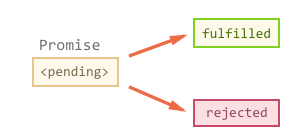
ლაბორატორიული 8

1. **Promises**

Promise ასინქრონული კოდის ორგანიზების ერთერთი საშუალებაა და წარმოადგენს ობიექტს 3 შესაძლო მდგომარეობით: მომდლოდინე, შესრულებული, უარყოფილი, რომლებიც აღწერენ ასინქრონული ოპერაციის მიმდინარე სტატუსს.



Promise-ზე შეიძლება 2 ტიპის callback-ის მიბმა:

onFulfilled - გამოიძახება როდესაც ოპერაცია წარმატებით დასრულდა

onRejected - გამოიძახება როდესაც ოპერაცია შეცდომით დასრულდა

მუშაობის ზოგადი პრინციპი შემდეგია:

1. ფუნქცია რომელიც ასრულებს რაღაცას ასინქრონულად, ქმნის Promise-ს და აბრუნებს მას, შედეგის ნაცვლად.
2. გარე კოდი რომელიც იყენებს ამ ფქუნციას, დაბრუნებულ promise-ზე აბავს callback-ებს
3. როდესაც ოპერაციის შედეგი ხელმისაწვდომია, ფუნქციას გადაყავს Promise შესაბამის მდგომარეობაში (fulfilled ან rejected). ამ დროს გარე კოდის მიერ მიბმული callback-ები გამოიძახება ავტომატურად

მაგალითი:

var promise = new Promise(function(resolve, reject) {

// აქ შეიძლება ნებისმიერი ასინრონული ოპერაციის შესრულება

// რომლის დასრულებისთანავაე უნდა გამოვიძახოთ:

// resolve(data) წარმატების შემთხვეავში და გადავცეთ შედეგად მიღებული მონაცემები

// reject(error) შეცდომის შემთხვევაში და გადავცეთ შეცდომის შესახებ ინფორმაცია

})

Promise-ზე callback-ის მისაბმელად უნივერსალური ფუნქცია then:

promise.then(onFulfilled, onRejected)

ასევე მხოლოდ შეცდომის callback-ის მისაბმელად catch:  
promise.catch(onRejected)

მაგალითად:

let p = new Promise((resolve, reject) => {

// throw-ს შედეგი reject(new Error("შეცდომა"))-ის გამოძახების ანალოგიურია

throw new Error("შეცდომა");

})

p.catch(alert); // Error: შეცდომა

promise-ს გამოყენების სრული კოდი (setTimeout-ის გამოყენებით):

let promise = new Promise((resolve, reject) => {

setTimeout(() => {

// გადაიყვანს fulfilled მდომარეობაში შედეგით "წარმატება"

resolve("წარმატება");

}, 1000);

});

// promise.then-ის გამოყენებით callback-ების მიბმა

promise

.then(

result => {

// წარმატების შემთხვევაში - როდესაც გამოიძახება resolve

alert("Fulfilled: " + result); // result – resolve-სთვის გადაცემული არგუმენტი

},

error => {

// შეცდომის შემთხვევაში, როდესაც გამოიძახება reject

alert("Rejected: " + error); // error – reject-სთვის გადაცემული არგუმენტი

}

);

Promise-ის შეცდომით შესრულების მაგალითი:

var promise = new Promise((resolve, reject) => {

setTimeout(() => {

reject(new Error("შეცდომა!"));

}, 1000);

});

promise

.then(

result => alert("Fulfilled: " + result),

error => alert("Rejected: " + error.message) // Rejected: შეცდომა!

);

დავალება

1. **L801** დაწერეთ ფუნქცია რომელიც დააბრუნებს promise-ს რომელიც 10 წამის დაყოვნების შემდეგ სრულდება წარმატებით და აბრუნებს მიმდინარე დროიდან აღებულ წუთებს თუ მიმდინარე დროიდან აღებული წუთები ლუწია, და შეცდომით („არაკორექტული დრო“) როდესაც მიმდინარე დროიდან აღებული წუთები კენტია. მაგალითად თუ ფუნქციის გამოძახებისას იყო 15:55:50, 10 წამის შემდგე იქნება 15:56:00, 56 ლუწი რიცხვია ამიტომ უნდა დასრულდეს წარმატებით და დააბრუნის 56.
2. **L802** დაწერეთ 3 ფუნქცია რომლებიდანაც პირველი აბრუნებს 10 წამის გავლის შემდეგ მიმდინარე დროიდან აღებული წამებს, ხოლო მეორე 20 წამის გვალის შემდეგ მიმდინარე დროდიან აღებული წუთებს, ხოლო მესამე მიმდინარე დროიდან აღებული საათებს, გამოიტანეთ 3-ვე ფუნქციის მიერ დაბრუნებული რიცხვების ჯამი.