

JavaScript Gyakorló Feladatok

Tömb Műveletek és Matematikai Függvények

Leibinger Bence

Általános Tudnivalók

- A feladatok megoldása során használja a strict mode-ot ('use strict';).
- A megoldásokat külön JavaScript fájlba írja (pl. `szamTombMuveletek.js`).
- A JavaScript fájlt csatolja egy HTML dokumentumhoz!
- Az eredményeket a konzolra írassa ki!
- Az **1. feladatban** létrehozott tömböt használja a további feladatok megoldásához.

1. Tömb Inicializálás és Random Számok

- **Feladat:** Hozzon létre egy függvényt `randomSzamok` néven!
- A függvénynek legyen egy paramétere (`tomb`), amelynek alapértelmezett értéke egy üres tömb.
 - Feltételezheti, hogy mindig tömböt kap (ha a meghívó kód mégsem így tesz, az nem gond ebben a feladatban).
- Ha a kapott tömb üres, akkor generáljon 5 darab véletlen (random) egész számot **1 és 100 között**, és tegye ezeket a tömbbe.
- Írassa ki a konzolra a tömböt.

Példa (lehetséges) kimenetre:

Tömb: [15, 28, 42, 95, 33]

2. Négyzetgyök Számítás

- **Feladat:** Számítsa ki az **1. feladatban** kapott tömb minden elemének négyzetgyökét!
- Az eredményeket **két tizedesjegyre** kerekítse, majd **string** formában tárolja el egy új tömbben.
- Írassa ki a konzolra az így kapott négyzetgyökök tömbjét.

Példa (lehetséges) kimenet:

Négyzetgyökök: ["3.87", "5.29", "6.48", "9.75", "5.74"]

3. Szélsőérték Keresés

- **Feladat:** Keresse meg az **1. feladatban** kapott tömb legnagyobb és legkisebb elemét!
- Hozzon létre egy új függvényt `getMinMax` néven, amelynek két paramétere legyen:
 1. Egy tömb (például `numbers`).
 2. Egy szöveges érték (például `tipus`), amely `"min"` vagy `"max"` lehet.
 - Ha `"min"`, akkor térjen vissza a tömb legkisebb elemével,
 - ha `"max"`, akkor a legnagyobb elemével.
- A `Math.min()` és `Math.max()` függvényeknél **használja a spread operátort** (`...numbers`), különben `NaN` eredményt kaphat!
- Hívja meg a függvényt mindkét módon (`"min"` és `"max"`), és a kapott eredményeket írassa ki a konzolra.

Példa (lehetséges) kimenet:

```
Legkisebb szám: 15  
Legnagyobb szám: 95
```

(Megjegyzés: Mivel a feladat mindkét szélsőértéket kéri, egyszerűen kétszer hívja meg a függvényt különböző paraméterrel.)

4. Átlagszámítás

- **Feladat:** Számítsa ki az **1. feladatban** kapott tömb elemeinek átlagát!
- Az eredményt **matematikai módon** (0.5-től felfelé) kerekítse egész számra
- Írassa ki a konzolra a kapott átlagot!

Példa (lehetséges) kimenet:

```
Átlag: 43
```

5. Páros-Páratlan Osztályozás

- **Feladat:** Válogassa szét a **1. feladatban** kapott tömb számait páros és páratlan kategóriákba!
- A páros és páratlan számokat külön tömbökben tárolja.
- Írassa ki a konzolra a két tömböt.

Példa (lehetséges) kimenet:

```
Páros számok: [28, 42]  
Páratlan számok: [15, 95, 33]
```

6. Hatványozás

- **Feladat:** Minden elemet hatványozzon meg a tömbbeli **indexének** megfelelő értékkel! Az indexelést most kivételesen ne nullától, hanem egytől kezdjük. Ügyeljen hogy az index ne essen a tömbhatáron kívülre.
 - Azaz az első (index 1) elem elem^1 lesz, a második (index 2) elem^2 , stb.
- Az eredményeket tárolja egy új tömbben, és írassa ki a konzolra.

Példa (lehetséges) kimenet:

```
Számok hatványai: [15, 784, 74088, 857375, 1185921]
```

7. Kategorizálás

- **Feladat:** Kategorizálja a **1. feladatban** kapott tömb számait három csoportra:
 - `alacsony` (< 30)
 - `közepes` ($30-69$)
 - `magas` (≥ 70)
- A feladatban használhat if-else if-else szerkezetet
- Az eredményeket három külön tömbben tárolja el, majd írassa ki őket a konzolra

Példa (lehetséges) kimenet:

```
alacsony: [15, 28] közepes: [42, 33] magas: [95]
```