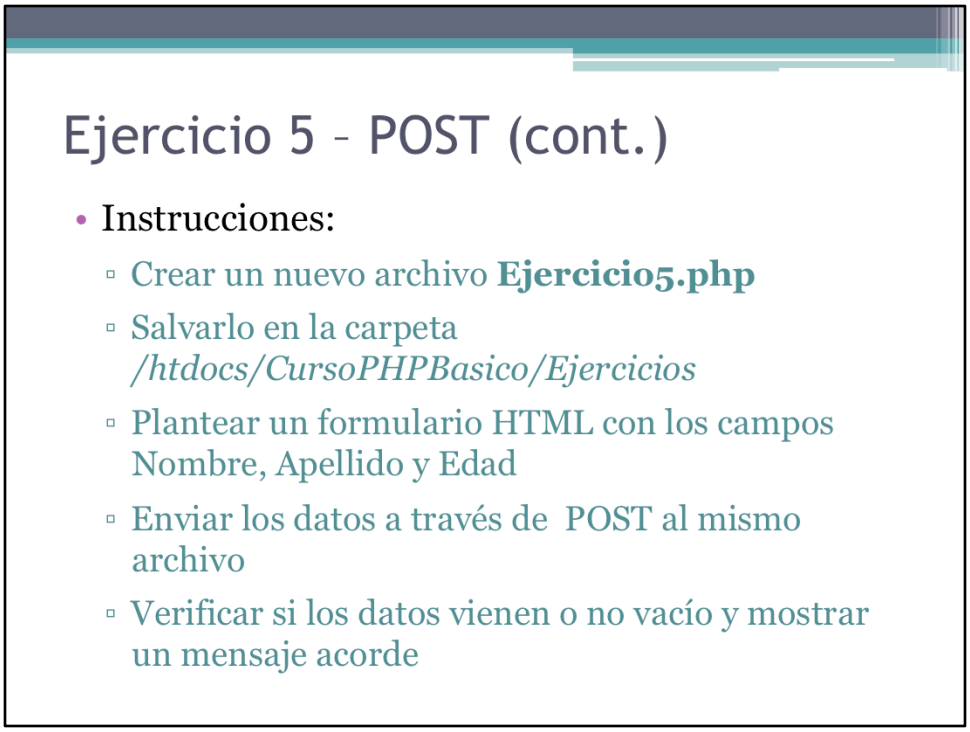
Parámetros – POST- Ejemplo

```
<html>
    <head>
        <title>POST</title>
    </head>
    <h1>Simple ejemplo usando POST</h1>
    <body>
        <?php if(isset($_POST["cantidad"])):?>
            <h1>Se seleccionaron <?=$_POST["cantidad"]?> unidades</h1>
            <a href="javascript:history.back()">Volver</a>
            <?php print_r($_POST) ?>
        <?php else:?>
            <form method="post">
                Seleccionar la cantidad de unidades que desea:
                <select name="cantidad">
                    <?php for($i=0;$i<100;$i++):?>
                        <option value="<?=$i?>"><?=$i?></option>
                    <?php endfor;?>
                </select>
                <input type="submit" name="Enviar" value="Enviar">
            </form>
        <?php endif;?>
    </body>
</html>
```


Parámetros – POST- Ejemplo
POST
http://localhost:8888/POST.php
Most Visited
Getting Started
Latest Headlines
YouTube – Sir Ken R...
Apple
Yahoo! Finance
POST
Simple ejemplo usando POST
Seleccionar la cantidad de unidades que desea:
52
Enviar
Done
POST
http://localhost:8888/POST.php
Most Visited
Getting Started
Latest Headlines
YouTube – Sir Ken R...
Apple
Yahoo! Finance
POST
Simple ejemplo usando POST
Se seleccionaron 52 unidades
Volver
Array ([cantidad] => 52 [Enviar] => Enviar)
Done

Ejercicio 5 – POST

- Duración: de 15 a 20 minutos
- Objetivos:
  - Entender el funcionamiento de un formulario HTTP
  - Manipular los parámetros recibidos por POST
- Especificaciones:
  - Hacer un formulario con los campos Nombre, Apellido y Edad.
  - Escribir un script que valide si estos campos fueron ingresados o no.

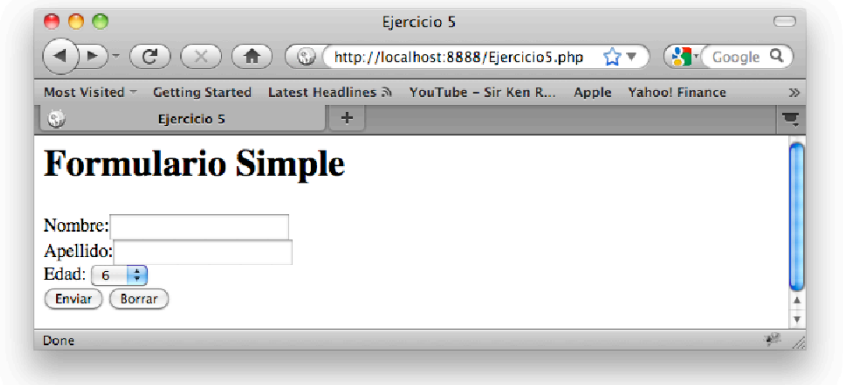


Seguir las instrucciones.

Ejercicio 5 – POST (cont.)

- Al momento de hacer el ejercicio, pregúntese:
  - ¿Existirá alguna forma más eficiente de verificar todos los campos, sin repetir código?
  - ¿Existirá algún campo para el cuál se puede generar valores predeterminados?
    - ¿De que forma es más eficiente generar estos valores predeterminandos?

