Iterators

Advanced Python Programming

—

Valerio Velardo - The Sound of AI

What’s an Iterator?

An object that can be iterated upon -> it’s possible to traverse all of its values

Defining an iterator

Defining an iterator

- Must have __iter__(): initialization + return the iterator

Defining an iterator

- Must have __iter__(): initialization + return the iterator
- Must have __next__(): return next item in the sequence

Defining an iterator

- Must have __iter__(): initialization + return the iterator
- Must have __next__(): return next item in the sequence
- It should raise *StopIteration* error when the we're over the last element in the sequence

Iterables vs iterators

- List, tuples, dictionaries, strings are *iterables*
- You can get an iterator out of them using *iter()*

Why should you define a custom iterator?

Why should you define a custom iterator?

Save memory!