U.T. 3: Fundamentos de la inserción de código en páginas web (II): Cadenas

Cadenas en PHP

- *PHP nos ofrece cientos de funciones predefinidas:*
  - *Unas pertenecen al núcleo de PHP*
  - *Otras las aportan los módulos que cargamos sobre el intérprete PHP*
- *http://php.net/manual/es/funcref.php*
- *Leer muy cuidadosamente la declaración de las funciones:*
  - *bool shuffle (array &$array)*

Cadenas en PHP: Delimitación

- *Delimitador " ": inserta el valor de las variables*
- *Delimitador ' ': no inserta el valor de las variables.*
- *Escape: \$, \\, \', ….*
- *Tipo 'heredoc' de Perl: se comporta como "".*
- *$cadena = <<<DELIMITADOR*

text

text

……

DELIMITADOR

- *Concatenación con el operador punto '.'*
- *Se puede acceder a un carácter de una cadena con [ ] (empieza en cero)*

Cadenas en PHP: Delimitación

- Con comillas dobles:
  - $cad1 = “hola.\n\$var1=$var1”;
- Sintaxis ‘ heredoc’ de Perl:

  $cad2 = <<<FICAD

  Esto es un ejemplo de cadena.

  La variable \$a vale $a.

  Ahora termina la cadena:

  FICAD
- Concatenación:
  - $cad1 = $cad1 . $cad2;
- Acceso a caracteres:
  - $caracter5 = $cad1[4];

Cadenas en PHP: Visualización

- **Echo y print**
- No permiten formatear la salida
- print $variable == echo $variable.
  - echo "hola1","hola2"; → admite parámetros
  - print ("hola1","hola2"); → error
- **printf (formato [, argumentos])**
  - Imprime una cadena con formato
  - La cadena de formato compuesta por :
    - Caracteres ordinarios (excluyendo %) se copian directamente
    - Especificaciones de conversión: dan el formato a los argumentos
- **sprintf** : Igual que printf pero devuelve una cadena formateada.

Cadenas en PHP: Visualización

- **printf("%02d/%02d/%04d", $dia, $mes, $anno)**
  - $dia= 5;
  - $mes=3;
  - $anno=12;
  - Escribe: 05/03/0012

```
$pago1 = 68.75;
$pago2 = 54.35;
$pago = $pago1 + $pago2;
//echo $pago mostraría "123.1"
//Mostrar al menos un dígito entero y
//exactamente 2 decimales, rellenando
//con ceros
printf ("%01.2f", $pago); // Muestra: 123.10
```

% → carácter %
u → número decimal sin signo, se trata como un entero
d → Decimales, como entero decimal
b → Binarios, como entero binario
o → Octales como entero
x → Hexadecimales (letras minúsculas) como entero
X → Hexadecimales (letras mayúsculas) como entero
c → Caracteres ASCII como carácter
f → Punto flotante (signo decimal) como float o decimales
e → Punto flotante (notación exponencial) como float o decimales
s → Cadenas como string

Cadenas en PHP: Visualización

- $number= 255.8;
- printf("En decimal: %d
",$number);
  - //"En decimal: 255"
- printf("En hexadecimal: %x",$number);
  - //"En hexadecimal: ff"
- printf("En hexadecimal: %X",$number);
  - //"En hexadecimal: FF"
- printf("En octal: %o",$number);
  - //"En octal: 377"
- printf("Decimal exponencial: %e",$number);
  - //"Decimal exponencial: 2.55000e+2"
- printf("En binario: %b",$number);
  - //"En binario: 11111111"

Cadenas en PHP : Búsqueda

- ***strstr(cadena, cadBusq, false)***
  - *Devuelve desde la **primera** aparición de la cadena (**incluyéndola - false**) hasta el final. Busca coincidencia total. Case-sensitive.*

strstr("carmenduran@domenicoscarlatti.es","@domenico");

// @domenicoscarlatti.es

- ***strrchr(cadena, carBusq)***
  - *Devuelve desde la **última** aparición del 1er carácter (**incluido**) hasta el final (Sólo se utiliza el primer carácter). Case-sensitive*

strrchr ("Esto es muy bonito","so"); // s muy bonito

- ***stristr(cadena, carBusq)***
  - *Igual que strstr pero no es case-sensitive.*