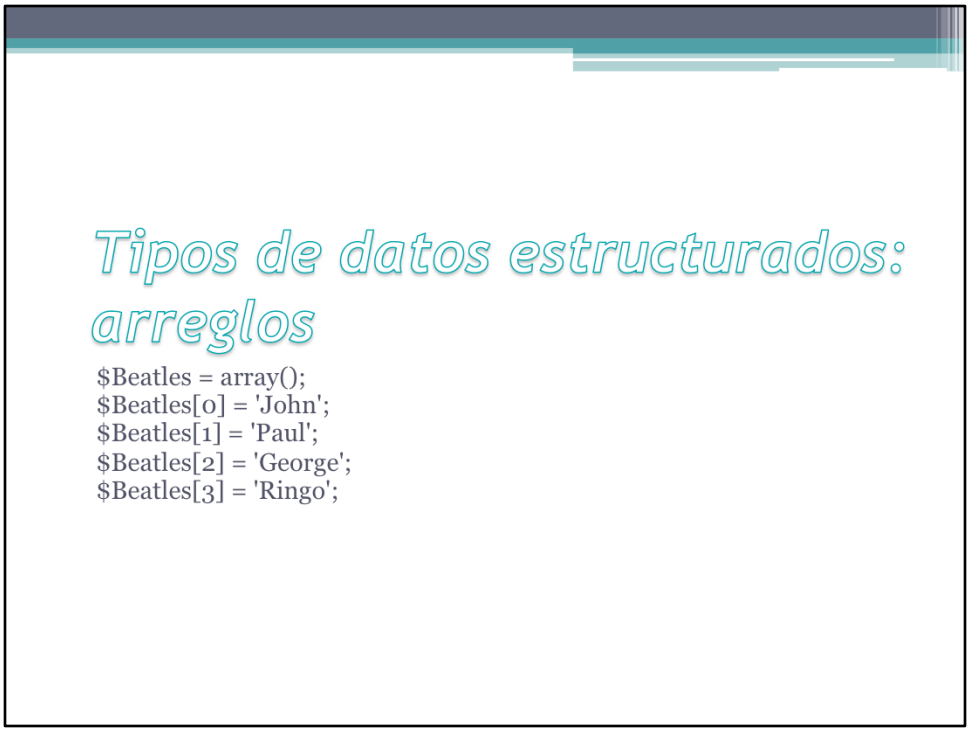
Ejercicio 5- Resultado



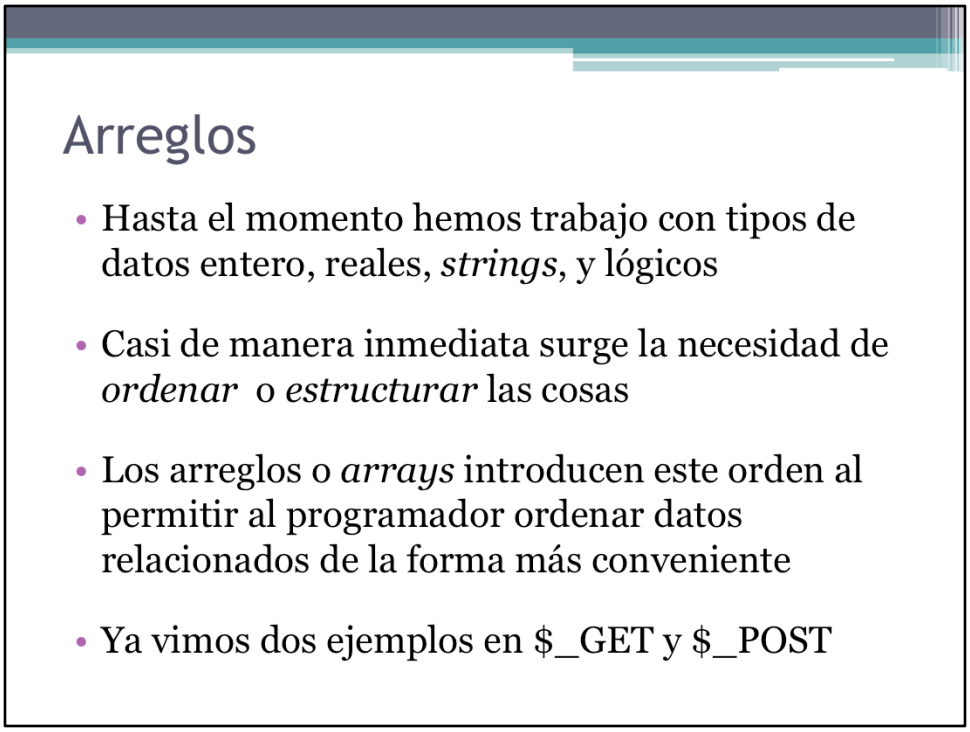
Ejercicio 5- Resultado (cont.)

Ejercicio 5- Solución

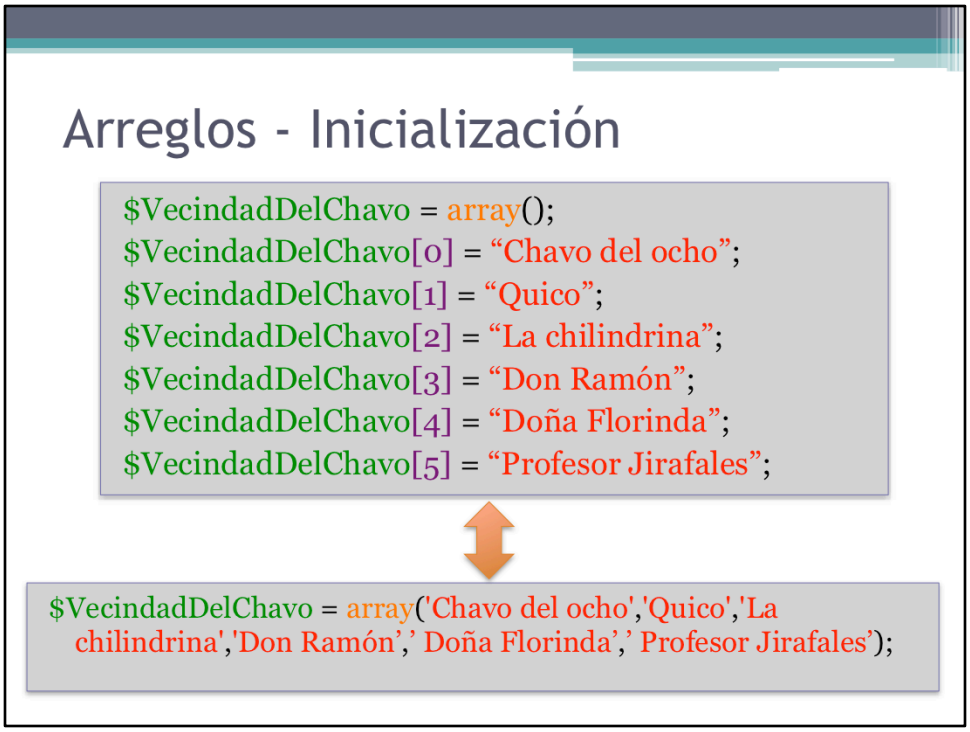
- Ver archivo Ejercicio5.php



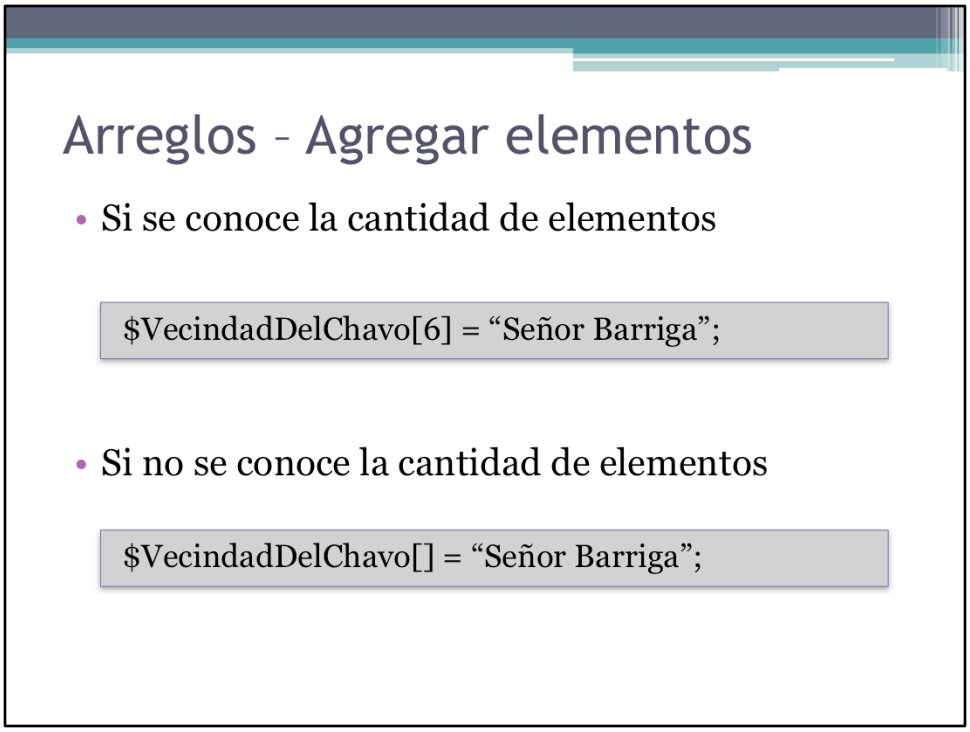
Los tipos de datos estructurados elevan el nivel de abstracción del lenguaje y permiten un uso más eficiente de los recursos del sistema, así como una mayor legibilidad y mantenimiento del código. Un nivel aún mayor de abstracción lo posee la programación orientado de objetos. En este curso sólo vamos a tratas con datos estructurados del tipo arreglos. En el curso siguiente trataremos con programación orientada a objetos en PHP.



