

Utazási iroda

Az alábbi feladatban egy utazási iroda ajánlatait megjelenítő oldalt kell elkészítened. A megoldás során az adatok adottak, valamint az alap HTML és CSS.

1. Készíts egy függvényt `createSortOptions()` néven. Az `options` tömb alapján készíts a `option` elemeket egy új tömbbe, és a függvény ezt a tömböt adja vissza.
2. Az oldal betöltésére add hozzá a legördülő listához az előbbi elemeket.
3. Készíts egy `isAvailable(before, after)` függvényt, amely megadja, hogy az adott ajánlat elérhető-e. Ennek feltétele, hogy a mai nap a `before` és az `after` érték között legyen.
4. Hozz létre egy `createAvailableBadge(isAvailable)` függvényt, amely visszaad egy `span` elemet, `badge` osztállyal és a kapott logikai érték alapján:
 1. Ha igaz: `badge-new` osztályt is megkapja, valamint szövege legyen "Elérhető"
 2. Ha hamis: `badge-premium` osztályt kapjon és "Nem elérhető" szöveget
5. Legyen egy `createFeatureTags(features)`, amely egy tömböt kap paraméterül. A tömböt képezd le (`map`) `span` elemekre, amelyek a `tag` osztályt kapják és szövegük az adott elem értéke.
6. Hozd létre a `createPriceTag(cost)` függvényt, amely létrehoz egy `price` osztállyal ellátott tárolót, amelynek a tartalma legyen a `cost` értéke `toLocaleString` segítségével formázva. Legyen `hu-HU` string az első paramétere, második paramétere pedig egy objektum az alábbi tulajdonságokkal:
 1. Stílusa legyen: `currency`
 2. A pénznem: `HUF`,
 3. `maxFractionDigits` értéke pedig 0
7. A formázott érték után fűzd hozzá a `" /fő/éj"` szöveget is és az így elkészült tárolót add vissza.
8. Az előző függvényt felhasználva, készíts egy `createFeatureList(features)` függvényt, amely létrehoz egy általános tárolót, amely a `feature-tags` osztályt kapja és gyermek elemei legyenek az előbb létrehozott `span` elemek. A kész tárolót adja vissza a függvény.
9. Hozz létre egy `filtered` változót, amely kezdetértéke az `offers` tömb lesz.
10. Készítsd el a `loadOffers()` függvényt, amely a `#offers` tárolót fogja feltölteni
 1. Töröljük a tároló összes elemét, ezt követően hozzunk létre kártyákat a `filtered` tömb alapján.
 1. Legyen egy új tárolónk `offer-card` osztállyal:
 1. Amely egy képet tartalmazzon, amelynek a forrása `assets/images/` mappából és az ajánlatban található hotel képéből megadva, alternatív szövege legyen a hotel neve.
 2. Valamint egy harmadik szintű címsort, melynek tartalma szintén a hotel neve legyen.
 3. Végül egy bekezdést is hozz létre, amely tartalma az ajánlat összefoglaló szövege legyen.
 2. Hozz létre egy `card-body` osztályú tárolót.
 2. A `card-body` tároló gyerekei az alábbiak legyenek sorrendben:
 1. Az elérhetőséget mutató badge
 2. A hotel neve
 3. Az ajánlat szövege
 4. A hotel felszerettsége (features)
 5. Ára
 3. Az `offer-card` gyerekei legyen a kép és a `card-body`
11. Az oldal betöltésére hívd meg a `loadOffers()` függvényt.
12. Készíts `filter()` függvényt, amelyek a két beviteli mezőből kiolvassa a minimum és maximum árakat és ez alapján szűri ki megfelelő ajánlatokat. Ezeket a `filtered` változóba tároljuk el. Majd hívjuk meg a `loadOffers()` függvényt. Ez a művelet fusson le a "Szűrés" gombra kattintva.
13. Készíts `clearFilter()` függvényt, amely törli a megadott mezők értékét és a `filtered` értékét visszaállítja az `offers`-re, ez az "Újrakezdem" gombra kattintva fusson le.
14. Készíts `sortOffers()` függvényt, amely a `filtered`-be rendezi a megadott szempont alapján a tömböt, figyelj rá, hogy az eredeti tömböt ne módosítsd. Ha az "Alapértelmezett" érték kerül kiválasztásra, akkor az eredeti `offers` tömböt kapja meg.
15. Amikor a felhasználó kiválaszt valamit a legördülő listában, akkor fusson le az előző függvény.