

# Package ‘iteratoR’

October 13, 2022

**Type** Package

**Title** Print Loop Iterations at Exponentially Disparate Intervals

**Version** 0.1.1

**Maintainer** Steve Condylis <steve.condylis@gmail.com>

**BugReports** <https://github.com/stevecondylis/iteratoR/issues>

**License** MIT + file LICENSE

**URL** <https://github.com/stevecondylis/iteratoR>

**Description** Know which loop iteration the code execution is up to by including a single, convenient function call inside the loop.

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**NeedsCompilation** no

**Author** Steve Condylis [aut, cre, cph]  
(<<https://orcid.org/0000-0003-0599-844X>>),  
Zane Dax [ctb]

**Repository** CRAN

**Date/Publication** 2022-08-30 15:20:02 UTC

## R topics documented:

iteration . . . . .	2
<b>Index</b>	<b>4</b>

---

iteration

*Conveniently print loop iterations to console*

---

### Description

Place inside a loop to automatically and conveniently print the current loop iteration at exponentially disparate (or custom) intervals.

### Usage

```
iteration(iterator_name, iteration_values)
```

### Arguments

`iterator_name` The name of the loop iterator (e.g. "i")

`iteration_values`

An integer vector specifying loop iterations (defaults to the sequence 1, 2, 5, 10, 20, 50, 100, 200, 500 ....)

### Value

`iteration()` is a non-value-returning function. As such, it will not return anything, and instead print to console the value representing the current loop iteration.

### Examples

```
# For a loop that would otherwise give no feedback as to where it is up to,  
# simply include iteration() anywhere inside the loop to show progress
```

```
for(i in 1:10000) {  
  2 * 2  
  iteration()  
}  
# 10  
# 20  
# 50  
# 100  
# 200  
# 500  
# 1,000  
# 2,000  
# 5,000  
# 10,000  
# 20,000  
# 50,000
```

```
# To use an iterator other than 'i' (example: 'page')
for(page in 1:10000) {
  2 * 2
  iteration("page")
}
# 10
# 20
# 50
# 100
# 200
# 500
# 1,000
# 2,000
# 5,000
# 10,000

# Use custom iteration intervals
for(i in 1:10000) {
  2 * 2
  iteration(iteration_values = seq(0, 1e4, 1e3))
}

# 1,000
# 2,000
# 3,000
# 4,000
# 5,000
# 6,000
# 7,000
# 8,000
# 9,000
# 10,000
```

# Index

iteration, [2](#)