

Funciones PHP 8.2

Antonio Espín Herranz

Contenidos

- Funciones sobre cadenas.
- Funciones sobre Arrays.
- Funciones sobre fechas y horas.
- Conversiones de tipos
- De uso general.
- Expresiones Regulares

Funciones del Lenguaje (cadenas)

- Cadenas de caracteres:
 - `string chr(int valor)`: Devuelve una cadena que representa el carácter correspondiente en la tabla de códigos ASCII. De 0 a 255.
 - `int ord(string cadena)`: La función inversa.
- Visualizar cadenas:
 - `echo string`
 - `echo "el valor de las variables es $var"` → expande la variable.
 - La suma de `<?=$valor1 ?>` y `<?=$valor2?>` es ...

Funciones del Lenguaje (cadenas)

- Visualizar cadenas:
 - `print (string cadena);`
 - `print("El valor del número es: $numero
");`
 - `printf(string formato [, valores, ...]);`
 - `printf("%'03d * '%02d = '%05d
\n", 13, $i, 13 * $i);`
013 * 01 = 00013
013 * 02 = 00026

Funciones del Lenguaje (cadenas)

- `string chop(string cadena)`: Elimina los caracteres en blanco y `\n`, se puede usar después de leer de un fichero.
- `ltrim`, `rtrim`, `trim`: recortan blancos.
- `str_pad(string cadena, int longitud [, string relleno [, int lugar]])`:
 - Rellena la cadena, hasta alcanzar la longitud con el relleno indicado y en un lugar (`STR_PAD_RIGHT`, `STR_PAD_LEFT`, `STR_PAD_BOTH`)

Funciones del Lenguaje (cadenas)

- `string str_repeat(string cadena, int veces)`: repite tantas veces el carácter indicado.
- `strtolower(string cadena)`: convierte a minúsculas.
- `string strtoupper(string cadena)`: convierte a mayúsculas.
- `str_replace(subcadena1, subcadena2, cadena)`: reemplaza subcadena2 por subcadena1 dentro de cadena.

Funciones del Lenguaje (cadenas)

- Localizar subcadenas y devuelven pos:
 - `int strpos(string cadena, string subcadena [, int posición])`: localizar dentro de una cadena por el principio.
 - `int strrpos(string cadena, string subcadena)`: localizar por el final de la cadena.
- Localizar caracteres y devuelve string:
 - `string strpos(string cadena, char carácter)`: la primera aparición.
 - `string strrchr(string cadena, char caracter)`: la última aparición.

Funciones del Lenguaje (cadenas)

- `int strlen(string cadena)`: Devuelve la longitud de la cadena.
- `string substr(string cadena, int comienzo [, int longitud])`: Extraer una subcadena.
- `int strcmp(string cadena1, string cadena2)`
 - retorna 0 si `cad1 == cad2`
 - retorna ≥ 1 si `cad1 > cad2`
 - retorna ≤ -1 si `cad1 < cad2`
- `int strcasecmp(string cadena1, string cadena2)`: idem de la anterior con case sensitive.

Funciones del Lenguaje (cadena)

- Para partir cadenas en trozos:
 - `String chunk_split(string cadena [, int longitud [, string separador]])`;
 - Añade un carácter de separación cada cierto número de caracteres, por defecto longitud → 76 y el carácter separador `\r\n`. No modifica el original.
 - `array explode(string separador, string cadena [,int limite])`;
 - Parte una cadena en trozos, metiendo cada trozo en una posición del array, parte por el separador, podemos establecer un limite.
 - `string implode(string separador, array elementos)`;
 - La contraria a la anterior. A partir del array monta la cadena y conecta los elementos mediante el separador.

Funciones del Lenguaje (cadena)

- `void parse_str(string cadena);` Permite crear variables que forman parte de una cadena que se corresponde con 'query string'.
- `$cadena = "nombre=agustin&dni=125556"`
- `parse_str($cadena);`
 - Crea las variables `$nombre` y `$dni`.
 - Con los valores `agustin` y `125556`.

