Kurzusok

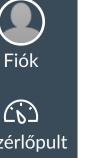
Naptár

Bejövő

üzenetek

StudyCoach

 \leftarrow



Kezdőlap Feladatok

Értékelések

2022/23/1

Due No Due Date **Pont** Nincs

Tudnivalók

- A zárthelyi megoldására 120 perc áll rendelkezésre.
- Humán segítségen kívül minden segédanyag használható (EA slideok, órai kódok, Google), de egy az egyben
- internetről másolt megoldást nem fogadunk el! • A feladatokat a Canvas rendszeren keresztül kell beadni. A rendszer pontban 18:00-kor lezár, ezután nincs
- lehetőség beadásra.
- A feladatok nem épülnek egymásra, tetszőleges sorrendben megoldhatók. • A feladatokat külön fileokba készítsd el. Ezeken kívül természetesen használhatsz más fájlokat is az adatok

PHP javító évfolyam ZH minta (2019-20-2)

PHP javító évfolyam ZH minta (2019-20-2)

- tárolására, a kódod további szervezésére, illetve a szerveroldali AJAX/Fetch funkcionalitás megvalósítására, de az alább megadott 4 néven kell hogy elérhetők legyenek a megoldásaid.
- 1. feladat: f1.php
- 2. feladat: f2.php
- 3. feladat: f3.php
- Egy külön README.md fájlban szerepeljen a következő kijelentés (behelyettsítve a megfelelő adatokat, a <> jeleket nem kell beleírni):

<Hallgató neve> <Neptun kódja> <Tárgy neve> - PHP javító évfolyam ZH Ezt a megoldást <Hallgató neve, Neptun kódja> küldte be és készítette a <Tárgy neve> kurzus PHP javító évfolyam ZH-jáho Kijelentem, hogy ez a megoldás a saját munkám. Nem másoltam vagy használtam harmadik féltől származó megoldásokat. Nem t ovábbítottam megoldást hallgatótársaimnak, és nem is tettem közzé. Az Eötvös Loránd Tudományegyetem Hallgatói Követelmén yrendszere (ELTE szervezeti és működési szabályzata, II. Kötet, 74/C. §) kimondja, hogy mindaddig, amíg egy hallgató egy másik hallgató munkáját - vagy legalábbis annak jelentős részét - saját munkájaként mutatja be, az fegyelmi vétségnek sz ámít. A fegyelmi vétség legsúlyosabb következménye a hallgató elbocsátása az egyetemről.

• A README.md fájlban a kijelentés alatt egy üres sorral elválasztva szerepeljen az alábbi lista. Az egyes [] közötti szóközt cseréld le x-re azokra a feladatokra, amit sikerült (akár részben) megoldanod!

- [] 1. feladat

- [] 2. feladat - [] 3. feladat

A megfelelően kitöltött `README.md` fájl nélkül a megoldást nem fogadjuk el! Minden feladathoz tartozik kiinduló kód (HTML, JSON, PHP): kiinduló kód letöltése

A kiinduló HTML kódot illetve a JSON adatfájlt a kiinduló csomagban találod.

1. feladat (f1.php - 10 pont)

Egy adatfájlban (data.json) tároljuk egy felhasználó heti munkanyilvántartását. Az adatfájlban az alábbi formátumú rekordokat tárolunk:

- day: a hét napja (1..7)
- from: a munka kezdetének órája: (0..24) • to: a munka végének órája: (0..24)

Tudjuk, hogy minden rekordban a from kisebb, mint a to mező értéke. Egy napon több munka is lehetett (pl. egy reggel, egy délután). Számítsd ki, hogy az egyes napokon összesen hány órát dolgozott az alkalmazott. (2 pont)

Készíts vizualizációt a munkanyilvántartás adataihoz: A hét napjaihoz (H-V) tartozó munkaórákat táblázatban vizualizáljuk színek segítségével. Minden napra az előírt munkamennyiség 8 óra. Ezt egy színskálán jelöljük a zöld árnyalataival. Az egyes napokhoz tartozó táblázatcellák színe hsl(120, 50%, <mennyiség>%) legyen, ahol a mennyiség a napi szükséges 8 óra munkából az elvégzett munka %-ban. (2 pont) Ha valamelyik napon a munkavállaló túlórázott, akkor azt a napot piros ("red") színnel jelöljük. (1 pont)



A táblázat alatt jelenjen meg egy űrlap, amivel újabb munkaórákat tudunk felvenni az adatfájlba. Az űrlapon egy legördülő listából lehet kiválasztani a napot, a kezdés és a befejezés időpontját szám beviteli mezők segítségével adjuk meg. Az űrlapot **POST** metódussal lehet elküldeni a szervernek. A szerveren végezz ellenőrzést a bemeneten:

- A kezdés és a befejezés időpontja pozitív egész szám 0 és 24 között. (2 pont)
- A kezdés időpontja korábbi, mint a befejezés időpontja. (1 pont)

Az esetleges átfedéseket a munkaórák között nem kell figyelned. Ha minden feltétel teljesül, akkor szúrd be az adatfájlba az új rekordot a megfelelő formátumban! (2 pont) Hiba esetén jelenjenek meg a hibaüzenetek a felületen! (1 pont)

2. feladat (f2.php - 10 pont)

Az oldal HTML kódját a kiinduló csomagban találod.

