

PHP PROGRAMOZÁS

6. forduló



A kategória támogatója: Human Priority Kft.

Ismertető a feladathoz

Felhasznált idő: 25:00/25:00

Elért pontszám: 0.5/15

1. feladat 0.5/1 pont

Melyek létező Superglobal tömbök?

Válaszok

☒ **`$_GLOBALS`**
Ez a válasz helytelen, de megjelölted.

☒ **`$_SERVER`**
Ez a válasz helyes, és meg is jelölted.

☒ **`$_POST`**
Ez a válasz helyes, és meg is jelölted.

☒ **`$_GET`**
Ez a válasz helyes, és meg is jelölted.

☒ **`$_FILE`**
Ez a válasz helytelen, de megjelölted.

☐ **`$_RESPONSE`**

☒ **`$_SESSION`**
Ez a válasz helyes, és meg is jelölted.

Magyarázat

A **`$_GLOBALS`** nem létezik, helyesen **`$GLOBALS`** lenne (nincs benne aláhúzás)

A **`$_FILE`** nem létezik, helyesen **`$_FILES`** lenne (többesszám)

`$_RESPONSE` egyáltalán nem létezik (**`$_REQUEST`** igen)

2. feladat 0/2 pont

Mely állítások igazak a következő kifejezésre?

5 << 2

Válaszok

- ☒ Bitwise Left Shift utasítás, mely balra tolja az első operandus bitjeit a második operandusban megadott számú helyiértékkel.
Ez a válasz helyes, de nem jelölted meg.
- ☒ Bitwise Left Shift utasítás, mely balra tolja mindkét operandus bitjeit, majd összeszorozza az így kapott két értéket.
Ez a válasz helytelen, de megjelölted.
- ☐ A művelet az első operandus kettővel való szorzását valósítja meg.
- ☒ A művelet az első operandus négygyel való szorzását valósítja meg.
Ez a válasz helyes, és meg is jelölted.

Magyarázat

Valóban egy ún. bitszintű operátorról van szó, amely az első paramétert (operandust) bit szinten balra tolja még hozzá annyi helyiértékkel, amennyit a második paraméterben adunk meg (esetünkben a második paraméter értéke 2, tehát kétszer tolja balra el az első paraméter bitjeit). Ez a művelet egyenértékű az első paraméter négygyel való szorzásával, ugyanis bit szinten, azaz kettes számrendszerben egy balra tolás az érték duplázódását jelenti. (Tehát két balra tolás az két duplázódást, azaz négyezereződést jelent.)

3. feladat 0/3 pont

Az alábbi kódrészlettel azt szeretnénk megállapítani egy képről, hogy álló (portrait) vagy fekvő (landscape) tájolású.

```
echo $width >= $height ? 'Landscape' : 'Portrait';
```

A felsorolt válaszok közül melyek biztosítják a megfelelő **\$width** és **\$height** paramétereket a kiértékeléshez?

Válaszok

☐

```
$details =  
getimagesize('images/test.png');  
$width = $details['width'];  
$height = $details['height'];
```



```
list($width, $height) =  
getimagesize('images/test.png');
```

Ez a válasz helyes, de nem jelölted meg.



```
$details = $_FILES['test_image'];  
$width = $details['width'];  
$height = $details['height'];
```



```
$details =  
getimagesize('images/test.png');  
$width = $details[0];  
$height = $details[1];
```

Ez a válasz helyes, de nem jelölted meg.

```
list($width, $height) =  
$_FILES['test_image']['size'];
```

Magyarázat

Csak a `getimagesize()` paranccsal kapjuk meg egy kép dimenzióit, a `$_FILES` tömb ezen információkat nem tartalmazza. A `getimagesize()` metódus egy tömböt ad vissza, melynek első paramétere a szélesség, második paramétere a magasság érték pixelben.

A `list()` parancs a tömb elemeinek sorrendjében deklarálja a paraméterül megadott változókat, s azok felveszik a megfelelő értékeket.

4. feladat 0/4 pont

Email címekből szeretnénk kinyerni a domain neveket, de csak azokban az esetekben, ahol a domain név tartalmazza az *'example'* stringet.

