

REACT

1. forduló



A kategória támogatója: TCS - Tata Consultancy
Services

Ismertető a feladathoz

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.
- 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.
- **Badge-ke**t szintén a 4.forduló után kapsz majd először.
- 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 hét forduló során az egyes kategóriákban (de nem feltétlenül mindegyikben) **könnyű-közepes-nehéz kérdésekkel** egyaránt találkozhatasz majd.

Jó versenyzést kívánunk!

Felhasznált idő: 00:00/10:00

Elért pontszám: 0/4

1. feladat 0/1 pont

Mely állítások **igazak** a Virtual DOM (VDOM)-ra?

Válaszok

☒ DOM reprezentációja a JS memóriában
Ez a válasz helyes, de nem jelölted meg.

☒ Fa struktúra a memóriában
Ez a válasz helyes, de nem jelölted meg.

☐ Csak development time használja a rendszer

☐ Ugyanaz, mint a Shadow DOM

☐ A Fiber nem használja

Magyarázat

A Virtual DOM egy, a React által használt koncepció, miszerint a felhasználói interfész egy ideális reprezentációját a memóriában tartja, és ez alapján tartja szinkronban a tényleges DOM-ot a böngészőben.

Ezért a "DOM reprezentációja a JS memóriában" válasz helyes.

A Virtual DOM fa szerű struktúrában tárolja az adatokat hasonlóan, ahogy azok a DOM-ban is tárolva vannak.

Ezért a "Fa struktúra a memóriában" válasz helyes.

A Virtual DOM egy alapvető koncepciója a React-nak ezért ez development és production time is használva van.

Ezért a "Csak development time használja a rendszer" helytelen.

A Shadow DOM egy böngésző funkció, ami lehetővé teszi DOM elemek (pl. CSS stílusok) megjelenítését anélkül, hogy azokat a DOM-fába helyezné, nincs közvetlenül köze a React-hoz.

Ezért az "Ugyanaz, mint a Shadow DOM" válasz helytelen.

Fiber a React egy alrendszere, ami a Virtual DOM fokozatos építésért felelős.

Ezért a "A Fiber nem használja" válasz helytelen.

2. feladat 0/2 pont

Melyek a JSX buktatók?

Válaszok

☒ **for**-t nem lehet használni pl.

```
<label for="test">
```

helyette **htmlFor** kell

```
<label htmlFor="test">
```

Ez a válasz helyes, de nem jelölted meg.

☒ Nem lehet visszaadni szomszédos elemeket pl:

```
return (  
  <h1>Hello</h1>  
  <p>There</p>  
)
```

Ez a válasz helyes, de nem jelölted meg.

☒ Nem rendereli a

false, **true**, **null**, vagy **undefined** értékeket, pl:

```
<div>{false}</div>
```

Ez a válasz helyes, de nem jelölted meg.

☒ **class** nem működik pl.

```
<div class="sidebar" />
```

Ez a válasz helyes, de nem jelölted meg.

☒ Kisbetű nagybetű számít pl.:

```
<article> vs <Article />
```

Ez a válasz helyes, de nem jelölted meg.

☐ Kisbetű nagybetű nem számít pl.:

```
<article> vs <Article />
```

Magyarázat

A JSX a JavaScript egy nyelvi kiterjesztése, ami szándékosan hasonlít a HTML-hez, megkönnyítve a vele való UI készítést, de közelebb áll a JavaScript-hez, mint a HTML-hez.

Ezért nem használhatóak a különböző JavaScript kulcsszavak mint a **for** vagy a **class** közvetlenül, [lásd dokumentáció](#).

A JSX egy belső limitációja, hogy nem adható vissza több szomszédos elem. Ennek elkerülésére használható csomagoló elem (pl. `div`), vagy a `Fragment`, [lásd dokumentáció](#).

A JSX-ben a boolean, null és undefined értékek nem jelennek meg, [lásd dokumentáció](#).

A React a kisbetűvel kezdődő komponens neveket DOM tagként kezeli, ezért JSX-ben az elemek neve kis és nagybetű érzékeny, [lásd dokumentáció](#).

3. feladat 0/1 pont

Mire jó a

```
<React.StrictMode>
```

?

Válaszok

☐ jelez

```
var
```

hasznalatára

☒ Csak development time fut
Ez a válasz helyes, de nem jelölted meg.

☐ Jelez nullával osztásnál

☒ Jelez string ref API használatánál
Ez a válasz helyes, de nem jelölted meg.

Magyarázat

A Strick mód egy eszköz, amivel a potenciális problémákat lehet egy React alkalmazásban megfogni.

A **var** használata **const** vagy **let** helyett tisztán JavaScript nyelvi probléma lehet, erre a különböző linterek figyelhetnek, de nem React specifikus ezért a Strict mód nem ellenőrzni.

Ezért a **"a var használta"** válasz helytelen.

A Strict mód csak fejlesztési módban fut, ezért a production módban történő futást nem érinti.

Ezért a **"Csak development time fut"** válasz helyes.

Nullával való osztás egy futás idejű hiba, ami szintén nem React specifikus ezért a Strict mód nem ellenőrzni.

Ezért a **"Jelez nullával osztásnál"** válasz helytelen.

A Strict mód [több ellenőrzést](#) végez. Ezek közé tartozik a különböző régi, már nem használt funkciók használatára vonatkozó figyelmeztetés mint pl. a string ref API használata.

Ezért a **"Jelez string ref API használatánál"** válasz helyes.



[Legfontosabb tudnivalók](#)

[Kapcsolat](#)

[Versenyszabályzat](#)

[Adatvédelem](#)

© 2023 Human Priority Kft.

KÉSZÍTETTE **cone**

Megjelenés

Világos