

Tömbök, függvények, elágazások

JavaScript

Készítsd el az alábbi feladatokat. A megoldásaidat fel is pusholhatod a GitHubra. A feladatok egyre nehezednek, többek között azért, mert egyre kevesebb instrukciót kapsz a leírások mellé. Nem gond, ha nem érsz a feladatok végére.

1. Dublázd meg!

```
/*  
Írj egy függvényt, ami paraméterként kap egy tömböt és megduplázza annak  
minden elemét, majd visszaadja az új tömböt. Az új tömb értékeit írasd  
ki a konzolra!  
*/  
let testArray = [1, 2, 3, 4, 5, 6, 7];  
  
function doubleArray(arr) {  
    // ide írhatod a kódot  
}  
  
doubleArray(testArray);
```

2. Egészítsd ki!

```
/*  
Írj egy függvényt, ami egy string tömböt kap, és minden elemének  
elejéhez hozzáadja az "a" karaktert. Az új tömb elemeit írasd ki a  
konzolra!  
*/  
let testArray = ["lma", "nanász", "szta1", "kkord", "utó", "nakonda"];  
  
function addACharacter(arr) {  
    // ide írhatod a kódot  
}  
  
addACharacter(testArray);
```

3. Páros és páratlan

```
/*  
Írj egy függvényt, ami paraméterként egy integerekből álló tömböt kap,  
és annak elemeit szétválogatja két másik tömbbe. Egyikbe kerüljenek a  
páros, a másikba a páratlan számok.  
  
Plusz pontért old meg, hogy a függvény returnölje a két új tömböket!
```

```

*/

let testArray = [12, 55, 234, 8765, 4, 77, 196, 133, 759, 211, 17, 45,
7698, 3];

function sortNumbers(arr) {
    // ide írhatod a kódot
}

sortNumbers(testArray);

```

4. Töltsd fel a tömböt X-ig!

```

/*
Írj egy függvényt, ami bekér a usertől két pozitív egész számot, és egy
tömböt feltölti azokkal az egész számokkal, amelyek a két szám között
szerepelnek.

Második lépésben módosítsd a függvényt úgy, hogy ha az első szám nagyobb
mint a második, akkor csökkenő sorrendben töltsd fel a tömböt, ha
kisebb, akkor növekvő. Például:
első szám: 1
második szám: 4
tömb: [1, 2, 3, 4]

első szám: 8
második szám: 3
tömb: [8, 7, 6, 5, 4, 3]
*/

```

5. Classroom menedzsment program

```

/*
Felkértek, hogy írd egy programot, ami nyomon követi, kik vesznek részt
egy online programozó kurzuson. A résztvevőket egy tömbben tárold. Az
alábbi függvényeket kell elkészítened:
addParticipant() -> stringet kap, és hozzáadja a résztvevőt a listához;
removeParticipant() -> stringet kap és eltávolítja a hallgatót a
listáról,
printParticipants() -> a konzolra kiírja egyesével a résztvevők nevét

Egyel nehezebb lehet a feladat, ha az addParticipant() és a
removeParticipant() függvényekbe írsz egy validátort, ami azt ellenőrzi,
hogy valóban stringet adtunk-e meg. Amennyiben nem, akkor hibaüzenetet ír
ki. Szintén hibaüzenetet kapunk, ha a removeParticipant() függvény
meghívásakor egy nemlétező résztvevő nevét adjuk meg.

```

```
*/
```

6. Bazinga!

```
/*
```

Írj egy függvényt, ami bekér egy inputot (pozitív, egész szám), és egy üres tömböt feltölt 1-től addig a számig, amit a user megadott.

Ezt követően meghív egy másik függvényt, amely a tömbben minden 3-mal osztható számot "Baz"-ra, minden 5-tel osztható számot "inga"-ra, és minden 15-tel osztható számot "Bazinga"-ra cserél.

Ezt követően meghív egy harmadik függvényt, ami végigiterál a tömbön és kiírja annak elemeit.

```
*/
```

7. Hasáb felülete és térfogata

```
/*
```

Írj programot, amely bekér három pozitív integert a user-től, ezeket eltárolja egy tömbben, és kiszámolja annak a hasábnak a térfogatát és felületét, amelynek élei olyan hosszúak, mint a három szám. Az eredményeket írasd ki a konzolra.

