REVERSED ITERATION

Iterating a sequence in reverse order

If we have a sequence type, then iterating over the sequence in reverse order is quite simple:

```
for item in seq[::-1]:
    print(item)
```

This works, but is wasteful because it makes a copy of the sequence

```
for i in range(len(seq)):
    print(seq[len(seq) – i – 1])
```

```
for i in range(len(seq)-1, -1, -1):
    print(seq[i])
```

This is more efficient, but the syntax is messy

```
for item in reversed(seq)
    print(item)
```

This is cleaner and just as efficient, because it creates an iterator that will iterate backwards over the sequence – it does not copy the data like the first example

Both `__getitem__` and `__len__` must be implemented

We can override how reversed works by implementing the `__reversed__` special method

Iterating an iterable in reverse

Unfortunately, `reversed()` will not work with custom iterables without a little bit of extra work

When we call `reversed()` on a custom iterable, Python will look for and call the `__reversed__` function

That function should return an iterator that will be used to perform the reversed iteration

So basically we have to implement a reverse iterator ourselves

Just like the `iter()` method, when we call `reversed()` on an object:

- looks for and calls `__reversed__` method
- if it's not there, uses `__getitem__` and `__len__` to create an iterator for us
- exception otherwise

Card Deck Example

In the code exercises I am going to build an iterable containing a deck of 52 sorted cards

2 Spades … Ace Spades, 2 Hearts … Ace Hearts, 2 Diamonds … Ace Diamonds, 2 Clubs … Ace Clubs

But I don't want to create a list containing all the pre-created cards → Lazy evaluation

So I want my iterator to figure out the suit and card name for a given index in the sorted deck

```
SUITS = ['Spades', 'Hearts', 'Diamonds', 'Clubs']
RANKS = [2, 3, …, 10, 'J', 'Q', 'K', 'A']
```

We assume the deck is sorted as follows:

iterate over `SUITS`

for each suit iterate over `RANKS`

card = combination of suit and rank

Card Deck Example

```
SUITS = ['Spades', 'Hearts', 'Diamonds', 'Clubs']

RANKS = [2, 3, …, 10, 'J', 'Q', 'K', 'A']
```

2S … AS 2H … AH 2D … AD 2C … AC

There are `len(SUITS)` suits 4

There are `len(RANKS)` ranks 13

The deck has a length of: `len(SUITS) * len(RANKS)` 52

Each card in this deck has a positional index: a number from `0` to `len(deck) - 1` 0 - 51

To find the suit index of a card at index `i`:

```
i // len(RANKS)
```

Examples

5th card (6S) → index 4

→ `4 // 13 → 0`

16th card (4H) → index 15

→ `15 // 13 → 1`

To find the rank index of a card at index `i`:

```
i % len(RANKS)
```

Examples

5th card (6S) → index 4

→ `4 % 13 → 4`

16th card (4H) → index 15

→ `15 % 13 → 2`

Code Exercises