Promises

JavaScript/ES6

1. Írj egy egyszerű függvényt **asyncTrueOrFalse** néven, ami visszaad egy Promise-t.  
   Ha a függvény  
    - igaz értéket kap, akkor a Promise resolve() hívódik meg a beadott értékkel  
    - hamis értéket kap, akkor a Promise reject() hívódik meg a beadott értékkel!  
     
   Teszteld a **.then** és .**catch** ággal!  
   Példa a .then -hez

|  |
| --- |
| **let** trueOrFalsePromise = asyncTrueOrFalse(**true**);  trueOrFalsePromise.then(data => console.log("promise was resolved:", data)); // true |
|  |

1. **Írd át a TODO http lekérést Promise verzióra!**XmlHttpRequest kérést csomagold egy Promise objectbe! → lásd slide példák:  
   - hiba eseén pl. hibás URL esetén (404 kód) a Promise kerüljön “rejected” státuszba→ catch metódus fut le  
   - teszteld a kódot!  
   - teszteld a getTodo hívást egy lokális mockTodo.json fájllal is, amiben JSON formában létezik pár todo object. Hozzd létre ajson fájlt és töltsd meg adattal!

|  |
| --- |
| getTodo("API\_URL").then(....).catch(....); |

(Bonus feladat) The built-in function setTimeout uses callbacks. Create a promise-based alternative.  
The function delay(ms) should return a promise. That promise should resolve after ms milliseconds, so that we can add .then to it, like this:

|  |
| --- |
| **function** **delay**(ms) {  // your code }  delay(3000).then(() => alert('runs after 3 seconds')); |