

Ambientes no propietarios

Introducción

Edwin Salvador

06 de octubre de 2015

Sesión 2

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

Contenido II

- Comentarios
- Sintaxis básica
- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

- Comentarios
- Sintaxis básica

Contenido II

- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

Temas de proyectos Bimestre 1

- Cada uno elegir una tecnología, framework, lenguaje, etc con el que tenga algo de experiencia o que desee aprender y preparar una presentación, ejercicios, clase práctica.
- Por ejemplo: Los que hayan trabajado con Wordpress, Angular y Laravel
- Grupos de tres.
- Informe escrito de máximo 10 páginas. Formato informes. Vía Turnitin.
- Presentaciones de 30 minutos, debe ser práctico, ejercicios.

- Propuestas?
- ¿Les gustaría continuar con el proyecto de datos abiertos para llegar a algo más interesante?

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

- Comentarios
- Sintaxis básica

Contenido II

- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

- Ingresar a http://peerwise.cs.auckland.ac.nz/at/?epn_ec
- Crear una cuenta. Nombre de usuario: nombre.apellido
- El correo debe ser el registrado en el SAEw.
- ID del curso **12062**

Confirmar invitación a Turnitin.

¿Interesados en Datos Abiertos?

La comunidad de datos abiertos Ecuador organiza un taller previo a la expedición de datos.

El taller será el Sábado 17 de octubre a las 10am en el BuenTrip.

La expedición de datos será el 7 de noviembre.

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

Contenido II

- Comentarios
- Sintaxis básica
- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

- ¿Qué es la WWW? *World Wide Web*

- ¿Qué es la WWW? *World Wide Web*
- URI? URL?

- ¿Qué es la WWW? *World Wide Web*
- URI? URL?
- HTTP?

- ¿Qué es la WWW? *World Wide Web*
- URI? URL?
- HTTP?
- HTML?

- ¿Qué es la WWW? *World Wide Web*
- URI? URL?
- HTTP?
- HTML?
- XML?

- ¿Qué es la WWW? *World Wide Web*
- URI? URL?
- HTTP?
- HTML?
- XML?
- **servidores web** y **navegadores web** unen el HTTP y el HTML.

- ¿Qué es la WWW? *World Wide Web*
- URI? URL?
- HTTP?
- HTML?
- XML?
- **servidores web** y **navegadores web** unen el HTTP y el HTML.
- El resultado son las páginas web, documentos HTML que se transfieren por HTTP.

- El usuario debía a daptarse a lo que se ofrecia en la página.

- El usuario debía adaptarse a lo que se ofrecía en la página.
- Contenido de Internet era plano y no llamaba mucho la atención.

Contenido estático

- El usuario debía adaptarse a lo que se ofrecía en la página.
- Contenido de Internet era plano y no llamaba mucho la atención.
- La actualización de contenidos era muy tediosa.

Contenido estático

- El usuario debía adaptarse a lo que se ofrecía en la página.
- Contenido de Internet era plano y no llamaba mucho la atención.
- La actualización de contenidos era muy tediosa.
- No existía personalización para cada usuario.

Contenido estático

- El usuario debía adaptarse a lo que se ofrecía en la página.
- Contenido de Internet era plano y no llamaba mucho la atención.
- La actualización de contenidos era muy tediosa.
- No existía personalización para cada usuario.
- ¿Alguien puede nombrar más desventajas?

- Ha cambiado completamente como utilizamos el Internet.

- Ha cambiado completamente como utilizamos el Internet.
- ¿Cuáles son los principales elementos que han permitido la revolución del contenido de Internet?

- Ha cambiado completamente como utilizamos el Internet.
- ¿Cuáles son los principales elementos que han permitido la revolución del contenido de Internet?
- Lenguajes de servidor, BDD, JS, Ajax, HTML5, CSS3.

- Ha cambiado completamente como utilizamos el Internet.
- ¿Cuáles son los principales elementos que han permitido la revolución del contenido de Internet?
- Languages de servidor, BDD, JS, Ajax, HTML5, CSS3.
- Otros elementos importantes son: los applets de Java, Flash, JScript (variante de Microsoft de JS), ActiveX, WebGL (<https://www.chromeexperiments.com/webgl>)

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

- Comentarios
- Sintaxis básica

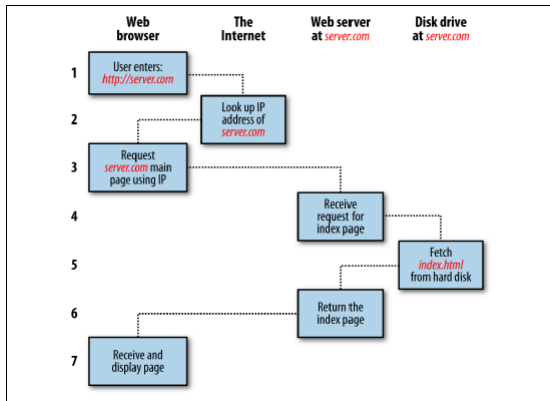
Contenido II

- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

- HTTP es un estándar de comunicación (protocolo).
- Maneja las peticiones y repuestas entre clientes y servidores (exploradores y servidores web).
- ¿Cuál es el trabajo del servidor? aceptar peticiones de clientes y responder acorde a la solicitud, recursos como páginas web, imágenes, archivos, etc.
- En el camino entre cliente y servidor normalmente tenemos otros dispositivos como routers, proxies, gateways, etc.
- Un servidor normalmente puede aceptar múltiples peticiones de varios clientes. Y pasa el tiempo escuchando por un canal de comunicación hasta recibir peticiones.

El protocolo petición/respuesta

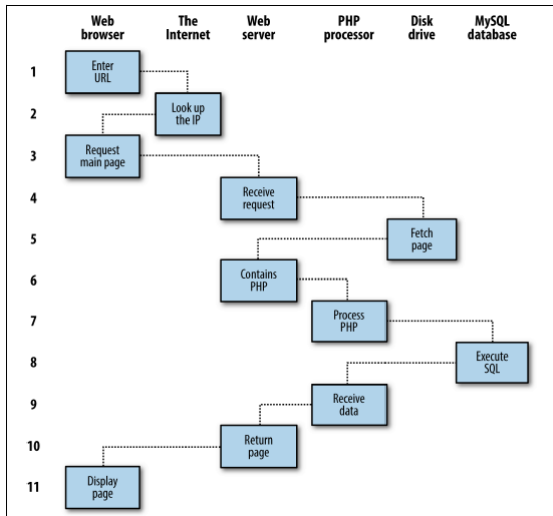
- Un explorador web realiza peticiones al servidor web que debe enviar como respuestas los recursos pedidos.
- El cliente (navegador) se encarga de desplegar el contenido recibido (la página web).



Protocolo petición/respuesta

Páginas dinámicas

Procesamiento de lenguaje del servidor y BDD.



Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- **Herramientas**
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

- Comentarios
- Sintaxis básica

Contenido II

- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

Las herramientas que utilizaremos

- De todas las tecnologías que existen o existieron dentro de la web finalmente destacaron unas pocas que lideran actualmente.
- Estas son:
 - **PHP** por su facilidad de programación, y conexiones simples a **MySQL**.
 - <https://www.similartech.com/technologies/php>
 - **JavaScript** que es esencial que cualquier página web hoy en día. Y permite modificar los **CSS** muy fácilmente.
 - **HTML** y su reciente evolución a HTML5.
 - **Ajax** que permite manipular datos y enviar peticiones al servidor en segundo plano. Sin que el usuario sepa que algo está pasando.
- La combinación de todas estas tecnologías nos provee de las herramientas suficientes para desarrollar aplicaciones web muy dinámicas.

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- **PHP**
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

- Comentarios
- Sintaxis básica

Contenido II

- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

- Nos permite introducir contenido dinámico en páginas web HTML.
- Un archivo con extensión `.php` tiene acceso automático al lenguaje.
- Desde un punto de vista del programador se podría escribir el siguiente código:

```
<?php  
echo "Hoy es " . date("l") . ".";  
?>  
Les presentamos las noticias de hoy.
```

- En el código podemos ver las etiquetas de apertura `<?php` que le dicen al servidor web que el código que sigue debe ser interpretado por el interpretador de PHP hasta la etiqueta de cierre `?>`
- El texto “Les presentamos las noticias de hoy.” es enviado al explorador como HTML.

- PHP es un lenguaje muy flexible y gracias a eso ha tenido tanto éxito.

- PHP es un lenguaje muy flexible y gracias a eso ha tenido tanto éxito.
- Se puede mezclar HTML directamente con código PHP así:

```
Hoy es <?php echo date("l"); ?>. Les presentamos las  
noticias de hoy.
```

- PHP es un lenguaje muy flexible y gracias a eso ha tenido tanto éxito.
- Se puede mezclar HTML directamente con código PHP así:

```
Hoy es <?php echo date("l"); ?>. Les presentamos las  
noticias de hoy.
```

- La hora del servidor está configurada en Europa
(<http://php.net/manual/es/timezones.america.php>)

- PHP es un lenguaje muy flexible y gracias a eso ha tenido tanto éxito.
- Se puede mezclar HTML directamente con código PHP así:

```
Hoy es <?php echo date("l"); ?>. Les presentamos las  
noticias de hoy.
```

- La hora del servidor está configurada en Europa
(<http://php.net/manual/es/timezones.america.php>)
- Existen más maneras de dar formato
<http://php.net/manual/es/function.date.php>.

- PHP es un lenguaje muy flexible y gracias a eso ha tenido tanto éxito.
- Se puede mezclar HTML directamente con código PHP así:

```
Hoy es <?php echo date("l"); ?>. Les presentamos las  
noticias de hoy.
```

- La hora del servidor está configurada en Europa (<http://php.net/manual/es/timezones.america.php>)
- Existen más maneras de dar formato <http://php.net/manual/es/function.date.php>.
- El principal beneficio de PHP es la manera en la que se integra con HTML.

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

- Comentarios
- Sintaxis básica

Contenido II

- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

- Muchos sitios web solían utilizar archivos de texto para almacenar datos (incluso usuarios y claves).
- Era difícil proteger estos archivos y eran propensos a sufrir daños por los accesos simultáneos.
- Muchos problemas cuando los archivos crecían mucho al hacer consultas complejas.
- Por este motivo las BDD relacionales se han apoderado del mercado.
- Dentro de estas BDD relacionales MySQL destaca sobretodo por ser muy potente, robusto, rápido y **gratuito**.

- Un rápido ejemplo de uso de MySQL.
- Ingresar a <http://localhost/phpmyadmin>
- Creamos una base de datos “prueba”
- Creamos una tabla llamada “usuarios” con los campos:
 - apellido (varchar 30)
 - nombre (varchar 30)
 - email (varchar 50)
- Vamos a la pestaña “SQL” y ejecutamos:
`INSERT INTO usuarios VALUES('Perez', 'Juan', 'jperez@mail.com');`
- Para buscar un dato ejecutamos:
`SELECT apellido,nombre FROM usuarios WHERE email='jperez@mail.com';`
- En los siguientes capítulos veremos más detalles sobre MySQL.

- Podemos ejecutar muchos más comandos SQL en MySQL: JOIN, ORDERBY, buscar por coincidencias parciales de texto, etc.

- Podemos ejecutar muchos más comandos SQL en MySQL: JOIN, ORDERBY, buscar por coincidencias parciales de texto, etc.
- PHP puede interactuar con MySQL directamente.

- Podemos ejecutar muchos más comandos SQL en MySQL: JOIN, ORDERBY, buscar por coincidencias parciales de texto, etc.
- PHP puede interactuar con MySQL directamente.
- Podríamos utilizar arreglos de PHP (array) para mantener los registros, modificarlos, realizar búsquedas, etc directamente desde PHP.

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- **JavaScript**
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

- Comentarios
- Sintaxis básica

Contenido II

- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

- Provee una interacción dinámica con el usuario: validación de datos como email, textos, mostrar mensajes de confirmación, etc.

- Provee una interacción dinámica con el usuario: validación de datos como email, textos, mostrar mensajes de confirmación, etc.
- Las validaciones de JS siempre deben ser verificadas en el servidor.

- Provee una interacción dinámica con el usuario: validación de datos como email, textos, mostrar mensajes de confirmación, etc.
- Las validaciones de JS siempre deben ser verificadas en el servidor.
- Con JS y CSS se pueden crear páginas muy dinámicas.

- Provee una interacción dinámica con el usuario: validación de datos como email, textos, mostrar mensajes de confirmación, etc.
- Las validaciones de JS siempre deben ser verificadas en el servidor.
- Con JS y CSS se pueden crear páginas muy dinámicas.
- La incompatibilidad entre exploradores.

- Provee una interacción dinámica con el usuario: validación de datos como email, textos, mostrar mensajes de confirmación, etc.
- Las validaciones de JS siempre deben ser verificadas en el servidor.
- Con JS y CSS se pueden crear páginas muy dinámicas.
- La incompatibilidad entre exploradores.
- Versiones antiguas aún en funcionamiento.

- Provee una interacción dinámica con el usuario: validación de datos como email, textos, mostrar mensajes de confirmación, etc.
- Las validaciones de JS siempre deben ser verificadas en el servidor.
- Con JS y CSS se pueden crear páginas muy dinámicas.
- La incompatibilidad entre exploradores.
- Versiones antiguas aún en funcionamiento.
- Estos problemas se alivian gracias a librerías como jQuery y Bootstrap.

