

Zadaci za učenje i vježbu

Predavanje 7. JavaScript 3. dio

Svi primjeri s predavanja dostupni su u repozitoriju <https://gitlab.com/fer-web1/docs>.

1. Objasnite razliku sinkronog i asinkronog kôda.
2. Nabrojite asinkrone načine izvedbe kôda u JavaScriptu.
3. Objasnite ulogu sistemskog stoga, WEB APIs, reda čekanja poruka i petlje događaja u izvođenju asinkronog kôda u JavaScriptu.
4. Objasnite što su obećanja (eng. promises) u JavaScriptu i u kojim stanjima se mogu nalaziti.
5. Navedite čemu služi izvršna funkcija (eng. executor function) u okviru obećanja u JavaScriptu. Napišite primjer takve funkcije.
6. Objasnite čemu služi funkcija `setTimeout` u JavaScriptu, koje parametre ima, te da li i kada je sinkrona i/ili asinkrona.
7. Proučite slajd 12 predavanja JavaScript 3. dio i modificirajte primjer na način da u slučaju stvaranja parnog broja baca dvije vrste greške:
 - a. Ako se radi o parnom broju `< 5`: "Error 1: This is generated from the promise executor!"
 - b. Ako se radi o parnom broju `> 5`: "Error 2: This is generated from the promise executor!"Također, modificirajte odsječak kôda na način da u slučaju generiranja neparnog broja umjesto poziva `resolve("An odd random number was generated.")`; koristi `resolve` kako bi se vratio generirani neparni broj.
8. Proučite slajd 14 predavanja JavaScript 3. dio i modificirajte primjer na način da unaprijedite rad funkcija u bloku `then`, tako da i one vraćaju nova stvorena obećanja. Promjene koje treba napraviti i mjesta na kojima ih treba napraviti su označene komentarima.

```
promise.catch(  
  function(error) {  
    return new Promise(function(resolve, reject) {  
      reject(error);  
    });  
  })  
  .then(  
    function(result) {  
      console.log("Resolve:" + result);  
      //ovdje dodati stvaranje i povratak obećanja koje prosljeđuje  
      //result  
    },  
    function(result) {  
      console.log("Reject:" + result);  
      //ovdje dodati stvaranje i povratak obećanja koje prosljeđuje  
      //result  
    }  
  )  
);
```

- 9.** Proučite slajd 17 predavanja JavaScript 3.dio i izmijenite primjer na način da se u slučaju nepostojanja datoteke `s12.promise.json`, pokuša otvoriti nova datoteka `s12.promise2.json`. Prilikom navedenoga koristite novi `fetch`, te sve potrebne pretplate s `then`, kao i u primjeru sa slajda 17.
- 10.** Proučite slajd 19 predavanja JavaScript 3.dio i izmijenite primjer na način da se u slučaju nepostojanja datoteke `s12.promise.json`, pokuša otvoriti nova datoteka `s12.promise2.json`. Prilikom navedenoga koristite novi `fetch` u kombinaciji s `await`, kao i u primjeru sa slajda 19.
- 11.** Proširite modul `SimpleModule2.js` sa slajda 22 na način da dodate funkcije za oduzimanje, množenje i dijeljenje. Navedene funkcije pozovite iz `SimpleModule.html` i iz `SimpleModule1.js`.