

El origen de la semilla

El valor Unix Epoch

El conocido como *tiempo UNIX* –o también *Unix Epoch*– es un sistema referencia de tiempo cuya unidad son los segundos y que tiene su valor cero a las **0:00:00 horas (GMT)** del día **uno de enero de 1970**.

La función time()

La función `time()` devuelve una **cadena** con el número de segundos transcurridos desde el comienzo de la **Unix Epoch**.

¿Que cuantos son? Pues mira, hasta este mismo instante han transcurrido:

1 121 786 388 segundos

desde el comienzo del *tiempo UNIX*.

La función microtime()

Usando la función `microtime()` se obtiene una **cadena** a la que, además del número de segundos transcurridos desde el comienzo de la **Unix Epoch**, se añade – al comienzo de ella y separado por un espacio– la parte decimal de ese tiempo expresada en microsegundos.

Este es el valor de la cadena devuelta por `microtime()` en este instante:

0.54022100 1121786388

donde, como verás, aparece la fracción decimal del tiempo *Unix Epoch* delante de su valor entero.

(double)microtime()

Dado que con `microtime()` obtenemos una **cadena**, es posible convertirla en número de coma flotante anteponiendo a esa función **(double)**.

Una vez cambiado el tipo de variable el valor devuelto por `microtime` se convierte en:

0.543142

que como puedes observar es un número con *seis decimales* (si las últimas cifras son ceros no se visualizarán).

Si se multiplica el valor anterior por **1.000.000** obtenemos un número de **seis cifras** y valor *entero* que puede cambiar de valor cada **millonésima de segundo**.

543280

¡Fíjate que ha cambiado! La diferencia no es otra cosa que el tiempo transcurrido entre los instantes en que se ejecutaron ambas instrucciones.

Este número entero es utilizado por `srand` y `mt_rand` como *semilla* generadora de números aleatorios.

Números aleatorios

PHP dispone de dos funciones capaces de generar números aleatorios. Se trata de la función `rand()` y de la *función mejorada* `mt_rand()`.

El valor **mínimo** del intervalo que contiene los números aleatorios que pueden ser generados es **CERO** en ambos casos y los valores **máximos** de cada opción pueden determinarse mediante las funciones `getrandmax()`, para el primer generador, y `mt_getrandmax()`, para el segundo.

Veamos cuales son esos valores en cada uno de los casos.

Valores máximos de los generadores de números aleatorios			
Generador <code>rand()</code>		Generador <code>mt_rand()</code>	
Sintaxis	Valor máximo	Sintaxis	Valor máximo
<code>echo getrandmax()</code>	32767	<code>echo mt_getrandmax()</code>	2147483647

Según las librerías que esté usando PHP, puede ocurrir que los valores máximos con ambos generadores sean iguales. No es ese nuestro caso, como puedes comprobar.

La forma más simple

La forma más simple -y más desaconsejable- de generación de un número aleatorio es esta:

Generación de un número aleatorio			
Generador <code>rand()</code>		Generador <code>mt_rand()</code>	
Sintaxis	Nº aleatorio	Sintaxis	Nº aleatorio
<code>echo rand()</code>	16193	<code>echo mt_rand()</code>	2064588981

Una semilla que mejora la aleatoriedad

Si antes de escribir la función `rand()` en un caso, o `mt_rand()` en el otro escribimos:

`srand((double)microtime()*1000000)`, en el primer caso y/o

`mt_srand((double)microtime()*1000000)`, en el segundo

estaremos introduciendo una *semilla* (al margen comentamos la forma en que se genera) que mejora sustancialmente la aleatoriedad de los números obtenidos.

Aquí tenemos un ejemplo:

Generación de un número aleatorio con semilla	
Generador rand()	
Sintaxis	Nº aleatorio
srand((double)microtime()*1000000); echo rand()	9639
Generador mt_rand()	
Sintaxis	Nº aleatorio
mt_srand((double)microtime()*1000000); echo mt_rand()	633782837

Manteniendo la sintaxis anterior -no te olvides de las *semillitas famosas*- se pueden generar números aleatorios comprendidos dentro del intervalo que preestablezcamos.

Bastaría con añadir los valores de los **extremos** de ese **intervalo** como parámetros de la función. La sintaxis sería esta:

`rand(extremo inferior , extremo superior)`

y para la *función mejorada*

`mt_rand(extremo inferior , extremo superior)`

Generación de un número aleatorio delimitando intervalos	
Generador <code>rand()</code>	
Sintaxis	Nº aleatorio
<code>srand((double)microtime()*1000000); echo rand(1,300)</code>	18
Generador <code>mt_rand()</code>	
Sintaxis	Nº aleatorio
<code>mt_srand((double)microtime()*1000000); echo mt_rand(1,300)</code>	262