Ejemplo JS

- Un pequeño ejemplo JS, dentro del mismo archivo `index.php` al final escribimos el siguiente código:

```
<script type="text/javascript">  
    document.write("Hoy es " + Date() );  
</script>
```

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- **Ajax**
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

- Comentarios
- Sintaxis básica

Contenido II

- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

- *Asynchronous JavaScript and XML* (aunque ya esta definición está desactualizada).

- *Asynchronous JavaScript and XML* (aunque ya esta definición está desactualizada).
- Mejora la experiencia del usuario y optimiza la comunicación con el servidor.

- *Asynchronous JavaScript and XML* (aunque ya esta definición está desactualizada).
- Mejora la experiencia del usuario y optimiza la comunicación con el servidor.
- Hoy en día se utiliza JSON, ya no es común XML con Ajax.

- *Asynchronous JavaScript and XML* (aunque ya esta definición está desactualizada).
- Mejora la experiencia del usuario y optimiza la comunicación con el servidor.
- Hoy en día se utiliza JSON, ya no es común XML con Ajax.
- Ajax a sido clave para el paso a la Web 2.0, donde las páginas web se han vuelto prácticamente programas independientes.

- *Asynchronous JavaScript and XML* (aunque ya esta definición está desactualizada).
- Mejora la experiencia del usuario y optimiza la comunicación con el servidor.
- Hoy en día se utiliza JSON, ya no es común XML con Ajax.
- Ajax a sido clave para el paso a la Web 2.0, donde las páginas web se han vuelto prácticamente programas independientes.
- Una llamada Ajax es capaz de cargar contenido y actualizar un solo elemento de la página web como un div, tabla, p, a, etc. Ej. cargar una foto, reemplazar un botón, mostrar una barra de progreso, etc.

- *Asynchronous JavaScript and XML* (aunque ya esta definición está desactualizada).
- Mejora la experiencia del usuario y optimiza la comunicación con el servidor.
- Hoy en día se utiliza JSON, ya no es común XML con Ajax.
- Ajax a sido clave para el paso a la Web 2.0, donde las páginas web se han vuelto prácticamente programas independientes.
- Una llamada Ajax es capaz de cargar contenido y actualizar un solo elemento de la página web como un div, tabla, p, a, etc. Ej. cargar una foto, reemplazar un botón, mostrar una barra de progreso, etc.
- **Mostrar ejemplo** HTML, PHP, jQuery, Ajax.

- *Asynchronous JavaScript and XML* (aunque ya esta definición está desactualizada).
- Mejora la experiencia del usuario y optimiza la comunicación con el servidor.
- Hoy en día se utiliza JSON, ya no es común XML con Ajax.
- Ajax a sido clave para el paso a la Web 2.0, donde las páginas web se han vuelto prácticamente programas independientes.
- Una llamada Ajax es capaz de cargar contenido y actualizar un solo elemento de la página web como un div, tabla, p, a, etc. Ej. cargar una foto, reemplazar un botón, mostrar una barra de progreso, etc.
- **Mostrar ejemplo** HTML, PHP, jQuery, Ajax.
- CheatSheet jQuery: <http://oscarotero.com/jquery/>

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

- Comentarios
- Sintaxis básica

Contenido II

- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

- El desarrollo del estándar CSS3 ha permitido que los CSS ofrezcan un mayor nivel de interactividad que antes era posible solo con JS.

- El desarrollo del estándar CSS3 ha permitido que los CSS ofrezcan un mayor nivel de interactividad que antes era posible solo con JS.
- CSS3 ofrece no solo la opción de cambiar tamaño, color, bordes, espaciado si no que permite realizar animaciones de transiciones y transformaciones a los elementos.

- El desarrollo del estándar CSS3 ha permitido que los CSS ofrezcan un mayor nivel de interactividad que antes era posible solo con JS.
- CSS3 ofrece no solo la opción de cambiar tamaño, color, bordes, espaciado si no que permite realizar animaciones de transiciones y transformaciones a los elementos.
- Un ejemplo de CSS (podemos añadir un elemento <p> a nuestro index.php) y añadir el siguiente código al inicio:

```
<style>
  p {
    text-align: justify;
    font-family: Helvetica;
    color: red;
  }
</style>
```

- Los CSS pueden escribirse directamente en las etiquetas HTML en el atributo `style`

- Los CSS pueden escribirse directamente en las etiquetas HTML en el atributo `style`
- Es recomendable escribir un archivo aparte con las reglas de estilo e incluir el archivo en la cabecera del HTML.

- Los CSS pueden escribirse directamente en las etiquetas HTML en el atributo `style`
- Es recomendable escribir un archivo aparte con las reglas de estilo e incluir el archivo en la cabecera del HTML.
- Con CSS se pueden manejar eventos del mouse para cambiar los estilos de los elementos (`:hover`).

- Los CSS pueden escribirse directamente en las etiquetas HTML en el atributo `style`
- Es recomendable escribir un archivo aparte con las reglas de estilo e incluir el archivo en la cabecera del HTML.
- Con CSS se pueden manejar eventos del mouse para cambiar los estilos de los elementos (`:hover`).
- Se puede modificar los estilos CSS a través de JS.

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- **HTML5**
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

- Comentarios
- Sintaxis básica

Contenido II

- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

- Aunque todos los estándares y elementos que hemos mencionado son de gran utilidad, no ha bastado para las exigencias de los usuarios.

- Aunque todos los estándares y elementos que hemos mencionado son de gran utilidad, no ha bastado para las exigencias de los usuarios.
- Se necesitaban mayor facilidad para manipular gráficos en un explorador sin necesidad de instalar plug-ins como Flash.

- Aunque todos los estándares y elementos que hemos mencionado son de gran utilidad, no ha bastado para las exigencias de los usuarios.
- Se necesitaban mayor facilidad para manipular gráficos en un explorador sin necesidad de instalar plug-ins como Flash.
- Otro problema era la inserción de audio y video en las páginas web.

- Aunque todos los estándares y elementos que hemos mencionado son de gran utilidad, no ha bastado para las exigencias de los usuarios.
- Se necesitaban mayor facilidad para manipular gráficos en un explorador sin necesidad de instalar plug-ins como Flash.
- Otro problema era la inserción de audio y video en las páginas web.
- Todos estos problemas han llevado a una evolución del HTML.

- Aunque todos los estándares y elementos que hemos mencionado son de gran utilidad, no ha bastado para las exigencias de los usuarios.
- Se necesitaban mayor facilidad para manipular gráficos en un explorador sin necesidad de instalar plug-ins como Flash.
- Otro problema era la inserción de audio y video en las páginas web.
- Todos estos problemas han llevado a una evolución del HTML.
- El desarrollo de HTML5 empezó en 2004 y fue liderado por Mozilla Foundation y Opera Software.

