

Könyvek

JS Objektumok

Rostagni Csaba

Az alábbi feladatban egy könyvesbolt termékeit bemutató oldalt kell elkészítenie.

A `data` mappában található `books.js` tartalmazza a könyv objektumok tömbjét az alábbiak szerint. Az opcionális mezők `null` értéket vesznek fel, ha nincs értékük.

Kulcs	típus	Érték
<code>title</code>	szöveg	A könyv címe
<code>author</code>	szöveg	A szerző neve
<code>publisher</code>	szöveg	A kiadó neve
<code>price</code>	egész	A könyv ára
<code>category</code>	szöveg	A könyv kategóriája
<code>hardcover</code>	logikai	Igaz, ha keménytáblás, míg puhatáblás könyv esetében hamis
<code>series</code>	szöveg	A könyvsorozat megnevezése (opcionális)
<code>sequence</code>	szöveg	Elhelyezkedése a sorozatban (opcionális)
<code>adaptation</code>	szöveg	Készítő megnevezése, ha készítettek belőle feldolgozást (opcionális)
<code>cover</code>	szöveg	A borítóképet tartalmazó fájl neve (opcionális)

Előkészületek

1. Másolja le a `forras` mappát `books-vezeteknev-keresztnev` néven, ahol `vezeteknev-keresztnev` helyére a saját nevét írja! Nyissa meg az így létrehozott projektmappát!
2. A `page.js` fájlban hozzon létre egy objektumot `classes` néven a badge osztályainak, ami az alábbi kulcs-érték párokat tartalmazza:

Kulcs	Érték
<code>hard</code>	<code>hard-cover</code>
<code>soft</code>	<code>paperback</code>
<code>series</code>	<code>series</code>
<code>adaptation</code>	<code>adaptation</code>

Badge

A könyvekhez tartozó badge-ek létrehozása

3. Hozzon létre egy függvényt `createBadge` néven, ami egy objektumot kap paraméterül.
 - Az objektumot már a paraméterben bontsa szét destrukturálással.
 - Az objektum két értéket tartalmaz a badge típusát (`type`) és a szöveges tartalmát (`content`).
 - A badge-nek hozzon létre egy inline általános tárolóelemet (`span`).
 - Mindenképpen adja hozzá a `badge` osztályt, továbbá a paraméterként kapott típus és a `classes` objektum alapján meghatározott osztálynevet.
 - Állítsa be tartalmának a paraméterül kapott objektum `content` értékét.
 - Adja vissza az elkészített elemet.
4. Hozzon létre egy függvényt `createBadgeRow` néven, ami egy könyvet kap paraméterül, azaz a `book` objektumot.
 - A badge-ek listájához hozzon létre egy blokk szintű általános tárolóelemet (`div`).
 - Adja hozzá a `badge-row` osztályt.
 - Generálja le az alábbi badge-eket és adja hozzá az imént létrehozott elemhez.
 - Borító
 - Amennyiben keményborítású a könyv, úgy adjon hozzá egy `hard` típusú badge-et *“Keménytáblás”* szöveggel,
 - különben adjon hozzá egy `soft` típusú badge-et *“Puhatáblás”* szöveggel.
 - Sorozat
 - Amennyiben valamilyen sorozat része, úgy a badge típusa legyen `series`,
 - míg a tartalma a sorozat neve és a sorszám a következő minta szerint: *“Sorozat neve - X. könyv”*
 - Például *“Gyűrűk ura - 1. könyv”*
 - Feldolgozás
 - Amennyiben a könyvből készítettek feldolgozást, úgy adjon hozzá egy badge-et `adaptation` típussal, a tartalom pedig legyen az adaptációt készítő megnevezése.
 - Végezetül adja vissza az elkészített elemet.

Könyvek megjelenítése

5. Hozzon létre egy függvényt `createBook` néven, ami paraméterként egy könyv objektumot kap.
6. Hozzon létre egy blokk szintű általános tárolóelemet (`div`).
 - Adja hozzá a `book` osztályt.
7. Hozzon létre egy képet, ami az `assets/img/covers` mappában található képek közül megjeleníti azt, ami az objektum szerinti, hozzá tartozó borító.
 - A képhez tartoz alternatív szöveg legyen a könyv címe.
8. A könyv tartalmi részéhez hozzon létre egy blokk szintű általános tárolóelemet (`div`).
 - Adja hozzá a `book-body` osztályt.
9. A könyv címét egy Címsor 2-es elembe jelenítse meg.
10. Hozzon létre egy számozatlan felsorolást a könyv adatainak tárolásához.
 - A felsorolás az alábbi tulajdonságokat jelenítse meg a minta szerint: *“Szerző”*, *“Kiadó”* és *“Kategória”*

11. A könyv árát egy bekezdésben helyezze el.
 - A formázáshoz alkalmazza a `toLocaleString()` függvényt:
 - legyen a formátuma (első paramétere) magyar (`hu-HU`)
 - a második paraméternek a következő objektumot adja át:
 - * A stílusa legyen `currency`
 - * a pénznem `HUF`, míg
 - * a `maximumFractionDigits` értéke legyen 0.
12. Adja hozzá a `book-body`-hoz a címet, a leírást, az árat és a badge-eket tartalmazó blokkot.
13. Adja hozzá a kártyához a képet és a tartalmi részét, majd adja vissza az elkészült elemet.
14. Készítsen egy függvényt `renderBooks` néven, ami megkapja a könyvek (már esetlegesen szűrt) listáját paraméterként.
 - Amennyiben a tömb mérete 0, úgy legyen a `booksContainer` szöveges tartalma: *“Nincs találat.”*
 - Egyéb esetben a `booksContainer`-ben jelenítse meg a könyveket a `createBook` függvény felhasználásával.
15. Biztosítsa, hogy az oldal betöltődésekor jelenjen meg az összes könyv.

Szűrés

A szűrés gombra kattintva ki kell olvasni a szűrési feltételeket és kiválogatni a szükséges könyveket. A Törlés gomb alapállapotba állítja az űrlapot.

16. Készítsen egy eseménykezelő függvényt `filterBooks` néven, aminek a bemeneti paraméter maga az esemény.
 - Biztosítsa, hogy ne hajtódjon végre az alapértelmezett esemény.
17. Olvassa ki a `filter_title`, `filter_min` és `filter_max` azonosítójú űrlapelemek értékeit.
18. Hozzon létre egy változót `filtered` néven, ami a szűrt eredményt tartalmazza az alábbiak szerint:
 - Amennyiben a cím mezőbe bármit is beleírtak, csak azok a könyvek jelenjenek meg, melyek címében megtalálható a szövegrészlet. Ügyeljen rá, hogy a kis- és nagybetűk ne számítsanak!
 - Amennyiben szám lett megadva a minimum értéknek (nem kell validálnia), úgy a legalább annyiba kerülő könyveket jelenítse meg.
 - A maximum hasonlóan viselkedjen, legfeljebb a megadott értékű könyvek jelenjenek meg.
19. Hívja meg a `renderBooks` függvényt a szűrt eredménnyel.
20. Állítsa be, ha a `books_filter` azonosítójú űrlap elküldésre kerül, akkor fusson le az előző lépésekben létrehozott eseménykezelő függvény.
21. A `reset_filters` azonosítójú gombra kattintva állítsa az űrlapot alaphelyzetbe.