

PHP PROGRAMOZÁS

3. forduló



A kategória támogatója: Human Priority Kft.

Ismertető a feladathoz

A 3.forduló feladatait a hosszú hétvége miatt kivételesen szerda (11.02.) éjfélig tudod megoldani!

Érdemes ebben a fordulóban is játszani, mert a következő forduló kezdetekor, 11.03-án 18 órától kiosztjuk az 1.-2.-3. fordulóban megszerzett badgeket!

A verseny közben az alábbi teljesítményeket díjazzuk:

- fordulógyőztes
- átlagnál jobb időeredmény
- átlag feletti pontszám
- hibátlan forduló

Szeretnénk rá felhívni figyelmedet, hogy az egyszer megkapott badge-eket nem vonjuk vissza, akkor sem, ha esetleg az adott fordulóban a visszajelzések alapján változások vannak.

Jó játékot!

A feladatok értelmezéséhez a [PHP 7.4.32](#) verziót javasoljuk!

Felhasznált idő: 12:27/25:00

Elért pontszám: 9/14

1. feladat 1/1 pont

Az alábbiak közül mely parancs konvertál csupa nagybetűssé egy stringet?

Válasz

☐ `str_toupper()`

☒ `strtoupper()`
Ez a válasz helyes, és meg is jelölted.

☐ `strtouppercase()`

☐ `str_uppercase()`

Magyarázat

2. feladat 0/2 pont

Mi lesz a ***\$variable*** változó értéke?

```
  
$variable = 5;  
  
function doSomeFunc(&$var) {  
    $var += $var * 2;  
    $variable = $var + 1;  
}  
  
doSomeFunc($variable);  
echo $variable;
```

Válasz



5

Ez a válasz helytelen, de megjelölted.



6



15

Ez a válasz helyes, de nem jelölted meg.



10

Magyarázat

A ***doSomeFunc()*** metódusban a ***\$var*** paraméter referenciaként lett átadva. Ilyen esetben bármit is csinálunk azzal a változóval a metóduson belül, az felül fogja írni annak a változónak az értékét a metóduson kívül is.

A metóduson belül az alábbi történik:

```
$var += $var * 2; // $var += 5 * 2, ahol a += operátor a meglévő értékhez hozzáadja azt, amit az egyenlőség után írtunk. Tehát:  
$var = 5 + 5 * 2 = 5 + 10 = 15
```

Mivel a metódus paraméterezésénél a ***\$var*** nevet használtuk, így a metóduson belüli ***\$variable*** változó használata nincs rá hatással, hiszen az csak a metóduson belüli lokális változó.

3. feladat 3/3 pont

Milyen logikai értéket vesz fel a **\$var** változó?



```
$var = 3 > 4 && (5 > 2 || true);
```

Válasz

☐ true

☒ false

Ez a válasz helyes, és meg is jelölted.

Magyarázat

A zárójelben lévő rész két feltételből áll, mindkettő IGAZ ($5 > 2 = \text{igaz}$, $\text{true} = \text{igaz}$). Köztük VAGY logikai kapcsolat van, így akár csak egy IGAZ feltétel esetén is IGAZ lenne a részeredmény, de esetünkben mindkét részfeltétel IGAZ, tehát a zárójel felbontása után ez a rész IGAZ.

A zárójelen kívüli másik rész azonban hamis ($3 > 4 = \text{hamis}$).

A két feltétel rész között ÉS logikai kapcsolat van, amely csak akkor ad IGAZ eredményt, ha minden tagja IGAZ. Esetünkben tehát a zárójel felbontása után a feltétel így néz ki: HAMIS && IGAZ = HAMIS

4. feladat 0/3 pont

Az alábbiak közül mely feltétel **IGAZ**?

Válasz



`empty(['']) === false`

Ez a válasz helyes, de nem jelölted meg.



`empty([]) === false`



`empty((object)'foo') === true`

Ez a válasz helytelen, de megjelölted.



`empty('') === false`

Magyarázat

A `empty(['']) === false` válasz esetén a tömbnek van egy eleme, mely egy üres stringet tartalmaz. Erre az `empty()` függvény HAMIS értéket ad (hiszen a tömb nem üres), így ez az egyetlen helyes válasz.

`empty([]) === false` válasz esetén a tömb üres, így nem kaphatunk HAMIS értéket.

`empty((object)'foo') === true` válasz esetén az objektum nem üres, így nem kaphatunk IGAZ értéket.

5. feladat 5/5 pont

Mi lesz az értéke a **`$result`** változónak az alábbi kódrészletben?

```
$first = [  
    'firstname' => 'Bela',  
    'lastname' => 'Teszt',  
    'telephone' => '123456',  
    'street' => 'Teszt utca 11',  
    'postcode' => '1111',  
    'city' => 'Budapest',  
    'country' => 'HU',  
];  
  
$second = [  
    'firstname' => 'Béla',  
    'lastname' => 'Teszt',  
    'telephone' => 123456,  
    'street' => 'Teszt utca ' . 11,  
    'postcode' => 1111,  
    'city' => 'Budapest',  
    'country' => 'HU',  
];  
  
$result = array_diff($first,  
$second);
```

Válasz

☒ `array(1) { ["firstname"]=> string(4) "Bela" }`

Ez a válasz helyes, és meg is jelölted.

☐ `array(1) { ["firstname"]=> string(4) "Béla" }`

☐ `array(4) { ["firstname"]=> string(4) "Bela" ["telephone"]=> string(6) "123456" ["street"]=> string(13) "Teszt utca 11" ["postcode"]=> string(4) "1111" }`

☐ `array(0) { }`

Magyarázat

Az `array_diff()` tömbösszehasonlítás esetén úgynevezett loose comparison történik, mely nem foglalkozik az adatok típusával, csak az értékükkel. Eszerint az egyetlen különbség a 'firstname' tömbelemében lévő Bela - Béla lesz. A 'postcode', 'telephone' és 'street' adatokban nincs különbség annak ellenére, hogy típusuk nem egyezik.

Az `array_diff()` esetünkben az első tömbhöz hasonlítja a másodikat, és ez utóbbiban talált különbségeket fogja kiírni.



[Legfontosabb tudnivalók](#)

[Kapcsolat](#)

[Versenyszabályzat](#)

[Adatvédelem](#)

© 2023 Human Priority Kft.

KÉSZÍTETTE 

Megjelenés

 Világos 