Servicio Nacional de Aprendizaje – SENA

Centro de Diseño y Metrologia

informeFundamentosPHP

Aprendiz:

Juan Esteban Castro Escamilla

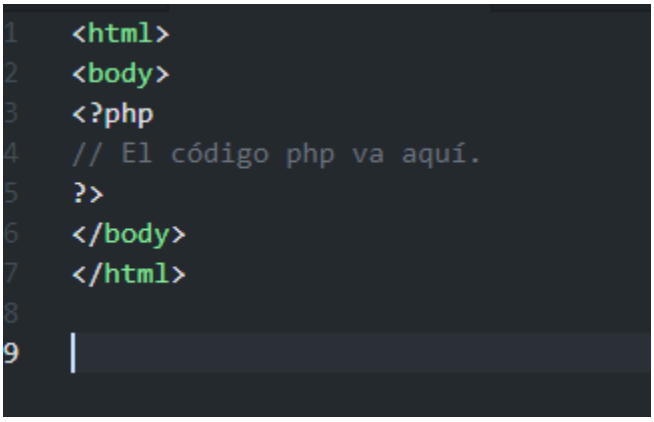
No Ficha 2848530-A

Tabla de contenido

FUNDAMENTOS PHP

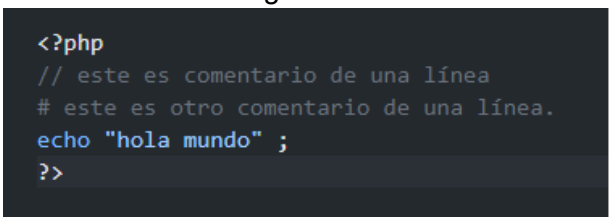
- Características del lenguaje:

El archivo debe tener la extensión .php para que el servidor procese el código PHP embebido, de lo contrario lo ignorará.
El tag de inicio estándar es “<? php” y el tag de cierre es “?>”.
Existe la posibilidad de usar “tags” abreviados pero no se aconsejan para que el código quede más entendible.



- Comentarios:

Comentarios de una sola línea: se puede realizar anteponiendo el símbolo “#” o dos barras inclinadas seguidas “//” al inicio de la línea. No necesitan cerrarse.



Comentarios de más de una línea: se pueden realizar comentarios que abarquen más de una línea usando los símbolos “/*” al inicio y “*/” al final.

```php
<?php
/*
este es comentario abarca varias líneas.
Los comentarios son útiles para documentar los
programas.
*/
echo "hola mundo" ;
?>
```

- Modo incrustado en páginas web:

Para ejecutar un programa PHP embebido en una página web se requiere que el servidor web como por ejemplo Apache, Nginx, IIS tengan instalada la extensión del lenguaje.

Se mostrará un programa en que se mostrará un texto “Hola Mundo”

```php
<html>
    <body>
        <?php
            echo "Hola Mundo";
        ?>
    </body>
</html>
```

Estas líneas de código se guardan con el nombre: “holamundo.php” en el directorio htdocs de la instalación XAMPP, generalmente “d:/xammp/htdocs/holamundo.php”.
Una vez puesto el archivo en el directorio se puede visualizar en el navegador como se muestra a continuación:

- Variables:

En PHP, las variables se representan con el signo dólar ($), seguido del respectivo nombre de variable. Se Utilizan los estándares de los lenguajes de programación para nombrar variables y PHP es sensible a mayúsculas y minúsculas.

```php
<?php
// Ejemplos de declaración de variables.
$nombre = "Pedro"; // variable tipo texto;
$apellido = "Perez"; // variable tipo texto;
$edad = 45; // variable numérica ;
echo "El señor ". $nombre . " " . $apellido . " tiene una
edad de ". $edad . " años. " ;
```

- Constantes

Son valores que no cambian y siempre serán los mismos

```php
<?php

echo "Ejemplo de una constante";
define(constant_name: "SENA", value: "Servicio Nacional De Aprendizaje");
echo SENA;

?>
```

Usamos la palabra define para declarar dicha constante y usamos echo para mostrarla por pantalla, misma función que haría normalmente un PRINT.

- Tipos de datos:

Como todos los lenguajes de programación, PHP nos permite utilizar diferentes tipos de datos, ya sea simples y compuestos.

- Tipos de datos escalares:

Aquellos datos que solo contienen una magnitud, es decir solamente dicho dato en concreto como los puede ser un numérico, decimal, string entre otros.
En esta imagen se mostrará un ejemplo de variable de texto (String) y de misma manera como imprimirla con un echo y concatenar estas variables con el símbolo “.”.

```php
<?php
$nombre = "Pedro";
$apellido = "Perez";
echo "Buenos días" . $nombre . " " . $apellido ;
?>
```

- Datos compuestos: aquellos que poseen diferentes magnitudes o contenidos dentro de este por ejemplo un Arreglo u objeto.

