

introduccion a la programacion en PHP

1. primeros pasos
2. variables y tipos
3. constantes
4. operadores
5. condicionales
6. bucles
7. ejercicios
8. funciones
9. includes
10. arrays
11. ejercicios 2

preparacion del entorno de desarrollo

1. navegador web : chrome, firefox, opera, brave
2. server web : xampp, wamp
3. ide dev : VSCode

variables de entorno

1. propiedades de windows
2. variables de entorno
3. path
4. pegar ruta

configuraciones y modulos

que es php

1. lenguaje mas usado

por que usar php

1. php mas usado del lado del servidor para la web

Alternativas

1. python
2. ruby
3. java

PHP

1. localhost
2. www
3. project-en-php
4. project here
5. index.php

estructura

```
<?php // code... ?>
```

cargar proyecto

1. localhost/master-php

sintaxis de php

imprimir en pantalla

1. echo

concatenacion

2. "texto con \$variable" . 'texto sin variables con punto'

comentarios

3. // comentario en linea

4. /* */ comentarios en bloque

variables y tipos de datos

Un variable es un contenedor de informacion scope global.

1. \$name = "datoDeLaVariable";

tipos de datos que acepta una variable

1. entero (int / integer) = 99

2. coma flotante o decimales (float / double) = 3.65

3. cadenas (string) = "Hola soy un string"

4. booleano (bool) = true, false

5. null = sin contenido

6. Array (coleccion de datos)

7. Objetos

como vamos que tipo de variable es

gettype(\$numero)

```
$numero = 10;  
$decimal = 27.9;  
$texto = "hola mundo";  
$verdadero = true; // 1  
$falso = false; // 0  
$nula;
```

```
echo gettype($nula);
```

Las variables solo se pueden definir con

1. guiones bajos
2. texto
3. camelCase
4. SnakeCase

debugging de variables, para saber que tipo de variable es

```
$miNombre = "wizard";
```

var_dump(\$miNombre)

me da informacion completa de la variable

salto de linea

1. \n en consola
2. < br/> en html
3. \ para escapar simbolos

constantes

el valor no varia en una constante
se declara con define('var','valor');

```
define('nombre','jared');  
echo nombre;
```

constantes predefinidas en php

con estas funciones podemos ver que tiene nuestro servidor

1. PHP_OS;
2. PHP_version;
3. __LINE__
4. __FILE__

operadores aritméticos

1. + - * / %

```
$num1 = 55;  
$num2 = 33;  
echo 'suma'.($num1 + $num2);  
echo 'resta'.($num1 - $num2);  
echo 'multiplicacion'.($num1 * $num2);  
echo 'division'.($num1 / $num2);  
echo 'modulo o resto de una disivision'.($num1 % $num2);
```

Incremento y decremento

1. ++

2. --

```
$year = 2020;  
$year++; // $year = $year + 1;  
$year--; // $year = $year - 1;
```

pre-incremento y pre-decremento

1. ++

2. --

```
$year = 2021;  
++$year; // 1 + $year;  
--$year; // 1 - $year;
```

operadores asignacion

1. =

2. +=

3. -=

4. *=
5. /=

operadores de comparación y lógicos

1. == igualdad
2. === igualdad y tipo de dato
3. != diferente
4. <> diferente
5. !== diferente de tipo y valor
6. <
7. >
8. <=
9. >=
10. && AND condiciona true y true
11. || OR condiciona true o false
12. ! NOT negacion

variables superglobales

variables de servidor

```
echo $_SERVER['SERVER_ADDR']; //'localhost'  
echo $_SERVER['SERVER_NAME']; //'nombre del servidor'  
echo $_SERVER['SERVER_SOFTWARE']; //' '  
echo $_SERVER['HTTP_USER_AGENT']; //'navegador'  
echo $_SERVER['REMOTE_ADDR']; //'ip del cliente'
```

metodos _GET

por parametros get, se envian los datos por la url, de esta manera los datos son manipulables y no seguros para nuestro proyecto.

```
form method="GET" action="recibir.php"
```

