Informe PHP

Planear Actividades De Construcción Del Software De Acuerdo Con El Diseño Establecido

Aprendiz:

Santiago Carranza Carrillo

Ficha:

2848530-A

Instructor:

Julio Roberto Galvis Cardozo

SENA

Centro De Diseño Y Metrología

9 De diciembre 2024

Índice

Introducción

En este informe, se presentan diversos conceptos fundamentales de PHP, un lenguaje de programación ampliamente utilizado para el desarrollo web. A través de una serie de ejemplos de código, se exploran diversas funciones y estructuras que forman la base de la programación en PHP, tales como la manipulación de variables, estructuras de control (como if, for, y switch), y la interacción con bases de datos MySQL. Los ejemplos incluyen desde la declaración de variables hasta la ejecución de consultas SQL, así como la correcta gestión de conexiones a la base de datos.

Códigos 1- 28

Código 1

```
<html>
<body>
<?php
// El código php va aquí.
?>
</body>
</html>
```

Explicación del código:

Ejemplo básico de una página web con un archivo HTML que incluye código PHP incrustado.

Código 2

```
<?php
// El código php va aquí.
// Cuando solamente hay código PHP se puede omitir el tag de
cierre.
```

Explicación del código:

En PHP, si un archivo contiene solo código PHP, se puede omitir la etiqueta de cierre “?>”.

Código 3

```php
<?php
// este es comentario de una línea
# este es otro comentario de una línea.
echo "hola mundo" ;
?>
```

Explicación del código:

En PHP se puede utilizar tanto el “//” como el “#” para realizar comentarios de una línea, también se encuentra un “echo” se utiliza para imprimir algún contenido en una pagina web

Código 4

```php
<?php
/*
este es comentario abarca varias líneas.
Los comentarios son útiles para documentar los
programas.

*/
echo "hola mundo" ;
?>
```

Explicación del código:

En PHP se puede utilizar el “/*” y el “*/” para abrir y cerrar un comentario que utilice varias líneas (Se utiliza el “echo” ya explicado).

Código 5

```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. " ;
?>
```

Explicación del código:

En PHP se utiliza el “$” para comenzar una variable, así mismo en el código se le asignan valores con el (= “”) en caso del texto y solo el “=” en caso de números (Se utiliza el “echo” ya explicado).

Código 6

```php
<?php
define( "<Nombre de la constante>", "<valor>" );
?>
```

Explicación del código:

En este código se utiliza la función define de PHP que permite definir una constante, se asigna el nombre de la misma y su valor el cual no cambia durante la ejecución del código.

Código 7

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

Explicación del código:

En este código se definen variables y se utilizan en el “echo” al cerrar las comillas en el comando se utiliza un “.” Seguido de la variable definida para ser visualizada.

Código 8

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

Explicación del código:

Este código crea un arreglo asociativo en PHP, donde cada elemento tiene una clave única asociada a un valor.

Código 9

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

echo “Buenos días". $arreglo["Nombre"] . " " .
$arreglo["Apellido"] ;
?>
```

Explicación del código:

En este código se define la variable “$arreglo” la cual tiene dos claves, Nombre y Apellido, se le asignan valores y se utiliza un “echo” para la salida de los datos accediendo al valor asociado a las respectivas claves.

Código 10

```php
<?php
if (expr)
Sentencia; // en caso de condición verdadera
else
```

```
Sentencia; // en caso de condición falsa
?>
```

Explicación del código:

Se utiliza un “If” dando salida a una sentencia, de lo contrario (“else”) se dará salida a una sentencia.

Código 11

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

Explicación del código:

En este código se utiliza un “While” de manera que mientras se cumpla la condición establecida en el proceso se repetirán unas instrucciones especificas.

Código 12

```
<?php
do
{
<instruccion 1> ;
<instruccion 2> ;
<instruccion n> ;
}
while (<expresión lógica>) // evalúa condición y repite ciclo en
//caso de ser verdadera
?>
```

Explicación del código:

En este código se establecen cierto número de instrucciones a ejecutar ligadas a una condición determinada en un “While” siendo que repetirá el ciclo de instrucciones mientras esa condición se cumpla.

Código 13

```
<?php
for (expresión 1; expresión 2; expresión 3)
{
<sentencia 1> ;
<sentencia 2> ;
<sentencia n> ;

} ;
?>
```

Explicación del código:

En este código se utiliza un bucle “for” este se utilizara ejecutando repetidamente el bloque de código establecido siempre y cuando la condición se cumpla.

Código 14

```
<?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> ;
}
?>
```

Explicación del código:

En este código se utiliza el “switch”, este permite evaluar una variable o expresión, esta misma se compara con cada valor establecido en las clausulas “case”, si esa variable coincide con alguna de las clausulas se ejecutara la sentencia, de no corresponder con ninguna de estas se dará salida a una sentencia “default”.

Código 15

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

Explicación del código:

Este código muestra cómo usar un bucle “foreach” en PHP, que es una forma de recorrer arreglos

Código 16

```
<?php
$arreglo = [ "Jacinto", "Jose", "Pepita", "Mendieta" ] ;
```

```php
$j = 0;
foreach($arreglo as $elemento)
{
echo "$j: $elemento \n";
++$j ;
}
?>
```

Explicación del código:

Este código recorre el arreglo $arreglo, mostrando el índice de cada elemento seguido por su valor. La variable $j se utiliza para mostrar el índice de cada elemento, y se incrementa al final de cada ciclo del bucle.

Código 17

```php
<?php
function(<parámetros>){
// bloque de código

return <variable>
};
?>
```

Explicación del código:

Este código muestra la sintaxis básica para definir una función en PHP.

Código 18