Se mostrará un ejemplo de cómo definir un arreglo:

```php
$arreglo = array(
"llave1" => "Valor1",
"llave2" => "Valor2",
);
```

Con valores:

```php
<?php
$arreglo = [
"Nombre" => "Pedro",
"Apellido" => "Perez",
];
echo "Buenos días". $arreglo["Nombre"] . " " .
$arreglo["Apellido"] ;
?>
```

- Estructuras de control:

Permiten establecer una condición para el funcionamiento de una parte del código en específico.

- Condicional simple (IF):

Permite realizar una instrucción de acuerdo a la condición o decisión a evaluar.

```
<html>
<body>
<p>Ejemplo de una estructura condicional sencilla.<p><br>
<?php
echo "Codigo PHP de un condicional sencillo <br>" ;
$a = 8;
$b = 3;
if( $a < $b ){
echo "a es menor que b" ;
} else {
echo "a es mayor que b" ;
};
?>
</body>
</html>
```

- Ciclo mientras (WHILE):

Dependiendo de una condición numérica se hará una iteración de esa parte de código o instrucción del proceso.

```php
<?php
while (expr) //evalúa la condición
{
//instrucción que se repite mientras
// condición sea verdadera
};
?>
```

- Ciclo hacer mientras (DO WHILE):

Similar al while, se realizará un ciclo dependiendo de la condición que se plantee al final del proceso.

```php
<?php
do
{

// Instrucciones a realizar

}
while //(condicion numerica)// evalúa condición y repite ciclo en
//caso de ser verdadera
```

- Ciclo repita por veces (FOR):

Permite realizar un proceso iterado, conociendo la condición numérica antes de ejecutarse.

```php
<?php
for // (expresión 1; expresión 2; expresión
3)
{

// Sentencias procesos }

} ;
?>
```

- Decisiones múltiples condicionales (SWITCH)

Dependiendo del valor numérico o variable a evaluar, se harán distintas acciones o sentencias.

```php
<?php
/*
switch (variable_a_evaluar)
{
case <valor1>:
<sentencias> ;
break;
case <valor2>:
<sentencias> ;
break;
case <valorn:
<sentencias> ;
break;
default // si no corresponde con ninguno de los
// valores anteriores
<sentencias> ;
}
*/
?>
```

- Ciclo Foreach:

Permite trabajar con arreglos y objetos, evaluandolos con una variable de ayuda, es decir auxiliar que nos permite realizar dicho ciclo.

```php
<?php
/*
foreach (<nombre_del_arreglo> as <variable_auxiliar>)
{
<sentencia 1> ;
<sentencia 2> ;
<sentencia n> ;
} ;
*/
?>
```

- Funciones:

Permiten realizar un bloque de código con ciertos parámetros o no y acciones a realizar, evitando la repetición y manteniendo el código más limpio y entendible

En la siguiente imagen se muestra un ejemplo de sintaxis de una función:

```php
<?php
/*
function(<parámetros>){
// bloque de código
return <variable>
};
*/
?>
```

- Incluir librerias de codigo:

Accediendo a estas librerías o bibliotecas nos va permitir ejecutar las diferentes funciones que ésta provee.

Ejemplo:

```
<?php
/*
include_once ''<nombre_biblioteca.php>'' ;
*/
?>
```

- Pasar variables a un programa en PHP:

Los metodos get y post nos permite pasar variables desde una página web a PHP

- Metodo GET usando una URL

Por medio de la URL se pasan las variables a un programa PHP

```
<a href="holamundo.php?nombre=pedro&apellido=perez> Enlace a
Pedro Pérez </a>
```

- Metodo get a traves de un formulario

Por medio de un formulario HTML podremos procesar y pasar las variables.

```
<form action="<nombre_programa>.php">
Enunciado del campo:
<input type="text" name="campo1">
<input type="submit" value="Enviar">
</form>
```

}

- Metodo POST con formulario:

En el método POST las variables no se anexan a la URL sino que se anexan al cuerpo de la solicitud del protocolo HTTP (“HTTP request”) y por tanto no son visibles en la barra de navegación.

```
<!--formulario.html -->
<html>
    body>
        <form action="programa1.php"method=post>
            Nombre: <input type=text name="nombre"><br>
            Apellido: <input type=text name="apellido"><br>
             input type=submit value="Enviar">
        </form>
    /body>
</html>
```

```
<!--programa1php-->
<html>
    <body>
        <?php">
            echo "Hola".$_POST["nombre"]."".$_POST["apellido"]."<br>";
        ?>
    </body>
</html>
```