

# FRONT-END FEJLESZTÉS

1. forduló

cl'ck

A kategória támogatója: Click Clock by BCS -  
Business Consulting Services Group

## Ismertető a feladatlaphoz

Kérjük, hogy a feladatlap indítása előtt mindenképp olvasd el az alábbi útmutatót:

MINDEN kérdésre van helyes válasz.

Olyan kérdés NINCS, amire az összes válasz helyes, ha mégis az összes választ bejelölöd, arra a feladatra automatikusan 0 pont jár.

Több válaszlehetőség esetén a helytelen válasz megjelölése mínusz pontot ér.

A radio button-os kérdésekre egy helyes válasz van.

Ha lejár a feladatlap ideje, a rendszer AUTOMATIKUSAN beküldi azt az addig megjelölt válaszokkal.

Azokat a feladatlapokat, amelyekhez csatolmány tartozik, javasoljuk NEM mobilon elindítani, erre az érintett feladatlapok előtt külön felhívjuk a figyelmet.

Az adatbekérős feladatokra NEM jár részpontszám, csak a feleletválasztósakra.

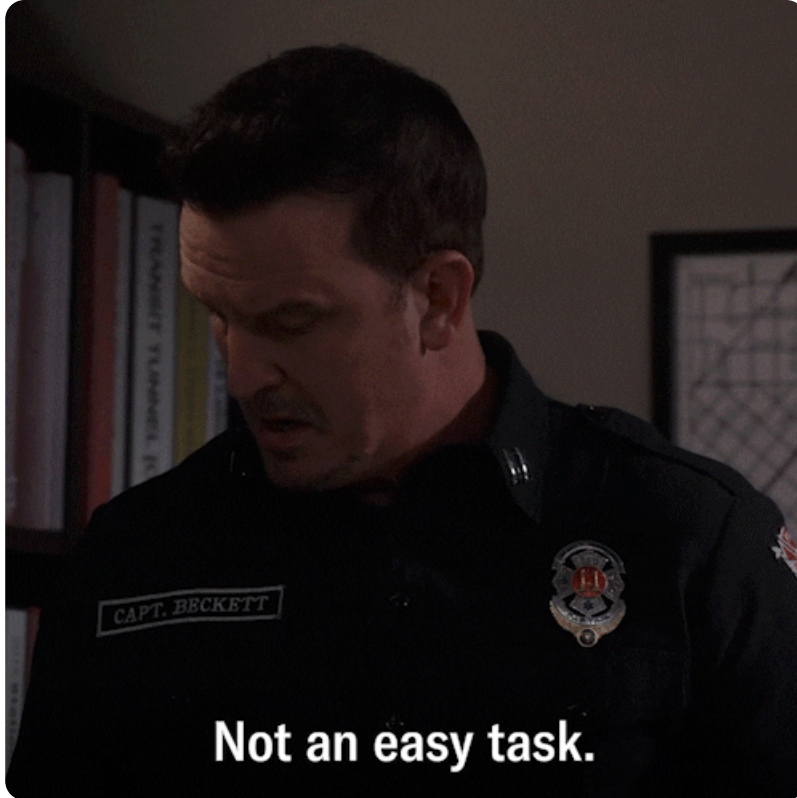
Helyezéseket a 4. forduló után mutatunk, százalékos formában: adott kategóriában a TOP 20-40-60%-hoz tartozol.

Ha egyszerre több böngészőből, több ablakban vagy több eszközről megnyitod ugyanazt a feladatlapot, nem tudjuk vállalni az adatmentéssel kapcsolatban esetlegesen felmerülő anomáliákért a felelősséget!

A ChatGPT használata nem tiltott, de az arra való hivatkozással észrevételt NEM fogadunk el!

A feltűnően rövid idő alatt megoldott feladatlapok kizárást vonnak maguk után, bármilyen más gyanús esetben fenntartjuk a jogot a forduló érvénytelenítésére!

Jó versenyzést kívánunk!



---

Az első fordulóban a JavaScript és TypeScript kapcsán lesz pár kérdés.

A large yellow square with the letters "JS" in bold black font, representing JavaScript.

A large blue square with the letters "TS" in bold white font, representing TypeScript.

## 1. feladat 1 pont

Az alábbiak közül melyik a legrövidebb mód arra modern JavaScript vagy TypeScript esetén, hogy biztonságosan (hiba nélkül) meghívjuk egy objektum memberjét, ha a member lehet függvény és null is?

## Válasz

☐

```
if(obj.fn) {  
  obj.fn();  
}
```

☐

```
obj.fn();
```

☐

```
obj?.fn()
```

☐

```
obj.fn?.()
```

## 2. feladat 2 pont

Mely állítások igazak az alábbi kódra?

```
async function query() {  
  throw "it is a bug";  
}  
  
async function runQuery() {  
  try {  
    return query();  
  } catch(e) {  
    console.error("Error happened:", e);  
    return null;  
  }  
}  
  
async function myCoolApp() {  
  console.log(await runQuery());  
}  
  
myCoolApp();
```

## Válaszok

☐

A dobott hiba kezeletlen marad, mert nincs await-elve a query() hívás eredménye a try-on belül

☐

A dobott hiba kezeletlen marad, mert nem Error típusú értéket dobtunk a throw-val

☐

A console-on megjelenő hiba stack trace-e nem lesz teljes, mert nem Error-t dobtunk, hanem stringet

- ☐ A console-ra kiíródik ez: Error happened: it is a bug
- ☐ A console-ra kiíródik ez: null
- ☐ A console-ra nem íródik ki ez: null

### 3. feladat 3 pont

Mely állítások igazak az alábbi kódra?

```
interface BaseResult<TData, TError> {
  data: TData | undefined
  error: TError | null
  status: string
}

interface SuccessResult<TData> extends BaseResult<TData, null> {
  data: TData
  error: null
  status: "success"
}

interface ErrorResult<TError> extends BaseResult<undefined, TError> {
  data: undefined
  error: TError
  status: "error"
}

type Result<TData, TError> = SuccessResult<TData> | ErrorResult<TError>;

const result: Result<string, Error> = query();

if(result.status === "success") {
  const data: string = result.data;
  const error: null = result.error;
  console.log("Data is: " + data + " , error is " + error);
} else {
  const data: undefined = result.data;
  const error: Error = result.error;
  console.log("Data is: " + data + " , error is " + error);
}
```

### Válaszok

- ☐ Az if ágban ez a hiba jelentkezik: Type 'string | undefined' is not assignable to type 'string'.
- ☐ Az if ágban ez a hiba jelentkezik: Type 'Error | null' is not assignable to type 'null'.
- ☐ Az else ágban ez a hiba jelentkezik: Type 'string | undefined' is not assignable to type 'undefined'.
- ☐ Az else ágban ez a hiba jelentkezik: Type 'Error | null' is not assignable to type 'Error'.
- ☐ Az elágazásnál nem kapunk semmilyen TypeScript fordítási hibát.
- ☐ Az if ágban a TypeScript ki tudja következtetni, hogy a data változó értéke csak string típusú lehet, az error változó értéke pedig csak null típusú lehet



Az else ágban a TypeScript ki tudja következtetni, hogy az error változó értéke csak Error típusú lehet, a data változó értéke pedig csak undefined típusú lehet

**Megoldások beküldése**