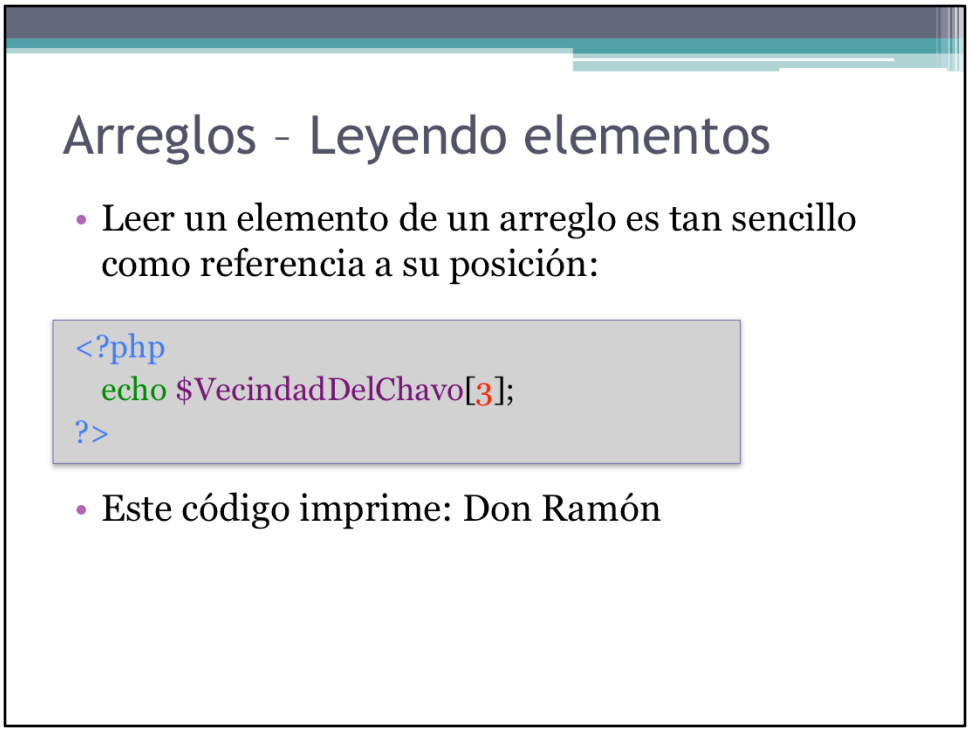
Utilizar un arreglo para ordenar los datos que provienen como parámetros del usuario ($_GET y $_POST), provee de una forma lógica de agrupar datos relacionados con un solo punto de acceso.



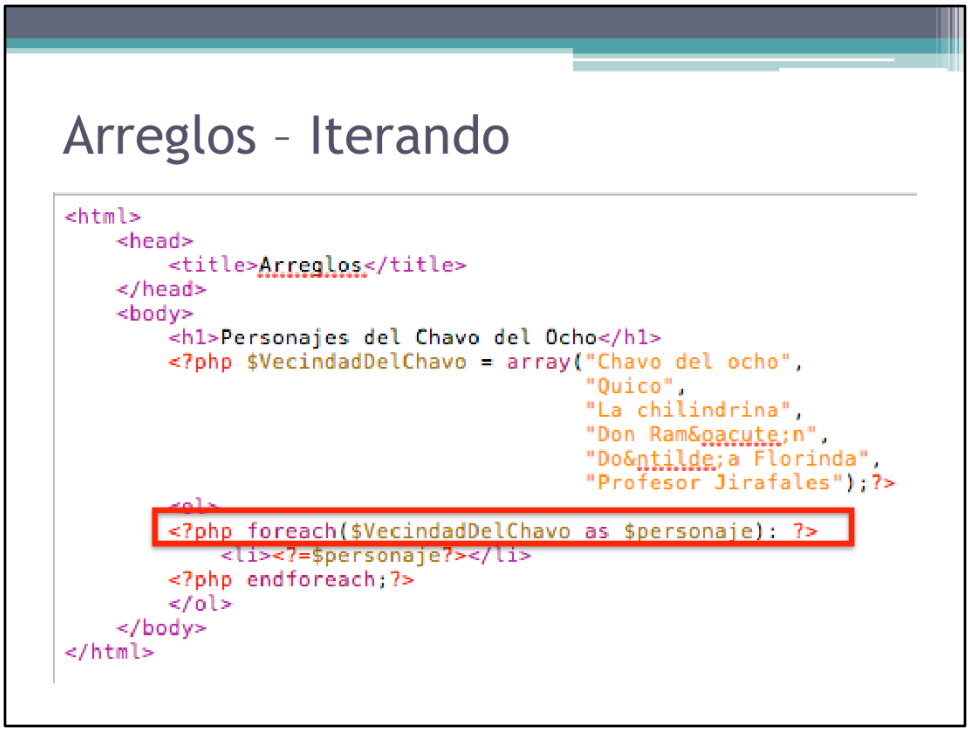
Ejemplo de organización lógica de datos en arreglos.



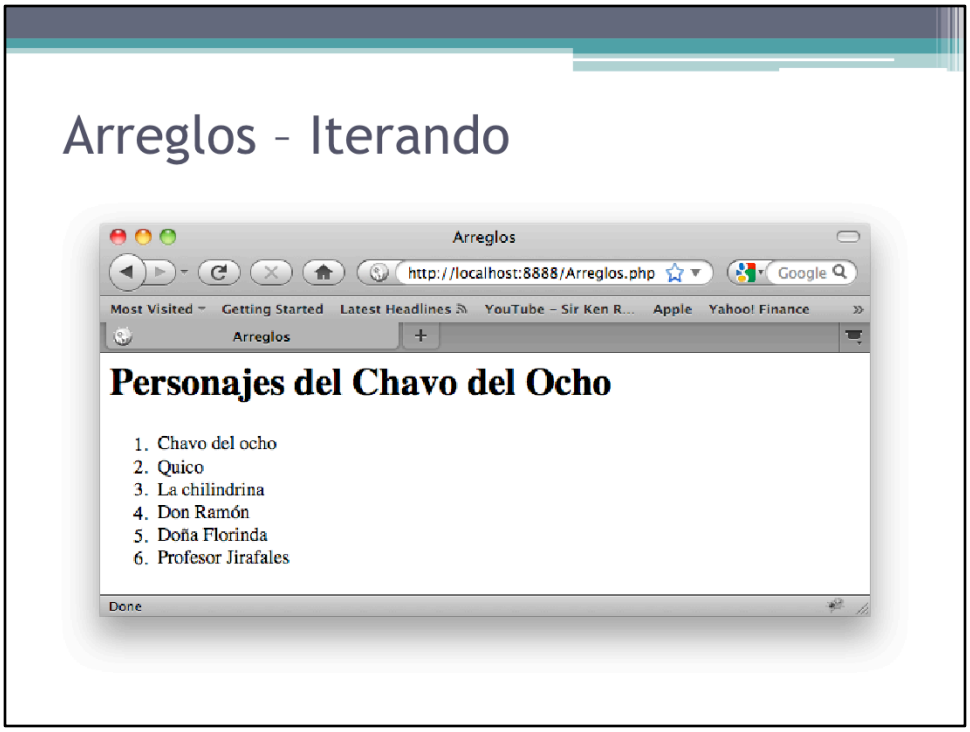
Agregar datos a un arreglo es muy sencillo, otra característica interesante de PHP.



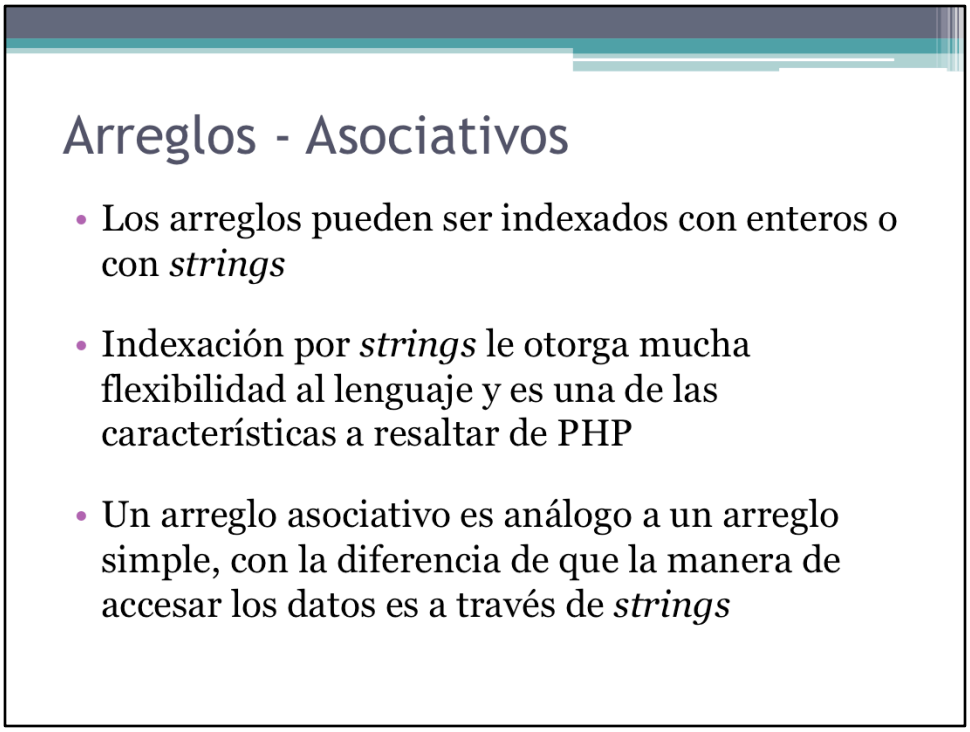
Acceder a un elemento de un arreglo es tan fácil como referenciar su posición.



