The Home Task can be done using Typescript or Vanilla JS.

You need to implement:

1. class Page:
   1. property “pageNumber”
   2. property “pageType” (“with text” for Book, “with article” for Magazine, “with images” for Comics)
   3. property “pageMaterial” (“simple paper” for Book, “glossy paper” for Magazine and Comics)
2. class Pages which is wrapper for array of Page instances
   1. should provide appropriate methods to communicate with iterator
3. PagesIterable mixin:
   1. You can use 3rd library or your own approach
   2. Use Symbol.iterator to implement PagesIterable mixin
   3. the **“for..of”**, **spread operator** (“…”) and **other data consumers** should work with your iterables (Book, Magazine, Comics)
4. class Book (should be Iterable):
   1. properties: “pages”, “title”, “author”
   2. setters and getters for “title” and “author”
   3. “toString” -> “Book: {book title} by {author} with number of pages: {number}”
   4. “pages” property keeps instance of Pages class
5. class Magazine (should be Iterable):
   1. properties: “pages”, “title”
   2. setters and getters for “title”
   3. “toString” -> “Magazine: {title} with number of pages: {number}”
   4. “pages” property keeps instance of Pages class
6. class Comics (should be Iterable):
   1. properties: “pages”, “title”, “author”, “artist”
   2. setters and getters for “title”, “author” and “artist”
   3. “toString” -> “Comics: {title} by {author}, the artist is {artist}, number of pages: {number}”
   4. pages” property keeps instance of Pages class
7. abstract class Item with abstract toString method. Make Book, Magazine and Comics inherited from Item class. Mix Iterable behavior into Item’s prototype

Usage:



As an extra mile you can try to implement PageFactory, so it will incapsulate pages creation logic.