Universidad Santo Tomas

Construcción de software

Backend

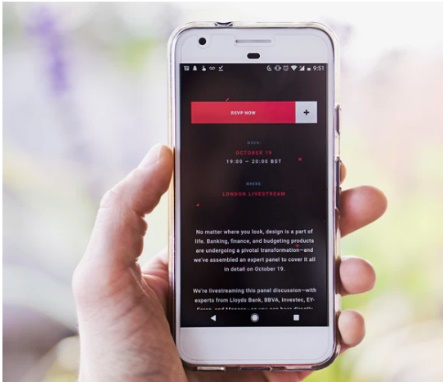
PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

El código php se ejecuta en el lado del servidor y genera una salida en html que es enviada al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era.

(PHP, 2019).





petición

respuesta

cliente

server

## Uno de los elementos importantes al iniciar nuestro trabajo con php es conocer como está configurado, para esto ejecutamos el siguiente comando:

## configuración

<?php  
  
echo phpinfo();

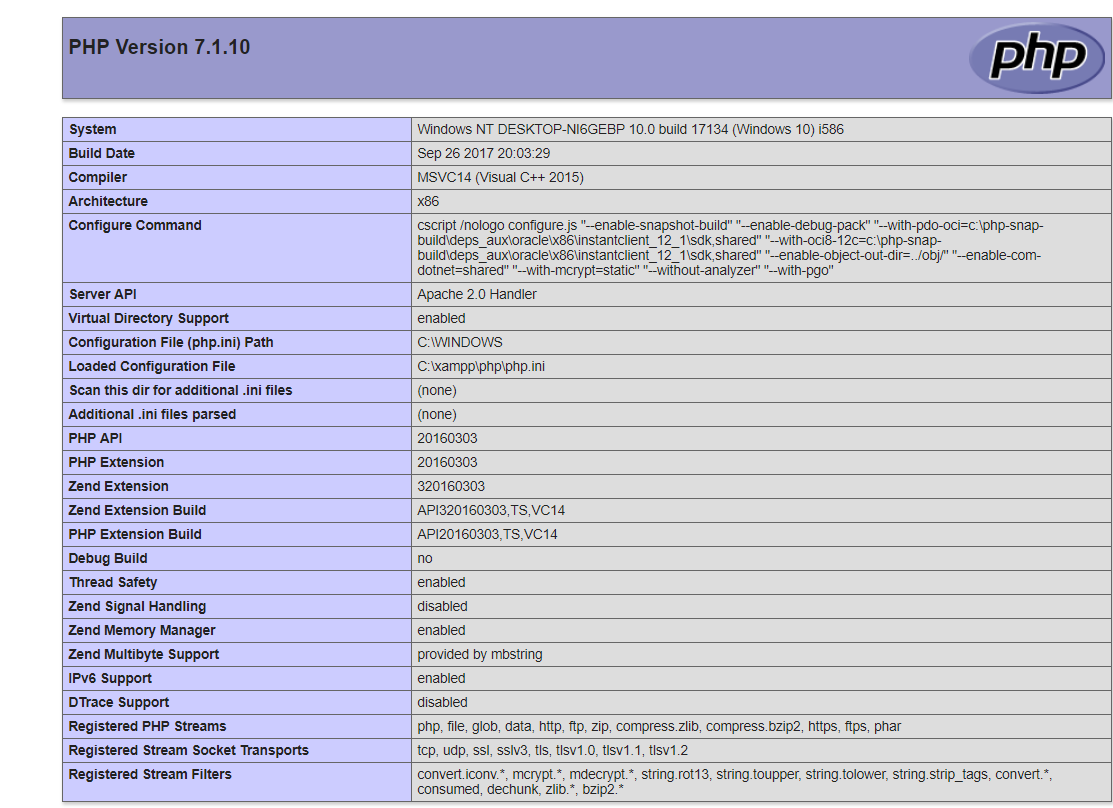


Imagen 1 Salida phpinfo

En la salida web se encuentra toda la información de opciones de compilación y extensiones de PHP, versión de PHP, información del servidor y entorno (si se compiló como módulo), entorno PHP, versión del OS, rutas, valor de las opciones de configuración locales y generales, cabeceras HTTP y licencia de PHP.

