

Fundamentos

| | |
|--|-------------------------------------|
| inicio de script | <code><?php</code> |
| fin de script | <code>?></code> |
| fin de comando | <code>;</code> |
| comentario en línea | <code>// comentario simple</code> |
| comentario en bloque | <code>/* múltiples líneas */</code> |
| ejecutar script de archivo (arroja advertencia si hay error) | <code>include('script.php');</code> |
| ejecutar script de archivo (los errores son fatales) | <code>require('script.php');</code> |

Variables

| | |
|----------------------------------|---|
| declaración sin valor | <code>\$variableName;</code> |
| variable de tipo cadena | <code>\$string = "some text";</code> |
| variable de tipo entero | <code>\$int = 123;</code> |
| variable de tipo double | <code>\$doble = 1.2</code> |
| declaración de arreglo vacío | <code>\$array = array();</code> |
| declaración de arreglo con datos | <code>\$a = array("text",1,2.3);</code> |
| variable de tipo lógica | <code>\$bool = true;</code> |
| declaración nula | <code>\$empty = null;</code> |

Salida a pantalla

| | |
|--|---|
| salida simple ej.1 | <code>echo(\$variable . " y algo de texto");</code> |
| salida simple ej.2 | <code>echo 1;</code> |
| salida simple ej.3 (tiene que ir sin paréntesis) | <code>echo 1,2; // 12</code> |

Salida a pantalla

| | |
|--|--|
| salida en múltiples líneas | <code>echo("línea1\nlínea2\n");</code> |
| imprime el tipo de variable y los datos | <code>print_r(\$var);</code> |
| imprime el tipo de variable y los datos (detallado). | <code>var_dump(\$var);</code> |

Operadores

| | |
|---------------------------------------|-------------------------------------|
| suma | <code>\$int1 + \$int2</code> |
| división | <code>6 / 3 // 2</code> |
| multiplicación | <code>6 * 6 // 36</code> |
| exponente | <code>2 ** 4 // 16</code> |
| resta | <code>2 - 1 // 1</code> |
| string + int (convierte la cadena) #1 | <code>"13abcd" + 13 // 26</code> |
| string + int (convierte la cadena) #2 | <code>"abcd13" + 13 // 13</code> |
| string + string | <code>"2abcd" + "5abcd" // 7</code> |
| concatenación | <code>\$str1 . \$str2</code> |
| Concatenación cadena y entero | <code>"número uno - " . 1;</code> |
| unset - borra variable | <code>unset(\$variable);</code> |

Ciclos

| | |
|---|--|
| for - ciclo clásico (estado inicial; condición de paro; incremento o decremento) | <code>for (\$i = 1; \$i <= 10; \$i++) { echo \$i; }</code> |
| foreach - realiza una tarea por cada elemento de la colección o arreglo | <code>foreach(\$array as \$arrayElement){ var_dump(\$arrayElement); }</code> |
| while - realiza una tarea si la condición es verdadera y repítela mientras lo siga siendo. | <code>while(true){ echo("infinito!"); }</code> |

Ciclos

| | |
|--|------------------------|
| break - rompe el ciclo | <code>break;</code> |
| continue - se brinca la actual iteración. | <code>continue;</code> |

Control de flujo - IF

| | |
|---|---|
| condición simple | <code>if (true){ echo("condition pass"); }</code> |
| ¿es \$var1 igual a \$var2? | <code>if (\$var1 == \$var2){ echo("son iguales"); } else { echo("no son iguales"); }</code> |
| comparación de dos variables | <code>(\$var1 == \$var2)</code> |
| ¿es \$var1 igual a 4? | <code>(\$var1 == 4)</code> |
| ¿es \$var1 menor que 4? | <code>(\$var1 < 4)</code> |
| ¿es \$var1 mayor que 4? | <code>(\$var1 > 4)</code> |
| ¿es \$var1 menor que 5 y diferente de cero? | <code>(\$var1 < 5 && \$var1 != 0)</code> |
| ¿es \$str igual a "texto"? | <code>(\$str == "texto")</code> |
| ¿es \$num 1 o 4? | <code>(\$num == 1 \$num == 4)</code> |
| ¿es \$num1 igual a "4" y del mismo tipo? | <code>\$num1 === "4" // falso si \$num es 4</code> |