Készíts dátumkitalálós játékot! A felhasználói felületen egy beviteli mező van, melybe egy dátumot lehet beírni. A szerver az oldal első betöltésekor "gondol egy dátumra" véletlenszerűen:

- A véletlenszerűen kiválasztott dátum 2000.01.01. és 2020.12.31. között van. (2 pont)
- Csak a nap számít, órával, perccel, másodperccel nem foglalkozik.
- A véletlenszerűen kiválasztott dátumot a szerver munkamenetben tárolja. (1 pont)

A felületen található gombra kattintva AJAX/Fetch kérést küldünk a szervenek. A szerveroldalon egy PHP script fogadja a klienstől az AJAX/Fetch kéréseket. A kliens egy darab értétéket, a dátum beviteli mező értékét küldi el a szervernek (1 pont)! A szerveroldali PHP szkript HTTP metódusok és JSON formátumú adat segítségével válaszol.

A szerver fogadja az AJAX/Fetch kérésben beérkezett adatot és ellenőrzni annak helyességét. Amennyiben a beérkezett adat nem dátum formátumú, azt a szerver 300-as HTTP hibakódban jelezze és egy JSON objektumot küld vissza, aminek egyetlen "error" mezőjében az "Invalid input format" szöveg szerepel (1 pont). Helyes formátum esetén:

- Ha a dátum korábbi, akkor a válasz: HTTP: 200 OK, JSON: { "guess": "Later" } (1 pont)
- ha a dátum később, mint a szerveren tárolt, akkor a válasz: HTTP: 200 OK, JSON: { "guess": "Earlier" } (1 pont) • ha napra pontosan eltaláltuk a dátumot, akkor a válasz: HTTP: 200 OK, JSON: { "guess": "That's it!" } (1 pont)

A szerver által adott válasz szövege megjelenik a felhasználói felületen az output elemben. (1 pont) Ha a felhasználó eltalálta a választ, akkor a szerver új dátumot választ. (1 pont)

3. feladat (f3.php - 10 pont)

A kiinduló HTML kódot, illetve a word_diff fügvényt tartalmazó word_diff.php fált a kiinduló csomagban találod. A word_diff használathoz segítséget a word_diff.example.md fájlban találsz.

Adott egy fájlban egy word_diff nevű PHP függvény, ami képes két szöveg közötti különbségét meghatározni. Készíts egy PHP oldalt, mely két szövegbeviteli mezőt (egy "bemeneti" és egy "kimeneti") tartalmaz. A bemeneti mezőbe be lehet írni egy szöveget és egy gomb lenyomására el lehet küldeni a szervernek. Az adatok elküldése után a "bemeneti" mezőbe írd vissza a szervernek elküldött szöveget. (1 pont) A "kimeneti" mező tartalmazza a felküldött szöveg és az azt megelőző szöveg közötti különbséget az alábbiak szerint:

- 1. Számítsd ki a különbséget a bemenet és az előző bemenet között a word_diff függvény segítségével (3 pont). A word_diff bemenete két string: a régi szöveg és az új szöveg. Kimenete egy tömb, melynek elemei az alábbiak lehetnek:
- egy string: ebben az esetben ez egy szó, ami a két szövegben megegyezik • egy két elemű tömb tömb:
- "del": törölt szöveg
- "ins": beszúrt szöveg
- 2. A word_diff függvény tömb kimenetéből állíts elő egy szöveget az alábbiak szerint: • ha az aktuális tömbelem string, akkor kiírjuk (1 pont),
 - ha az aktuális tömbelem *tömb*, akkor: a tömb "del" mezőjében található szöveget kjírjuk egy "<dels" HTML elemben (1 5 nont)

	PHP javító ZH				
 összes kiírandó elem szóközökkel elválasztva jelenik meg (2 pont) 					
	 a tömb "ins" mezőjében található szöveget kiírjuk egy "<ins>" HTML elemben (1.5 pont)</ins> 				
	• a tomb "del "mezojeben talalnato szöveget kilrjuk egy " " H i ML elemben (1.5 pont)				

PHP javító ZH					
Kritérium	Érté	Értékelések			
1. kiszámítja naponta hány órát dolgozott	2.0 pont Teljes megoldás	0.0 pont Nem értékelhető	2.0 pont		
1. meghatározza a színt minden naphoz	2.0 pont Teljes megoldás	0.0 pont Nem értékelhető	2.0 pont		
1. túlóra esetén piros	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont		
1. ellenőrzés: szám, 0-24	2.0 pont Teljes megoldás	0.0 pont Nem értékelhető	2.0 pont		
1. kezdés < befejezés	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont		
1. új rekordot beszúrja	2.0 pont Teljes megoldás	0.0 pont Nem értékelhető	2.0 pont		
1. hibaüzenetek megjelennek	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont		
2. generál random dátumot	2.0 pont Teljes megoldás	0.0 pont Nem értékelhető	2.0 pont		
2. SESSION-ben tárolja	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont		
2. AJAX-szal felküldi a tippet	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont		
2. formátum hiba -> 300 error	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont		
2. korábbi -> 200 ok	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont		
2. későbbi -> 200 ok	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont		
2. eltalálta -> 200 ok	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont		
2. válasz megjelenik	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont		
2. ha eltalálta akkor újat generál	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont		
3. visszaírja az elküldött szöveget	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont		
3. kiszámolja a diff-et + tudja, mi volt az előző kérés	3.0 pont Teljes megoldás	0.0 pont Nem értékelhető	3.0 pont		
3. string érték belekerül a kimenetbe	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont		
3. del-t megfelelően generál	1.5 pont Teljes megoldás	0.0 pont Nem értékelhető	1.5 pont		
3. ins-et megfelelően generál	1.5 pont Teljes megoldás	0.0 pont Nem értékelhető	1.5 pont		

◆ Előző Következő ▶

2.0 pont

Teljes megoldás

0.0 pont

Nem értékelhető

2.0 pont

Összpontszám: 31.0 pontból

3. a teljes output OK