

11 algoritm 02-magyarázattal

1. Mi a fájlkezelés első lépése?

- A) Fájl bezárása
- B) Fájl megnyitása
- C) Fájl feldolgozása
- D) Fájl törlése

☒ **Helyes válasz: B) Fájl megnyitása**

Miért ez a jó válasz?

Mielőtt bármilyen műveletet végeznénk egy fájlban (pl. olvasás, írás, feldolgozás), először meg kell nyitni azt a `open()` függvénnyel. Ez minden fájlkezelési művelet kiindulópontja.

✗ Miért rosszak a többi válaszok?

- **A) Fájl bezárása** – Ez csak a legvégén történik, amikor már nem kell több művelet a fájlban.
- **C) Fájl feldolgozása** – Feldolgozni csak már megnyitott fájlt lehet.
- **D) Fájl törlése** – A törlés nem része a szokásos fájlkezelésnek, inkább kivételes művelet.

2. A Pythonban az `input()` függvénnyel szöveget és számot is be tudunk kérni.

Igaz / Hamis

☒ **Helyes válasz: Igaz**

Miért ez a jó válasz?

Az `input()` függvény mindig szöveggként (string) olvas be adatot, de ezt a szöveget át tudjuk alakítani pl. számmá (`int()`, `float()`), ha szükséges. Így **számot is be lehet kérni**, csak konvertálni kell.

✗ Miért lenne rossz a „Hamis”?

Mert tévesen azt feltételezné, hogy nem kérhetünk be számot. Valójában a szöveges bemenet konvertálható számmá, ezért ez hamis állítás lenne.

3. Melyik függvénnyel lehet Pythonban beolvasni a felhasználótól adatot?

- A) `read()`
- B) `input()`
- C) `open()`
- D) `strip()`

☒ **Helyes válasz: B) input()**

Miért ez a jó válasz?

Az input() függvény felhasználói adatbekérésre szolgál – amikor a program futása közben adatot kérünk a felhasználótól, ezt ezzel a függvénnyel tesszük.

☒ **Miért rosszak a többi válaszok?**

- **A) read()** – Ez egy fájlból olvas be adatot, nem a felhasználótól.
- **C) open()** – Ez fájlokat nyit meg, nem olvas be adatot.
- **D) strip()** – Ez csak egy szöveg formázását (pl. szóközök levágása) végzi, nem adatkérésre való.

4. Melyek fájlműveletek Pythonban? *(Többszörös választás)*

- A) open()
- B) close()
- C) enumerate()
- D) read()

☒ **Helyes válasz: A), B), D)**

Miért ezek a jó válaszok?

- **A) open()** – Fájl megnyitása
- **B) close()** – Fájl bezárása
- **D) read()** – Fájl tartalmának beolvasása

☒ **Miért rossz a C) enumerate()?**

Az enumerate() nem fájlművelet, hanem iterálásra szolgál például listák vagy más bejárható objektumok bejárására, indexeléssel együtt.

5. Mit jelent az eldöntés algoritmus?

- A) Eldöntjük, hogy hány sor van egy fájlban
- B) Eldöntjük, hogy van-e adott tulajdonságú elem
- C) Eldöntjük, hogy mikor kell kilépni a ciklusból
- D) Eldöntjük, hogy mi legyen a program neve

☒ **Helyes válasz: B) Eldöntjük, hogy van-e adott tulajdonságú elem**

Miért ez a jó válasz?

Az eldöntés algoritmus lényege, hogy megvizsgáljuk: létezik-e olyan elem egy adathalmazban, amely kielégít egy bizonyos feltételt.

✗ Miért rosszak a többi válaszok?

- A) – Ez számlálás lenne, nem eldöntés.
 - C) – Ez ciklusvezérlés, nem eldöntés.
 - D) – A program elnevezése nem algoritmikus probléma.
-

6. A for ciklus mindig előltesztelő ciklus.

Igaz / Hamis

☒ Helyes válasz: Igaz

Miért ez a jó válasz?

A Python for ciklusa az iterálás előtt ellenőrzi a bejárható objektumot, tehát **előltesztelő**. Ez azt jelenti, hogy csak akkor hajtódik végre a ciklusmag, ha még van bejárható elem.

✗ Miért lenne rossz a „Hamis”?

Mert ez tagadná a Python for ciklusának működését – az valóban mindig előbb tesztel.

7. Mit jelent a kiválasztás algoritmus?

- A) Megszámoljuk a listában szereplő elemek számát
- B) Megkeressük az adott tulajdonságú elem helyét
- C) Összeadjuk a listában található számokat
- D) Végigjárunk minden elemet

☒ Helyes válasz: B) Megkeressük az adott tulajdonságú elem helyét

Miért ez a jó válasz?