Operaciones con Arreglos

| | |
|--|--|
| obtiene un elemento del arreglo por su índice | <code>\$array[index]</code> |
| agrega un elemento al final del arreglo | <code>array_push(\$array, \$data); OR \$array[] = \$data;</code> |
| agrega un elemento en la posición indicada del arreglo | <code>\$array['myColor'] = "#FF- F";</code> |
| elimina y obtiene el último elemento | <code>array_pop(\$array);</code> |

Operaciones con Arreglos

| | |
|---|---|
| cuenta el número de elementos del arreglo | <code>count(\$array);</code> |
| verifica si existe \$dato en el arreglo | <code>in_array(\$dato, \$array);</code> |

Constantes predefinidas

| | |
|---|---|
| Información del servidor y del entorno de ejecución. | <code>\$_SERVER</code> |
| IP del servidor | <code>\$_SERVER['SERVER- ADDR']</code> |
| Obtiene el método http que se uso para acceder a la página; ej. 'GET', 'HEAD', 'POST', 'PUT' | <code>\$_SERVER['REQU- EST_METHOD']</code> |
| Encabezado User-Agent de la petición actual | <code>\$_SERVER['HTTP- _USER_AGENT']</code> |
| Dirección IP del usuario actual | <code>\$_SERVER['REMO- TE_ADDR']</code> |
| Nombre de archivo y ruta completa del script actual | <code>__FILE__</code> |
| Directorio del script actual | <code>__DIR__</code> |
| Línea de código actual en el script | <code>__LINE__</code> |
| Arreglo asociativo con los valores recibidos vía parámetros de URL. | <code>\$_GET</code> |
| Arreglo asociativo con los valores recibidos por vía del método POST de HTTP, cuando se utiliza la opción application/x-www-form-urlencoded ó multipart/form-data como tipo de contenido en la solicitud. | <code>\$_POST</code> |
| Arreglo asociativo con los archivos subidos por el script actual vía el método POST. | <code>\$_FILES</code> |



Control de flujo - Switch

| | |
|--------------------------------|--|
| Condición con múltiples casos. | <pre><?php switch (\$i) { case 0: echo "cero"; break; case 1: echo "uno"; break; case 2: echo "dos"; break; default: echo "otro"; }</pre> |
|--------------------------------|--|

Control de flujo - Estructuras alternativas

| | |
|-------------------|--|
| Condición IF | <pre><?php if (\$a == 5): ?> A es igual a 5 <?php endif; ?></pre> |
| Condición IF-ELSE | <pre><?php if (\$a == 'auth'): ?> <p>Usuario autenticado</p> <?php else: ?> <p>Login</p> <?php endif; ?></pre> |
| FOREACH | <pre><?php foreach (\$productos as \$aactual): ?> <td>\$actual->nombre</td> <td>\$actual->codigo</td> <?php endforeach; ?></pre> |
| FOR | <pre> <?php for (\$i=0; \$i<10; \$i++): ?> \$i <?php endfor; ?> </pre> |
| WHILE | <pre><?php while(\$var<10): ?> ... <?php endwhile; ?></pre> |

Control de flujo - Estructuras alternativas

| | |
|--------|---|
| SWITCH | <pre><?php switch(\$variable): case 1: ?> <div>Inicio</div> <?php break;?> <?php case 2: ?> <div>Foro</div> <?php break;?> <?php endswitch;?></pre> |
|--------|---|

Operadores de comparación

| | |
|-------------|--------------------------|
| \$a == \$b | Equal |
| \$a === \$b | Identical |
| \$a != \$b | Not equal |
| \$a <> \$b | Not equal |
| \$a !== \$b | Not identical |
| \$a < \$b | Less than |
| \$a > \$b | Greater than |
| \$a <= \$b | Less than or equal to |
| \$a >= \$b | Greater than or equal to |

Operadores Lógicos

| | |
|-------------|-----|
| \$a and \$b | And |
| \$a or \$b | Or |
| \$a xor \$b | Xor |
| ! \$a | Not |
| \$a && \$b | And |
| \$a \$b | Or |