

## 1. Feladat:

Készítsen egy JavaScript alkalmazást, amely egy aszinkron függvényt valósít meg!

- A függvény egy linket vár paraméterként, amely azt az API-t szolgáltatja, ahonnan adatokat lefoglalja: [Link](#)
- A függvény megvalósítása során kezelje az API elérhetetlenségéből vagy annak átalakításából származó hibákat a tanult hibakezelő módszerek alkalmazásával!
  - o Hiba esetén írja ki a hibát tartalmazó változó értékét és az alábbi szöveget is a konzolra: „Hiba történt: ”
- Ellenőrizze a kérés feldolgozásának helyességét az *ok* adattag vizsgálatával, dobjon új hibaüzenetet a konzolra, ha a helyességvizsgálat sikertelen volt!
- Jelenítse meg a feladathoz csatolt weboldalon az 1-es, 3-as és 5-ös *id* tulajdonsággal rendelkező adatokból a hozzájuk tartozó *email* tulajdonságok értékeit!
- Ne felejtse el kipróbálni a programot, azaz meghívni az aszinkron függvényt!

**Elvárt kimenet:**

### Adatok az API-ból:

```
ID: 1, Email: Eliseo@gardner.biz
```

```
ID: 3, Email: Nikita@garfield.biz
```

```
ID: 5, Email: Hayden@althea.biz
```

---

## 2. Feladat:

Készítsen másolatot az előző programból **Feladat2.js** néven, majd bővítse ki a kódot az alábbiak szerint:

- Hozzon létre egy **Comment** nevű osztályt
- Egy kommentről az alábbi tulajdonságokat (mezőket) tároljuk el:
  - o **id** - egyedi azonosító, minden kommentnek eltérő
  - o **nev** - komment neve, akár csak egy levél esetén a tárgy mező
  - o **email** - a komment írójának email címe
  - o **tartalom** - a komment tényleges üzenete

- Egy kommenthez az alábbi viselkedések (metódusok) tartoznak:
  - o **commentInfok()** – A függvény visszaadja a kommentekhez tartozó tulajdonságokat az alábbi módon: „A komment azonosítója: *id*, neve: *nev*, hozzátársuló email cím: *email*, komment tartalma: *tartalom*”
  
- Készítsen egy **CommentGyujtemeny** nevű osztályt!
  - o Az osztály tartalmazzon egy **commentek** nevű tömböt, amely **Comment** objektumokat fog tárolni
  - o Hozzon létre egy **commentBegyujtes(comment)** metódust, amely egy **Comment** objektumot ad hozzá a tömbhöz!
  - o Készítsen egy **megjelenites()** metódust, amely a **commentek** nevű tömbben található összes bejegyzés részleteit megjeleníti a feladathoz csatolt weboldalon!
    - A kiíráshoz használja fel a **Comment** osztályban megvalósított **commentInfok()** függvényt!
  
- Alakítsa át úgy az első feladatból származó kódot, hogy az API-ból származó kommentekből új **Comment** példányokat hozzon létre, és a kommenteket adja hozzá a **commentek** nevű tömbhöz!

### Elvárt kimenet:

### Adatok az API-ból:

A komment azonosítója: 1, neve: id labore ex et quam laborum, hozzátársuló email cím: Eliseo@gardner.biz, komment tartalma: laudantium enim quasi est quidem magnam voluptate ipsam eos tempora quo necessitatibus dolor quam autem quasi reiciendis et nam sapiente accusantium

A komment azonosítója: 3, neve: odio adipisci rerum aut animi, hozzátársuló email cím: Nikita@garfield.biz, komment tartalma: quia molestiae reprehenderit quasi aspernatur aut expedita occaecati aliquam eveniet laudantium omnis quibusdam delectus saepe quia accusamus maiores nam est cum et ducimus et vero voluptates excepturi deleniti ratione

A komment azonosítója: 5, neve: vero eaque aliquod doloribus et culpa, hozzátársuló email cím: Hayden@althea.biz, komment tartalma: harum non quasi et ratione tempore iure ex voluptates in ratione harum architecto fugit inventore cupiditate voluptates magni quo et

## 3. Feladat:

Készítsen egy JavaScript programot, amely az Objektum Orientált Programozás alapelveit valósítja meg az alábbiak szerint:

- Készítsen egy **Uvegautomata** nevű osztályt
- Egy **statikus uvegDarabszam** nevű változóban eltároljuk az összes bedobott üveg darabszámát! – Kezdőértéke: 0
- Egy automatáról az alábbi tulajdonságokat (mezőket) tároljuk el:
  - o **maxKapacitas** - Ennyi üveget képes fogadni az automata, az értéket az osztály példányosításakor kell meghatározni (Pl.: 5)
  - o **uvegek** – Objektum, amely két mezőt tartalmaz
    - **szines** – Kezdőértéke: 0

- **atlatsz**o – Kezdőértéke: 0
- Egy kommenthez az alábbi viselkedések (metódusok) tartoznak:
  - **üvegBehelyez(típus)** – Paraméterként az üveg típusát várja („színes” vagy „átlászó”)
    - Ha az össz-üvegmennyiség nagyobb-egyenlő, mint a maximális kapacitása az automatának, a függvény írja ki konzolra az alábbi: „Az automata tele van!”
      - Továbbá az automata ne vegye be az üveget ilyenkor!
    - Ha az automata beveszi az üveget, akkor a megfelelő változók értékeit növelje meg egyel!
  - **kapacitasEllenorzes()** – Visszatér egy szöveggel, amely a bedobott üvegeket összdarabszámát és az automatában rendelkezésre álló helyet is kiírja az alábbi formában: „Bedobott üvegek: xy, Ennyi hely van még az automatában: yz”
  - **kiurit()** – Kiüríti az automatát, azaz az üvegek mennyiségét nullára állítja és a függvény visszatér az alábbi szöveggel: „Automata kiürítve!”
- Az osztály és minden függvényét próbálja ki!

**Elvart kimenet 10-es kapacitás esetén, színes és átlászó üvegeket behelyezve, *kapacitasEllenorzes()*, *kiurit()*, *kapacitasEllenorzes()* függvények meghívása után:**

```
Bedobott üvegek: 2, Ennyi hely van még az automatában: 8
Automata kiürítve!
Bedobott üvegek: 0, Ennyi hely van még az automatában: 10
```