

---

# **Generalized Halton Module**

*Release 0.6.1*

**Aug 24, 2017**



---

## Contents

---

**Bibliography**

**3**



**class** `ghalton.GeneralizedHalton` (*dim*, [*seed*] | *config*)

Creates a generalized Halton generator. It takes at least one argument, either the dimensionality, or a configuration. When the dimensionality is given, an optional argument can be used to seed for the random permutations created. The configuration is a series of permutations each of  $n_i$  numbers, where  $n_i$  is the  $n_i$ 'th prime number. In that last case, the dimensionality is inferred from the number of sublists given.

#### Parameters

- **dim** (*integer*) – Dimensionality of the points to create
- **seed** (*integer*) – Seed to create the permutations
- **config** (*list of lists of integers*) – List of permutations to scramble the halton digits

**get** (*n*)

Retrieve the  $n$  next points from this sequence. Each point is a `list` containing each value for each coordinates and the points are returned in a `list` of  $n$  elements even if  $n$  is 1.

**reset** ()

Reset the generator to its initial state, i.e. before it generated the first point.

**seed** ([*config*])

Seed the generator with a new seed or configuration. Seeding a generator automatically call `reset()`.

**Parameters config** (None, integer, or list of lists of integers) – The config to seed the sequencer

**class** `ghalton.Halton` (*dim*)

Creates a Halton generator of dimensionality *dim*. This is similar to creating a `GeneralizedHalton` sequence with the identity permutations.

**get** (*n*)

Retrieve the  $n$  next points from this sequence. Each point is a `list` containing each value for each coordinates and the points are returned in a `list` of  $n$  elements even if  $n$  is 1.

**reset** ()

Reset the generator to its initial state, i.e. before it generated the first point.

`ghalton.EA_PERMS`

Permutations described in [DeRainville2012] for the 100 first dimensions of the Generalized Halton sequence.

`ghalton.PRIMES`

Prime numbers lower than 10000.



---

## Bibliography

---

[DeRainville2012] F.-M. De Rainville, C. Gagné, O. Teytaud, D. Laurendeau. *Evolutionary optimization of low-discrepancy sequences*. *ACM Trans. Model. Comput. Simul.*, 22(2):1-25, 2012.



## G

get() (ghalton.GeneralizedHalton method), 1  
get() (ghalton.Halton method), 1  
ghalton.EA\_PERMS (built-in variable), 1  
ghalton.GeneralizedHalton (built-in class), 1  
ghalton.Halton (built-in class), 1  
ghalton.PRIMES (built-in variable), 1

## R

reset() (ghalton.GeneralizedHalton method), 1  
reset() (ghalton.Halton method), 1

## S

seed() (ghalton.GeneralizedHalton method), 1