

Get-Random

Der Befehl `Get-Random` erzeugt einen zufälligen Wert beziehungsweise wählt zufällig ein Element aus einer angegebenen Menge aus.

Mit `Get-Random` lassen sich unter anderem:

- zufällige Ganzzahlen erzeugen
- Zufallswerte innerhalb eines bestimmten Bereichs erzeugen
- zufällig ein Element aus einer Liste auswählen
- mehrere zufällige Elemente auswählen
- die Reihenfolge einer Sammlung zufällig verändern
- reproduzierbare Zufallsfolgen mit einem festen Seed erzeugen

Der Befehl gehört zu den Standard-Cmdlets von PowerShell.

Grundlagen

Zufällige Zahl erzeugen

Wird `Get-Random` ohne weitere Parameter aufgerufen, wird eine zufällige Ganzzahl erzeugt.

```
Get-Random
```

Beispielausgabe:

```
137582946
```

Bei jedem Aufruf kann ein anderer Wert zurückgegeben werden.

Zufallszahl innerhalb eines Bereichs

Mit `-Minimum` und `-Maximum` lässt sich der Wertebereich festlegen.

```
Get-Random -Minimum 1 -Maximum 100
```

Mögliche Ausgabe:

73

“ Hinweis

Der Wert von `-Maximum` ist **exklusiv**. Bei `-Maximum 100` können Werte von `1` bis `99` erzeugt werden.

Soll beispielsweise eine Zahl zwischen **1 und 100 einschließlich** erzeugt werden, muss `-Maximum` auf `101` gesetzt werden.

```
Get-Random -Minimum 1 -Maximum 101
```

Syntax

Zufällige Zahl

```
Get-Random
```

Zufällige Zahl mit Bereich

```
Get-Random -Minimum <Int32> -Maximum <Int32>
```

Zufälliges Element auswählen

```
Get-Random -InputObject <Object[]>
```

Mehrere zufällige Elemente auswählen

```
Get-Random -InputObject <Object[]> -Count <Int32>
```

Sammlung zufällig mischen

```
Get-Random -InputObject <Object[]> -Shuffle
```

Zufallsfolge reproduzierbar machen

```
Get-Random -SetSeed <Int32>
```

Parameter

Parameter	Typ	Beschreibung
-InputObject	Object[]	Gibt die Objekte an, aus denen zufällig ausgewählt wird.
-Minimum	Int32	Legt die untere Grenze für eine Zufallszahl fest.
-Maximum	Int32	Legt die obere Grenze für eine Zufallszahl fest.
-Count	Int32	Gibt an, wie viele zufällige Elemente ausgewählt werden sollen.
-Shuffle	SwitchParameter	Gibt die angegebenen Elemente in zufälliger Reihenfolge zurück.
-SetSeed	Int32	Legt den Startwert des Zufallszahlengenerators fest.

-InputObject

Beschreibung

Der Parameter `-InputObject` gibt eine Sammlung von Objekten an, aus der `Get-Random` zufällig ein Element auswählen kann.

```
Get-Random -InputObject <Objekte>
```

Beispiel:

```
$Farben = "Rot", "Grün", "Blau", "Gelb"
```

```
Get-Random -InputObject $Farben
```

Mögliche Ausgabe:

```
Blau
```

Bei jedem Aufruf kann eine andere Farbe ausgewählt werden.

Beispiel mit Zahlen

```
$Zahlen = 10, 20, 30, 40, 50
```

```
Get-Random -InputObject $Zahlen
```

Mögliche Ausgabe:

```
30
```

Beispiel mit Strings

```
$Namen = "Anna", "Ben", "Clara", "David"
```

```
Get-Random -InputObject $Namen
```

Mögliche Ausgabe:

```
Clara
```

“ Hinweis

`Get-Random` kann grundsätzlich mit beliebigen PowerShell-Objekten arbeiten, nicht nur mit Zahlen und Zeichenketten.

-Minimum

Beschreibung

`-Minimum` legt den kleinsten möglichen Wert einer Zufallszahl fest.

```
Get-Random -Minimum 10 -Maximum 20
```

Mögliche Ausgabe:

```
17
```

Der Minimalwert ist **inklusiv**.

Das bedeutet:

```
Get-Random -Minimum 10 -Maximum 20
```

kann unter anderem `10` zurückgeben, aber niemals `20`.

-Maximum

Beschreibung

`-Maximum` legt die obere Grenze des Zufallsbereichs fest.

Der angegebene Wert ist **exklusiv**.

```
Get-Random -Minimum 1 -Maximum 10
```

Mögliche Werte:

```
1  
2  
3  
4  
5  
6  
7  
8  
9
```

Der Wert `10` wird nicht erzeugt.