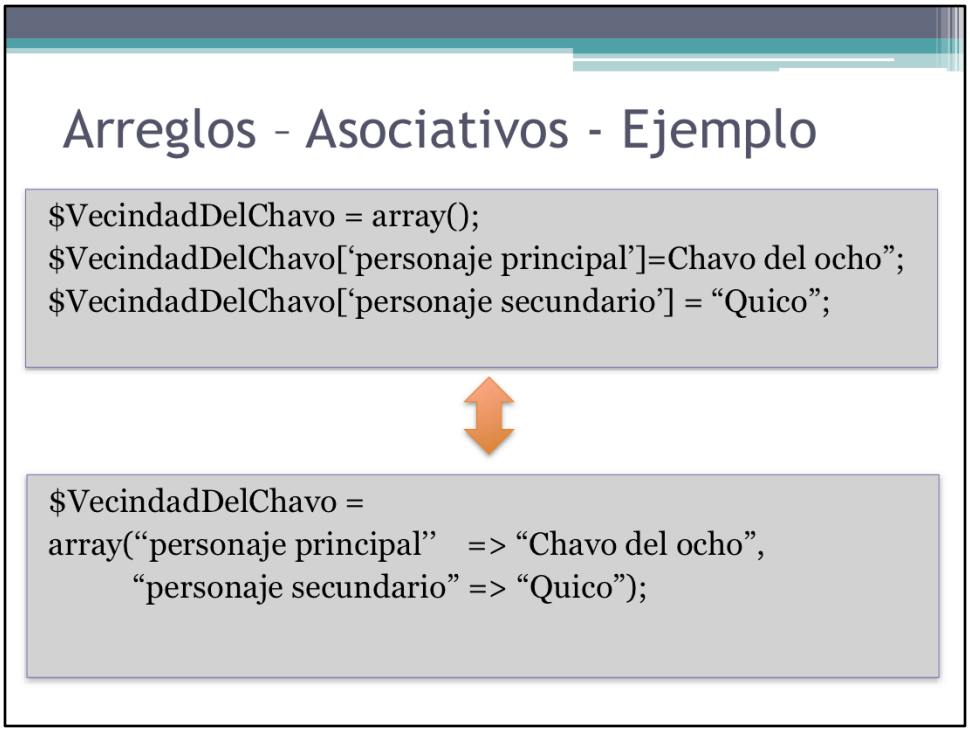
Una vez que se tienen los datos en un arreglo, con PHP se puede iterar sobre dichos datos muy cómodamente.



Ejemplo típico de datos provenientes de un arreglo: una lista de elementos.



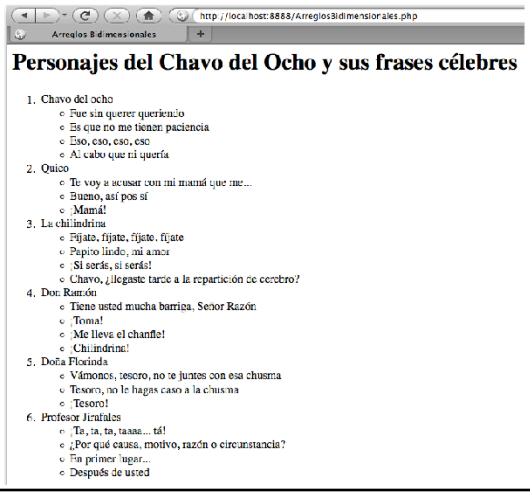
Los arreglos asociativos se asemejan más al lenguaje natural y pueden resultar más útiles en cantidad de ocasiones.



Ejemplo de arreglo asociativo.

Arreglos - Bidimensionales

- Ver archivo ArreglosBidimensionales.php

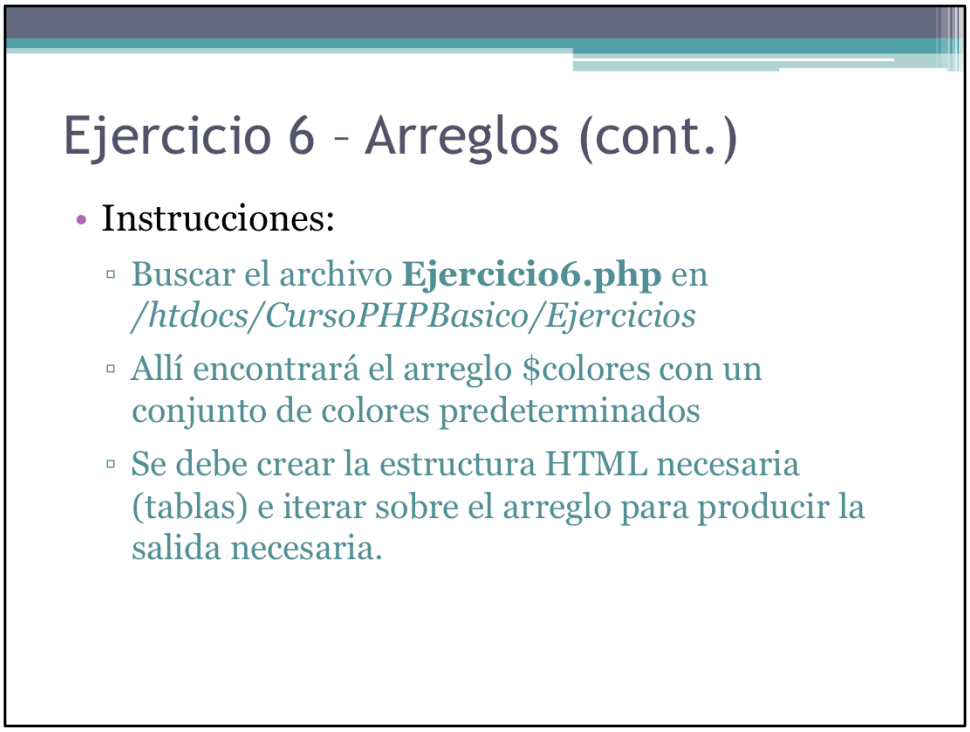


Arreglos – Funciones nativas PHP

- Para manipular arreglos existen un sin fin de funciones nativas en PHP
  - sort()
  - asort()
  - shuffle()
  - count()
  - explode()
  - Etc
- Revisar documentación
  http://php.net/manual/en language.types.array.php

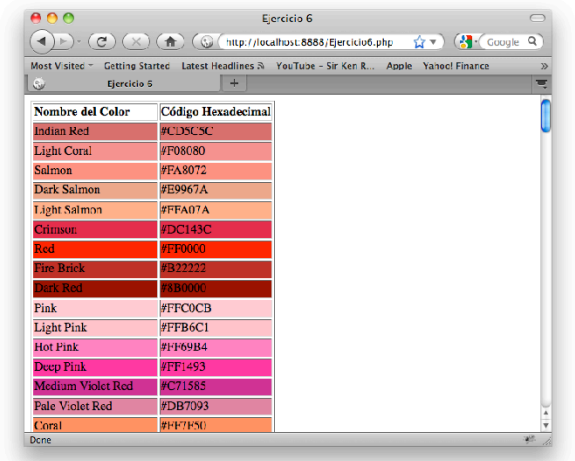
Ejercicio 6 – Arreglos

- Duración: de 15 a 20 minutos
- Objetivos:
  - Utilizar arreglos en PHP
- Especificaciones:
  - Crear una tabla en la que se listen un conjunto de colores, su nombre y código hexadecimal
  - El color fondo de cada fila debe coincidir con el color en cuestión



Seguir las instrucciones.

