

PYTHON'S BUILT-IN ITERABLES AND ITERATORS

Python provides many functions that return iterables or iterators

Additionally, the iterators perform **lazy** evaluation

You should always be aware of whether you are dealing with an iterable or an iterator

why?

if an object is an **iterable** (but **not** an **iterator**) you can iterate over it many times

if an object is an **iterator** you can iterate over it only once

`range(10)` → iterable

`zip(l1, l2)` → iterator

`enumerate(l1)` → iterator

`open('cars.csv')` → iterator

`dictionary.keys()` → iterable

`dictionary.values()` → iterable

`dictionary.items()` → iterable

and many more...

Code Exercises

© 2018 Matplotlib Academy