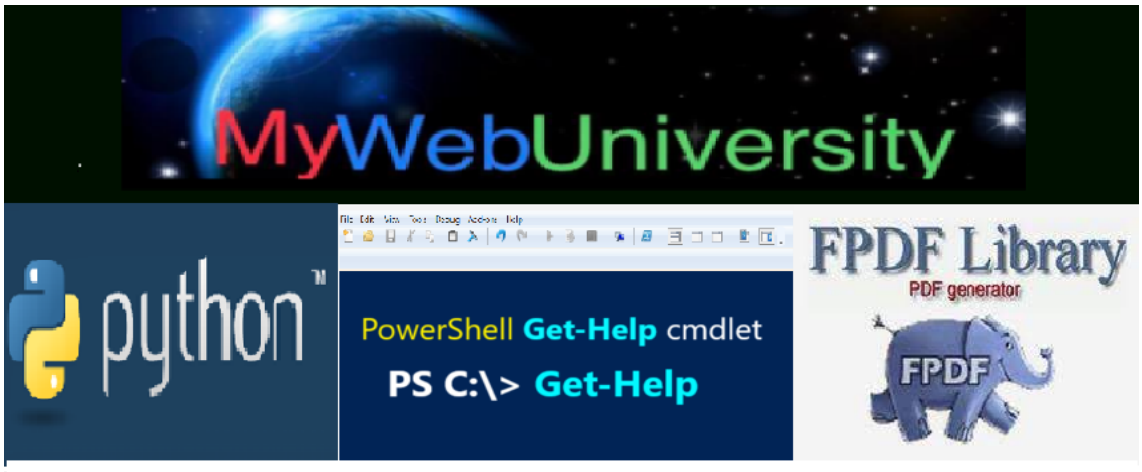

Full credit is given to all the above companies including the Operating System that this PDF file was generated!

Windows PowerShell Get-Help on Cmdlet 'Get-Random'

PS:\>Get-HELP Get-Random -Full

NAME

Get-Random

SYNOPSIS

Gets a random number, or selects objects randomly from a collection.

SYNTAX

Get-Random [-InputObject] <System.Object[]> [-Count <System.Int32>] [-SetSeed <System.Nullable`1[System.Int32]>] [<CommonParameters>]

Get-Random [[-Maximum] <System.Object>] [-Minimum <System.Object>] [-SetSeed <System.Nullable`1[System.Int32]>] [<CommonParameters>]

DESCRIPTION

The `Get-Random` cmdlet gets a randomly selected number. If you submit a collection of objects to `Get-Random`, it gets one or more randomly selected objects from the

collection.

Without parameters or input, a `Get-Random` command returns a randomly selected 32-bit unsigned integer between 0 (zero) and `[int32]::MaxValue`.

You can use the parameters of `Get-Random` to specify the minimum and maximum values, the number of objects returned from a collection, or a seed number.

> [!CAUTION] > `Get-Random` doesn't ensure cryptographically secure randomness. The seed value is used for the > current command and for all subsequent `Get-Random`

commands in the current session until you use > SetSeed again or close the session. You can't reset the seed to its default value. > > Deliberately setting the seed

results in non-random, repeatable behavior. It should only be used > when trying to reproduce behavior, such as when debugging or analyzing a script that includes >

`Get-Random` commands. Be aware that the seed value could be set by other code in the same > session, such as an imported module. > > PowerShell 7.4 includes

`Get-SecureRandom`, which ensures cryptographically secure randomness.

PARAMETERS

-Count <System.Int32>

Specifies the number of random objects to return. The default is 1.

When used with `InputObject` containing a collection:

- Each randomly selected item is returned only once.

- If the value of Count exceeds the number of objects in the collection, all objects in the collection are returned in random order.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-InputObject <System.Object[]>

Specifies a collection of objects. `Get-Random` gets randomly selected objects in random order from the collection up to the number specified by Count . Enter the

objects, a variable that contains the objects, or a command or expression that gets the objects. You can also pipe a collection of objects to `Get-Random`.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-Maximum <System.Object>

Specifies a maximum value for the random number. `Get-Random` returns a value that's less than the maximum (not equal). Enter an integer, a double-precision

floating-point number, or an object that can be converted to an integer or double, such as a numeric string ("100").

The value of Maximum must be greater than (not equal to) the value of Minimum . If the value of Maximum or Minimum is a floating-point number, `Get-Random`

returns a randomly selected floating-point number.

On a 64-bit computer, if the value of Minimum is a 32-bit integer, the default value of Maximum is Int32.MaxValue .

