# Part 6

You are required to implement a custom iterator in Python that extends the functionality of a standard iterator by allowing you to peek at the next item without advancing the iterator.

Create a class PeekableIterator that wraps around any iterable. This class should support the following methods:

__init__(self, iterable): Initialize the iterator with an iterable.

__iter__(self): Returns the iterator object itself.

__next__(self): Returns the next item from the iterator.

peek(self): Returns the next item without advancing the iterator.

has_next(self): Returns True if there are more items to iterate, otherwise False.