

Callback gyakorló

1. Definiálj egy osztályt, aminek bemeneti paramétere egy string tulajdonságot és egy number tulajdonságot tartalmazó tömb. A bemeneti paramétert tárold el egy privát tulajdonságban. Definiálj az osztálynak egy metódust, amelynek bemeneti paramétere egy callback, amit privát tulajdonságban tárolt tömb minden elemén meghívunk. Példányosítsd az osztályt és hívd meg a metódusát egy olyan callback-el, ami a tömb minden elemének a szám tulajdonságát 0-val teszi egyenlővé. (callback paraméterei: elem)

Adott az alábbi tömb: <https://gist.github.com/gomszab/b215068bd87e4b3f9b7e25f9044bc47f>
Csinálj egy osztályt, amiben eltárolod.

1. Az osztálynak definiálj egy render metódust, amely egy callback-et vár, valamint egy tbody elemet, és a tömb minden egyes elemén meghívjuk. Hívd meg a render metódust egy olyan callbackel, ami hozzacsatolja bemeneti tbody-hoz a táblázat sorait, valamint amennyiben van concepts2 és title2 tulajdonság is, akkor „rowspanos” sort fűz hozzá a táblázathoz! Mielőtt példányosítod az osztályt, hozz létre egy 3 oszlopos táblázatot a html oldalon. (callback paraméterei: tbody, elem)

2. Hozz létre egy újabb 3 oszlopos táblázatot a html oldalon. Ezúttal hívd meg a render metódust úgy hogy csak az author, title1 és title2 tulajdonságot kell kirendelned. A callback-ben implementáld, a táblázat sorának hozzacsatolását a bemeneti tablebody-hoz. Amennyiben a title2 tulajdonságnak nincs értéke, a title1-es cell colspan 2-es legyen. (callback paraméterei: tbody, elem)

3. Definiálj az osztálynak egy renderOptionalCell függvényt, amelynek bemeneti paramétere egy callback, valamint egy tablebody. A metódus törzsében fűzd hozzá a tbody-hoz a kötelező tulajdonságokkal rendelkező cellákat egy sorban, majd hívd meg callbacket. Hozz létre egy újabb táblázatot, majd hívd meg a példányon az új metódust és callbackben implementáld az objektum vizsgálatát, valamint ha az objektumnak definiálva van a title2 concepts2 tulajdonság, akkor fűzzön hozzá egy új sort a táblázathoz, állítsa az első sor első cellájának rowspan-ját 2-re és hozzon létre a title2 concepts2 tulajdonságnak 1-1 cellát. (callback paraméterei: elem, tbody, első cella)

4. Definiálj egy újabb metódust az osztálynak. A metódusnak 5 paramétere van. 1 tablebody és 4 callback. Mind a 3 callbacket minden ciklus iterációjában hívjuk meg. Az első callbacket az elemmel és a tbody-val hívjuk meg, majd visszatérési értékét eltároljuk egy változóban. A második callback-et egy elágazásban hívjuk meg az aktuális elemmel, a harmadik callback-et az elágazás igaz ágában a tablebody és az aktuális elemmel, és az első callback visszatérési értékét tartalmazó változóval, a negyedik callback-et a hamis ágban a tablebody és az aktuális elemmel, és az első callback visszatérési értékét tartalmazó változóval. Hozz létre egy újabb táblázatot, majd hívd meg a metódust az osztály példányán, majd add meg a létrehozott táblázat törzsét az első paraméternek. A második paraméterben implementáld, hogy egy elem alapján hozzáfűzünk egy sort a táblázat törzséhez, majd létrehozunk 3 cellát (author, title1, concepts1), és visszatérünk az első létrehozott cellával. A harmadik paraméterben implementáld, hogy igazzal tér vissza ha a paraméterben kapott objektum title2 és concepts2 tulajdonsága definiálva van. A negyedik paraméterben implementáld, implementáld a „rowspanos” táblázat logikáját, a kapott html elemnek kell 2-es rowspan állítani. az ötödik paraméterben egy üres függvényt adj be.

5. Hívd meg az előző pontban definiált metódust egy újabb táblázattal, és a callback-eket implementáld a korábbiakban ismertetett colspanos logika alapján.

6. Definiálj egy függvényt ami egy bemeneti tömb alapján létrehoz egy dropdownlistet, valamint a második paraméterben megadunk egy callbacket, aminek a segítségével a dropdownlist értékét

tudjuk ellenőrizni. A callbacket a függvényben először a létrehozott dropdownListen hívjuk meg, majd a dropdownlistnek definiálunk egy change eseménykezelőt és az eseménykezelőn belül az esemény target tulajdonságával is. Hívjuk meg létrehozott függvényt, és implementáljunk egy olyan logikát, ami ellenőrzi a dropdownlist értékét és eszerint futtat valamilyen logikát.

7. Definiálj egy függvényt ami létrehoz egy checkboxot, és beállítja neki labelnek a függvény első paraméterét, majd a második paraméterben megadott callback-et meghívja a létrehozott elemen. A checkboxra ezután regisztráljunk egy change eseménykezelőt és az eseménykezelőn belül is hívjuk meg a függvényünk második paraméterben megadott callbackjét az esemény targetjén. Hívjuk meg létrehozott függvényt, és implementáljunk egy olyan logikát, ami ellenőrzi a checkbox értékét és eszerint futtat valamilyen logikát.

8. Implementáljunk egy függvényt ami létrehoz egy gombot, valamint egy checkboxot. A függvénynek 4 paramétere legyen. Az első paraméter a checkbox labeljének értéke, a második a gomb felirata. A függvényen belül iratkozzunk fel a gomb click eseményére, és azon belül vizsgáljuk a checkbox értékét. Ha a checkbox értéke igaz, akkor a 3. paraméterben megadott callback fusson le, ha hamis, a 4. paraméterben megadott callback.