

Javascript module gyakorlás

Készíts egy projektet, amelyben JavaScript osztályokat és modulokat gyakorolsz, Git verziókezeléssel!

Ajánlott projektmappa: `htdocs/js-mod-gyak-13a-20250909-TeNeved/`

Követelmények

1. Hozz létre egy `Person` osztályt, amelynek van egy `_name` tulajdonsága, amit a konstruktor paraméterként kap meg.
2. Valósítsd meg az accessorokat (`get/set`) `name` számára.
3. Hozz létre egy `Student` osztályt, ami öröklí a `Person` tulajdonságait és kiegészül egy `marks` tömbbel (frontend jegyek).
4. A `Student` konstruktorában add meg a kezdő jegyeket, valamint a jegyek módosítása/lekérdezése `spread (...)` szintaxissal történjen.
5. Hozz létre egy `Teacher` osztályt, ami szintén öröklí a `Person`-ből, tartalmaz egy `students` tömböt a tanított diákokról, `spread` szintaxissal kezelve.
6. Az osztályokat exportáld külön modulban (pl. `person.js`, `student.js`, `teacher.js`), majd importáld egy handler modulba, ahol mindháromból példányokat hozol létre, minden mezőhöz legalább egy értéket rendel.
7. A handler modul exportálja tovább az osztálypéldányokat az `index.html`-nek.
8. Az `index.html`-ben, a `<body>` alatt JavaScript-ben példányosítsd mindhárom osztályt, hozz létre egy "Mutasd" gombot, amely kattintásra, külön `div` vagy `p` elemekben, igényes formázással megjelenít minden adatot, nem `JSON.stringify`-val.
9. Plusz opciónak készíthetsz:
 - új tanár felvétele
 - meglévő tanár törlése
 - új diák felvétele
 - meglévő diák törlése
 - meglévő diák tanárhoz rendelése (egyszerű formokkal)

Javaslatok a gyakorláshoz

Modul szervezés, exportálás

- A modulokat szervezd funkciók szerint külön fájlba (pl. modellek, handler, fő script), és named exportokat részesíts előnyben, így jobban támogatja az automatikus kódkiegészítést a fejlesztőkörnyezet.
- Az indexfájlban (pl. index.js) egyszerűsítsd a re-exportokat, hogy importáláskor könnyebb legyen elérni a főbb funkciókat.

Osztályok és öröklés

- A modern ES6 szintaxist használd: `class Student extends Person {}` és mindig hívd meg a szülő konstruktorát a konstruktorban `super()` hívással.
- Az accessorokat (getter/setter) célszerű privát mezővel (`_name`) együtt megvalósítani.

Spread szintaxis használata

- A spread szintaxis főként tömbmásolásra és új értékek illesztésére alkalmas, pl.: `this.marks = [...marks]`.
- Az adatmódosításnál használd spread-et, például új jegyek vagy új tanulók hozzáadásánál: `this.students = [...this.students, newStudent]`.

Továbbgondolás, haladó ötletek

- Minden "művelet" (diák hozzáadása, tanár törlése, stb.) legyen külön funkcióba vagy modulba szervezve.
- Formoknál validációt, hibakezelést is ajánlott beépíteni.
- A tanár-diák összerendelést objektum referenciával vagy ID-vel lehet megoldani (pl. relációs adatmodell).