*An object-oriented book-list!*

* Create a class BookList
* Create another class called Book
* BookLists should have the following properties:
  + Number of books marked as read
  + Number of books marked not read yet
  + A reference to the next book to read (book object)
  + A reference to the current book being read (book object)
  + A reference to the last book read (book object)
  + An array of all the Books
* Each Book should have several properties:
  + Title
  + Genre
  + Author
  + Read (true or false)
  + Read date, can be blank, otherwise needs to be a [JS Date() object](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date)
* Every Booklist should have a few methods:
  + .add(book)
    - should add a book to the books list.
  + .finishCurrentBook()
    - should mark the book that is currently being read as "read"
    - Give it a read date of new Date(Date.now())
    - Change the last book read to be the book that just got finished
    - Change the current book to be the next book to be read
    - Change the next book to be read property to be the first unread book you find in the list of books
* Booklists and Books might need more methods than that. Try to think of more that might be useful.

// This isn't the whole solution - just a really, really big hint.

var Book = function(title, genre, author, read, readdate) {

this.title = title || "No Title";

this.genre = genre "Fiction";

this.author = author || "No Author";

this.read = read || False; //short circuit evaluation makes this work as a default

this.readDate = new Date(readdate);

}

var BookList = function(books) {

//books should be an array.

if (!typeof books == "array") {

books = false;

}

this.bookShelf = books || [];

//check to see if we have books to process

if (this.bookShelf.length > 0) {

//Loop through all of the books in the "books" argument

//Use the .add function to handle adding books, so we keep counts updated.

}

this.add = function(book){

this.bookShelf.push(book);

//If a book has been read:

//See if we have a .lastBook property - set if we don't.

//Update the count of read books

//If a book hasn't been read:

//See if we have a .currentBook property. Set if we don't.

//See if we have a .nextBook property as well - if we didn't set the .currentBook property, set the .nextBook property.

//it should also update the number of books marked as read / unread

}

this.finishCurrentBook = function(){

//We're going to have to modify the state of this thing pretty heavily.

if (this.currentBook) {

this.currentBook.read = true;

this.lastBook = this.currentBook

this.currentBook = this.nextBook

// Now, we need to loop through and get the next book in the array that has read marked as "false"

//we also need to go through and update the number of books marked as read, and as not read.

} else {

//error handling - put something here that says "you don't have a current book!"

}

}

}