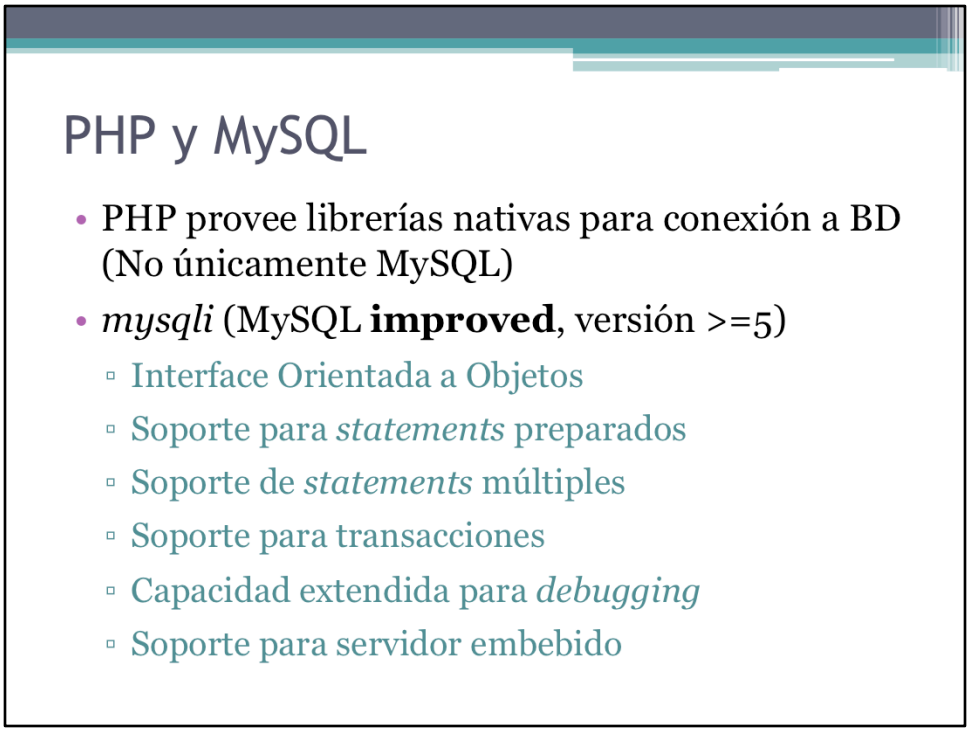
Ejercicio 6- Resultado



Ejercicio 6- Solución

```
<html>
    <head>
        <title>Ejercicio 6</title>
    </head>
    <body>
        <table border="1">
            <tr>
                <td><b>Nombre del Color</b></td>
                <td><b>C&oacute;digo Hexadecimal</b></td>
            </tr>
            <?php foreach($colores as $nombre=>$codigohex):?>
                <tr style="background-color:<?=$codigohex?>">
                    <td><?=$nombre?></td>
                    <td><?=$codigohex?></td>
                </tr>
            <?php endforeach;?>
        </table>
    </body>
</html>
```

Otro pivote sobre el que giran las aplicaciones web es el uso de base de datos. En este curso exploramos como es la interacción entre PHP y las base de datos tipo MySQL para generar aplicaciones web dinámicas.

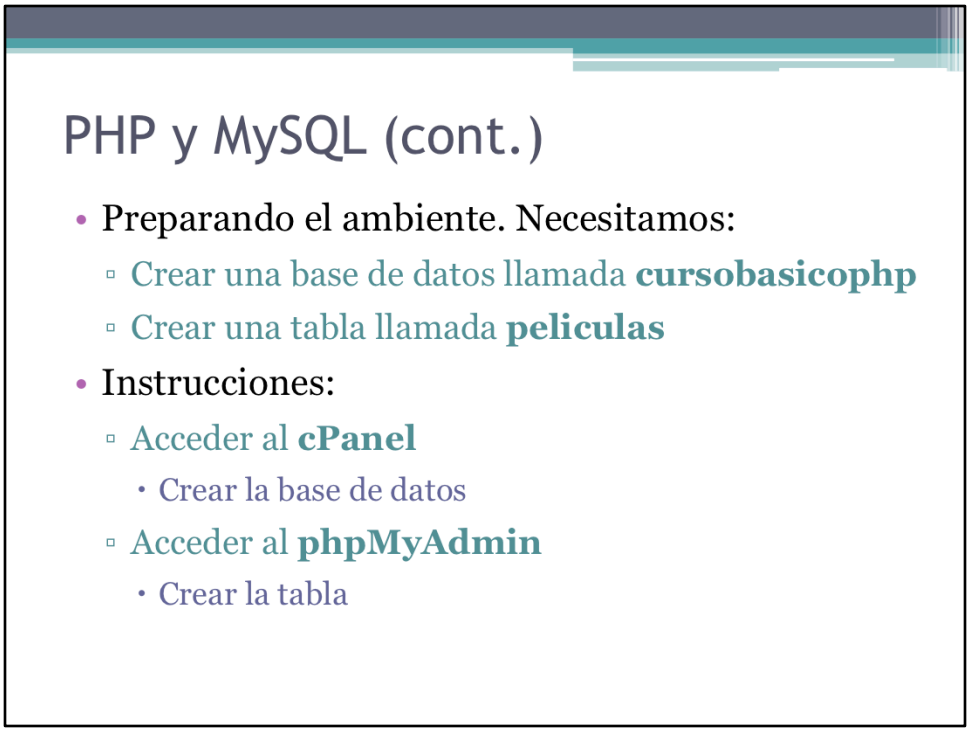


PHP posee extensiones para varios tipos de base de datos. Sin embargo, en este curso sólo utilizaremos base de datos MySQL.

En esta lámina se muestran algunas de las características de la nueva extensión de PHP para conexión con MySQL. En este curso utilizaremos mysql desde un enfoque procedimental. En cursos siguientes utilizaremos la extensión orientada a objetos.

PHP y MySQL (cont.)

- Crearemos una aplicación Web para manipular (agregar, eliminar y editar) información de películas
- Necesitaremos lo siguiente
  - Una base de datos
  - Una o más tablas dependiendo del modelo de datos
  - Uno o más scripts PHP que interactúen con la base de datos

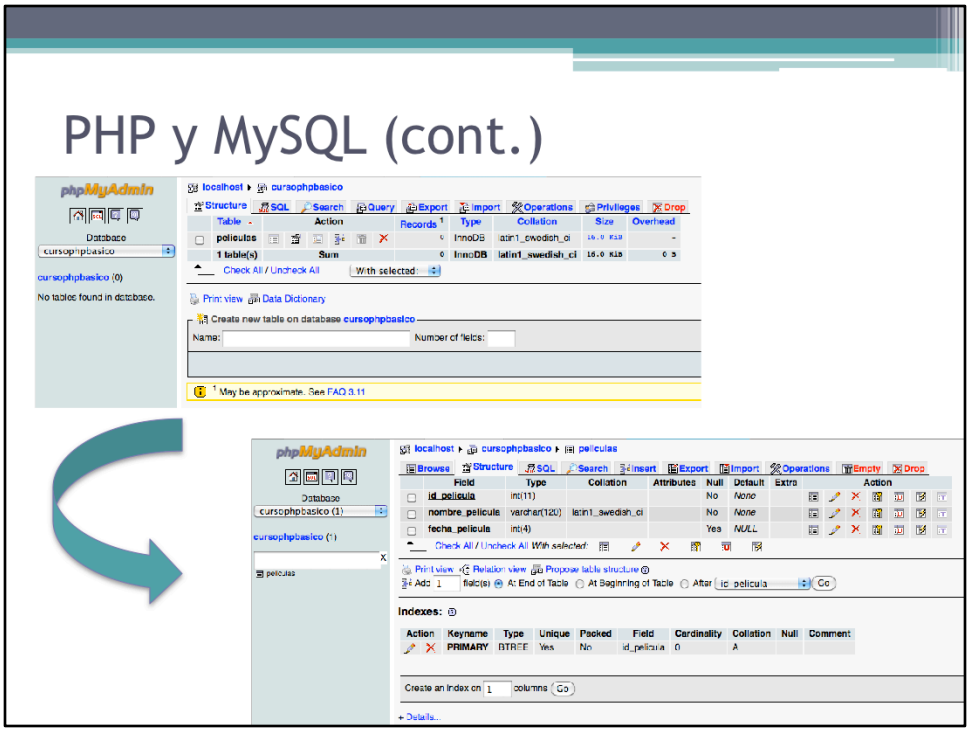


La implementación de las bases de datos está sujeta a las características del ambiente de desarrollo. Aquí se presenta un esquema común a la fecha, para sistema linux.

PHP y MySQL (cont.)