- Aunque todos los estándares y elementos que hemos mencionado son de gran utilidad, no ha bastado para las exigencias de los usuarios.
- Se necesitaban mayor facilidad para manipular gráficos en un explorador sin necesidad de instalar plug-ins como Flash.
- Otro problema era la inserción de audio y video en las páginas web.
- Todos estos problemas han llevado a una evolución del HTML.
- El desarrollo de HTML5 empezó en 2004 y fue liderado por Mozilla Foundation y Opera Software.
- Recién en 2013 se emitió un documento final a la W3C (el encargado de controlar los estándares de la web).

- Aunque todos los estándares y elementos que hemos mencionado son de gran utilidad, no ha bastado para las exigencias de los usuarios.
- Se necesitaban mayor facilidad para manipular gráficos en un explorador sin necesidad de instalar plug-ins como Flash.
- Otro problema era la inserción de audio y video en las páginas web.
- Todos estos problemas han llevado a una evolución del HTML.
- El desarrollo de HTML5 empezó en 2004 y fue liderado por Mozilla Foundation y Opera Software.
- Recién en 2013 se emitió un documento final a la W3C (el encargado de controlar los estándares de la web).
- El desarrollo de nuevos estándares y mejoras de las especificaciones es continuo y es un ciclo sin fin. Pero por ahora todos los sitios web deben ser compatibles con HTML5.

Novedades en HTML5

- *Markup* nuevos elementos como `<nav>` y `<footer>` y elementos obsoletos como `` y `<center>`.

Novedades en HTML5

- *Markup* nuevos elementos como `<nav>` y `<footer>` y elementos obsoletos como `` y `<center>`.
- *Nuevas APIs* como `<canvas>` (para escribir y dibujar), `<audio>` y `<video>`, aplicaciones web offline, microdata y almacenamiento local.

Novedades en HTML5

- *Markup* nuevos elementos como `<nav>` y `<footer>` y elementos obsoletos como `` y `<center>`.
- *Nuevas APIs* como `<canvas>` (para escribir y dibujar), `<audio>` y `<video>`, aplicaciones web offline, microdata y almacenamiento local.
- *Aplicaciones* incluye nuevas tecnologías de renderizado: MathML (Math Markup Language) para fórmulas y SVG (Scalable Vector Graphics) para crear elementos fuera del `<canvas>`.

Novedades en HTML5

- *Markup* nuevos elementos como `<nav>` y `<footer>` y elementos obsoletos como `` y `<center>`.
- *Nuevas APIs* como `<canvas>` (para escribir y dibujar), `<audio>` y `<video>`, aplicaciones web offline, microdata y almacenamiento local.
- *Aplicaciones* incluye nuevas tecnologías de renderizado: MathML (Math Markup Language) para fórmulas y SVG (Scalable Vector Graphics) para crear elementos fuera del `<canvas>`.
- Con HTML5 ya no es necesario utilizar las etiquetas de compatibilidad con XHTML. Es decir los elementos que no tienen etiquetas de cerrado como el `
` ya no necesitan escribirse así: `
`.

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

- Comentarios
- Sintaxis básica

Contenido II

- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

Otras características de Apache

- Apache no solamente sirve como un servidor web que provee recursos HTML.

Otras características de Apache

- Apache no solamente sirve como un servidor web que provee recursos HTML.
- Es capaz de manejar documentos como imágenes, archivos Flash, MP3, RSS, etc.

Otras características de Apache

- Apache no solamente sirve como un servidor web que provee recursos HTML.
- Es capaz de manejar documentos como imágenes, archivos Flash, MP3, RSS, etc.
- Apache y PHP proporcionan también módulos precompilados que pueden ser llamados en tiempo de ejecución para generar imágenes u otros archivos. Ej: GD (Graphics Draw).

Otras características de Apache

- Apache no solamente sirve como un servidor web que provee recursos HTML.
- Es capaz de manejar documentos como imágenes, archivos Flash, MP3, RSS, etc.
- Apache y PHP proporcionan también módulos precompilados que pueden ser llamados en tiempo de ejecución para generar imágenes u otros archivos. Ej: GD (Graphics Draw).
- Módulos importantes que proporciona PHP es el de seguridad, Rewrite (opción para el servidor web de manejar tipos de URL y reescribirlos a un estándar interno).

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

Contenido II

- Comentarios
- Sintaxis básica
- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

Deber Consulta

El software libre

- Herramientas como PHP, MySQL y Apache son los más populares dentro de sus categorías.
- Cabe destacar que son de código abierto y han sido desarrollados por comunidades de programadores.
- ¿Cuál es la principal ventaja del código abierto?
- Consultar sobre las ventajas y desventajas del software libre, quienes son sus principales promotores y sus principales productos en el mercado.
- Máximo 3 hojas según el formato de deberes publicado.
- Entrega Martes 13 de octubre via Turnitin.

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

Contenido II

- Comentarios
- Sintaxis básica
- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

Ejercicio

- Escribir un script PHP que genere la siguiente tabla de numeros en HTML, indicando con un color diferente los números primos:

TABLA 100 PRIMEROS NÚMEROS NATURALES									
NÚMEROS PRIMOS EN ROJO									
0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

Contenido II

- Comentarios
- Sintaxis básica
- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

Introducción a PHP

- El servidor automáticamente pasa los archivos `.php` al interprete de PHP.

Introducción a PHP

- El servidor automáticamente pasa los archivos `.php` al interprete de PHP.
- Para empezar a escribir código PHP necesitamos la etiqueta: `<?php`

Introducción a PHP

- El servidor automáticamente pasa los archivos `.php` al interprete de PHP.
- Para empezar a escribir código PHP necesitamos la etiqueta: `<?php`
- Para terminar de escribir PHP usamos la etiqueta de cierre `?>`.

Introducción a PHP

- El servidor automáticamente pasa los archivos .php al interprete de PHP.
- Para empezar a escribir código PHP necesitamos la etiqueta: <?php
- Para terminar de escribir PHP usamos la etiqueta de cierre ?>.
- Ejemplo:

```
<?php  
    echo "Hello world";  
?>
```

Variación de etiquetas

- Ocasionalmente encontraremos (en Internet) que el código PHP está dentro de las etiquetas <? ?>

```
<?  
    echo "Hello world";  
?>
```

Variación de etiquetas

- Ocasionalmente encontraremos (en Internet) que el código PHP está dentro de las etiquetas `<? ?>`

```
<?  
    echo "Hello world";  
?>
```

- Aunque son aceptadas y funcionan, no se recomiendan por incompatibilidad. Esta opción será removida en futuras versiones de PHP.

Etiqueta de cierre es opcional

- Cuando tenemos solamente código PHP en nuestro archivo, se puede omitir la etiqueta de cierre `?>`.

Etiqueta de cierre es opcional

- Cuando tenemos solamente código PHP en nuestro archivo, se puede omitir la etiqueta de cierre `?>`.
- Es es incluso recomendado y una buena práctica ya que nos aseguramos que no existen espacios es blanco después del código PHP lo que generaría errores cuando se utiliza código orientado a objetos.

