

Forum:[Hack Forums](#)

Thread: [Fairest way to randomly generate a number?](#)

Kedox - April 19, 2019, 3:07 pm

I've been using mt_rand, is there any more fair ways to generate a number however?

Thanks

Omniscient - April 19, 2019, 4:41 pm

(Apr 19, 2019 03:07 PM)Kedox Wrote: [\[CITE\]](#) I've been using mt_rand, is there any more fair ways to generate a number however?

Thanks

What's wrong with mt_rand? Does it not do something you need?

Kedox - April 19, 2019, 4:45 pm

(Apr 19, 2019 04:41 PM)Omniscient Wrote: [\[CITE\]](#) What's wrong with mt_rand? Does it not do something you need?

Trying to most fairly generate a number between 0 and 100. I was curious if it was the most fair or if there is better options

gianni - April 19, 2019, 4:47 pm

mt_rand is completely fine for what you need.

Edit:

<https://stackoverflow.com/questions/28760650/difference-between-mt-rand-and-rand>

enjoy the quick read on why its good

Kedox - April 19, 2019, 4:49 pm

(Apr 19, 2019 04:47 PM)gianni Wrote: [\[CITE\]](#) mt_rand is completely fine for what you need. What I need it for is dice gambling, so most fair option is important

SuperiorUser - April 19, 2019, 5:18 pm

mt_rand eventually produces predictable data overtime.

<https://www.soliantconsulting.com/blog/secure-randomness-php>

"If you have the mcrypt library installed you could use mcrypt_create_iv() as it's believed to be a secure source of random bytes. If you are on PHP 7 you can call either random_bytes() or random_int() "

also

<https://www.openwall.com/lists/announce/2013/11/04/1>

Disease - April 19, 2019, 11:07 pm

(Apr 19, 2019 05:18 PM)SuperiorUser Wrote: [\[CITE\]](#) ...If you are on PHP 7 you can call either random_bytes() or random_int() "...

If you're not, please update or contact your host immediately and demand they update. The entire PHP 5.x branch is unsupported as of January 1st of this year. In fact, 7.0 turned end-of-life a couple months ago as well. Being on anything less than 7.1, though preferably 7.2, is not good. But yes, random_bytes() or random_int() would be a solid way to go. Both generate cryptographically secure

psuedo-random return. Obviously `random_int()` returns a number, making it more suitable for the OP's need.

Charger - April 20, 2019, 7:14 am

Anything you do outside of a built in function will lead to less randomness and more predictability.

QRCode - April 23, 2019, 6:15 am

Squito - April 24, 2019, 3:54 am

QRCode:

Quote:`str_shuffle`

This function does not generate cryptographically secure values, and should not be used for cryptographic purposes.

<https://www.php.net/manual/en/function.str-shuffle.php>

Please stop giving people bad advices/solutions.