

PHP PROGRAMOZÁS

4. forduló



A kategória támogatója: Human Priority Kft.

Ismertető a feladathoz

A 4. forduló után elérhetőek lesznek a helyezések %-os formában: azaz kiderül, hogy a kategóriában a versenyzők TOP 20% - 40% - 60% -ához tartozol-e!

Szeretnénk rá felhívni figyelmedet, hogy a játék nem Forma-1-es verseny! Ha a gyorsaságod miatt kilököd a rendesen haladó versenyzőket, kizárást vonhat maga után!

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

Felhasznált idő: 10:13/25:00

Elért pontszám: 10/15

1. feladat 1/1 pont

Melyik függvény segítségével kapjuk meg egy string típusú változó hosszát?

Válasz

- ☐ `count()`
- ☐ `strcount()`
- ☐ `length()`
- ☒ `strlen()`
Ez a válasz helyes, és meg is jelölted.
- ☐ `strlen()`

Magyarázat

A PHP nyelvben string hosszának meghatározására csak a **`strlen()`** metódus létezik. A **`count()`** metódus is létezik, azonban hibát adna, ha paraméterként stringet adnánk át neki, hiszen az tömbelemek számlálására alkalmas.

2. feladat 2/2 pont

Mi az eredménye az alábbi számításnak?



```
abs(-5) - abs(floor(-2.14)) + pow(2,
4) - sqrt(9)
```

Válasz

☐ 7

☐ 5

☒ 15

Ez a válasz helyes, és meg is jelölted.

☐ 9

Magyarázat

A számításban szereplő funkciók:

Az **abs()** az abszolút értéket adja vissza, azaz negatív szám esetén annak pozitív megfelelőjét.

A **pow()** a PHP-ban a hatványozásra használatos metódus, első paramétere a hatványalap, második pedig a hatványkitevő. Ez esetben 24, azaz $2 * 2 * 2 * 2 = 16$

A **floor()** metódus egész értékre való kerekítést végez, mégpedig mindig lefelé (-2.14 értékből -3 lesz)

Végül a **sqrt()** a négyzetgyökvonás metódusa (9 négyzetgyöke 3)

A teljes kalkuláció tehát: $5 - 3 + 16 - 3 = 15$

3. feladat 3/3 pont

Mit fog megjeleníteni az alábbi kódrészlet?



```
var_dump(0123 == 123);
var_dump('0123' == 123);
var_dump('0123' === 123);
```

Válasz

- ☐ bool(true)
bool(true)
bool(false)

- ☒ bool(false)
bool(true)
bool(false)

Ez a válasz helyes, és meg is jelölted.

- ☐ bool(false)
bool(false)
bool(false)

Magyarázat

Az **első** kifejezésben `bool(false)` logikai értéket kapunk, mert a `0123` számsorban a bevezető `0` arra utasítja a PHP értelmezőt, hogy az értéket oktális (nem decimális) értékként kezelje, a `123` oktális pedig `83` decimális értékkel egyenlő, tehát a két összehasonlított érték nem egyenlő.

A **második** kifejezés `bool(true)` logikai értékkel tér vissza, a `'0123'` karakterlánc ugyanis automatikusan egész konvertálódik, melyet aztán egy azzal azonos értékű egész számmal hasonlítunk össze. (Az automatikus string-int konverzió végrehajtásakor a rendszer figyelmen kívül hagyja a kezdő `0`-t, és az értéket decimális (nem pedig oktális) értékként kezeli).

A **harmadik** kifejezés ismét `bool(false)` logikai értékkel tér vissza, mert ez esetben szigorú összehasonlítást végeztünk, ilyenkor nem történik automatikus típuskonverzió, s a kifejezésben a típusok nem fognak egyezni.

