

# Dashboard

*Kurityák Dániel*

Készítse el a dashboard-ot a leírás és a minta alapján!

Az API kéréseknél ügyeljen arra, hogy a szervertől csak JSON formátumú adatokat fogadjon el!

## Előkészületek

1. Klónozza le az aktuális alapot!
2. Másolja be a forrásban kapott `index.html` fájlt az alapban lévő helyére, valamint helyezze a `helpers.js` fájlt az `src/js` mappába (a mappát hozza létre, ha még nem létezik)!

## Név

3. Kérje le az adatokat az alábbi API végpontról, majd vizsgálja meg a válaszban kapott adatok szerkezetét!  
`https://randomuser.me/api?nat=us,gb`
4. A kapott adatokat feldolgozva jelenítse meg a válaszban lévő keresztnévet (first name) és utónevet (last name) a `name` azonosítójú elem szövegeként, a mintán látható módon!

## Emoji

5. Kérje le az adatokat az alábbi API végpontról, majd vizsgálja meg a válaszban kapott adatok szerkezetét!  
`https://emojihub.yurace.pro/api/random/category/smileys-and-people`
6. A válaszban kapott adatokból nyerve ki a `unicode` tömb első elemét, majd a `helpers.js` fájlból kiexportált `formatEmoji()` függvénnyel formázza ezt meg, és állítsa be az emoji elem szövegének!

## Árfolyam

7. Kérje le az adatokat az alábbi API végpontról, **a teljes URL-t az `urls.txt` fájlban találja**, majd vizsgálja meg a válaszban kapott adatok szerkezetét!  
`https://api.fiscaldata.treasury.gov/`
8. A visszakapott adatokban csak a `data` tömbben lévő(k) lesznek relevánsok. Ennek a tömbnek az első elemét, abból is a `exchange_rate` értékét jelenítse meg a `huf` azonosítójú elem szövegeként, a matematika szabályai szerint, a mintán látható tizedesjegyekre kerekítve!

## Hírek

9. A `news-card.html` tartalmazza egy kártyának a felépítését. Helyezze el az oldalon egy `template` elembe, adjon neki azonosítót!
10. Kérje le az adatokat az alábbi API végpontról, majd vizsgálja meg a válaszban kapott adatok szerkezetét!  
`https://api.spaceflightnewsapi.net/v4/articles/`
11. Módosítsa úgy a fenti URL címet, hogy a query paraméterek között szerepeljen a `limit`, melynek értéke legyen 6!
12. A `results` tömb tartalmazza a híreket. Menjen végig ezen a tömbön, majd készítsen minden eleméhez egy másolatot a fent létrehozott `template` tartalmából, amit töltsön ki az alábbiak szerint:
  - a. Az `img` elem forrás legyen az `image_url`, az alternatív szövege legyen a `title`!
  - b. A `h3` elem szövege legyen a `title`!
  - c. A `p` elem szövege legyen a `summary`!
  - d. A kártyát adja hozzá a `news` azonosítójú elem végére!

## Időjárás

13. A `weather-card.html` tartalmazza egy kártyának a felépítését. Helyezze el az oldalon egy `template` elemben, adjon neki azonosítót!
14. Kérje le az adatokat az alábbi API végpontról, **a teljes URL-t az `urls.txt` fájlban találja**, majd vizsgálja meg a válaszban kapott adatok szerkezetét!

`https://api.open-meteo.com/`

15. A `daily` objektum tartalmazza az egyes napoknak az adatát. Egy adott napnak azonosak az indexei mindegyik tömbben.
  - Például: Az alábbi válasz esetén 2026-02-03 dátummal a maximum hőmérséklet  $-6.6^{\circ}\text{C}$ , a minimum  $-13.3^{\circ}\text{C}$ , az összcsapadék 0mm (mert mindegyik indexe 0).

```
"daily": {
  "time": [
    "2026-02-03",
    "2026-02-04",
    "2026-02-05",
    "2026-02-06",
    "2026-02-07",
    "2026-02-08",
    "2026-02-09"
  ],
  "temperature_2m_max": [-6.6, -2.4, -1.2, -1.1, 0.1, 2.6, -2.7],
  "temperature_2m_min": [-13.3, -9.6, -2.1, -3, -2, -2.1, -9.3],
  "rain_sum": [0, 0, 0.6, 4.5, 1, 0, 0]
}
```

16. Minden naphoz készítsen a fent létrehozott `template` tartalmából, amit töltsön ki az alábbiak szerint:
  - a. Az `img` osztályú elem szövege legyen a dátum!
  - b. A `max` osztályú elem szövege legyen az aznapi maximum hőmérséklet a mintán látható módon, matematika szabályai szerint kerekítve!
  - c. A `min` osztályú elem szövege legyen az aznapi minimum hőmérséklet a mintán látható módon, matematika szabályai szerint kerekítve!
  - d. A `rain` osztályú elem szövege legyen az aznapi összcsapadék mennyisége!
  - e. A kártyát adja hozzá a `weather` azonosítójú elem végére!