

Beyond the Textbook (Zelle 3e - Chapter 8)

Loop Structures and Booleans

For–In is the preferred Python looping structure.

- It is supported by all Python **iterables**, including:
 - `str`
 - `list`
 - `tuple`
 - `range`
 - `file`
- `For–In` handles any number of items in the underlying data structure.
- It behaves appropriately even when the data structure is **empty**.

`while` is a better choice for some looping use cases

- Interacting flexibly with the user at the console
- Searching a file for a matching value
- Processing records from a data store that does not support `For-In` (not a Python iterable)
- Simulating game play
- Controlling devices

Interacting flexibly with the user at the console

- The preferred design pattern for flexible interaction using the console is the **sentinel loop**.
- See:
 - *_05_flexible_user_interaction_using_while.py*
 - *_08_recovering_from_bad_user_input_using_while.py*

Searching a file for a matching value

- When you are searching for just 1 match in a file, finding the match is finding a **sentinel value**.
- So, a sentinel loop is a good solution here, too.
- See:
 - *_10_searching_file_records_using_while.py*
 - *_12_searching_file_records_using_forin_and_break.py*

Processing records from a data store that does not support **For-In**

- In our course, the only data store that we cover is the Python `file`.
- `file` is a Python iterable. It supports `For-In`.
- So, we must demonstrate this design pattern using a method of the Python `file` class that does not make uses of the `iterable` features of `file`:
 - `readline()`
- See:
 - *`_15_processing_data_records_using_while.py`*
 - *`_16_processing_data_records_using_while_with_break.py`*
 - *`_17_processing_data_records_using_while_with_break_and_continue.py`*

Simulating game play

- Designing and building game simulation programs is good practice for designing and building complex programs.
- We will be doing this when we get to Zelle 3e Chapter 9.
- Games continue until one player has won.
- It is a natural use case for `while`.
- See:
 - *_20_playing_a_game_using_while.py*

Controlling Devices

- Controlling electronic devices using a computer requires constant communication between the program and the device.
- Because devices are being controlled for an indefinite period, the `while` is the natural construct to use.
- See:
 - *`_30_controlling_devices_using_while.py`*

Last Revised 2022-09-25