Nehezebb a feladat, ha validációt is építesz bele. Azaz a program ellenőrizni, hogy integert adtunk-e meg (nem pedig stringet, booleant...stb.), illetve ellenőrzi, hogy a megadott számok pozitívak.

```
*/
```

8. Ki az ügyeletes?

```
/*
```

A Móka Miki Általános Iskolának három párhuzamos osztálya van. Minden osztálynak van egy-egy ügyelete. Írj egy függvényt, amelynek két paramétere van: a három osztály névsorát tartalmazó tömb, ill. egy három számot tartalmazó tömb, amely azt mutatja meg, a névsorban hányadik gyerek az aznapi ügyeletes. A függvény egy harmadik tömbbel tér vissza, amely az ügyeletes gyerekek neveit tartalmazza. Például:

Input tömb #1:

```
[
```

```
  ["Anna", "Béla", "Csabi", "Gergő", "Hajni", "Jani", "Kata"],  
  ["Ákos", "Cili", "Dénes", "Gitta", "Karola", "Lili", "Zotya"],  
  ["Ferkó", "Imre", "Márti", "Nóra", "Polly", "Szabi", "Titti"]
```

```
]
```

Input tömb #2

```
[1, 5, 3]
```

Output tömb

```
["Anna", "Karola", "Márti"]
```

```
*/
```

9. Közös kurzus

```
/*  
Írj egy függvényt, amely paraméterként két tömböt kap, két hallgató  
által felvett kurzusok listáját tartalmazza. A függvény egy harmadik  
tömböt ad vissza, amely tartalmazza azokat a kurzusokat, amelyeket  
mindketten felvettek. Például:  
input #1  
["Geography", "History", "Physics", "English"]  
  
input #2  
["Geography", "History", "Music", "Drama"]  
  
output  
["Geography", "History"]  
  
A megoldásodat a megadott tesztömbökkel ellenőrizd!  
*/  
// test #1  
let student1 = ["Math", "Music", "Biology"];  
let student2 = ["Drama", "Music", "Biology"];  
  
// test #2  
let student3 = ["Chemistry", "Sport", "History"];  
let student4 = ["Sport", "Physics", "English"];  
  
// test #3  
let student5 = ["French", "Math", "Programming"];  
let student6 = ["Psychology", "Literature", "Geometry"];
```

10. Az utolsó n szám

```
/*  
Írj egy függvényt, ami egy paraméterként kap egy tömböt és egy számot  
(n), és visszaadja a tömb utolsó n elemét.  
  
input #1  
["Star Wars", 12, 449, true, "jedi", false, -10, 0.12, "force"]  
  
input #2  
2  
  
output  
["force", 0.12]
```

```
*/
```

11. Forint formázás

```
/*  
Írj egy függvényt, ami paraméterként egy pozitív, egész számot kap és  
visszaad egy stringet, amit hármassával tagol és a végére illeszt, hogy  
",- forint".  
  
input  
176432454  
output  
"176 432 454,- forint"  
  
input  
84532  
output  
"84 532,- forint"  
*/
```

12. Leggyakoribb elem

```
/*  
Írj egy függvényt, amely paraméterként kap egy tömböt és visszaadja a  
leggyakoribb elemét.  
  
Input:  
[1, 5, 6, 1, 1, 4, 3, 1, 7, 1, 1, 5]  
  
output  
1  
*/
```

13. Unique values

```
/*  
Írj egy függvényt, ami egy paraméterként egy tömböt kap, és visszatérési  
értéke szintén egy tömb, ami az input tömb egyedi értékeit tartalmazza.  
  
input  
[1, 4, 7, 7, 8, 10, 10]  
  
output  
[1, 4, 7, 8, 10]  
*/
```

14. Szigorúan csökken/növekszik vagy sem?

```
/*  
Írj egy függvényt, ami paraméterként egy számokat tartalmazó tömböt kap,  
és egy boolean-nak tér vissza aszerint, hogy a tömb szigorúan csökken  
vagy növekszik.  
Szigorúan csökken/növekszik egy számsor akkor, ha egy szám mindig  
nagyobb/kisebb mint a közvetlenül utána jövő.  
  
input  
[1, 2, 5, 7, 8, 9]  
output  
true -> a számsor szigorúan növekszik  
  
input  
[10, 8, 8, 5, 4, 2, 1]  
output  
false -> a számsor csökken, de nem szigorúan  
*/
```

15. Cseréld ki

```
/*  
Írj egy függvényt, ami paraméterként kap egy 5 elemű tömböt, és  
kicseréli a 1 és 3 indexű elemeit.  
*/
```

16. Cseréld ki++

```
/*  
Alakítsd át ezt a függvényt úgy, hogy a tömb mellett kapjon egy második  
és harmadik paramétert, ami azt határozza meg, hogy a tömb mely két  
elemét cserélje ki.  
*/
```