

Question2

February 10, 2024

```
[51]: #Q1. Create a dictionary for the above table. Each book should have the
      ↪ following
      #information: title, author, publication year, and availability status (whether
      ↪ the book is
      #currently available or checked out).

books = {
    'Book1': {'Title': 'The Great Gatsby', 'Author': 'F. Scott Fitzgerald',
    ↪ 'Year': 1925, 'Availability': True},
    'Book2': {'Title': 'To Kill a Mockingbird', 'Author': 'Harper Lee', 'Year':
    ↪ 1960, 'Availability': False},
    'Book3': {'Title': '1984', 'Author': 'George Orwell', 'year': 1949,
    ↪ 'Availability': True},
    'Book4': {'Title': 'Pride and Prejudice', 'Author': 'Jane Austen', 'Year':
    ↪ 1813, 'Availability': True},
    'Book5': {'Title': 'The Catcher in the Rye', 'Author': 'J.D. Salinger',
    ↪ 'Year': 1951, 'Availability': True},
}
```

```
[17]: #Q2. Write a Python function that takes this dictionary as input and prints out
      ↪ a message
      #indicating whether the book is available or checked out. Check if book2 is
      ↪ available or
      #not

def check_availability(books, book_key):
    if book_key in books:
        if books[book_key]['Availability']:
            book_status = 'Available'
            print(f'The book is {book_status}')
        else:
            print('Checked out')
    else:
        print(f'Book with key "{book_key}" not found in the inventory.')

check_availability(books, 'Book2')
```

Checked out

```
[54]: #Q3. Imagine a situation where a library user returns a checked-out book.
      ↪Modify the
      ↪function to take an additional parameter indicating whether the book is being
      ↪returned
      ↪or borrowed. If the book is being returned, update the book's availability
      ↪status
      ↪accordingly. Update status of book2

def Book_status(books, book_key, returned):
    if returned:
        if not books[book_key]['Availability']:
            print('book is returned')
            books[book_key]['Availability'] = True
        else:
            print('Book is already available and can not be returned')
    else:
        print('Book is borrowed')
        books[book_key]['Availability'] = False

Book_status(books, 'Book1', True)
```

Book is already available and can not be returned

```
[53]: def search_by_author(books, author):
      books_found = []

      for key, book in books.items():
          if book['Author'] == author:
              books_found.append(book)

      for book in books_found:
          print(f"{book['Title']} - {'Available' if book['Availability'] else
      ↪'Checked Out'}")

found_books = search_by_author(books, 'F. Scott Fitzgerald')
```

The Great Gatsby - Available

```
[ ]:
```