Zufällige Zahl zwischen 0 und einem Maximum

```
Get-Random -Maximum 50
```

Mögliche werte:

```
0
1
2
...
49
```

-Count

Beschreibung

`-Count` legt fest, wie viele zufällige Elemente zurückgegeben werden sollen.

```
$Farben = "Rot", "Grün", "Blau", "Gelb", "Orange"
```

```
Get-Random -InputObject $Farben -Count 3
```

Mögliche Ausgabe:

```
Gelb
Rot
Blau
```

Die Elemente werden dabei standardmäßig ohne Wiederholung aus der Sammlung ausgewählt.

Beispiel

```
$Zahlen = 1..10
```

```
Get-Random -InputObject $Zahlen -Count 3
```

Mögliche Ausgabe:

7
2
9

“ Hinweis

`-Count` bezieht sich auf die Anzahl der zurückgegebenen Elemente, nicht auf die Anzahl der durchzuführenden Zufallsoperationen.

-Shuffle

Beschreibung

`-Shuffle` gibt alle über `-InputObject` übergebenen Elemente in einer zufälligen Reihenfolge zurück.

```
$Zahlen = 1..10
```

```
Get-Random -InputObject $Zahlen -Shuffle
```

Mögliche Ausgabe:

```
4  
9  
1  
7  
3  
10  
2  
8  
5  
6
```

Alle Elemente bleiben erhalten, lediglich ihre Reihenfolge wird verändert.

Beispiel mit Namen

```
$Namen = "Anna", "Ben", "Clara", "David"
```

```
Get-Random -InputObject $Namen -Shuffle
```

Mögliche Ausgabe:

```
Clara  
Anna  
David  
Ben
```

Das ist praktisch, wenn beispielsweise eine Liste zufällig sortiert werden soll.

-SetSeed

Beschreibung

Mit `-SetSeed` kann der Startwert des Zufallszahlengenerators festgelegt werden.

```
Get-Random -SetSeed 12345
```

Der Seed beeinflusst die erzeugte Zufallsfolge.

Ein identischer Seed kann dazu verwendet werden, eine reproduzierbare Zufallsfolge zu erzeugen.

```
Get-Random -SetSeed 12345
```

Anschließend erzeugte Zufallswerte folgen dann derselben deterministischen Sequenz.

“ Hinweis

Ein fester Seed bedeutet nicht, dass die Werte tatsächlich zufällig und unvorhersehbar sind. Für kryptografische Zwecke ist `Get-Random` daher nicht geeignet. Zufall ist schließlich nicht automatisch Sicherheit, auch wenn Menschen das erstaunlich gerne verwechseln.

Beispiele

Zufällige Zahlen

Zahl zwischen 0 und einem Maximum

```
Get-Random -Maximum 50
```

Mögliche Werte:

```
0  
1  
2  
...  
49
```

Zahl zwischen zwei Grenzen

```
Get-Random -Minimum 50 -Maximum 100
```

Mögliche Werte:

```
50  
51  
52  
...  
99
```

Zufällige Würfelzahl

Ein Würfel besitzt die Werte `1` bis `6`.

```
Get-Random -Minimum 1 -Maximum 7
```

Mögliche Ausgabe:

4

Mehrfach würfeln

```
1..10 | ForEach-Object {  
    Get-Random -Minimum 1 -Maximum 7  
}
```

Mögliche Ausgabe:

```
3  
6  
1  
4  
2  
6  
5  
3  
1  
4
```

Zufällige Elemente auswählen

Zufälliges Element aus einem Array

```
$Tiere = "Hund", "Katze", "Maus", "Pferd"
```

```
Get-Random -InputObject $Tiere
```

Mögliche Ausgabe:

Katze

Zufälliges Element aus einer Pipeline

`Get-Random` kann auch Objekte aus der Pipeline entgegennehmen.

```
"Rot", "Grün", "Blau", "Gelb" | Get-Random
```

Mögliche Ausgabe:

```
Grün
```

Das funktioniert beispielsweise auch mit Dateien:

```
Get-ChildItem | Get-Random
```

Hier wird eine zufällige Datei beziehungsweise ein zufälliges Verzeichnis aus dem aktuellen Verzeichnis ausgewählt.

Zufällige Datei auswählen

```
Get-ChildItem "C:\Temp" -File | Get-Random
```

Damit wird ein zufälliges Dateiobjekt aus `C:\Temp` ausgewählt.

Zufälligen Namen auswählen

```
$Namen = @(
    "Anna"
    "Ben"
    "Clara"
    "David"
    "Eva"
)

$Name = $Namen | Get-Random

$Name
```

Mögliche Ausgabe:

```
David
```

Mehrere Elemente auswählen

Mit `-Count` können mehrere Elemente aus einer Sammlung ausgewählt werden.

```
$Namen = "Anna", "Ben", "Clara", "David", "Eva"
```

```
Get-Random -InputObject $Namen -Count 2
```

Mögliche Ausgabe:

```
David
```

```
Anna
```

Drei zufällige Dateien auswählen

```
Get-ChildItem "C:\Temp" -File |
```

```
Get-Random -Count 3
```

Damit werden drei zufällige Dateien ausgewählt.