Funciones del Lenguaje (tipos)

- `string gettype($var) →` Devuelve el tipo de la variable (integer, string, boolean, array, object, null).
- `$otra_var = settype($var, tipo) →` Convierte al tipo indicado la variable.
- Casting o moldeo de tipos:
 - `$numero = 3;`
 - `$numero2 = (int) ($numero / 2);`
 - `$unArray = (array)$numero;`

Funciones del Lenguaje (tipos)

- Devuelve true en caso de que la variable sea del tipo por el que se pregunta:
 - `boolean is_array(variable);`
 - `boolean is_bool(variable);`
 - `boolean is_double(variable);`
 - `boolean is_float(variable);`
 - `boolean is_int(variable);`
 - `boolean is_integer(variable);`
 - `boolean is_long(variable);`
 - `boolean is_null(variable);`
 - `boolean is_numeric(variable);`

Funciones del Lenguaje (definición)

- `bool isset(variable)` Comprueba si está definida la variable.
 - Podemos utilizarla para validación de formularios, preguntando si hemos recibido o no algún campo del mismo.
 - `if (isset($_POST['login']))`
 - ...

Funciones del Lenguaje (arrays)

- Creación de arrays:
 - `array()`: crea un array vacío.
 - `array(1,2,3,5);`
 - Podemos indicar su índice y valor:
 - `array(10 => 'diez', 20=>'veinte');`
- Contar: `count()` devuelve el número de elementos del array.
- `print_r($un_array)`: Devuelve todo el contenido del array.

Funciones del Lenguaje (arrays)

- Posicionamiento en arrays:
 - `reset($matriz)`: Se posiciona en el 1 er elemento.
 - `next($matriz)`: Avanza al siguiente elemento.
 - `prev($matriz)`: Retrocede 1.
 - `end($matriz)`: En el último.
 - `current($matriz)`: El valor actual.
 - `key($matriz)`: La clave actual.

Funciones del Lenguaje (arrays)

- `array_walk($matriz, 'nombre_funcion')`: recorre y aplica la función a todos los elementos del array.
- Ejemplo: La función debe recibir dos parámetros,

```
function miFuncion($valor, $indice){  
    echo "$indice = $valor";  
}
```

```
array_walk($matriz, 'miFuncion' );
```


Funciones del Lenguaje (arrays)

- Otro ejemplo, modificando los elementos del array:

```
function modificar_elemento(&$valor, $indice){  
    $valor = $valor + 5;  
}
```

```
array_walk($matriz, 'modificar_elemento');
```

Funciones del Lenguaje (arrays)

- `unset`: Para eliminar elementos del array.
- `unset($matriz['unIndice']);` Elimina el elemento del array y lo recoloca.
- `unset($matriz);` Destruye la matriz.
- Si queremos simplemente vaciarla, podemos usar `array()`;

Funciones del Lenguaje (arrays)

- `in_array(valor_buscado, array, [estricto])`: localizar elementos dentro del array.
- Si `estricto` es `true` hace que las búsquedas de valores tengan que coincidir en tipo.
- `$matriz = array(1,2,3,4);`
- `$encontrado1 = in_array('2', $matriz); // true.`
- `$encontrado2 = in_array('2', $matriz, true); // false`

Funciones del Lenguaje (arrays)

- `array_search(valor, array, [con tipo])` Devuelve el índice del valor buscado. Con tipo indica que también compare por tipo.
- `array_key_exists(indice, array)` : Comprueba si existe dentro del array el índice pasado por argumento.

Funciones del Lenguaje (arrays)

- `array_rand($matriz [,numero])`: Devuelve un índice aleatorio de un array.
 - Con `numero` indicamos el número de claves que queremos que nos devuelva. En este caso el resultado se almacena en un array.
- `array_walk_recursive`: Idem de `array_walk` pero se aplica de forma recursiva.
- `array_keys($matriz)`: Devuelve un array con los nombres de las claves.

Funciones del Lenguaje (arrays)

- Insertar elementos en el array:
 - Podemos usar la forma dinámica:
 - `$matriz[] = "valor";`
 - `int array_push($array, elem1 [,elem2]);` Añade al final, pueden ser varios y devuelve el nuevo número de elementos.
 - `int array_unshift($array, elem1 [,elem2]);` Añade al principio. Los índices numéricos se reasignan los de texto no.
 - `array array_pad(array entrada, int tamaño, relleno);`
 - Rellena la matriz por la izquierda o derecha según tamaño sea > 0 o < 0 , si el número de elementos \geq que `abs(tamaño)` no añade nada.