Ejercicio:

1. getpost.html
2. recibir.php

metodos _POST

de esta manera se envian los datos de manera oculta

```
<form method="POST" action="recibir.php">
  <p>
    <label for="nombre">Nombre</label>
    <input type="text" name="nombre" />
  </p>

  <p>
    <label for="apellido">Apellido</label>
    <input type="text" name="apellido" />
  </p>

  <p>
    <input type="submit" value="enviar formulario" />
  </p>
</form>
```

recibe los datos y los imprime en pantalla

```
<?php
echo '<h1>' .$_POST['nombre']. '</h1>';
echo '<h1>' .$_POST['apellido']. '</h1>';

var_dump($_POST);
```

var_dump me trae la estructura del array

```
array(2) { ["nombre"]=> string(5) "jared" ["apellido"]=> string(6) "mimimi" }
```

Estructuras de control

if

la condicional if permite controlar el flujo del programa

1. if
2. else
3. elseif

Sintaxis

```
if (condicion true) {  
    si es true  
} elseif (otra condicion true) {  
    si es true  
} else {  
    si no se culplio  
}
```

switch

Sintaxis

```
$dia = 4;  
  
switch ($dia) {  
    case 1:  
        echo "lunes";  
        break;  
    case 2:  
        echo "martes";  
        break;  
    case 3:  
        echo "miercoles";  
        break;  
    case 4:  
        echo "jueves";  
        break;  
    case 5:  
        echo "viernes";  
        break;  
    case 6:  
        echo "sabado";  
        break;  
    default:  
        echo "domingo";  
}
```


GOTO

es un operado que se usa para saltar a otra parte del programa saltando el código siguiente.

```
goto marca;
echo "<h3>Instruccion 1</h3>";
echo "<h3>Instruccion 2</h3>";
echo "<h3>Instruccion 3</h3>";
echo "<h3>Instruccion 4</h3>";

marca:
echo "<h1>Me he saltado 4 echos</h1>";
```

Iteradores y bucles

while

Estructura de control que itera o repite la ejecución de una serie de instrucciones tantas veces como sea necesario, en base a una condición.

```
while (condicion) {
    bloque de instrucciones
}
```

del 1 al 100

```
$numero = 0;

while ($numero <= 100) {
    echo "Imprime linea $numero";
    $numero++;
}
```

ejemplo de while

```
// isset comprueba si existe una variable
if (isset($_GET['numero'])) {
    // casting, cambia tipo de dato a entero
```

```

    $numero = (int)$_GET['numero'];
} else {
    $numero = 4;
}

var_dump($numero);

echo "<h1>Tabla de multiplicar del numero $numero</h1>"
$contador = 0;

while ($contador <= 10) {
    echo "$numero x $contador = " . ($numero*$contador)."<br/>";
    $contador++;
}

```

do while

Sintaxis

```

do {
    // bloque de instrucciones
} while (condicion);

```

ejemplo

```

$edad = 18;
$contador = 1;
do {
    echo " tienes acceso al local privado $contador";
} while ($edad >= 18 && $contador <= 10);

```

se ejecutara siempre la primera vez

ciclo for

sintaxis

```

for (contador; condicion; incremento++) {
    //code...
}

```

ejemplo

```
$resultado = 0;

for ($i= 0;$i <= 100;i++) {
    $resultado = $resultado + $i;
}
echo "el resultado es $resultado";
```

break, dentro de los bucles

si se requiere evaluar algo, si se cumple o no, y si es el caso, con break se sale y deja de ejecutar el bucle.

modulo 13 de ejercicios

1. mostrar valores de variables
2. lista de numeros del 1 al 100 e imprimir los pares
3. imprimir del 1 al 40 elevado al cuadrado
4. recibir por get dos numero y hacer operaciones basicas
5. rango de numeros e imprimir
6. tablas del multiplicar en html
7. imprimir impares y pares con modulo