

# Funciones del lenguaje

## Funciones para el tratamiento de los numeros

### 1. Redondear un numero

- round()

Ejemplo redondeo hacia arriba 5.6

Ejemplo redondeo hacia abajo 2.3

La funcion acepta un float o un int pero si es float, solo se muestra el int si añadimos una coma y otro numero despues esto indica el numero de decimales que queremos que se muestren y será el redondeo. Si ademas añadimos un tercer parametro es una constante, que indica como queremos que se redondee el numero, por defecto redondea hacia arriba.

- ceil()

Ejemplo de redondeo hacia el entero superior 5

Solo tiene un parametro de entrada y lo que hace es redondear hacia arriba

- floor()

Ejemplo de redondeo hacia el entero inferior 4

solo tiene un parametro de entrada y lo que hace es redondear hacia abajo

### 2. Valor absoluto

Funcion abs()

Este ejemplo es un -5 y da como resultado: 5

Calcula el valor absoluto del numero sin el signo

### 3. Calcular potencias

Funcion pow()

En este ejemplo ponemos pow(2,5), que es 2 elevada a la 5 y da como resultado: 32

Esta funcion pide dos parametros, un numero que sera la base y el segundo sera el exponente de dicho numero

### 4. Calcular raices

funcion sqrt()

La raiz cuadrada de 16 es: 4

La raiz cuadrada de 7 es: 2.6457513110646

Pero si usamos round tambien es: 2.65

Solo admite un parametro y calcula su raiz cuadrada si ponemos un numero que su raiz no de un numero entero exacto como en el segundo ejemplo, podemos unir la funcion sqrt con la funcion round para que lo redondee

### 5. Generar aleatorios

Funcion rand()

devuelve un numero aleatorio: 860033413

devuelve un numero aleatorio entre el 1 y el 10: 8

se pueden poner hasta dos parametros, pero ambos son opcionales, es decir, que si lo ejecutamos sin ningun parametro, devuelve un numero aleatorio. Cada vez que recargues la pagina ese numero sera diferente. Es posible indicar un rango en el que debe de estar ese numero aleatorio, por ejemplo del 1 al 10 como en el ejemplo 2.

## 6. Dar formato a un Número

Funcion `number_format()`

Para el numero 1005.4562 con 2 decimales y la , para separar los decimales y el . para delimitar los miles nos da: 1.005,46

Con esta funcion podemos cambiar los parametros de las comas flotantes y los . para los miles asi como indicar el numero de decimales que queremos que se muestren

# Funciones para el tratamiento de cadenas

## 1. Contar en una cadena

- o Caracteres

funcion `strlen()`

En la frase Hola Pipsicola hay estos caracteres: 14

cuenta todos los caracteres del string hay que tener en cuenta que tambien contabiliza los espacios en blanco como un caracter mas

- o Palabras

funcion `str_word_count()`

De nuevo en la frase Hola Pipsicola hay estas palabras: 2

esta funcion sirve para contar las Palabras que hay en un string, en este caso no tiene en cuenta los espacios en blanco.

## 2. Invertir la cadena

funcion `strrev()`

La frase Hola Pipsicola al revés es: aloCispiP aloH

Esta funcion invierte toda la cadena caracter por caracter, quedando totalmente al reves que la cadena original

## 3. Repetir la cadena

funcion `str_repeat()`

Vamos a repetir el simbolo \* 10 veces: \*\*\*\*\*

en los parentesis se colocan dos parametros el primero es el caracter que queremos que se repita y el segundo es el numero de veces que queremos que se repita .

## 4. Dividir la cadena

funcion `wordwrap()`

Dividimos por cada 2 caracteres la palabra Pipsicola:

Pi  
ps  
ic  
ol  
a

Esta funcion admite cuatro parametros El primero indica el string que queremos Dividir El segundo indica cada cuantos caracteres el tercero con que lo vamos a dividir el cuarto por defecto esta en false asi que lo podemos activar poniendo true

## 5. Encontrar un caracter

funcion `strpos()`

vamos a encontrar la posicion de la palabra buenos en la frase Hola, buenos dias: 6

esta funcion admite dos parametros, el primero es el string completo que queremos analizar y el segundo es la palabra o el caracter del que queremos descubrir la posicion de su primera aparicion, nos

devuelve el numero entero de la posicion

## 6. Reemplazar un caracter

funcion `str_replace()`

Vamos a sustituir las letras i de la palabra Pipsicola por la letra e: Pepsecola

en esta funcion tenemos que pasar 3 parametros, el primero es el caracter del string que queremos reemplazar, el segundo es el caracter por el que lo queremos reemplazar y por ultimo el string dentro del que se encuentra ese caracter

# Funciones para el tratamiento de arrays

## 1. Agregar elementos

funcion `array_push()`

Primer ejemplo array original:

Array ( [0] => 3 [1] => 5 )

Segundo ejemplo añadiendo un 1 con push:

Array ( [0] => 3 [1] => 5 [2] => 1 )

funcion `array_unshift()`

Tercer ejemplo añadiendo un 1 con unshift al principio:

Array ( [0] => 2 [1] => 3 [2] => 5 [3] => 1 )

con `array_push()` en el primer elemento le pasamos el array y en el segundo el elemento que queremos añadir, que por defecto se añade al final del todo si queremos añadirlo al principio, tenemos que usar otra funcion llamada `array_unshift()` En ambos podemos añadir los elementos que queramos siempre que los separemos mediante comas

## 2. Eliminar elementos

funcion `array_pop()`

Primer ejemplo array original:

Array ( [0] => a [1] => b [2] => c [3] => d )

Segundo ejemplo borrando con pop la ultima posicion:

Array ( [0] => a [1] => b [2] => c )

funcion `array_shift()`

Tercer ejemplo eliminando la primera posicion con shift:

Array ( [0] => b [1] => c )

se debe referenciar el array del que queremos eliminar el elemento, con `array_pop()` eliminamos la ultima posicion de la lista, mientras que con `array_shift()` eliminamos la primera posicion

## 3. Encontrar elementos

funcion `array_search()`

array original: Array ( [a] => 1 [b] => 2 [c] => 3 )

el valor 2 equivale a la clave: b

hemos creado un array asociativo con claves y valores, y queremos encontrar a que clave equivale el valor 2, que en este caso nos devuelve la clave b si no fuera un array asociativo, nos devolveria la posicion del array en la que se encuentra el numero indicado

## 4. Ordenar elementos

funcion `sort()`

Array original:

Array ( [0] => 4 [1] => 5 [2] => 2 [3] => 8 [4] => 1 [5] => 6 )

Array ordenado ascendente con sort:

Array ( [4] => 1 [2] => 2 [0] => 4 [1] => 5 [5] => 6 [3] => 8 )

Array ordenado descendente con asort:

Array ( [3] => 8 [5] => 6 [1] => 5 [0] => 4 [2] => 2 [4] => 1 )

utilizamos la función `asort` para ordenar los valores numéricos en orden ascendente en el caso de que sea un array asociativo es mejor usar la función `array_multisort()` para ordenar en orden descendente es `arsort()`

## 5. Contar elementos

función `count()`

Vamos a contar el array `d`: 6

en este caso en el array `d` había 6 elementos

## 6. Máximo y mínimo

funciones `max()` y `min()`

el valor máximo de `d` es: 8

el valor mínimo de `d` es: 1

con la función `max()` podemos sacar cuál es el valor más alto del array indicado, mientras que con `min()` podemos sacar el valor más pequeño