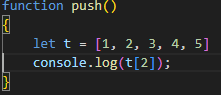
1-2.



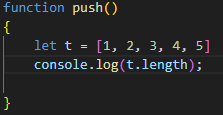
Létre hozzuk a tömböt az alábbi módon, majd vessző elválasztásával megadjuk az elemeket.

3.



Itt a konzolra kiíratjuk a ’console.log’ segítségével a tömb harmadik tagját. Mivel a tömb 0-tól indexelődik ezért a harmadik elem kiírásához a t tömb 2 indexű elemét kell vennünk.

4.



Hogy kiírassuk a tömb hosszát a .lenght nevezetű tömb tulajdonságot kell használni. Ez kiíratja a tömb összes elemét

5.



Hogy értéket módosítsunk az két féle képen lehet; .splice-szal vagy direktbe megadjuk. A splice-nál első helyen megadjuk melyik indextől, második helyen hány elemet és a harmadik helyen pedig az elemeket amiket hozzá szeretnénk adni a tömbhöz.

6.



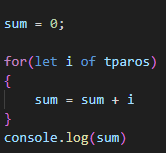
Hogy egy bizonyos helyről távolítsuk el elemet megint csak a .splice-ot kell használni, ugyanúgy mint előzőleg csak nem adunk meg új eleme(ke)t. A .pop és a .shift csak akkor jó, hogyha a tömb elejéről vagy hátuljáról akarunk elemeket eltávolítani. Viszont ezekből ki lehet nyerni az eltávolított adatot egy változóba.

7.



Az új tömbünk neve ”tparos”. Ebbe a tömbben az eredeti tömb páros számai lesznek benne. Ezt a . filter-rel és az megadott feltétellel érjük el.

8.



Ehhez egy ”sum” nevű változót hozok létre amiben majd az összeget eltárolom. Számolást pedig egy for ciklusban végzem el, a for ciklus végig megy a tömbön és a számokat hozzáadja a sum változóhoz. Majd console.log-gal kiíratunk.

9.



A növekvő sorrendbe állításhoz a .sort-ot használom. A sort függvény alapértelmezés szerint karakterláncokat rendez, így számok esetén az összehasonlítási logika kicsit eltérhet attól, amit az ember először gondolna. Viszont az ’(a, b) => a - b ’ miatt helyesen fogja rendezni őket.

10.



A ’szovegt’ nevezetű tömbbe fogjuk eltárolni . A map függvényt használva az eredeti tömb minden elemét átalakítjuk szöveggé a szam.toString() segítségével. Majd megint csak kiíratjuk a console.log-gal.