4. feladat 4/4 pont

Mi az értéke a **\$result** változónak a kód futását követően?

```
$a = 5 % -3;  
$b = -5 % 3;  
$c = 10 / 5;  
$d = 2 ** 3;  
$e = min(-3, 2, 0, -1) + 1;  
  
$result = $a - $b * $c + $d - $e;
```

Válasz

- ☒ 16

Ez a válasz helyes, és meg is jelölted.

- ☐ 17
- ☐ 18
- ☐ 7

Magyarázat

$\$a = 5 \% -3 = 2$

$\$b = -5 \% 3 = -2$

$\$c = 10 / 5 = 2$

$\$d = 2 ** 3 = 8$ (hatványozás, azaz $2 * 2 * 2$)

$\$e = \min(-3, 2, 0, 1) + 1 = -3 + 1 = -2$

$\$result = 2 - (-2) * 2 + 8 - (-2) = 2 - (-4) + 8 + 2 = 2 + 4 + 8 + 2 = 16$

5. feladat 0/5 pont

Az alábbi példában mi lesz a **print_r()** függvény kimenete?

```
$answers = [1];

$variable1 = intval('12.5John');
$variable2 = (float)'2.5';
$variable3 = 3.5;

$variable4 = ($variable1 * 2) + ceil($variable2) + (string)$variable3;

array_push($answers, $variable4);

$answers[] = 4;
$answers[] = 5.5;

array_pop($answers);

if (count($answers) == '3') {
    $answers[] = 6;
} elseif (count($answers) === 3) {
    array_unique($answers);
}

print_r($answers);
```

Válasz

- ☐ 1.válasz:



Array

```
(  
    [0] => 1  
    [1] => 6.5  
    [2] => 4  
    [3] => 6  
)
```

☐ 2.válasz:



Array

```
(  
  
    [0] => 1  
    [1] => 27.5  
    [2] => 4  
    [3] => 6  
)
```

☐ 3.válasz:



Array

```
(  
  [0] => 1  
  [1] => 30.5  
  [2] => 4  
)
```

☒ 4.válasz:



Array

```
(  
  [0] => 1  
  [1] => 30.5  
  [2] => 4  
  [3] => 6  
)
```

Ez a válasz helyes, de nem jelölted meg.

☐ 5.válasz:

Array

```
(  
    [0] => 1  
    [1] => 6.5  
    [2] => 4  
)
```

☐ 6.válasz:

Array

```
(  
    [0] => 1  
    [1] => 27.5  
    [2] => 4  
)
```

Magyarázat

Az egyetlen helyes válasz a 4.

A *\$variable1* típusa integer, értéke 12 lesz, hiszen a string elején lévő '12' egész értékre konvertálható.

A *\$variable2* típusa float, értéke 2,5

A *\$variable3* típusa float, értéke 3,5

A *\$variable4* típusa float, értéke 30,5 lesz a kalkuláció után

Ez utóbbi kifejezésben a *ceil()* metódus felfelé kerekíti a kapott értéket. A *(string)* utasítás ugyan stringgé alakítja a számértéket, az összeadási művelet miatt azonban a PHP automatikusan újra számértékké konvertálja azt.

Levezetve tehát a *\$variable4 = (12 * 2) + 3 + 3.5 = 30.5*

Az *array_push()* parancs a tömb végéhez illeszti a *\$variable4* változót, mint új tömbelemet.

A *array_pop()* ez esetben törli a tömb utolsó elemét.

A *count(\$answers) == '3'* feltétel kiértékelve IGAZ eredményt ad, mert a tömb ekkor valóban 3 elemet tartalmaz. A loose comparison összehasonlítás miatt nem számít, hogy stringet adtunk meg. A feltétel teljesülése miatt egy újabb értéket adunk a tömbhöz, azaz most már az utolsó érték a 6-os lesz.

Az **elseif** ágba már nem futunk be (annak ellenére, hogy igaz), mivel az **if** ág már teljesült.



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

© 2023 Human Priority Kft.

KÉSZÍTETTE 

Megjelenés

 Világos 