Nombre	Tipo
id_pelicula	INT (11), NOT NULL, PRIMARY
nombre_pelicula	VARCHAR (120), NOT NULL
fecha_pelicula	INT (4), NULL

Aquí se almacenará el año de estreno de la película, por ejemplo, 2004


PHP y MySQL (cont.)
phpMyAdmin
localhost ▸ cursophpbasico
Structure SQL Search Query Export Import Operations Privileges Drop
Table Action Records Type Collation Size Overhead
peliculas InnoDB latin1_swedish_ci
1 table(s) Sum InnoDB latin1_swedish_ci
Check All / Uncheck All With selected:
Print view Data Dictionary
Create new table on database cursophpbasico
Name: Number of fields:
May be approximate. See FAQ 3.11
Database
cursophpbasico (0)
No tables found in database.
localhost ▸ cursophpbasico ▸ peliculas
Browse Structure SQL Search Insert Export Import Operations Empty Drop
Field Type Collation Attributes Null Default Extra Action
id_pelicula int(11) No None
nombre_pelicula varchar(120) latin1_swedish_ci No None
fecha_pelicula int(4) Yes NULL
Check All / Uncheck All With selected:
Print view Relation view Propose table structure
Add 1 field(s) At End of Table At Beginning of Table After id_pelicula Go
Indexes:
Action Keyname Type Unique Packed Field Cardinality Collation Null Comment
PRIMARY BTREE Yes No id_pelicula 0 A
Create an index on 1 columns Go
cursophpbasico (1)
peliculas
Details...

PHP y MySQL (cont.)

- Para interactuar con MySQL desde PHP:
  - Conectarse a la base de datos
  - Enviar un *query* a la base de datos
  - Almacenar resultados obtenidos de la BD
  - Trabajar con los resultados
    - Posiblemente actualizar o eliminar datos de la BD
  - Mostrar resultados al cliente
  - Liberar la base de datos

Para chequear si la conexión fue exitosa utilizamos la función mysqli_connect_errno (), la cual devuelve un número de error en caso en que ocurra algún error o cero en caso contrario. Si ocurre un error, podemos obtener detalles del mismo a través de la función mysqli_connect_error().

PHP y MySQL – Buscar datos

```
<?php
    else:
    $Query = 'SELECT * FROM peliculas';
    $Result = mysqli_query($DB,$Query);
    $NumResults = mysqli_num_rows($Result);
    ?>
    <b><?=$NumResults?> Pel&iacute;culas</b>
```

PHP y MySQL – Mostrar los resultados

```
<table border="1">
    <tr>
        <th>Id Pel&iacute;cula</th>
        <th>Nombre</th>
        <th>A&ntilde;o</th>
    </tr>
    <?php while ($Row = mysqli_fetch_assoc($Result)):?>
        <tr>
            <td><?=$Row['id_pelicula']?></td>
            <td><?=$Row['nombre_pelicula']?></td>
            <td><?=$Row['fecha_pelicula']?></td>
        </tr>
    <?php endwhile;?>
</table>
```

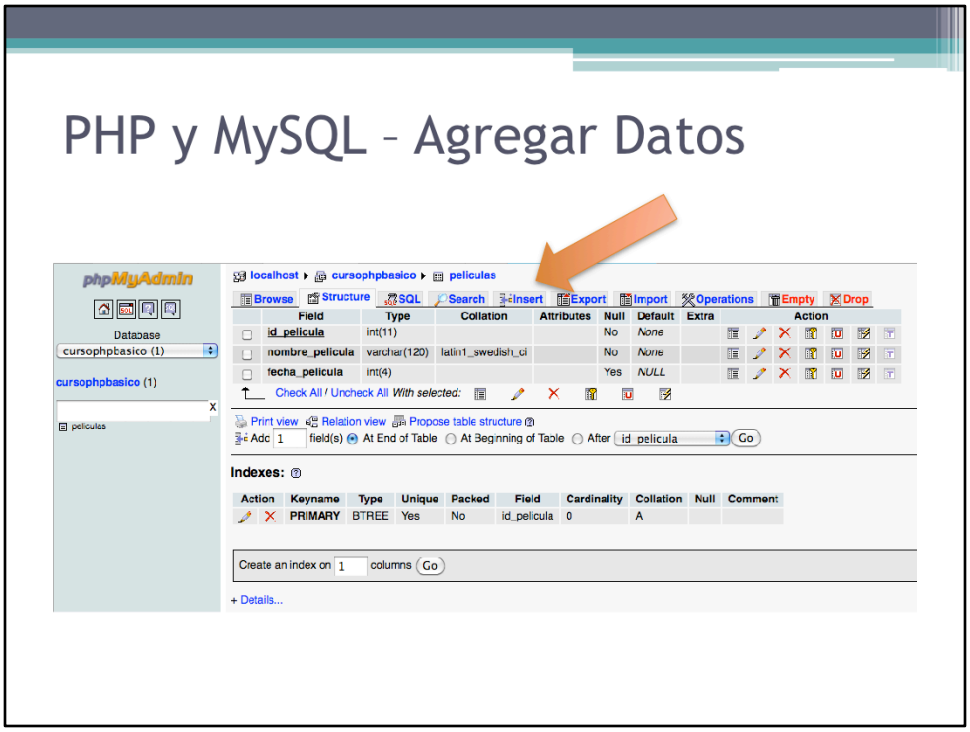
PHP y MySQL – Liberar los recursos

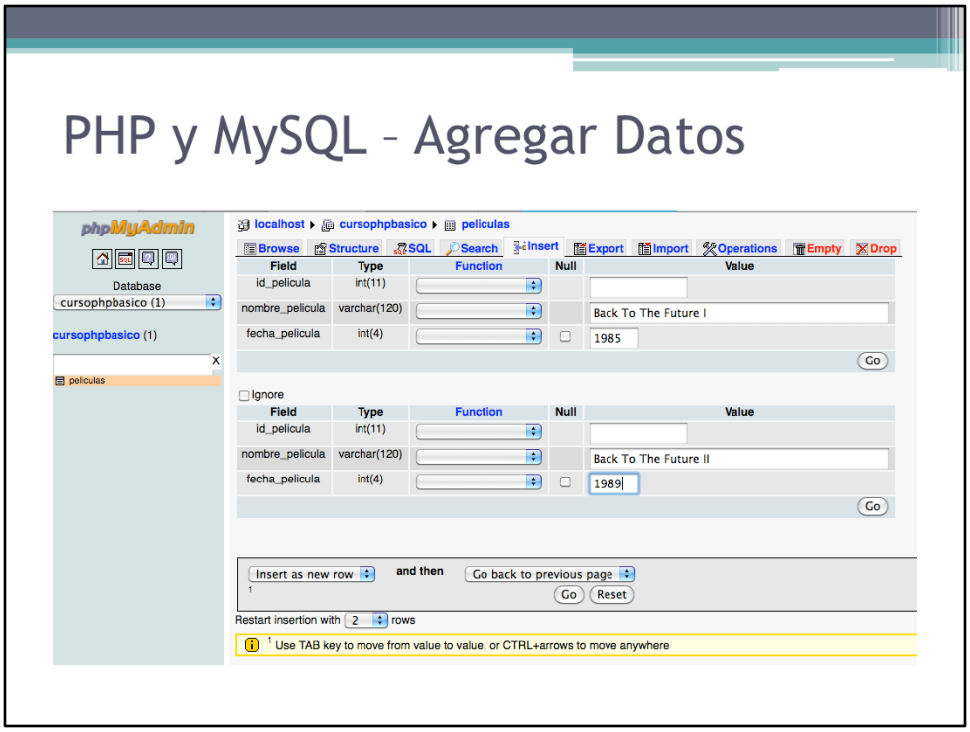
```
<?php
mysqli_free_result($Result);
mysqli_close($DB);
endif;
?>
```


PHP y MySQL – Todo Junto
<head>
<title>Películas</title>
</head>
<body>
<h1>Colección de películas</h1>
<?php
@$DB = mysqli_connect('localhost','root','root','cursophpbasico');
if (mysqli_connect_errno()):
?>
No se pudo conectar con la base de datos: <?=mysqli_connect_error();?>
<?php
else:
$Query = 'SELECT * FROM peliculas';
$Result = mysqli_query($DB,$Query);
$NumResults = mysqli_num_rows($Result);
?>
<?=$NumResults?> Películas
<table border="1">
<tr>
<th>Id Película</th>
<th>Nombre</th>
<th>Año</th>
</tr>
<?php while ($Row = mysqli_fetch_assoc($Result)):?>
<tr>
<td><?=$Row['id_pelicula']?></td>
<td><?=$Row['nombre_pelicula']?></td>
<td><?=$Row['fecha_pelicula']?></td>
</tr>
<?php endwhile;?>
</table>
<?php
mysqli_free_result($Result);
mysqli_close($DB);
endif;
?>
</body>

PHP y MySQL – Resultado...