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

- Comentarios
- Sintaxis básica

Contenido II

- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

- Existen dos formas de añadir comentarios:

`|| // Este un comentario de una linea`

- Existen dos formas de añadir comentarios:

```
|| // Este un comentario de una linea
```

- Útil para comentar lineas específicas de código:

```
|| // echo "X es igual a $x";
```

- Existen dos formas de añadir comentarios:

```
|| // Este un comentario de una linea
```

- Útil para comentar líneas específicas de código:

```
|| // echo "X es igual a $x";
```

- O para añadir información sobre una línea de código:

```
|| $x += 10; // Incrementar $x por 10
```

- Existen dos formas de añadir comentarios:

```
|| // Este un comentario de una linea
```

- Útil para comentar líneas específicas de código:

```
|| // echo "X es igual a $x";
```

- O para añadir información sobre una línea de código:

```
|| $x += 10; // Incrementar $x por 10
```

- Otra manera de comentar es con `/* */`

```
|| <?php
|| /* Esta es una seccion
||    de varias lineas
||    que esta comentada */
|| ?>
```


- Existen dos formas de añadir comentarios:

```
// Este un comentario de una linea
```

- Útil para comentar líneas específicas de código:

```
// echo "X es igual a $x";
```

- O para añadir información sobre una línea de código:

```
$x += 10; // Incrementar $x por 10
```

- Otra manera de comentar es con `/* */`

```
<?php
/* Esta es una seccion
   de varias lineas
   que esta comentada */
?>
```

- Común para comentar grandes fragmentos de código.

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

- Comentarios
- Sintaxis básica

Contenido II

- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

- PHP es un lenguaje simple, tiene sus raíces en C y Perl aunque se ve más como Java.

Sintaxis básica

- PHP es un lenguaje simple, tiene sus raíces en C y Perl aunque se ve más como Java.
- Es flexible, pero se deben seguir algunas reglas de sintaxis y estructura.

Sintaxis básica

- PHP es un lenguaje simple, tiene sus raíces en C y Perl aunque se ve más como Java.
- Es flexible, pero se deben seguir algunas reglas de sintaxis y estructura.
- Toda sentencia termina con (;). Ej.

```
|| $x += 10;
```

- PHP es un lenguaje simple, tiene sus raíces en C y Perl aunque se ve más como Java.
- Es flexible, pero se deben seguir algunas reglas de sintaxis y estructura.
- Toda sentencia termina con (;). Ej.

```
|| $x += 10;
```

- Cuando no se termina con ; PHP interpretará todo como una sola sentencia. Al no ser entendido mostrará un mensaje Parse error.

El símbolo \$

- Este símbolo (\$) se pone antes de cada variable.

El símbolo \$

- Este símbolo (\$) se pone antes de cada variable.
- Permite interpretar el lenguaje más rápido.

```
<?php
    $mycounter = 1;
    $mystring = "Hello";
    $myarray = array("One", "Two", "Three");
?>
```

Recomendaciones

- Usar indentación en el código.

Recomendaciones

- Usar indentación en el código.
- Los espacios en blanco hacen que el código sea más legible. Ej.

```
<?php
if ($x <= 10) {
    $a=3;
    $b=4;
    $i++;
}
?>
```

```
<?php
if( $x  <=  10 ){
    $a = 3;
    $b = 4;
    $i++;
}
?>
```

Ejemplo PHP

Crear el archivo prueba1.php, escribir el siguiente programa y visualizarlo en el navegador:

```
<?php // test1.php
    $username = "Fred Smith";
    echo $username;
    echo "<br>";
    $current_user = $username;
    echo $current_user;
?>
```

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

- Comentarios
- Sintaxis básica

Contenido II

- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

Variables String

- `$username = "Fred Smith";`

Variables String

- `|| $username = "Fred Smith";`

- Las comillas ("") nos indican que estamos almacenando un string (cadena de caracteres). Se puede utilizar ("") o (""). La diferencia la veremos luego.

Variables String

- `|| $username = "Fred Smith";`

- Las comillas ("") nos indican que estamos almacenando un string (cadena de caracteres). Se puede utilizar ("") o (""). La diferencia la veremos luego.
- Para ver el contenido de la variable:

```
|| echo $username;
```

Variables String

- `|| $username = "Fred Smith";`

- Las comillas ("") nos indican que estamos almacenando un string (cadena de caracteres). Se puede utilizar ("") o (""). La diferencia la veremos luego.

- Para ver el contenido de la variable:

```
|| echo $username;
```

- Podemos asignar el valor de la variable a otra variable:

```
|| $current_user = $username;
```

Variables numéricas

```
<?php  
$count = 17;  
$count = 17.5;  
echo $count;  
?>
```

Arreglos

```
● || $team = array('Bill', 'Mary', 'Mike', 'Chris', 'Anne');
```

Arreglos

- `$team = array('Bill', 'Mary', 'Mike', 'Chris', 'Anne');`
- La sintaxis de array es: `array()`;

Arreglos

- `|| $team = array('Bill', 'Mary', 'Mike', 'Chris', 'Anne');`

- La sintaxis de array es: `array()`;

- Si quisieramos sber el elemento en la posición 4:

```
|| echo $team[3]; // Imprime el nombre Chris
```

Arreglos

- `|| $team = array('Bill', 'Mary', 'Mike', 'Chris', 'Anne');`

- La sintaxis de array es: `array()`;

- Si quisieramos saber el elemento en la posición 4:

- `|| echo $team[3]; // Imprime el nombre Chris`

- El primer elemento es el 0.

Arrays multi-dimensionales

- Los arrays en PHP nos permiten hacer muchas cosas.

Arrays multi-dimensionales

- Los arrays en PHP nos permiten hacer muchas cosas.
- Los arrays pueden ser multidimensionales (bi, tri o más)

Arrays multi-dimensionales

- Los arrays en PHP nos permiten hacer muchas cosas.
- Los arrays pueden ser multidimensionales (bi, tri o más)
- Supongamos que queremos implementar un juego de tres en raya.

Arrays multi-dimensionales

- Los arrays en PHP nos permiten hacer muchas cosas.
- Los arrays pueden ser multidimensionales (bi, tri o más)
- Supongamos que queremos implementar un juego de tres en raya.
- Necesitamos 9 celdas en una matriz de 3x3. Un arreglo de dos dimensiones.



Array bidimensional

Con arreglo con un juego de tres en raya en progreso.

```
<?php
    $oxo = array(array('x', ' ', 'o'),
        array('o', 'o', 'x'),
        array('x', 'o', ' '));
?>
```

- Dos dimensiones: `array(array());`.