Cadenas en PHP : Búsqueda

- ***strpos (cad1, cad2 [, desplz])***
  - *Busca la primera posición de aparición de una cadena a partir de desplz (por defecto 0). Coincidencia total.*
  - *Si no la encuentra devuelve FALSE*

strpos ("Este espacio es muy bonito","es",7); // 13

- ***strrpos (cadena, cadena [, desplz])***

Devuelve la posición de la última aparición o FALSE. Coincidencia total

strrpos ("Este espacio es muy bonito","es"); // 13

Cadenas en PHP : Búsqueda

- ***strspn (cadena, máscara, comienzo, longitud)***
  - *Devuelve la longitud del segmento inicial de un string que consiste únicamente en caracteres contenidos dentro de una máscara dada. Case-sensitive*

strspn ("Este espacio es muy bonito","Estela"); // 4

strspn ("Este espacio es muy bonito","Estado"); // 3

strspn ("Este espacio es muy bonito","el");// 0

- ***strcspn (cadena, máscara, comienzo, longitud)***
  - *Devuelve la longitud de la subcadena más larga que está formada sólo por caracteres **no** contenidos en la máscara. Case-sensitive*

Cadenas en PHP : Comparación

- ***strcmp (cad1, cad2)***
  - *Compara dos cadenas (case-sensitive) y devuelve:*
    - *> 0, si cad1>cad2*
    - *< 0, si cad1<cad2*
    - *0, si ambas cadenas son iguales.*

strcmp("Hola","hola") → no iguales
strcasecmp("Hola","hola") → iguales

- ***strcasecmp(cad1, cad2)***
  - *Igual que strcmp() pero insensible a mayúsculas y minúsculas.*
- ***strncmp (cad1, cad2,longitud)***
  - *Sólo compara los longitud primeros caracteres*

strncmp("Paco","Paca",4) ⟶ distintos
strncasecmp("Paco","Paca",2) ⟶ iguales

Cadenas en PHP : Comparación

- ***strnatcmp (cad1, cad2)***
  - *Comparación "natural“*
    - *con strcmp(10,2) 10 es menor que 2*
    - *con strnatcmp(10,2) 10 es mayor que 2*

Cadenas en PHP : Operación

- ***int strlen (string $cadena)***
  - *Devuelve el número de caracteres que contiene una cadena.*

```
$cadena="Hola que tal";
echo strlen($cadenda);  //  12
```

Cadenas en PHP : Operación

- ***substr ($cadena, $inicio, $lon)***
  - *Devuelve una subcadena de longitud $lon a partir de la posición $inicio de la cadena $cadena.*

```
$cad="PATITOS";
echo substr($cad,2);      // TITOS
echo substr($cad,2,3);    // TIT
echo substr($cad,-2);     // OS
echo substr($cad,2,-3);   // TI  2º nº negativo = caracteres del
                          final que no se quitan
```

Cadenas en PHP : Operación

- ***substr_replace($cadena, $reemplazo, $ini, $charsABorrar)***
  - *Devuelve una cadena, resultado de reemplazar con el string dado en reemplazo, una copia de cadena empezando por el parámetro inicio hasta (opcionalmente) charsABorrar o hasta el final de cadena.*
  - *La cadena original no sufre ninguna modificación. Case sensitive*

```
$texto="Hola a todos los chicos";
$texto2=substr_replace($texto,"todas", 0) // todas
$texto2=substr_replace($texto,"todas", 7) // Hola a todas
$texto2=substr_replace($texto,"todas", 7,2)// Hola a todasdos los chicos
```

Cadenas en PHP : Operación

- ***str_replace ($cadBusq,$cadReempl,$texto)***
  - *Devuelve una cadena resultado de sustituir $cadBusq en $texto por $cadReempl.*
  - *La cadena original no sufre ninguna modificación.*
- ***strtr (cadena, cadBus, cadRee)***
  - *Sustituye carácter a carácter*
  - *La cadena original no sufre ninguna modificación.*

```
echo strtr("Hola a todos los presentes aquí","aeiou", "AEIOU");
HOlA A tOdOs lOs prEsEntEs AquÍ
```

  - *En lugar de dos argumentos se puede pasar un array.*

Cadenas en PHP : Operación

- ***substr_count ($cadena, $patron, $inicio, $longitud)***
  - *Devuelve el número de apariciones de $patron dentro de la cadena $cadena, (opcionalmente) empezando desde la posición $inicio hasta $longitud. Es case-sensitive.*

```
$cad="Hola a todos los presentes";
echo substr_count($cad,"as");          // 2
```

- ***mixed count_chars (string $string [, int $mode = 0])***
  - *Devuelve información sobre los caracteres usados en una cadena*
  - *Cuenta el número de apariciones de cada byte-value (0..255) en string y lo devuelve de varias maneras en función del parámetro opcional mode.*

Cadenas en PHP : Modificar

- ***rtrim($cadena, $listaChar)***
  - *Elimina los espacios en blanco y caracteres de fin de línea {" ", "\n", "\t", "\r", "\0", "\x0B" } del final de la cadena. Permite incluir una lista $listaChar con los caracteres a eliminar del final. Devuelve la cadena modificada.*
- ***chop($cadena)***
  - *Alias de rtrim().*
- ***ltrim(), trim()***
  - *Eliminan caracteres a la izquierda o por ambos lados de una cadena.*

