

NODE.JS FULLSTACK FEJLESZTÉS

4. forduló



A kategória támogatója: Emarsys

Ismertető a feladathoz

A 4. forduló után elérhetőek lesznek a helyezések %-os formában: azaz kiderül, hogy a kategóriában a versenyzők TOP 20% - 40% -60% -ához tartozol-e!

Szeretnénk rá felhívni figyelmedet, hogy a játék nem Forma-1-es verseny! Ha a gyorsaságod miatt kilököd a rendesen haladó versenyzőket, kizárást vonhat maga után!

Promise / Async

Felhasznált idő: 15:00/15:00

Elért pontszám: 0/6

1. feladat 0/2 pont

Jelöld be az **igaz** állításokat!

Válaszok

- ☐ A **new Promise(() => ...)** -beli függvény akkor fut le, amikor a promise resolved vagy rejected állapotba kerül.
- ☒ Ha egy async függvényben bevárt promise rejectel, akkor az await kivételt dob.
Ez a válasz helyes, de nem jelölted meg.
- ☐ Ha egy promise egyik **.catch** callbackje végrehajtódik, akkor ezután nem tud egyik **.then** callbackje se végrehajtódni.
- ☒ A **p = new Promise((res, rej) => { ... })** -beli res futtatásakor lefut **p.then(res)** -beli res függvény.
Ez a válasz helyes, de nem jelölted meg.
- ☒ A **then** egy két paraméteres függvény.
Ez a válasz helyes, de nem jelölted meg.

Magyarázat

A new Promise(() => ...) -beli függvény akkor fut le, amikor a promise resolved vagy rejected állapotba kerül: Hamis, mert a promise konstruktorral szinkron időben fut

Ha egy async függvényben bevárt promise rejectel, akkor a függvény hibát dob: Igaz

Ha egy promise elér az egyik .catch függvénybe, akkor ezután nem tud egyik .then függvénybe se eljutni: Hamis, ha a catch callback nem dob hibát, akkor az utána következő then-ek callback-je le tud futni

A p = new Promise((res, rej) => { ... }) -beli res és a p.then(res) -beli res függvény megegyeznek: Igaz

A **then** egy két paraméteres függvény: Igaz, az első paraméter az a függvény, ami resolve esetén, a második az, amelyik reject esetén fut le.

2. feladat 0/4 pont

Legyen adott a következő 4 függvény:

- fast: 1mp után sikeresen lefut és kiírja, hogy "fast success"
- slow: 3mp után sikeresen lefut és kiírja, hogy "slow success"
- fastFail: 2mp után hibát dob, "fast failure" szöveggel
- slowFail: 4mp után hibát dob, "slow failure" szöveggel

Mit ír ki az **await Promise.race([fast(), slow(), fastfail(), slowFail()])** függvényhívás?

Válasz

- ☐
 - fast success
 - Uncaught Error: "fast failure"
 - slow success
 - Uncaught Error: "fast failure"
- ☐
 - fast success
- ☒
 - fast success
 - slow success
- Ez a válasz helyes, de nem jelölted meg.**
- ☐
 - fast succes
 - Uncaught Error: "fast failure"

Magyarázat

A kód futtatása a leírt eredményt adja. A **promise.race** csak egy promise-t ad vissza.

3. feladat 0/0 pont

A 'totally-existing-website' sajnos nem stabil, ezért nem tudhatjuk, hogy amikor megpróbáljuk elérni, network error-ra futunk-e. Szerencsére a brute force megoldás működik, ezért úgy döntöttünk, hogy addig próbálkozunk, amíg vissza nem kapjuk a helyes választ.

Egészítsd ki az alábbi kódrészletet úgy, hogy a result értéke a válaszban kapott string legyen!

```
import {__1__} from '__2__'

const url = 'totally-existing-website';

let response;
do {
  response = __3__(url);
  await __1__(2000);
} while (__4__)
```

```
const result = response.__5__;
```

Válasz

- ☒ 1. setTimeout
2. util/timers
3. await fetch
4. !response.ok
5. text()

Ez a válasz helyes, de nem jelölted meg.

- ☐ 1. setImmediate
2. util/timers
3. await fetch
4. !response.ok
5. text

- ☐ 1. setTimeout
2. util/timers
3. fetch
4. !response.ok
5. toString()

- ☐ 1. setTimeout
2. timers
3. await fetch
4. !response.ok
5. toString()

- ☐ 1. setTimeout
2. util/timers
3. fetch
4. !response.success
5. text()

Magyarázat

Kedves Versenyzők!

A kérdést 0 pontosra kellett állítanunk: sajnos egy helytelen válaszlehetőség helyesnek lett megjelölve, ezért a kérdés, annak ellenére, hogy valójában csak egy helyes válaszlehetőség van, checkbox-szal jelent meg, amely alapján joggal jelölt valaki több válaszlehetőséget.

Elnézést kérünk a kellemetlenségért!

1. Helyesen működik a program
2. 'setImmediate' a node event loop végén fut le, nincs delay paramétere és 'text' egy futtatandó függvény
3. await hiányzik a 'fetch'-ről és 'toString()' nem létező függvény
4. 'timers' rossz modul név és 'toString()' nem létező függvény
5. await hiányzik a 'fetch'-ről és 'response.success' nem létező property