Hay que agregar datos!


PHP y MySQL – Agregar Datos
phpMyAdmin
Database
cursophpbasico (1)
cursophpbasico (1)
peliculas
localhost ▸ cursophpbasico ▸ peliculas
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Empty
Drop
Field
Type
Collation
Attributes
Null
Default
Extra
Action
id_pelicula
int(11)
No
None
nombre_pelicula
varchar(120)
latin1_swedish_ci
No
None
fecha_pelicula
int(4)
Yes
NULL
Check All / Uncheck All With selected:
Print view
Relation view
Propose table structure
Add 1 field(s) At End of Table At Beginning of Table After id_pelicula Go
Indexes:
Action
Keyname
Type
Unique
Packed
Field
Cardinality
Collation
Null
Comment
PRIMARY
BTREE
Yes
No
id_pelicula
0
A
Create an index on 1 columns Go
+ Details...


PHP y MySQL – Agregar Datos
phpMyAdmin
Database
cursophpbasico (1)
cursophpbasico (1)
peliculas
localhost ▸ cursophpbasico ▸ peliculas
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Empty
Drop
Field
Type
Function
Null
Value
id_pelicula
int(11)
nombre_pelicula
varchar(120)
Back To The Future I
fecha_pelicula
int(4)
1985
Go
Ignore
Field
Type
Function
Null
Value
id_pelicula
int(11)
nombre_pelicula
varchar(120)
Back To The Future II
fecha_pelicula
int(4)
1989
Go
Insert as new row
and then
Go back to previous page
Go
Reset
Restart insertion with 2 rows
Use TAB key to move from value to value, or CTRL+arrows to move anywhere

PHP y MySQL – Resultado…

PHP y MySQL - Lo que falta (básico)

- Se debe crear una interfaz desde nuestra aplicación, no desde phpMyAdmin, que nos permita:
  - Agregar registros
  - Modificar registros
  - Eliminar registros

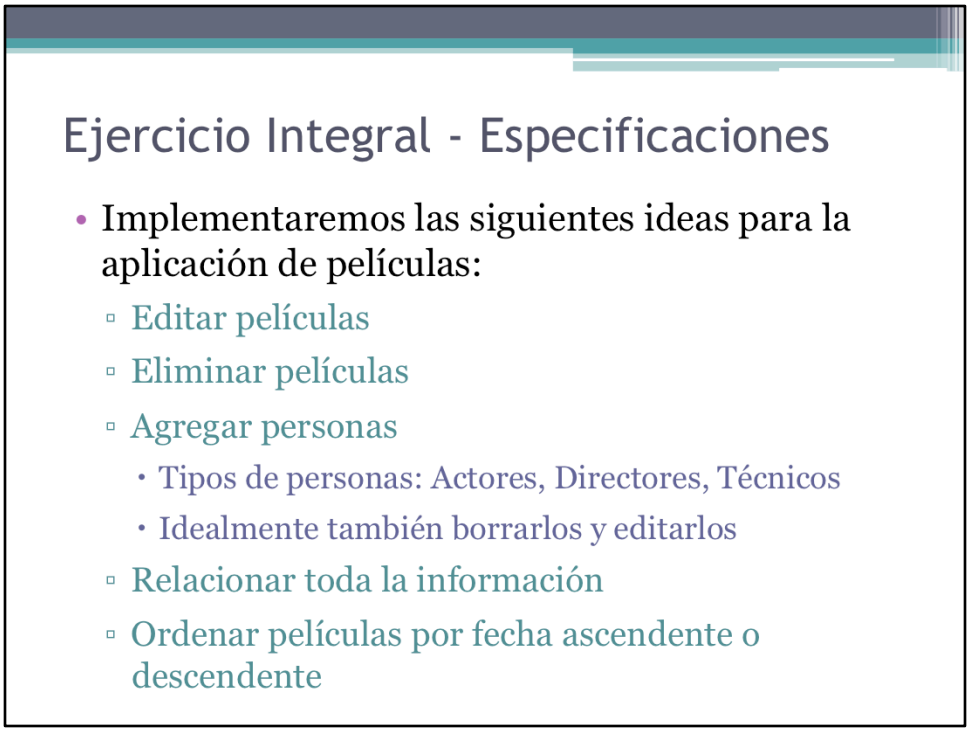
PHP y MySQL - Lo que falta…

- Ideas:
  - Dividir la aplicación en frontoffice y backoffice
    - Frontoffice: muestra información
    - Backoffice: permite editar información. Hay que autenticar al cliente.
  - Posibilidad de ordenar la lista de películas por campos
  - Directores
  - Actores
  - Premios
  - Trailers
  - Fotos
  - Comentarios y ratings de usuario
  - etc

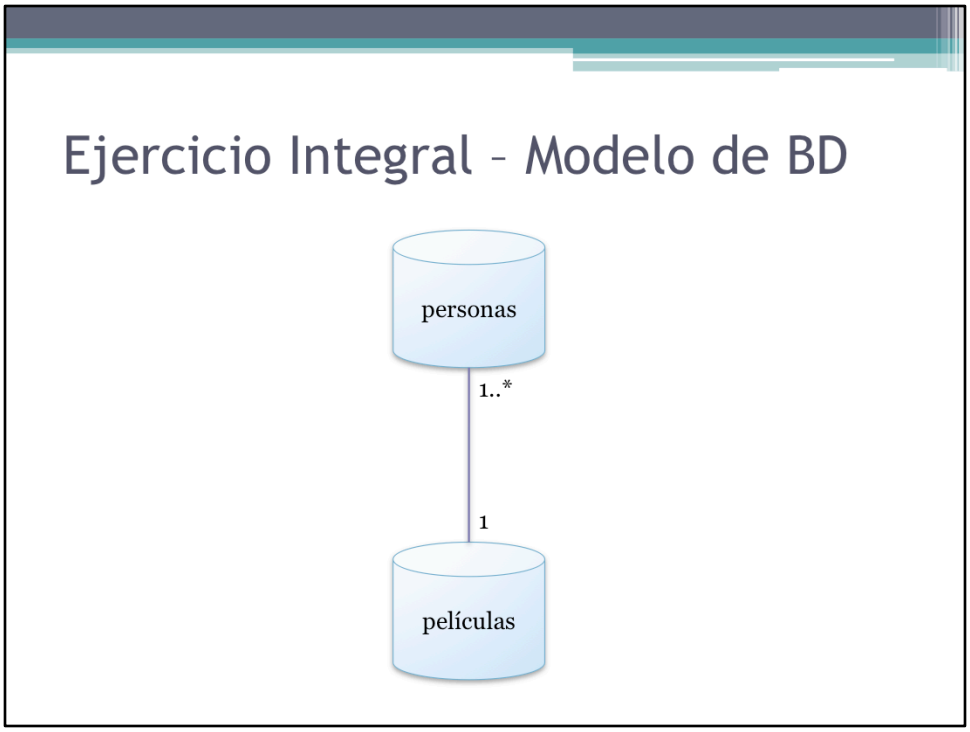
Este ejercicio, de larga duración, busca consolidar conocimientos y acercar al usuario a lo que en realidad sería el desarrollo con PHP

Ejercicio Integral - Descripción

- Duración: de 45 a 60 minutos o más
- Objetivos:
  - Consolidar lo aprendido en este curso
- Especificaciones:
  - Trabajaremos sobre nuestra aplicación de películas y la expandiremos para incorporar nuevas funcionalidades