Cadenas en PHP : Modificar

- ***str_pad($cadena, $long, $cadenaRelleno)***
  - *Rellena una cadena con un carácter de relleno (por defecto, es el espacio en blanco) hasta obtener $long caracteres en total.*
  - *Si $long es negativo o menor que la longitud de la cadena no rellena.*
  - *Parámetros: STR_PAD_RIGHT, STR_PAD_LEFT, STR_PAD_BOTH*
    - *Indican la opción para rellenar.*

Cadenas en PHP : Modificar

- ***strtolower($cad) / strtoupper($cad)***
  - *Devuelve una **string** con todos los caracteres alfabéticos convertidos a minúsculas/ mayúsculas.*
- ***ucfirst($texto)***
  - *Pone en mayúscula solo la 1ª letra de $texto*
- ***ucwords($texto)***
  - *Pone en mayúscula la 1ª letra de cada palabra contenida en $texto.*

Cadenas en PHP : Modificar

- ***addslashes($cadena)***
  - *Antepone barras de escape \ a: ("), (') , el NULL (el byte nulo) y (\).*
- ***stripslashes($cadena)***
  - *Quita barras de escape de: (\') .*
- ***nl2br($cadena)***
  - *Todos los saltos de línea serán convertidos a etiquetas
.*

echo nl2br($row['campo_BD']);

- ***addcslashes($cadena, $listcar)***
  - *Escapa cualquier carácter de $listcar.*
- ***stripcslashes($cadena)***
  - *Contraria a addcslashes().*

```
addcslashes("Hola","a,b");  // Hola\a
```

Cadenas en PHP : Modificar

- ***quotemeta($cadena)***
  - *Coloca una barra \ delante de : . , \, +, *, ? , [, ^,] , (, $,)*
- ***string htmlspecialchars ($string ,$flags , $codigo)***
  - *Traducen a entidades html*
  - *Las traducciones realizadas son:*
    - *'&' se convierte en '&'*
    - *‘ “ ‘ (comillas dobles) se convierte en '"‘ cuando **ENT_NOQUOTES** no está establecido.*
    - *“ ‘ " (comilla simple) se convierte en ''' (o ') sólo cuando **ENT_QUOTES** está establecido.*
    - *'<' (menor que) se convierte en '<'*
    - *'>' (mayor que) se convierte en '>‘*

Cadenas en PHP : Modificar

- ***string htmlentities ($string ,$flags , $codigo)***
  - *Esta función es idéntica a htmlspecialchars() en todos los aspectos, excepto que con htmlentities(), todos los caracteres que tienen equivalente HTML son convertidos a esas entidades.*
- ***Htmlspecialchars_decode()***
- ***Html_entity_decode()***

Cadenas en PHP : Modificar

- ***strtok($cadena, $delimitador)***
  - *Divide una cadena en subcadenas separadas por $delimitador. Devuelve la primera subcadena obtenida. Invocar desde un bucle para dividir toda una cadena.*

strtok("hola que tal"," ") → divide en palabras

- ***chunk_split($cadena [, $long [, $separador]])***
  - *Inserta la cadena separador cada $long caracteres*

chunk_split("hola que tal",3,'-'); → hol-a q-ue -tal

Cadenas en PHP : Modificar

- ***array str_split (string $string [, int $split_length = 1])***
  - *Convierte un string en un array.*
  - *Si el parámetro opcional split_length se específica, el array devuelto será separado en fragmentos los cuales cada uno tendrá una longitud de split_length, de otra manera cada fragmento tendrá una longitud de un carácter.*
- ***array explode (string $delimiter ,string $string [, int $limit])***
  - *Parte la cadena en función de $delimiter y crea un array con los elementos.*

```
$dir_correo="prueba@yahoo.com.ar";
$parte_array=explode("@",$dir_correo);
echo $parte_array[0]; // prueba
echo $parte_array[1]; // yahoo.com.ar
```

Cadenas en PHP : Modificar

- ***string implode (string $delimitador , array $matriz)***

  - *Convierte un array en una cadena de caracteres separando sus elementos con la cadena indicada en $delimitador.*

```
$matriz=array(7,'julio',2011);
echo implode(' de ',$matriz);

$unidos=implode("@",$parte_array);
```

Cadenas en PHP : Validación

- ***filter_var ($variable, $filtro)***
  - *Valida el contenido de la variable según el filtro indicado en la constante $filtro:*
  - *Algunos tipos de filtro:*

FILTER_VALIDATE_EMAIL: Valida emails según RFC 822

FILTER_VALIDATE_FLOAT: Valida números reales

FILTER_VALIDATE_IP : Valida IP's

FILTER_VALIDATE_URL : Valida URL's según RFC 2396