

# Tömbök, tömbfüggvények, destrukció

*Kurityák Dániel*

Készítse el az alábbi feladatokat a megfelelő JavaScript fájlokban!

## Étlap

Az alábbi feladatokat a `task1.js` fájlba végezze el!

1. Készítsen egy tömböt `foods` néven, melynek elemei szintén tömbök legyenek az alábbi adatokkal:

Tömb első eleme (név)	Tömb második eleme (ár)
Gulyásleves	1800
Roston csirke cézársalátával	2400
Hortobágyi palacsinta	2200
Mogyoró	0

2. Készítsen egy függvényt `createItem(name, price)` néven, amely a bemeneti paraméterei alapján legenerál egy listaelemet az alábbiak szerint:
  - a. A listaelem egy `li` legyen, amely kapja meg az `item` osztályt!
  - b. Az adott étel nevét `span` elem jeleníti meg, melynek a szövege legyen a `name` paraméter értéke!
  - c. Az étel árát szintén egy `span` elem jelenítse meg, amely kapja meg az `item-price` osztályt. Abban az esetben, ha a `price` paraméter nulla, úgy az elem szövege legyen *“ingyenes”*, ellenkező esetben a paraméter értéke.
  - d. Adja hozzá a két `span` elemet a listaelemhez, majd ez legyen a függvény visszatérési értéke!
3. Generálja le az ételekhez tartozó listaelemeket, majd adja hozzá a `container` változóban eltárolt elemhez! Ügyeljen rá, hogy az elemben lévő *“Betöltés...”* feliratú listaelem cserélje le!

## PetStore

Az alábbi feladatokat a `task2.js` fájlba végezze el! A `products` tömb az árucikkek nevét, árát, majd napra lebontottan az eladott mennyiséget tartalmazza.

4. Egészítse ki a `createCard(text, classes)` függvényt úgy, hogy a függvényben létrehozott elem kapja meg a `classes` paraméterként megadott tömbben lévő elemeket, mint osztály! Használjon destrukciót!
5. Készítsen egy változót `mapped` néven, melynek értéke legyen a `products` tömb elemei úgy módosítva, hogy az árucikkek eladási adatai után az árucikkből származó bevétel szerepeljen! Használjon tömbfüggvényeket!
6. Menjen végig a `mapped` tömbön, és a tömb elemeit adja hozzá az `averageSales` változóban eltárolt táblázat elemhez a `createTableRow(name, sum)` függvény segítségével!
7. Válogassa ki egy `low` változóba azokat az árucikkeket, amelynek összbevétele nem érte el a 150 000 forintot! Használjon tömbfüggvényt!
8. Menjen végig a `low` tömbön, és a tömb elemeit adja hozzá a `lowIncome` változóban eltárolt tárolóelemhez a `createCard(text, classes)` függvény segítségével (a `classes` jelen esetben nem kap értéket)!
9. Tárolja el a három legmagasabb összbevétellel rendelkező árucikket, a bevétel szerint csökkenő sorrendben egy `best3` változóba! Használjon tömbfüggvényeket!
10. Menjen végig a `best3` tömbön, és a tömb elemeit adja hozzá a `topPerformers` változóban eltárolt tárolóelemhez a `createCard(text, classes)` függvény segítségével, a kártya az alábbi osztályokat kapja meg:
  - mindegyik árucikk a `star` osztályt,
  - ha az árucikk eladása eléri a 400 000 forintot, akkor az `better-star` osztályt is
11. Számolja ki a bevételek összegát egy `sum` változóba! Használjon tömbfüggvényt!

12. Állítsa be a `sum` változó értékét az `overallIncome` változóban eltárolt elem szövegének! Az érték formázásához használja a `convertToHuf()` függvényt!
13. Egészítse ki a `search(name)` függvényt úgy, hogy a paraméterben kapott `name` nevű árucikk árát állítsa be a `findProduct` változóban eltárolt elem szövegének! Az átlagot az előző feladatban használt függvény segítségével formázza meg!
  - a. Amennyiben nem található az adott néven árucikk, úgy – jelenjen meg az ár helyett!
  - b. Feltételezheti, hogy nincs két azonos nevű árucikk!

## Crazy Story

Az alábbi feladatokat a `task3.js` fájlba végezze el!

14. Készítsen egy függvényt `capitalize(text)` néven, amely a paraméterben megadott szöveg első karakterét nagybetűssé teszi! Használjon destrukciót vagy szöveg-/tömbfüggvényeket!
15. Kérje le az összes bekezdés elemet az oldalról, majd mindegyik szövegére alkalmazza az előző feladatban létrehozott függvényt úgy, hogy **a bekezdésekben minden szó nagybetűvel kezdődjön!** Ne felejtse frissíteni erre az elem szövegét!