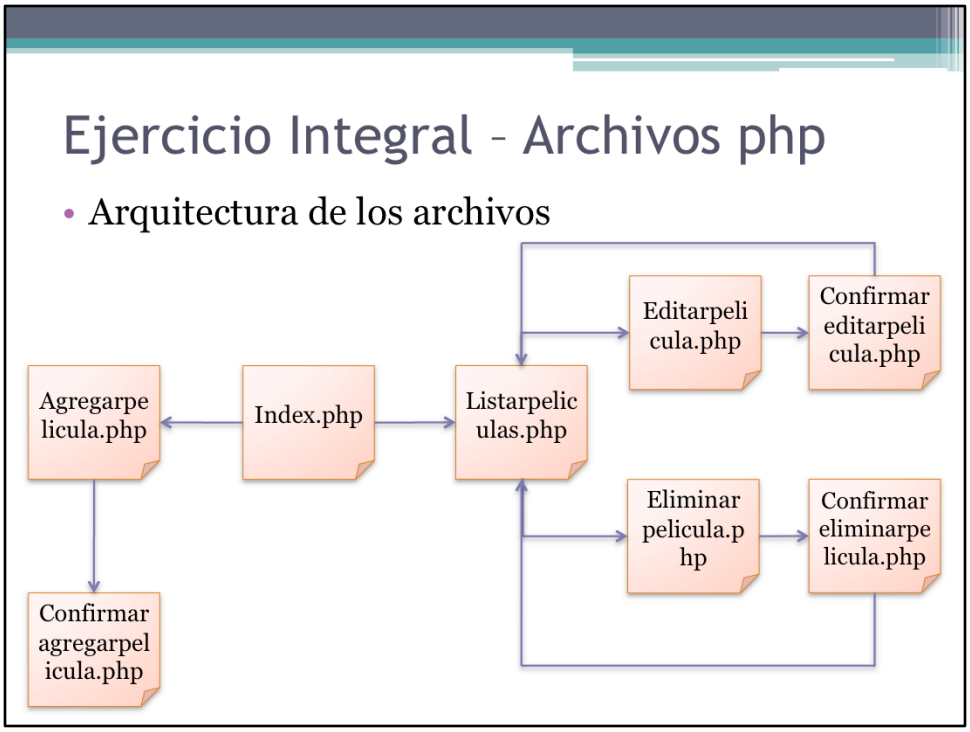
De las ideas plasmadas anteriormente vamos a implementar sólo algunas. Se deja como ejercicio al lector implementar el resto.



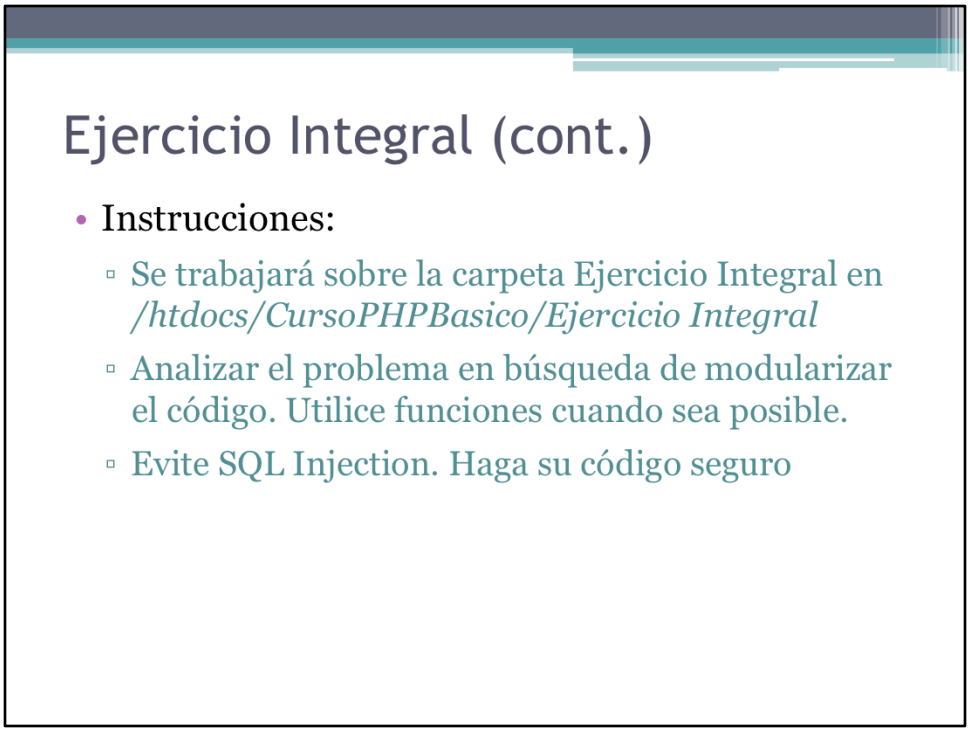
Modelo sencillo para nuestra aplicación

Ejercicio Integral – Modelo de BD

- Películas
  - Id
  - Nombre
  - Fecha
- Personas
  - Id
  - Nombre
  - Apellido
  - Fecha de Nacimiento
  - Profesión: Actor, Director, Técnico,...

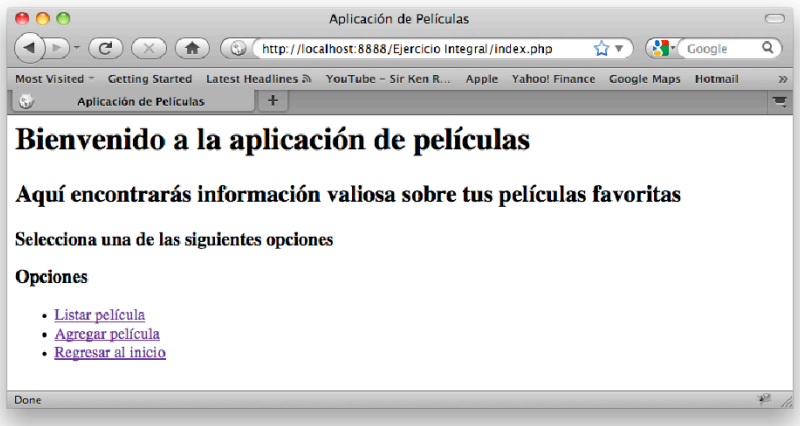


Posible planificación de la interacción y estructura de los archivos PHP



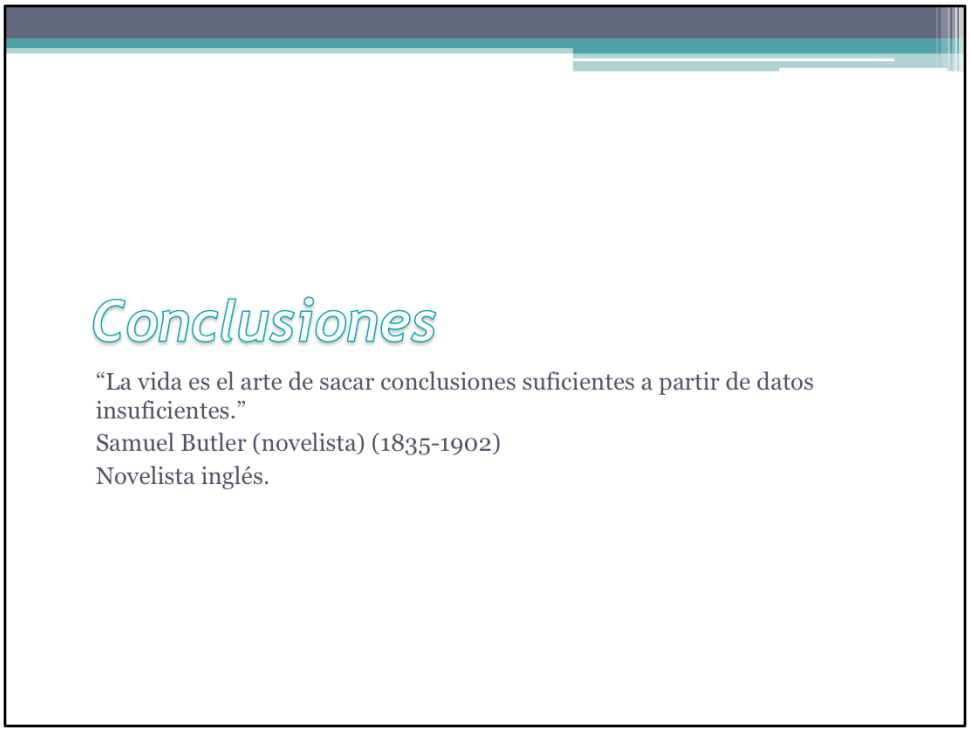
Seguir las instrucciones.

Ejercicio Integral - Resultado



Ejercicio Integral - Solución

- Ver carpeta con la solución


Conclusiones
“La vida es el arte de sacar conclusiones suficientes a partir de datos insuficientes.”
Samuel Butler (novelista) (1835-1902)
Novelista inglés.

Conclusiones

- PHP es un lenguaje flexible y versátil
- Aún con una aplicación sencilla es fácil ver el potencial para el desarrollo de aplicaciones complejas
- La flexibilidad introduce desorden estructural
- Necesitamos proveer al lenguaje de mayor estructura para facilitarnos la vida (PHP OO)