

Book Store Project Brief

This brief will guide you through building a **book store app**. Then, in the upcoming course lectures, we'll show you how I'd build it.

After completing the project, we'll extend it to store data in text files first, and then in a database. Each lecture builds upon earlier lectures, and everything assumes you've built this project yourself first.

By tackling this on your own, our explanations will make much more sense!

Your task

Create a console-based book store system that allows users to:

- Enter and retrieve book details.
- Mark books as read (meaning they've finished reading them).
- Delete existing books.

Like the previous project, we'll use an in-memory database (i.e. a Python list, which we're calling a database because it stores data).

What are books?

Just like movies in the last project, books will be **dictionaries**. You can define their structure to be anything you like, but here is my proposed structure:

```
{  
    'name': BOOK_NAME,  
    'author': BOOK_AUTHOR,  
    'read': False # or True  
}
```



Marking books as read

The property `read` of each book should be a boolean.

When a user wants to mark a book as read, we first should ask them for the name of the book they've read.

Then, loop through all the books and set the `read` property to `True` if the book name matches what the user typed.

Deleting books

Deleting books is something that might look complicated, but it can be really straightforward if we think of the Python constructs we've already seen. You can use a list comprehension to re-create the books list without the book that the user typed.

For example, let's say you have a list of three books, and you want to delete "Peter Pan".

You could have a list comprehension that adds each book to a new list if the name is not equal to "Peter Pan", like this:

```
def delete_book(name):  
    global books  
    books = [book for book in books if book['name'] !=  
name]  
# somewhere else in your application...  
delete_book('Peter Pan')
```

Bonus points

You'll get bonus points for saving your books to a file and loading them from a file too!

You can do this in two ways:

- Have an option in the menu to save the current list of books, and another option to load the list of books that you've previously saved.
- Another option: every time a book is added, read, changed, or deleted, change the file contents so that it matches what the application is showing.

I'm sure you can do it! If you're unsure, tackle it like the last project, and then try to add file storage afterwards.

By writing code yourself and working through these exercises, you're going to grasp everything in programming really quickly.

Happy coding!

Jose and the Teclado team