Fetch API

Implementacija asinkronih operacija putem metode fetch

- Kao alternativu objektu tipa XMLHttpRequest za realizaciju asinkronih operacija možemo koristiti metodu **fetch()**
  - daje nam veću fleksibilnost te pojednostavljuje definiciju skripte

Implementacija asinkronih operacija putem metode fetch

- Rad metode fetch() temelji se na objektu **Promise** (obećanje)
  - objekt koji predstavlja eventualni uspješni završetak ili neuspjeh asinkrone operacije
  - rezultirati će jednom vrijednosti koja može biti ili riješena ili neriješena (odbačena)

```
new Promise((resolve, reject) => {
  ...
  if (...) {
    return resolve;
  }
  return reject;
})
.then();
```

Implementacija asinkronih operacija putem metode fetch

- Rad metode fetch() temelji se na objektu **Promise** (obećanje)
  - 1. dohvaćanje JSON sadržaja iz odgovora
  - 2. prosljeđivanje dohvaćenih podataka funkciji koja će s njima rukovati
  - 3. ako Promise rezultira s neuspješnom obradom zahtjeva, opis greške šaljemo funkciji koja će s njime rukovati

```
fetch("http://frodo.ess.hr/api/svi-filmovi-json.php")
.then(odgovor => odgovor.json()) // 1.
.then(podaci => rukujPodacima(podaci, roditelj)) // 2.
.catch(greska => rukujGreskom(greska.toString(), roditelj)); // 3.
```

Implementacija asinkronih operacija putem metode fetch

- Funkcije koje rukuju s odgovorom ili greškom definiramo kao **Arrow** funkcije:
  - omogućuju kraću sintaksu za definiranje funkcija (bez zagrada, ključnih riječi function i return, ili kombinacija navedenoga)
  - standardna definicija

    ```
    function(odgovor) {return odgovor.json()}
    ```
  - arrow funkcija

    ```
    odgovor => odgovor.json()
    ```