

CLOUD BI

3. forduló



A kategória támogatója: DXC Technology

Ismertető a feladatlaphoz

Kérjük, hogy a feladatlap indítása előtt mindenképp olvasd el az alábbi útmutatót:

Amennyiben olyan kategóriában játszol, ahol van csatolmány, de hibába ütközöl a letöltésnél, ott valószínűleg a vírusirtó korlátoz, annak ideiglenes kikapcsolása megoldhatja a problémát. (Körülbelül minden 3000. letöltésnél fordul ez elő.)



Helyezéseket a 4. forduló után mutatunk, százalékos formában: adott kategóriában a TOP 20-40-60%-hoz tartozol.

A feltűnően rövid idő alatt megoldott feladatlapok kizárást vonnak maguk után, bármilyen más gyanús esetben fenntartjuk a jogot a forduló érvénytelenítésére!

Ebben a fordulóban elsősorban Python-nal kapcsolatos feladatokkal találkozhatasz.

Jó versenyzést kívánunk!

1. feladat 1 pont

Hogyan hozhatunk létre egy üres listát Pythonban?

Válasz

- ☐ `empty_list = [None]`
- ☐ `empty_list = {}`
- ☐ `empty_list = ()`
- ☐ `empty_list = []`
- ☐ `empty_list = [null]`

2. feladat 1 pont

Az alábbiak közül melyek nem Python vezérlési szerkezetek?

Válaszok

- ☐ `if`
- ☐ `def`
- ☐ `for`
- ☐ `and`
- ☐ `print`

3. feladat 1 pont

Az alábbiak közül mely Python adattípusok immutábilisek (azaz nem módosíthatóak) alapértelmezés szerint?

Válaszok

- ☐ `list`
- ☐ `set`
- ☐ `dict`
- ☐ `str`

☐ tuple

4. feladat 1 pont

Hogyan lehet Pythonban elérni egy osztály egyik szülőosztályát (parent class) az öröklés (inheritance) során?

Válasz

- ☐ A parent() metódussal.
- ☐ A self.parent attribútummal.
- ☐ A super() függvénnyel.
- ☐ A base() kulcsszóval.

5. feladat 1 pont

Hogyan definiálható egy paraméterezhető dekorátor Pythonban, amely lehetővé teszi a dekorátornak átadott argumentumok használatát?

Válasz

- ☐ A Python nem támogatja a paraméterezhető dekorátorokat.
- ☐ Definiálunk egy metódust, amely elfogadja az argumentumokat, és visszaad egy dekorátor függvényt.
- ☐ Definiálunk egy metódust, amely a dekorátor logikáját tartalmazza, és a dekorátor hívásakor átadjuk neki az argumentumokat.
- ☐ Alkalmazzuk a @decorator(param) szintaxist, ahol a param az argumentum.

6. feladat 1 pont

Hogyan hozhatunk létre egy olyan Python osztályt, amely "egyszeri használatos" (single-use) metódusokkal rendelkezik, azaz a metódusok csak egyszer hívhatók meg, és minden további hívásuk hibát dob?

Válasz

- ☐ Az ilyen osztály létrehozása nem lehetséges Pythonban, mivel a metódusok bármikor újra meghívhatók.

- ☐ Definiáljuk az osztályon belül egy speciális dekorátort, amely nyomon követi a metódusok meghívásait, és hibát dob, ha egy metódust már egyszer meghívtak.
- ☐ Használjuk az `@staticmethod` dekorátort minden metóduson, hogy megakadályozzuk a metódusok állapotának módosítását a hívások között.
- ☐ Az osztály konstruktorában használjuk a `setattr` függvényt, amivel letiltjuk a metódusokat a hívás után.

Megoldások beküldése