If the value of Minimum is a double (a floating-point number), the default value of Maximum is Double.MaxValue . Otherwise, the default value is Int32.MaxValue .

Required? false

Position? 0

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Minimum <System.Object>

Specifies a minimum value for the random number. Enter an integer, a double-precision floating-point number, or an object that can be converted to an integer or

double, such as a numeric string ("100"). The default value is 0 (zero).

The value of Minimum must be less than (not equal to) the value of Maximum . If the value of Maximum or Minimum is a floating-point number, `Get-Random` returns a

randomly selected floating-point number.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-SetSeed <System.Nullable`1[System.Int32]>

Specifies a seed value for the random number generator. When you use SetSeed , the cmdlet generates pseudorandom numbers, which isn't cryptographically secure.

> [!CAUTION] > Setting the seed results in non-random behavior. It should only be used when trying to reproduce > behavior, such as when debugging or analyzing a

script that includes `Get-Random` commands. > > This seed value is used for the current command and for all subsequent `Get-Random` commands in > the current

session until you use SetSeed again or close the session. You can't reset the seed > to its default value.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug,

ErrorAction, ErrorVariable, WarningAction, WarningVariable,

OutBuffer, PipelineVariable, and OutVariable. For more information, see

about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

INPUTS

System.Object

You can pipe any object to this cmdlet. It selects values randomly from the piped objects.

OUTPUTS

System.Int32

System.Int64

System.Double

System.Management.Automation.PSObject

This cmdlet returns an integer or floating-point number, or an object selected randomly from a submitted collection.

NOTES

`Get-Random` doesn't always return the same data type as the input value. The following table shows the output type for each of the numeric input types.

| Input Type | Output Type | | :--------: | :---------: | | SByte | Double | | Byte | Double | | Int16 | Double | | UInt16 | Double | | Int32 | Int32 | | UInt32 | Double | | Int64 | Int64 | | UInt64 | Double | | Double | Double | |

Single | Double |

Beginning in Windows PowerShell 3.0, `Get-Random` supports 64-bit integers.

---------- Example 1: Get a random integer ----------

```
Get-Random
```

```
3951433
```

------- Example 2: Get a random integer between 0 and 99 -------

```
Get-Random -Maximum 100
```

```
47
```

----- Example 3: Get a random integer between -100 and 99 -----

```
Get-Random -Minimum -100 -Maximum 100
```

```
56
```

-------- Example 4: Get a random floating-point number --------

```
Get-Random -Minimum 10.7 -Maximum 20.93
```

```
18.08467273887
```

-------- Example 5: Get a random integer from an array --------

```
1, 2, 3, 5, 8, 13 | Get-Random
```

```
8
```

```
----- Example 6: Get several random integers from an array -----
```

```
1, 2, 3, 5, 8, 13 | Get-Random -Count 3
```

```
3
1
13
```

```
---------- Example 7: Randomize an entire collection ----------
```

```
1, 2, 3, 5, 8, 13 | Get-Random -Count ([int]::MaxValue)
```

```
2
3
5
1
8
13
```

```
---------- Example 8: Get a random non-numeric value ----------
```

```
"red", "yellow", "blue" | Get-Random
```

```
yellow
```

---------- Example 9: Use the SetSeed parameter ----------

```
# Commands with the default seed are pseudorandom
Get-Random -Maximum 100 -SetSeed 23
Get-Random -Maximum 100
Get-Random -Maximum 100
Get-Random -Maximum 100
```

```
32
25
93
95
```

```
# Commands with the same seed aren't random
Get-Random -Maximum 100 -SetSeed 23
Get-Random -Maximum 100 -SetSeed 23
Get-Random -Maximum 100 -SetSeed 23
```

```
32
32
32
```

```
# SetSeed results in a repeatable series
Get-Random -Maximum 100 -SetSeed 23
Get-Random -Maximum 100
Get-Random -Maximum 100
Get-Random -Maximum 100
```

```
32
25
93
95
```

---------- Example 10: Get random files ----------

```
$Files = Get-ChildItem -Path C:\* -Recurse
$Sample = $Files | Get-Random -Count 50
```

---------- Example 11: Roll fair dice ----------

```
1..1200 | ForEach-Object {
    1..6 | Get-Random
} | Group-Object | Select-Object Name,Count
```

```
Name Count
---- -----
1    206
2    199
3    196
4    226
5    185
6    188
```

RELATED LINKS

Online Version:

https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/get-random?view=powershell-5.1&WT.mc_id=ps-gethelp