

Vježba 1

Callback funkcije:

1. Napisati funkciju `vecina()` koja kao argument prima niz i callback funkciju. Callback funkcija će vratiti `true` ili `false`. Funkcija `vecina()` će iterirati kroz niz i pozivati callback funkciju nad svakim elementom dokle god se ne može odrediti je li većina elemenata `true`. Ako je broj `true` povratnih vrijednosti, manji ili jednak broju `false` povratnih vrijednosti, funkcija `vecina()` vraća `false`. U protivnom vraća `true`.
2. Napisati funkciju `prioritiziraj()` koja prima niz i callback. Callback funkcija će vratiti `true` ili `false`. Funkcija `prioritiziraj()` će iterirati kroz niz i pozvati callback funkciju nad svakim elementom. Funkcija `prioritiziraj()` će vratiti novi niz u kojem su u prvom dijelu niza svi oni elementi za koje je callback vratila `true`, a u drugom dijelu svi oni elementi za koje je callback funkcija vratila `false`.
Proučiti funkciju `Array.unshift()`.
3. Kreirati funkciju `grupiraj()` koja prima niz i callback te vraća objekt. Funkcija `grupiraj()` će iterirati kroz niz i izvršiti callback funkciju nad svakim članom. Svaki `return` funkcije će postati ključ u objektu. Vrijednost na svakom ključu će biti niz čiji elementi su rezultat callback obrade.
Npr. Za niz `{1, 2, 3, 4, 5}` i `cb(input){if (input % 2 == 0) return true;}` povratna vrijednost funkcije `grupiraj()` je `{true:[2, 4], false:[1, 3, 5]}`.
4. Kreirati funkciju `komutativnost()` koja prima dvije callback funkcije i vrijednost. Funkcija `komutativnost()` će vratiti `true` ako prosljeđivanje vrijednosti u prvu callback funkciju i onda prenošenje rezultirajuće vrijednosti u drugu funkciju, daje isti rezultat kao prenošenje vrijednosti u drugu funkciju i prosljeđivanje rezultirajuće vrijednosti u prvu funkciju.
5. Kreirati funkciju `pipe()` koja prima niz (i to niz funkcija) i jednu vrijednost. Funkcija `pipe()` vrijednost ubacuje u prvu funkciju u nizu, te njen rezultat ubacuje u drugu funkciju u nizu, pa njen rezultat u treću i tako do zadnje funkcije u nizu. Funkcija `pipe()` vraća konačni rezultat.
6. Napisati svoju verziju JS funkcije `reduce()`. Funkcija prima 3 argumenta:
 - Niz
 - Callback funkciju koja će se primijeniti na svaki član niza
 - Akumulator koji će se vratiti, kao rezultat pozivanja callback funkcije na sve članove niza.

Za razliku od funkcije `map`, `reduce` ne vraća niz rezultata (za svaki element), već jedan zajednički rezultat (akumulator). Primjerice za niz `[1, 2, 3, 4, 5]` i callback funkciju `zbroji()`, rezultat funkcije `reduce` je 15.

`reduce([1, 2, 3, 4, 5], (acc, elem)=>acc+elem, 0);`

7. Koristeći izvedbu funkcije `reduce()` iz prethodnog zadatka, napisati funkciju `presjek()`. Funkcija `presjek`, iz dva niza koja prima kao argumente, vraća niz sa elementima koji se nalaze u oba niza.
8. Koristeći izvedbu funkcije `reduce()`, napisati funkciju `unija()`. Funkcija `unija()`, iz dva niza koja prima kao argumente, vraća niz objedinjenih elemenata iz oba niza.
9. Napisati svoju izvedbu funkcije `filter()`. Funkcija prima dva argumenta:
 - Ulazni Niz
 - Callback funkciju koja će se primjeniti na svaki element ulaznog niza, te će vratiti `true` ili `false`, na osnovu čega će se element odati u povratni niz (odnosno neće).Funkcija vraća povratni niz sa elementima koji su prošli filtriranje.