Funciones del Lenguaje (arrays)

- Eliminar elementos:
- `mixed array_pop(array)`: Elimina el último elemento y lo devuelve. Si no hay elems, devuelve NULL.
- `mixed array_shift(array)`: Elimina el primer elemento. Si no hay devuelve NULL.
- Para eliminar una parte o sustituir, tenemos:
 - `array array_splice(array matriz, int posicion [, int tamaño, [array_sustituidos]])`; Devuelve la porción eliminada o sustituida.
 - Posición: indica desde donde se va a operar > 0 por la izq., < 0 por la derecha.
 - Tamaño: cuanto, si no se indica con $posicion > 0$ desde, < 0 hasta.
 - Sustituidos: los valores de sustitución.

Funciones del Lenguaje (arrays)

- `array_reverse($matriz)`: da la vuelta a la matriz.
- `array_change_key_case($matriz, tipo)`: Cambia los índice de mayúsculas a minúsculas. Tipo: `CASE_UPPER` / `CASE_LOWER`.
- `array_flip($matriz)`: intercambia claves por valor.

Funciones del Lenguaje (arrays)

- Partes de matriz:
 - `array array_filter($matriz, 'funcion_filtro')`: Podemos definir una función de filtro, que reciba un valor y comprobar una condición. Cuando se devuelve true le incorpora a la matriz.
 - `array array_slice($matriz, int posicion, int tamaño)`: Extrae un trozo de matriz, desde posición (si es < 0 , empieza por el final), tamaño: el que sea.
 - `array array_chunk($matriz, numero, con_indices)`: Permite dividir la matriz en submatrices, obtenemos un array de dos dimensiones. Numero marca el número de submatrices, `con_indices`: mantiene los índices.

Funciones del Lenguaje (arrays)

- Ordenar matrices:
 - `bool sort (array $matriz [, int criterio]);`
 - Ordena pero no conserva índices. Criterio: `SORT_REGULAR` (según reglas PHP), `SORT_NUMERIC` (según números), `SORT_STRING` (según cadenas).
 - Ojo no conserva índices.
 - `bool rsort()` : igual que la anterior pero descendentemente.

Funciones del Lenguaje (arrays)

- Para ordenar y conservar índices:
 - `asort()` y `arsort()`
- Para ordenar con un criterio propio:
 - `usort($matriz, 'funcion')`
 - La función criterio recibirá dos parámetros.
- Para ordenar por claves:
 - `krsort(array, [int criterio])` → Ascendente.
 - `ksort(array, [int criterio])` → Descendente.
- `array file("nombreFichero.txt");` Carga el contenido del fichero en un array. Para limpiar luego cada dato, usar la función `chop()`.

Operadores sobre arrays

- `+` `$matriz1 + $matriz2`
 - Realiza la unión.
- `==` `$matriz1 == $matriz2`
 - ¿Son iguales? Mismas parejas claves / valor.
- `===` `$matriz1 === $matriz2`
 - Idénticas.
- `!=` o `<>` `$matriz1 != $matriz2`
 - No son iguales.
- `!==` `$matriz1 !== $matriz2`
 - No son idénticas.

Funciones del Lenguaje (Fechas)

- Comprobar si una fecha es correcta:
 - `checkdate(mes, dia, año)` Devuelve true si la fecha es correcta.
- `getdate()`:
 - Devuelve un array con la información de la fecha:
 - `seconds`, `minutes`, `hours`, `mday` (día del mes), `wday` (día de la semana), `mon` (mes en número), `year`, `yday` (día del año), `weekday`, `month`.
 - `$fecha = getdate();`
 - `print ("La hora es: " . $fecha["hours"]);`

Funciones del Lenguaje (Fechas)

- `mktime (hora, min, seg, mes, dia, año)`
 - Devuelve una variable con los valores pasados por argumento.
- `time()`
 - Devuelve el nº de segundos transcurridos desde el 1/1/1970.
- `date()`
 - Devuelve una cadena formateada según los códigos de formato.
 - `date("d-m-Y")` → 23-04-2003.
 - `date("d-m-y")` → 23-04-03
 - `date("Y-m-d", mktime(0,0,0, 3, 25, 2008));`

Funciones del lenguaje

- `boolean empty(var)`: Devuelve FALSE si var está definida y tiene un valor no-vacío o distinto de cero; en otro caso devuelve TRUE.
- `boolean isset(var)`: Devuelve true si la variable está definida.

Funciones del lenguaje

- **var_dump(\$variable)**
 - Se puede utilizar para depurar, explora una variable indicando: tipo, valor, etc.
 - Imprime toda la información de la variable.
- **phpinfo()**
 - Para comprobar la instalación de PHP.
- **header('location: pagina.php')**
 - Redirigir a otra página.
- **exit()**
 - Corta la ejecución del script.

Expresiones Regulares