Array bidimensional

Con arreglo con un juego de tres en raya en progreso.

```
<?php
    $oxo = array(array('x', ' ', 'o'),
        array('o', 'o', 'x'),
        array('x', 'o', ' '));
?>
```

- Dos dimensiones: `array(array());`.
- Para acceder al tercer elemento de la segunda fila:

```
|| echo $oxo[1][2];
```

Reglas para nombres de variables

Los nombres de variables:

- Empiezan con una letra del abecedario o guión bajo (_).

Reglas para nombres de variables

Los nombres de variables:

- Empiezan con una letra del abecedario o guión bajo (_).
- Contienen solamente caracteres de la a-z, A-Z, 0-9 y _

Reglas para nombres de variables

Los nombres de variables:

- Empiezan con una letra del abecedario o guión bajo (_).
- Contienen solamente caracteres de la a-z, A-Z, 0-9 y _
- No aceptan espacios. Las palabras se pueden separar con _

Reglas para nombres de variables

Los nombres de variables:

- Empiezan con una letra del abecedario o guión bajo (_).
- Contienen solamente caracteres de la a-z, A-Z, 0-9 y _
- No aceptan espacios. Las palabras se pueden separar con _
- Son sensibles a mayúsculas y minúsculas (case-sensitive). `$User_Name` no es igual que `$user_name`.

Reglas para nombres de variables

Los nombres de variables:

- Empiezan con una letra del abecedario o guión bajo (_).
- Contienen solamente caracteres de la a-z, A-Z, 0-9 y _
- No aceptan espacios. Las palabras se pueden separar con _
- Son sensibles a mayúsculas y minúsculas (case-sensitive). `$User_Name` no es igual que `$user_name`.
- Los acentos son aceptados pero es recomendable evitarlos para aumentar la compatibilidad con otros programadores.

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

- Comentarios
- Sintaxis básica

Contenido II

- Variables
- **Operadores**
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

- Todos los comandos matemáticos, string, de comparación y lógicos (+, -, *, /).

Operadores

- Todos los comandos matemáticos, string, de comparación y lógicos (+, -, *, /).
- `echo 6 + 2; //8`

Operadores

- Todos los comandos matemáticos, string, de comparación y lógicos (+, -, *, /).
- `echo 6 + 2; //8`
- Operadores Aritméticos

Operator	Description	Example
+	Addition	<code>\$j + 1</code>
-	Subtraction	<code>\$j - 6</code>
*	Multiplication	<code>\$j * 11</code>
/	Division	<code>\$j / 4</code>
%	Modulus (division remainder)	<code>\$j % 9</code>
++	Increment	<code>++\$j</code>
--	Decrement	<code>--\$j</code>

- ¿Cuál es la salida de la siguiente sentencia, si $\$x = 9$?

```
|| if (++$x == 10) echo $x;
```


Operadores ++ / --

- ¿Cuál es la salida de la siguiente sentencia, si $\$x = 9$?

```
|| if (++$x == 10) echo $x;
```

- ¿Y de esta, si $\$y = 0$?

```
|| if ($y-- == 0) echo $y;
```

Operadores de asignación

Operator	Example	Equivalent to
=	<code>\$j = 15</code>	<code>\$j = 15</code>
+=	<code>\$j += 5</code>	<code>\$j = \$j + 5</code>
-=	<code>\$j -= 3</code>	<code>\$j = \$j - 3</code>
*=	<code>\$j *= 8</code>	<code>\$j = \$j * 8</code>
/=	<code>\$j /= 16</code>	<code>\$j = \$j / 16</code>
.=	<code>\$j .= \$k</code>	<code>\$j = \$j . \$k</code>
%=	<code>\$j %= 4</code>	<code>\$j = \$j % 4</code>

Operadores de comparación

Operator	Description	Example
==	Is <i>equal</i> to	<code>\$j == 4</code>
!=	Is <i>not equal</i> to	<code>\$j != 21</code>
>	Is <i>greater than</i>	<code>\$j > 3</code>
<	Is <i>less than</i>	<code>\$j < 100</code>
>=	Is <i>greater than or equal</i> to	<code>\$j >= 15</code>
<=	Is <i>less than or equal</i> to	<code>\$j <= 8</code>

Operadores lógicos

Operator	Description	Example
<code>&&</code>	<i>And</i>	<code>\$j == 3 && \$k == 2</code>
<code>and</code>	Low-precedence <i>and</i>	<code>\$j == 3 and \$k == 2</code>
<code> </code>	<i>Or</i>	<code>\$j < 5 \$j > 10</code>
<code>or</code>	Low-precedence <i>or</i>	<code>\$j < 5 or \$j > 10</code>
<code>!</code>	<i>Not</i>	<code>! (\$j == \$k)</code>
<code>xor</code>	<i>Exclusive or</i>	<code>\$j xor \$k</code>

Concatenación de strings

- Para concatenar strings utilizamos en punto (.)

```
|| echo ‘‘Usted tiene ’’ . $msgs . ‘‘ mensajes.’’;
```

Concatenación de strings

- Para concatenar strings utilizamos en punto (.)

```
|| echo ‘‘Usted tiene ’’ . $msgs . ‘‘ mensajes.’’;
```

- Si asumimos que `\$msgs` tiene un valor de 5, la salida será: Usted tiene 5 mensajes.

Concatenación de strings

- Para concatenar strings utilizamos en punto (.)

```
|| echo ‘‘Usted tiene ’’ . $msgs . ‘‘ mensajes.’’;
```

- Si asumimos que `\$msgs` tiene un valor de 5, la salida será: Usted tiene 5 mensajes.
- Se puede utilizar también el operador: `.=`

```
|| $boletin .= $noticias;
```

Tipos de string

- Existen dos tipos de string en PHP. Se lo diferencia por el tipo de comillas: " o ""

Tipos de string

- Existen dos tipos de string en PHP. Se lo diferencia por el tipo de comillas: " o ""
- Si utilizamos comillas simples (') el valor del string será literal (contenido exacto). No intentará evaluar las variables dentro del string.

```
|| $info = 'Las variables llevan un prefijo $ asi:  
||     $variable';
```

Tipos de string

- Existen dos tipos de string en PHP. Se lo diferencia por el tipo de comillas: " o ""
- Si utilizamos comillas simples (") el valor del string será literal (contenido exacto). No intentará evaluar las variables dentro del string.

```
|| $info = 'Las variables llevan un prefijo $ asi:  
||     $variable';
```

- Si queremos incluir el valor de una variable en un string, utilizamos comillas dobles (""),. Esto facilita la concatenación de string con variables. *sustitución de variables*.

```
|| echo "Esta semana, $count personas han visto tu perfil";
```