Listen zufällig mischen

Eine Sammlung kann mit `-Shuffle` vollständig durchmischt werden.

```
$Karten = 1..10
```

```
$Karten | Get-Random -Shuffle
```

Mögliche Ausgabe:

```
7
```

```
2
```

```
10
```

```
4
```

```
1
```

```
9
```

```
6
```

3
8
5

Alternative mit `-Count`

Wenn eine bestimmte Anzahl zufälliger Elemente benötigt wird:

```
$Karten | Get-Random -Count 3
```

Mögliche Ausgabe:

8
2
5

Zufälligen Benutzer auswählen

```
$Benutzer = @(
    "Anna"
    "Ben"
    "Clara"
    "David"
)

$Zufall = $Benutzer | Get-Random

Write-Host "Ausgewählt: $Zufall"
```

Zufällige Farbe auswählen

```
$Farben = @(
    "Red"
    "Green"
    "Blue"
    "Yellow"
)
```

```
$Farbe = $Farben | Get-Random
```

```
Write-Host "Farbe: $Farbe"
```

Zufällige Zahl für einen Bereich

```
$Zahl = Get-Random -Minimum 1 -Maximum 101
```

```
Write-Host "Zufallszahl: $Zahl"
```

Zufällige Datei auswählen

```
$Datei = Get-ChildItem "C:\Temp" -File | Get-Random
```

```
Write-Host "Ausgewählte Datei: $($Datei.Name)"
```

Zufällige Reihenfolge erzeugen

```
$Teilnehmer = @(
    "Anna"
    "Ben"
    "Clara"
    "David"
    "Eva"
)
```

```
$Teilnehmer | Get-Random -Shuffle
```

Mögliche Ausgabe:

```
Clara
Eva
Anna
David
Ben
```

Häufige Einsatzgebiete

`Get-Random` wird häufig verwendet für:

- Zufallszahlen
- Würfelsimulationen
- zufällige Auswahl aus Listen
- zufällige Dateien oder Verzeichnisse
- Testdaten
- Simulationen
- Spiele
- zufällige Reihenfolgen
- Auswahl von Testfällen
- einfache Zufallsentscheidungen

Rückgabewerte

Der Rückgabewert hängt davon ab, welche Eingabe verwendet wird.

Verwendung	Rückgabewert
Zufallszahl	<code>System.Int32</code> bzw. numerischer Datentyp
<code>-InputObject</code>	Typ des ausgewählten Objekts
<code>-Count</code>	Mehrere Objekte
<code>-Shuffle</code>	Sammlung der ursprünglichen Objekte in zufälliger Reihenfolge

Beispiel:

```
$Wert = Get-Random -Minimum 1 -Maximum 100
```

```
$Wert.GetType().FullName
```

Mögliche Ausgabe:

```
System.Int32
```

Bei Objekten bleibt deren ursprünglicher Typ erhalten.

Hinweise

`-Maximum` ist exklusiv

Bei:

```
Get-Random -Minimum 1 -Maximum 10
```

liegt der Wertebereich zwischen `1` und `9`.

Für `1` bis `10` muss daher verwendet werden:

```
Get-Random -Minimum 1 -Maximum 11
```

`Get-Random` ist nicht für Kryptografie geeignet

`Get-Random` sollte nicht verwendet werden, wenn kryptografisch sichere Zufallswerte benötigt werden.

Für sicherheitsrelevante Anwendungen müssen dafür geeignete kryptografische Zufallszahlengeneratoren verwendet werden.

Zufälligkeit ist nicht Gleichverteilung in kleinen Stichproben

Auch wenn eine Zufallsfunktion verwendet wird, bedeutet das nicht, dass in einer kleinen Anzahl von Durchläufen jedes Ergebnis gleich häufig erscheint.

Beispielsweise kann:

```
1..10 | ForEach-Object {  
    Get-Random -Minimum 1 -Maximum 7  
}
```

```
}
```

zufällig fünfmal die `3` liefern.

Das ist kein Fehler. Zufall ist lediglich bemerkenswert schlecht darin, menschliche Erwartungen an "ausgewogen" zu erfüllen.

Kurzreferenz

Aufgabe	Befehl
Zufällige Zahl erzeugen	<code>Get-Random</code>
Zahl bis 99 erzeugen	<code>Get-Random -Maximum 100</code>
Zahl von 1 bis 100	<code>Get-Random -Minimum 1 -Maximum 101</code>
Zufälliges Element auswählen	<code>\$Liste Get-Random</code>
Mehrere Elemente auswählen	<code>\$Liste Get-Random -Count 3</code>
Liste mischen	<code>\$Liste Get-Random -Shuffle</code>
Seed setzen	<code>Get-Random -SetSeed 123</code>

Siehe auch

- `Get-Random`
- `System.Random`
- `System.Security.Cryptography.RandomNumberGenerator`
- `ForEach-Object`
- `Select-Object`

Revision #1

Created 2026-09-20 11:12:13 UTC by John-Andreas Borinas

Updated 2026-09-20 12:26:08 UTC by John-Andreas Borinas