A kiválasztás algoritmus célja megtalálni egy adott tulajdonságú elem **konkrét helyét vagy értékét** az adathalmazban.

✗ Miért rosszak a többi válaszok?

- A) – Ez a megszámlálás algoritmus.
 - C) – Ez az összegzés algoritmus.
 - D) – Ez a bejárás általános művelete, de nem specifikus a kiválasztásra.
-

8. Mely függvények adhatják vissza egy lista szélsőértékét? (Többszörös választás)

- A) sum()
- B) min()
- C) max()
- D) index()

☒ **Helyes válasz: B), C)**

Miért ezek a jó válaszok?

- **min()** – a legkisebb elemet adja vissza
- **max()** – a legnagyobb elemet adja vissza

☒ **Miért rosszak a többi válaszok?**

- **A) sum()** – Ez az elemek összegét adja vissza, nem szélsőértéket.
- **D) index()** – Ez egy adott elem első előfordulásának helyét adja meg.

9. Mit jelent az a kifejezés, hogy „bejárható objektum”?

- A) Olyan objektum, amely egyszerre több utasítást tárol
- B) Olyan objektum, amelyen ciklussal végig lehet menni
- C) Olyan objektum, amely csak számokat tartalmaz
- D) Olyan objektum, amely nem módosítható

☒ **Helyes válasz: B) Olyan objektum, amelyen ciklussal végig lehet menni**

Miért ez a jó válasz?

A bejárható (iterálható) objektumok, mint a lista, szöveg vagy tuple, olyan adattípusok, amelyeken a for ciklussal végig lehet haladni.

☒ **Miért rosszak a többi válaszok?**

- **A)** – Ez programlogikai egységekre jellemző, nem objektumokra.
- **C)** – A bejárhatóság nem függ az adatok típusától.
- **D)** – Ez az immutable (nem módosítható) fogalom, nem „bejárható”.

10. A megszámlálás algoritmus során mindig meg kell keresnünk a minimumot is.

Igaz / Hamis

☒ **Helyes válasz: Hamis**

Miért ez a jó válasz?

A megszámlálás algoritmus célja annak meghatározása, hogy hány elem elégít ki egy adott feltételt. A minimum (vagy maximum) érték keresése teljesen más algoritmus.

☒ **Miért lenne rossz az „Igaz”?**

Mert az állítás téves: megszámlálás során nem keresünk minimumot, csak számolunk.

11. Mit ad vissza a count() függvény?

- A) Egy adott szám előfordulásainak számát a listában
- B) Egy lista teljes hosszát
- C) Egy lista első elemét
- D) Az összes kulcsot egy szótárban

☒ **Helyes válasz: A) Egy adott szám előfordulásainak számát a listában**

Miért ez a jó válasz?

A count() függvény egy lista (vagy más szekvencia) adott értékének előfordulásait számolja meg. Például: [1, 2, 1, 3].count(1) eredménye 2.

☒ **Miért rosszak a többi válaszok?**

- **B)** – Ez a len() függvény feladata, nem a count()-é.
- **C)** – A lista első elemét az indexeléssel érjük el, pl. lista[0].
- **D)** – Az összes kulcs lekérésére a szótárban a dict.keys() szolgál.

12. Melyek jellemzők a dictionary-re? (Többszörös választás)

- A) Kulcs-érték párokból áll
- B) Az értékei lehetnek listák is
- C) Csak szöveget tárolhat
- D) Egy objektumban több „név” kulcs is lehet

☒ **Helyes válasz: A), B)**

Miért ezek a jó válaszok?

- A dictionary alapvetően **kulcs-érték párokból** áll (pl. {"név": "Anna"}).
- Az értékek lehetnek **listák**, más szótárak, számok stb.

☒ **Miért rosszak a többi válaszok?**

- **C)** – Egy dictionary bármilyen típusú értéket tárolhat, nem csak szöveget.
- **D)** – Nem lehet több azonos kulcs (pl. „név”), mert a kulcsoknak **egyedinek** kell lenniük.

13. A Pythonban az append() függvénnyel új elemet adhatunk egy listához.

Igaz / Hamis

☒ **Helyes válasz: Igaz**

Miért ez a jó válasz?

A `list.append(elem)` metódus új elemet fűz a lista végére. Ez a leggyakoribb módja egy lista bővítésének.

✗ Miért lenne rossz a „Hamis”?

Mert az állítás igaz: az `append()` valóban új elemet ad a listához. Ha valaki ezt hamisnak véli, félreérti a függvény célját.