Caracteres de escape

- Para poder utilizar ciertos caracteres con algún significado para PHP necesitamos utilizar el caracter de escape (\). Ej. la siguiente línea dará error:

```
|| $text = 'La comilla simple es ' esto dara error.'; //
```

error de sintaxis

Caracteres de escape

- Para poder utilizar ciertos caracteres con algún significado para PHP necesitamos utilizar el caracter de escape (`\`). Ej. la siguiente línea dará error:

```
|| $text = 'La comilla simple es ' esto dara error.'; //
```

error de sintaxis

- Deberíamos escribirlo así:

```
|| $text = 'La comilla simple es \' esto es correcto.'; //
```

correcto

Caracteres de escape

- Para poder utilizar ciertos caracteres con algún significado para PHP necesitamos utilizar el caracter de escape (\). Ej. la siguiente línea dará error:

```
|| $text = 'La comilla simple es ' esto dara error.'; //
```

error de sintaxis

- Deberíamos escribirlo así:

```
|| $text = 'La comilla simple es \' esto es correcto.'; //
```

correcto

- Lo mismo se puede aplicar a las comillas dobles:

```
|| $text = "La comilla doble es \" esto es correcto."; //
```

correcto

Caracteres de escape

- Para poder utilizar ciertos caracteres con algún significado para PHP necesitamos utilizar el caracter de escape (\). Ej. la siguiente línea dará error:

```
|| $text = 'La comilla simple es ' esto dara error.'; //
```

error de sintaxis

- Deberíamos escribirlo así:

```
|| $text = 'La comilla simple es \' esto es correcto.'; //
```

correcto

- Lo mismo se puede aplicar a las comillas dobles:

```
|| $text = "La comilla doble es \" esto es correcto.\"; //
```

correcto

- Los caracteres \t, \n, \r solo pueden ser interpretados utilizando (""). Las comillas simples lo presentarán literal. Las comillas simples solo interpretarán los caracteres \' y \\

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

- Comentarios
- Sintaxis básica

Contenido II

- Variables
- Operadores
- **Comandos de múltiples líneas**
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

Comandos de múltiples líneas

- Podemos escribir texto entre comillas en múltiples líneas

```
<?php
    $author = "Steve Ballmer";
    echo "Developers, Developers, developers, developers,
    developers, developers, developers, developers!

    - $author.";
?>
```

Comandos de múltiples líneas

- Podemos escribir texto entre comillas en múltiples líneas

```
<?php
    $author = "Steve Ballmer";
    echo "Developers, Developers, developers, developers,
    developers, developers, developers, developers!

    - $author.";
?>
```

- O asignar a un variable:

```
<?php
    $author = "Bill Gates";
    $text = "Measuring programming progress by lines of
    code is like
    Measuring aircraft building progress by weight.

    - $author.";
?>
```

Operador <<< (*Heredoc*)

```
<?php
$author = "Brian W. Kernighan";
echo <<<_END
Debugging is twice as hard as writing the code in the first
    place.
Therefore, if you write the code as cleverly as possible,
    you are,
by definition, not smart enough to debug it.

- $author.
_END;
?>
```

- PHP imprime todo literalmente, incluyendo espacios, y tabulaciones sin necesidad de añadir `\n`.
- La etiqueta de cierre solo debe ir acompañada de `(;)` ningún espacio en blanco adicional.
- También se puede asignar a variables.

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

- Comentarios
- Sintaxis básica

Contenido II

- Variables
- Operadores
- Comandos de múltiples líneas
- **Tipos de variables**
- Constantes
- Funciones
- Alcance de variables

Tipos de variables

- PHP es muy flexible con respecto a los tipos.

Tipos de variables

- PHP es muy flexible con respecto a los tipos.
- Las variables no tienen que ser declaradas antes de ser usadas.

Tipos de variables

- PHP es muy flexible con respecto a los tipos.
- Las variables no tienen que ser declaradas antes de ser usadas.
- PHP siempre convierte las variables al *tipo* requerido dependiendo del contexto.

```
<?php
    $number = 12345 * 67890; // numerico
    echo substr($number, 3, 1); // string
?>
```

```
<?php
    $pi = "3.1415927"; //string
    $radius = 5; //numerico
    echo $pi * ($radius * $radius); // numerico
?>
```


Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

- Comentarios
- Sintaxis básica

Contenido II

- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- **Constantes**
- Funciones
- Alcance de variables

Constantes

- Similares a las variables pero su contenido no se puede modificar.

Constantes

- Similares a las variables pero su contenido no se puede modificar.
- Una constante comúnmente declarada es el directorio de nuestro *document root*:

```
|| define("ROOT_LOCATION", "/usr/local/www/");
```

Constantes

- Similares a las variables pero su contenido no se puede modificar.
- Una constante comúnmente declarada es el directorio de nuestro *document root*:

```
|| define("ROOT_LOCATION", "/usr/local/www/");
```

- Para leer el valor:

```
|| $directory = ROOT_LOCATION;  
|| echo $directory;  
|| echo ROOT_LOCATION;
```

Constantes

- Similares a las variables pero su contenido no se puede modificar.
- Una constante comúnmente declarada es el directorio de nuestro *document root*:

```
|| define("ROOT_LOCATION", "/usr/local/www/");
```

- Para leer el valor:

```
|| $directory = ROOT_LOCATION;  
|| echo $directory;  
|| echo ROOT_LOCATION;
```

- No llevan el prefijo \$ y solo pueden ser definidas con `define()`; una sola vez.

Constantes mágicas

Constante mágica	Descripción
<code>__LINE__</code>	Línea actual del archivo.
<code>__FILE__</code>	El directorio completo (incluyendo nombre del archivo) del archivo actual.
<code>__DIR__</code>	El directorio del archivo.
<code>__FUNCTION__</code>	El nombre de la función actual (case-sensitive).
<code>__CLASS__</code>	El nombre de la clase (case-sensitive).
<code>__METHOD__</code>	El nombre del método (case-sensitive).
<code>__NAMESPACE__</code>	El nombre del namespace (case-sensitive).

```
echo "This is line " . __LINE__ . " of file " . __FILE__;
```

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

- Comentarios
- Sintaxis básica

Contenido II

- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- **Funciones**
- Alcance de variables

- Utilizadas para separar secciones de código con tareas distintas.

- Utilizadas para separar secciones de código con tareas distintas.
- Reutilización de código (con todos sus beneficios).

Funciones

- Utilizadas para separar secciones de código con tareas distintas.
- Reutilización de código (con todos sus beneficios).
- Reciben parámetros y retornan valores.

```
<?php
function longdate($timestamp)
{
    return date("l F jS Y", $timestamp);
}

echo longdate(time());
echo longdate(time() - 17 * 24 * 60 * 60); // la fecha
      hace 17 días (17 d, 24 h, 60 m, 60 s)

?>
```

Contenido I

1 Previa

- Temas de proyectos
- Acceso Peerwise

2 Contenido web dinámico

- HTTP y HTML
- Herramientas
- PHP
- MySQL
- JavaScript
- Ajax
- CSS
- HTML5
- Servidor Web Apache

3 Deber Consulta

4 Ejercicio

5 Introducción a PHP

- Comentarios
- Sintaxis básica

Contenido II

- Variables
- Operadores
- Comandos de múltiples líneas
- Tipos de variables
- Constantes
- Funciones
- Alcance de variables

Alcance de variables

- Las variables se pueden definir como locales o globales.