```php
<?php

include_once ’’<nombre_biblioteca.php>’’ ;
?>
```

Explicación del código:

Este código en PHP utiliza la instrucción “include_once”, que se emplea para incluir archivos externos en script.

Código 19

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

Explicación del código:

El código es un ejemplo de un enlace HTML que utiliza el método GET para enviar parámetros a un archivo PHP, La etiqueta <a> se utiliza para crear un enlace o hipervínculo en HTML.

Código 20

```php
<?php
```

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

Explicación del código:

Este código presenta un formulario HTML que envía los datos a otro archivo PHP cuando se hace clic en el botón de envío, PHP puede procesar los datos recibidos, por ejemplo, para mostrarlos o guardarlos en una base de datos.

Código 21

```
<?php
create database adsi ;
create table ciudades ( codigo text, nombre text ) ;
insert into ciudades ( codigo, nombre ) values ( "05001", "MEDELLIN" );
insert into ciudades ( codigo, nombre ) values ( "05002", "ABEJORRAL" );
insert into ciudades ( codigo, nombre ) values ( "05004", "ABRIAQUI" );
insert into ciudades ( codigo, nombre ) values ( "05021", "ALEJANDRIA" );
?>
```

Explicación del código:

Este código es un ejemplo básico para la creación de una base de datos, creación de tabla con sus respectivos campos y registros dentro de la tabla creada.

Código 22

```
<?php
$conn = new mysqli("<host>", "<usuario>", "<clave>", "<base_de_datos>" ) ;
if( $conn->connect_errno ) {
echo "Falla al conectarse a Mysql ( ". $conn->connect_errno . ") " .
$conn->connect_error ;
} else {
echo $conn->host_info. "\n" ;
} ;
?>
```

Explicación del código:

Este código en PHP muestra cómo conectarse a una base de datos MySQL usando la clase mysqli.

Conexión a la base de datos:

Usa new mysqli("<host>", "<usuario>", "<clave>", "<base_de_datos>") para conectar con el servidor MySQL.

Si la conexión es exitosa, se almacena en la variable $conn.

Manejo de errores:

Si hay un error en la conexión, se verifica con $conn->connect_errno. Si es diferente de cero, se muestra el error usando $conn->connect_error.

Conexión exitosa:

Si la conexión es exitosa, se muestra información del servidor MySQL con $conn->host_info.

Código 23

```php
<?php
$conn = new mysqli("localhost", "desarrollador", "adsi2017", "citas" ) ;
if( $conn->connect_errno) {
echo "Falla al conectarse a Mysql ( ". $conn->connect_errno . ") " .
$conn->connect_error ;
} else {
echo $conn->host_info. "\n" ;
} ;
?>
```

Explicación del código:

Hace referencia al código anterior pero remplazando datos.

Código 24

```php
<?php
if( $mysqli->query("<sentencia sql>") === TRUE ) {
    echo "Se ejecutó la sentencia con éxito";
    } else {
    echo "Hubo un error ..." ;
    };
?>
```

Explicación del código:

Este código ejecuta una sentencia SQL y muestra un mensaje si la ejecución es exitosa o si hay un error. Se usa comúnmente para consultas que no devuelven resultados, como "INSERT", "UPDATE", o "DELETE".

Código 25

```php
<?php

$conn = new mysqli("localhost", "desarrollador", "adsi2017", "adsi" ) ;
if( $conn->connect_errno) {
echo "Falla al conectarse a Mysql ( ". $conn->connect_errno . ") " .
$conn->connect_error ;
exit() ;
} ;

if($resultado = $conn->query("select codigo, nombre from ciudades ") ){
```

```php
while($registro = $resultado->fetch_assoc() ){

echo $registro["codigo"] . " " . $registro["nombre"] . "\n" ;

} ;
};

$resultado->free();
$conn->close();
?>
```

Explicación del código:

Primero se establece una conexión a la base de datos.

Luego se ejecuta una consulta SQL para obtener los datos de la tabla ciudades.

Después se procesan los resultados y se muestran los valores de las columnas codigo y nombre.

Y Finalmente, se liberan los recursos y se cierra la conexión a la base de datos.

Código 26

```php
<?php
if($resultado = $conn->query("select codigo, nombre from ciudades ") ){

    while($registro = $resultado->fetch_object() ){

    echo $registro->codigo . " " . $registro->nombre . "\n" ;
    } ;
    };
?>
```

Explicación del código:

Este código en PHP es similar al anterior, pero en lugar de utilizar fetch_assoc() para obtener los resultados de la consulta como un array asociativo, utiliza fetch_object() para obtener los resultados como un objeto.

Código 27

```php
<?php
$resultado->close();
?>
```

Explicación del código:

Este fragmento de código cierra el objeto de resultados “$resultado” después de haber terminado de usarlo.

Código 28

```php
<?php
$mysqli->close() ;
```

```
?>
```

Explicación del código:

Este código cierra la conexión a la base de datos MySQL que fue previamente establecida utilizando el objeto $mysqli.

Conclusión

A lo largo de este informe, se ha abordado una serie de conceptos y técnicas clave en PHP que permiten a los desarrolladores crear aplicaciones web dinámicas y robustas. Desde la gestión de variables y constantes, hasta el manejo de bases de datos y la optimización de recursos mediante el cierre adecuado de conexiones y resultados, estos elementos constituyen la base para desarrollar soluciones eficientes. Comprender estos conceptos es esencial para programadores que deseen trabajar con PHP, ya que dominarlos permite construir aplicaciones que no solo sean funcionales, sino también seguras y optimizadas. Este informe proporciona una base sólida sobre la cual se puede construir una comprensión más profunda del lenguaje y su aplicación en el desarrollo web.