14. Mit csinál az `index()` függvény?

- A) Eldönti, hogy van-e az elem a listában
- B) A keresett elem első előfordulásának helyét adja meg
- C) Az utolsó elem indexét adja vissza
- D) Egy listát rendez

☒ Helyes válasz: B) A keresett elem első előfordulásának helyét adja meg

Miért ez a jó válasz?

A `list.index(elem)` megkeresi az adott elem **első előfordulásának indexét**. Például: `[4, 5, 6, 4].index(4)` eredménye 0.

✗ Miért rosszak a többi válaszok?

- **A)** – Azt a `in` operátor dönti el: pl. `elem in lista`.
 - **C)** – Az utolsó elem indexét pl. `len(lista) - 1` adja meg.
 - **D)** – A rendezéshez a `sort()` vagy `sorted()` kell, nem az `index()`.
-

15. Mi a fájl megnyitásának alapértelmezett módja az `open()` függvénynél?

- A) Írás
- B) Olvasás
- C) Törlés
- D) Új fájl létrehozása

☒ Helyes válasz: B) Olvasás

Miért ez a jó válasz?

Az `open("fajl.txt")` alapértelmezés szerint `'r'` módban nyitja meg a fájlt, ami **olvasást** jelent. Ez a legbiztonságosabb és leggyakoribb mód.

✗ Miért rosszak a többi válaszok?

- **A)** – Az írás módhoz `open("fajl.txt", "w")` kell.
- **C)** – A fájl törlését nem az `open()` végzi, hanem pl. az `os.remove()`.

- **D)** – Az új fájl létrehozása x móddal történik ("x").
-

16. Melyek kulcsfontosságú ciklusfajták Pythonban? *(Többszörös választás)*

- A) while
- B) for
- C) until
- D) do-while

☒ **Helyes válasz: A), B)**

Miért ezek a jó válaszok?

A Pythonban két fő ciklus van:

- **for ciklus**, amely bejárható objektumokon iterál,
- **while ciklus**, amely addig fut, amíg a feltétel igaz.

☒ **Miért rosszak a többi válaszok?**

- **C) until** – Ez más nyelvekben (pl. Pascal) létezik, Pythonban nincs ilyen ciklus.
 - **D) do-while** – Pythonban nincs ilyen beépített ciklustípus. Helyette while True és break használható.
-

17. A Pythonban a függvényeket a program végén kell definiálni.

Igaz / Hamis

☒ **Helyes válasz: Hamis**

Miért ez a jó válasz?

A Pythonban a függvényeket **azelőtt kell definiálni**, hogy meghívnánk őket. Nem kell a program végén definiálni, bárhol lehet – de a futtatás sorrendjében értelmezve **a hívás előtt** kell lennie.

☒ **Miért lenne rossz az „Igaz”?**

Mert ez megtévesztő: a Python interpreter soronként értelmez, és ha egy még nem definiált függvényt próbálunk meghívni, hibát kapunk. Nem az elhelyezkedés számít, hanem a sorrend.

18. Melyik NEM sztring művelet?

- A) upper()
- B) strip()
- C) max()
- D) lower()

☒ **Helyes válasz: C) max()**

Miért ez a jó válasz?

A max() nem kizárólag sztringekre készült – minden iterálható típusnál működik. Sztring esetén is használható, de nem kifejezetten sztring művelet.

☒ **Miért rosszak a többi válaszok?**

- **A) upper()** – A szöveget nagybetűssé alakítja.
- **B) strip()** – Levágja a szóközöket a szöveg elejéről és végéről.
- **D) lower()** – A szöveget kisbetűssé alakítja. Mindhárom sztring specifikus művelet.

19. Mely utasításokkal lehet ciklusból kilépni vagy cikluson belül ugrani? (Többszörös választás)

- A) break
- B) continue
- C) return
- D) next

☒ **Helyes válasz: A), B), C)**

Miért ezek a jó válaszok?

- **break** – az egész ciklust megszakítja.
- **continue** – átugorja az aktuális cikluslépést.
- **return** – kilép a függvényből, akkor is, ha az egy cikluson belül van.

☒ **Miért rossz a D) next?**

A next() egy függvény, nem utasítás. Iterátorokkal működik, de **nem ciklusvezérlő parancs**, nem lehet vele kilépni vagy ugrani.

20. A Python a blokkhatárokat behúzásokkal jelöli.

Igaz / Hamis

☒ **Helyes válasz: Igaz**

Miért ez a jó válasz?

A Pythonban a kód blokkjait **nem kapcsos zárójelek** vagy kulcsszavak, hanem **behúzások (indentálás)** jelzik. Ez kötelező és szintaktikailag ellenőrzött.

☒ **Miért lenne rossz a „Hamis”?**

Mert ezt feltételeznék, hogy a blokkhatárokat más módon (pl. zárójelekkel) jelöljük – ez Pythonban **nem igaz**.