

# A csoport

Mi az osztály?

Mi a példányosítás?

Mi a kapcsolat az osztály és a példány között?

Mi a higher order function?

Mit nevezünk callbacknek?

Mit ír ki az alábbi program?

```
const foo = (a,b, callback) => {
  const c = 1;
  const result = callback(a,b)+1;
  return { er:result }
}
const result = foo("1", "2", (szam1, szam2) => {return szam1+szam2});
console.log(result.result);
console.log(result.er)
```

Hogyan módosítaná a programot, hogy short hand property alkalmazásával a visszatérési objektum result tulajdonsága tartalmazza kiszámított eredményt?

Hogyan tudná destructing alkalmazásával létrehozni azt a változót, ami foo függvény visszatérési értékének result tulajdonságát tartalmazza?

Javítsa a programot, hogy abban az esetben ha a és b paraméter nagyobb 10-nél, akkor az eredmény objektum result tulajdonsága 0 értéket vegyen fel.

Ha a két szám 0 és 10 között van, akkor az eredmény objektum result tulajdonsága a két szám összegét tartalmazza!

**A hibás functions.js függvényeinek javítása nélkül 0 pont dolgozat azon része, ahol használjuk őket. minden feladat után kötelező a commit (4 commit)**

A hibás függvények (1-1 sort módosítottam vagy töröltem) a következők: createTable, tbodyRenderRowspan, tbodyRenderColspan

Hozza létre a doga\_februar repót publikusan, cloneozza, majd hozza létre a csoport nevével megegyező nevű mappát (A, B, C, D) és másolja bele a következő gist tartalmát.

<https://gist.github.com/gomszab/88f6fecf39ab6f42d7fc82fe465cd4db>

1. Valósítsa meg a táblázat kirenderelését különálló Table és Manager osztályok létrehozásával. A managernél új elem létrehozásánál ügyeljen arra, hogy a táblázathoz kapcsolódó felületi elemek ne legyenek függésegek. Az új elem hozzáadása utáni elem hozzáfűzéshez használjon callback-et! A Table osztályt úgy alakítsa ki, hogy rowspanos, és colspanos táblázatokra is egyaránt alkalmazható legyen. Egy beadandó callback segítségével határozza meg, hogy a táblázat colspanos, vagy rowspanos.

2. Hozzon létre egy FormController osztályt, ami az űrlap megjelenítéséért és a beviteli mezőkből az adat kinyeréséért felel. A FormController függ a Manager osztálytól, az űrlap elküldésekor a Managernek csak egy metódusát használja. A beviteli mezők megjelenítésére használjon saját FormField osztályt!

3. Implementálja a validációt a FormController-en, hogy csak helyes adatbevitel esetén fűzzön hozzá elemet a Manager példány tömbjéhez.

4. Példányosítsa a Manager-t a Table-t és a FormController-t mind colspan, mind pedig rowspan értékekre a data.json alapján.