Például a john.doe@example.com esetén az example.com-ot szeretnénk visszakapni, azonban jane.doe@sample.com esetén nincs szükségünk a domain-re.

Melyik reguláris kifejezés valósítja meg ezt a feladatot?

Válasz

☒ `/@(example\lw+)$/i`
Ez a válasz helyes, de nem jelölted meg.

☐ `/@(example\w+)$/i`

☐ `/@(example\lw.)$/i`

☒ `/@(example\.[w]+)$/i`
Ez a válasz helytelen, de megjelölted.

Magyarázat

A `@` jel mindenképp kell a mintába. Ez jelzi, hogy inntől kezdődik a domain név. Ezután következik fixen az `'example'` szó, majd pedig a pont karakter, mely elé egy backslash karakter is kell, hiszen enélkül utasítást jelentene. A `'w'` string bármilyen alfanumerikus karakterre illeszkedik (szintén escape-elni kell), a `'+'` jel pedig azt határozza meg, hogy ezen karakterből bármennyi szerepelhet a vizsgálandó kifejezésben. A zárójelek csoportot alkotnak, így illeszkedés esetén visszakapjuk `@` nélkül az `'example'` szót tartalmazó domain neveket.

5. feladat 0/5 pont

Alább egy osztály privát metódusát láthatjuk, melynek feladata prímszámok meghatározása egy adott intervallumon belül. A teljes osztály ismerete nélkül azzal a feltételezéssel élünk, hogy a `$this->cycle()` metódus hívása előtt az osztályváltozók a következőképpen néznek ki:

```
$this->primes = [];
```

```
$this->stack = range(2, 100);
```

```
private function cycle(): void {
    $min = min($this->stack);
    $max = max($this->stack);
    if (!in_array($min, $this->primes)) {
        $this->primes[] = $min;

        /**
         * HIÁNYZÓ RÉSZ
         */

        if (!empty($this->stack)) {
            $this->cycle();
        }
    }
}
```

A `$this->cycle()` metódus meghívása után kiíratjuk a `$this->primes` változó tartalmát, melynek a `$this->stack` változóban deklarált intervallumban fellelhető prímszámok listáját kell tartalmaznia.

A felsoroltak közül mely kódrészlettel kell kiegészíteni a metódusban jelzett részt, hogy ezt a feladatot a leghatékonyabban megvalósíthassuk?

Válasz

☐ 1.opció:

```
for ($i = 1; $i <= $max; $i++) {
    if (($key = array_search($i * $min, $this->stack)) !== false) {
        unset($this->stack[$key]);
    }
}
```

☐ 2.opció:

```

for ($i = 1; $i * $min <= $max; $i++) {
    if (isset($this->stack[$i * $min])) {
        unset($this->stack[$i * $min]);
    }
}

```

☒ 3.opció:

```

for ($i = 1; $i * $min <= $max; $i++) {
    if (($key = array_search($i * $min, $this->stack)) !== false) {
        unset($this->stack[$key]);
    }
}

```

Ez a válasz helyes, de nem jelölted meg.

☐ 4.opció:

```

for ($i = 1; $i <= $max; $i++) {
    if (($key = array_search($min, $this->stack)) !== false) {
        unset($this->stack[$key]);
    }
}

```

Magyarázat

A helyes válasz a 3., ez a kódrészlet határozza meg leghatékonyabban a kívánt prímeket.

Végző soron az 1. válasz is ugyanerre az eredményre jut, azaz meghatározza a megfelelő prím-eket, ám a ciklusban nem csak a szükséges értékekkel számol, hanem mindig fixen 100 iterációt hajt végre, feleslegesen. Ez hibát nem okoz, de a felesleges körökben időt pazarol, így ha csak kis mértékben is, de kevésbé hatékony.

A 2. válasz azért hibás, mert a cikluson belül a **\$this->stack** tömbnek nem a kulcsával, hanem az értékével operál, a 4. válasz pedig helytelen kulcsot határoz meg a tömbben.



[Legfontosabb tudnivalók](#)  [Kapcsolat](#)  [Versenyszabályzat](#)  [Adatvédelem](#) 

© 2023 Human Priority Kft.

KÉSZÍTETTE 

Megjelenés

 Világos 