Alcance de variables

- Las variables se pueden definir como locales o globales.
- Por defecto, las variables creadas dentro de una función son **locales** a esa función y las variables creadas fuera de cualquier función son accesibles solo fuera de cualquier función.

```
<?php
function longdate($timestamp)
{
    $temp = date("l F jS Y", $timestamp);
    return "The date is $temp";
}
?>
```


Ejemplo (fallará)

```
<?php
    // MOSTRARA UNA NOTA DICIENDO QUE LA VARIABLE NO HA SIDO
    DEFINIDA
    $temp = "The date is ";
    echo longdate(time());
    function longdate($timestamp)
    {
        return $temp . date("l F jS Y", $timestamp);
    }
?>
```

Ejemplo (Dos alternativas de solución)

```
<?php
    $temp = "The date is ";
    echo $temp . longdate(time());
    function longdate($timestamp)
    {
        return date("l F jS Y", $timestamp);
    }
?>
```

O

```
<?php
    $temp = "The date is ";
    echo longdate($temp, time());
    function longdate($text, $timestamp)
    {
        return $text . date("l F jS Y", $timestamp);
    }
?>
```

Variables globales

- Cuando necesitamos que sea accedida desde cualquier parte del código.

Variables globales

- Cuando necesitamos que sea accedida desde cualquier parte del código.
- Cuando una variable contiene datos grandes y complejos y no se desea pasarla como argumento a varias funciones.

```
|| global $is_logged_in;
```

Variables globales

- Cuando necesitamos que sea accedida desde cualquier parte del código.
- Cuando una variable contiene datos grandes y complejos y no se desea pasarla como argumento a varias funciones.

```
|| global $is_logged_in;
```

- Se debe ser cuidadoso para no cambiar el valor de la variable accidentalmente o utilizar el mismo nombre para otra variable.

Variables globales

- Cuando necesitamos que sea accedida desde cualquier parte del código.
- Cuando una variable contiene datos grandes y complejos y no se desea pasarla como argumento a varias funciones.

```
|| global $is_logged_in;
```

- Se debe ser cuidadoso para no cambiar el valor de la variable accidentalmente o utilizar el mismo nombre para otra variable.
- Se recomienda que los nombre de las variables globales estén en mayúsculas para evitar confusiones.

Variables estáticas

- Sirven para mantener el valor de las variables a través de varias las llamadas a una función.

Variables estáticas

- Sirven para mantener el valor de las variables a través de varias las llamadas a una función.
- Ejemplo: deseamos mantener el valor de un contador en varias llamadas a la función:

```
<?php
function test()
{
    static $count = 0; // solo ejecutada primera llamada
    echo $count;
    $count++;
}
?>
```


Variables estáticas

- Sirven para mantener el valor de las variables a través de varias las llamadas a una función.
- Ejemplo: deseamos mantener el valor de un contador en varias llamadas a la función:

```
<?php
function test()
{
    static $count = 0; // solo ejecutada primera llamada
    echo $count;
    $count++;
}
?>
```

- Solo aceptan valores predefinidos (no el resultado de otra expresión).

```
<?php
static $int = 0; // Permitido
static $int = 1+2; // No permitido (Parse error)
static $int = sqrt(144); // No Permitido
?>
```

Variables superglobales

- Son variables predefinidas en el lenguaje y son accesibles desde cualquier parte.

Variables superglobales

- Son variables predefinidas en el lenguaje y son accesibles desde cualquier parte.
- Contienen información sobre el programa que está corriendo y su ambiente de ejecución.

Variables superglobales

- Son variables predefinidas en el lenguaje y son accesibles desde cualquier parte.
- Contienen información sobre el programa que está corriendo y su ambiente de ejecución.
- Definidas como arrays asociativos (que veremos luego).

Variables superglobales

Nombre	Contenido
\$GLOBALS	Todas las variables definidas como globales. Los nombres son los índices.
\$_SERVER	Info sobre headers, paths y ubicación de scripts. Creados por el servidor web puede variar.
\$_GET	Variables pasadas al script por el método HTTP Get.
\$_POST	Variables pasadas al script por el método HTTP Post.
\$_FILES	Items subidos al script por el método HTTP Post
\$_COOKIE	Variables pasadas al script por via HTTP cookies
\$_SESSION	Variables de sesión en el script actual
\$_REQUEST	Contenidos de información pasados por el navegador (\$_GET, \$_POST y \$_COOKIE)
\$_ENV	Variables pasadas al script por el método "environment"

Como usarlas?

```
print_r ($_SERVER);  
$came_from = $_SERVER['HTTP_REFERER'];
```

Superglobales (seguridad)

- Se debe ser cuidadoso con las superglobales ya que son utilizadas por hackers para romper seguridades en el sitio y acceder a información sensible.

Superglobales (seguridad)

- Se debe ser cuidadoso con las superglobales ya que son utilizadas por hackers para romper seguridades en el sitio y acceder a información sensible.
- Se pueden cargar comandos Unix o MySQL en `$_POST` o `$_GET`.

Superglobales (seguridad)

- Se debe ser cuidadoso con las superglobales ya que son utilizadas por hackers para romper seguridades en el sitio y acceder a información sensible.
- Se pueden cargar comandos Unix o MySQL en `$_POST` o `$_GET`.
- Siempre se debe limpiar las superglobales antes de usarlas.

Superglobales (seguridad)

- Se debe ser cuidadoso con las superglobales ya que son utilizadas por hackers para romper seguridades en el sitio y acceder a información sensible.
- Se pueden cargar comandos Unix o MySQL en `$_POST` o `$_GET`.
- Siempre se debe limpiar las superglobales antes de usarlas.
- Se puede utilizar la función `htmlentities()` que convierte todos los caracteres en entidades HTML (`< = <` y `> = >`).

```
|| $came_from = htmlentities($_SERVER['HTTP_REFERER']);
```

Superglobales (seguridad)

- Se debe ser cuidadoso con las superglobales ya que son utilizadas por hackers para romper seguridades en el sitio y acceder a información sensible.
- Se pueden cargar comandos Unix o MySQL en `$_POST` o `$_GET`.
- Siempre se debe limpiar las superglobales antes de usarlas.
- Se puede utilizar la función `htmlentities()` que convierte todos los caracteres en entidades HTML (`< = <`; y `> = >`).

```
|| $came_from = htmlentities($_SERVER['HTTP_REFERER']);
```

- `htmlentities` debe usarse siempre que se procese información enviada por los usuarios y que vaya a ser presentada (no solo con superglobales).