|  |
| --- |
| ¿Que librerías me permiten conectarme con la base de datos mysql? |
|  |
| ¿Qué archivo debo modificar para habilitar o deshabilitar librerías de php? |
|  |

## sintaxis

Con php resulta sencillo desarrollar código dinámico.

|  |
| --- |
| <h1>Bienvenidos a php</h1>  *<?php*  *$nombre=*"*Nicolas*"*;*  **echo** "<p>hola mundo ".*$nombre*. " </p>"**;**  *?>*  <p>párrafo en html</p> |

## Comentariar código

Una de las practicas más usadas en el desarrollo de código es crear comentarios para apoyar la lectura del código fuente.

para crear comentarios utilizamos:

|  |
| --- |
| //una linea  /\* varias lineas  \*/  # símbolo estilo perl  # autor: Iván  echo "hola mundo."; # resto de línea comentariada |

## Salida por pantalla

En el siguiente ejemplo se presenta la salida por pantalla de dos variables titulo y cuerpo utilizando la etiqueta echo.

|  |
| --- |
| *<?php*  $title **=** "Arqueólogos identifican una aguja para tatuar de 2.000 años "  **.** "de antigüedad"**;**    $body **=** "La herramienta está hecha a partir de un conjunto de espinas de "  **.** "cactus de nopal, las puntas están saturadas con pigmento oscuro,"  **.** " insertadas en un palo cortado de una planta de bayas de limonada "  **.** "y se ataban con fibra de yuca y atadas con fibra de yuca"**;**        **echo** $title**;**  **echo** "<br>"**;**  **echo** $body**;** |

## 

¿Qué pasa si deseo agregar las etiquetas html título y párrafo?

|  |
| --- |
| **echo** "<h1>"**.**$title**.**"</h1>"**;**  **echo** "<br>"**;**  **echo** "<p>"**.**$body**.**"</p>"**;** |

tendría que concatenar texto html con las variables.

o una alternativa más acertada es encerrado las variables dentro de llaves.

|  |
| --- |
| **echo** "<h1>{$title}</h1><p> {$body}</p>"**;** |

|  |
| --- |
| Como se utiliza la sentencia sprintf para presentar la variable $valor=4.2454654 con únicamente 2 dígitos. |
| $valor = 4.2454654;  echo sprintf("%1.2f",$valor); |

## Tipos de datos en php

Un tipo de datos es el nombre genérico asignado a cualquier dato que comparte un conjunto común de características. Los tipos de datos comunes incluyen Boolean, integer, float, string y array.

Complete el siguiente cuadro indicando si se trata de un valor true o false.

|  |  |
| --- | --- |
| if($valor) | Resultado |
| $valor=0 | False |
| $valor=5 | True |
| $valor=’a’ | True |
| $valor=’0’ | False |
| $valor=true | True |
| $valor=false | False |
| $valor=-5 | True |

Ejemplo

$valor=-5;

if($valor)

echo "true";

else echo "false";

Entero

representan números que no tienen fracciones

$entero=5;

$entero=5555;

float

se refiere a números que pueden contener decimales.

4.7

85.2454

1.23E+11

String

secuencia de caracteres

"hola mundo"

"445345#23"

"<h1>titulo</h1>"

## Array

Se define formalmente como una colección indexada de valores de datos.

|  |
| --- |
| $nombres**[0]** **=** "Juan"**;**  $nombres**[1]** **=** "Nicolas"**;**  $nombres**[2]** **=** "Sergio"**;**  $nombres**[3]** **=** "Jennifer"**;**  **echo** $nombres**[3];** |

## key value

Un array puede definirse también como elementos de llave y valor como se presenta en el siguiente ejemplo:

|  |
| --- |
| *<?php*  $departamentosCapitales**[**"Boyaca"**]** **=** "Tunja"**;**  $departamentosCapitales**[**"Caqueta"**]** **=** "Florencia"**;**  $departamentosCapitales**[**"Amazonas"**]** **=** "leticia"**;**  print\_r**(**$departamentosCapitales**);**  $departamentos**=**array\_keys**(**$departamentosCapitales**);**  **echo** "<br>"**;**  print\_r**(**$departamentos**);** |

|  |
| --- |
| ¿Qué utilidad proporciona la función array\_keys()? |
| Devuelve todas las claves de un array o un subconjunto de claves de un array |
| Imprima en pantalla la capital de Amazonas |
| print\_r($departamentosCapitales["Amazonas"]); |
| Imprima el departamento con índice 2 |
| print\_r($departamentos[2]); |

## Otra forma de declarar un array es utilizando la función array con índices es:

|  |
| --- |
| $estudiantes **=** **array(**"Diego"**,** "Oscar"**,** "Santiago"**,** "Nicolas"**);** |

Para agregar nuevos estudiantes al array utilizamos

array\_push**(**$estudiantes**,** "Tom"**,** "Jerry"**);**

para borrar un elemento

unset**(**$estudiantes**[4]);**

**para modificar un elemento**

$estudiantes**[0]** **=** "Jessie"**;**

|  |
| --- |
| //manejo de array  $estudiantes **=** **array(**"Diego"**,** "Oscar"**,** "Santiago"**,** "Nicolas"**);**  array\_push**(**$estudiantes**,** "Tom"**,** "Jerry"**);**  print\_r**(**$estudiantes**);**  **echo** "<br>"**;**  unset**(**$estudiantes**[4]);**  print\_r**(**$estudiantes**);**  **echo** "<br>"**;**  $estudiantes**[0]** **=** "Jessie"**;**  print\_r**(**$estudiantes**);** |

## recorrer un array con for

Vamos a recorrer el array estudiantes, utilizando el ciclo for.

|  |
| --- |
| $estudiantes **=** **array(**"Diego"**,** "Oscar"**,** "Santiago"**,** "Nicolas"**);**  array\_push**(**$estudiantes**,** "Tom"**,** "Jerry"**);**  $numeroElementos**=**count**(**$estudiantes**);**  **for(**$i**=0;**$i**<**$numeroElementos**;**$i**++){**  **echo** $estudiantes**[**$i**].**"<br>"**;**  **}** |

En el ejemplo es necesario utilizar la función count para contar el numero de elementos que contiene el array, con este valor ahora realizamos imprimimos el valor de cada índice de estudiantes desde 1 hasta el numero de elementos.

## Recorrer array con foreach

|  |
| --- |
| $estudiantes **=** **array(**"Diego"**,** "Oscar"**,** "Santiago"**,** "Nicolas"**);**  array\_push**(**$estudiantes**,** "Tom"**,** "Jerry"**);**  **foreach(**$estudiantes **as** $estudiante**){**    **echo** $estudiante**.**"<br>"**;**  **}** |

## recorrer un array con llave y valor.

|  |
| --- |
| $unEstudiante **=** **array(**  "edad" **=>** **20,**  "codigo" **=>** **211654,**  "genero" **=>** "masculino"**,**  "altura" **=>** **172,**  "nombre" **=>** "Armando"  **);**  **foreach** **(**$unEstudiante **as** $key **=>** $value**)** **{**  **echo** $key**.**" "**.**$value**.**"<br>"**;**  **}** |

En el ejemplo se presenta la forma para declarar un array de tipo llave, valor y la forma como podemos acceder a la información

## De String a array

|  |
| --- |
| $marcas**=**"mazda,nissan,toyota,chevrolet,fiat,renault"**;**  $arrayMarcas**=** explode**(**","**,** $marcas**);**  **echo** $arrayMarcas**[3];**  **echo** "<br>"**;**  var\_dump**(**$arrayMarcas**);**  **foreach** **(**$unEstudiante **as** $key **=>** $value**)** **{**  **echo** $key**.**" "**.**$value**.**"<br>"**;**  **}** |

|  |
| --- |
| Describa cómo funciona la función explode |
| Divide un string en varios string |
| ¿Como convertimos un array a un string? |
| Se usa el inverso de explote, el cual es implote  implode(",",$arrayMarcas); |

## Programación Orientada a Objetos

Seleccione una de las siguientes imágenes y transfórmelas a su representación en codigo php con programación orientada a objetos:

1.



