Python alapok.docx

5X20 darab (összesen 100 db) igaz–hamis tesztkérdés

---

1. Tesztsor (Változók, adattípusok):

1. A Pythonban a változónév kezdődhet számmal. –
2. A változónév lehet „_szam” –
3. A Python megköveteli, hogy a változó típusát előre meg kell adni. –
4. A „Szam” és a „szam” ugyanazt a változót jelentik. –
5. A „list” használata változónévként felülírja a beépített függvényt. –
6. A Python nem különbözteti meg a kis- és nagybetűket a változóneveknél. –
7. A „True” egy foglalt kulcsszó, nem lehet változónév. –
8. A változónevekben csak betűket lehet használni. –
9. Az aláhúzás „_” használható változónév részeként. –
10. A Python kulcsszavak, mint például „if” és „class”, érvényes változónevek. –
11. A Pythonban dinamikus típusosság van érvényben. –
12. A változónév nem lehet „await” vagy „lambda”. –
13. A „pelda_ertek” érvényes változónév. –
14. A Pythonban nincs különbség „x” és „X” változó között. –
15. Az értelmetlen, rövid változónevek használata mindig hibát okoz. –
16. A Pythonban a pont (.) a tizedes jel, nem a vessző (,). –
17. A str adattípus rózsaszín színkóddal van jelölve a dokumentumban. –
18. A változó lehet egyszer egész szám, majd később szöveg. –
19. A Pythonban nem lehet „import” nevű változót létrehozni. –
20. A változók elnevezésénél a beszédes nevek nem ajánlottak. –

---

2. Tesztsor (Adattípusok, operátorok):

1. A „bool” adattípus csak „True” vagy „False” értékeket vehet fel. –
2. A „list” egy nem módosítható adattípus. –
3. A „tuple” egy módosítható típus. –
4. A „dict” kulcs-érték párokból áll. –
5. A „set” adattípusban lehetnek ismétlődő elemek. –
6. A „complex” típus valós és képzetes részből áll. –
7. A „range” típust nem lehet listává alakítani. –
8. A Pythonban a type() függvény megmondja az érték típusát. –
9. A lebegőpontos számok típusa „int”. –
10. Az "alma"[0] karaktere 'a'. –
11. A „print” függvény alapból sortörést ír a sor végére. –
12. A sep paraméter a print függvényben a szóközt szabályozza. –
13. A „float” egész számot jelöl. –
14. A sztringek kezelhetők úgy, mint karakterlisták. –
15. A változók nem adhatók át a print függvénynek. –
16. A „**” művelet hatványozást jelent Pythonban. –
17. Az osztás (/) mindig egész számot ad eredményül. –
18. A „%” operátor az osztási hányadost adja meg. –
19. A „//” egészrészt ad vissza, levágva a tizedeseket. –
20. Az f-string csak Python 3.10-től elérhető. –

---

3. Tesztsor (Feltételek és logikai műveletek):

1. Az „if” szerkezet mindig kettősponttal végződik. –
2. Az „if – elif – else” szerkezet kizáró feltételek vizsgálatára jó. –
3. Az „if – if – if” szerkezet mindig csak az első igaz feltételt hajtja végre. –
4. A „not” logikai operátor megfordítja a feltétel értékét. –
5. A Pythonban létezik „switch” vezérlési szerkezet. –
6. A feltétel után nem kötelező behúzni a kódblokkot. –
7. A logikai kifejezések csak „True” és „False” értékeket adhatnak vissza. –
8. Az üres string logikai értéke True. –
9. A „and” operátor igazat ad vissza, ha mindkét oldal igaz. –
10. A „or” operátor csak akkor ad vissza True-t, ha mindkettő igaz. –
11. A Pythonban a feltételek sorrendje fontos lehet. –
12. Az „if 5 == 5” kifejezés eredménye False. –
13. Az „if 3 != 4” kifejezés igazat ad vissza. –
14. A „>=” relációs operátor vizsgálja, hogy nagyobb-e vagy egyenlő. –
15. A Python nem támogatja a logikai operátorokat. –
16. A „print(not(x == 7))” eredménye akkor igaz, ha x nem 7. –
17. Az „or” logika szerint legalább egy igaz elég. –
18. A „not(x > 5)” eredménye akkor lesz True, ha x <= 5. –
19. A Python nem enged meg több „elif”-et egy „if”-ben. –
20. A „True and False” eredménye True. –

---

4. Tesztsor (Véletlenszámok, input, ciklusok):

1. A random.randint(1, 5) visszaadhatja az 5-öt is. –
2. A random.random() csak egész számokat generál. –
3. A choice() véletlenszerűen kiválaszt egy listából egy elemet. –
4. A shuffle() véglegesen megváltoztatja a lista sorrendjét. –
5. Az input() függvény mindig string típust ad vissza. –
6. Az int(input()) segítségével egész szám olvasható be. –
7. A try – except blokkal elkerülhetők a típuskonverziós hibák. –
8. A „input()” visszatérési értéke mindig szám. –
9. A „print()” függvénynek megadható „sep” és „end” paraméter is. –
10. A while ciklus csak egyszer hajtódik végre. –
11. A while ciklus addig fut, amíg a feltétel igaz. –
12. A „break” kilépteti a ciklust. –
13. A „continue” megszakítja a teljes ciklust. –
14. A végtelen ciklust „Ctrl+C”-vel lehet megszakítani. –
15. A „while True” szerkezet mindig egyszer lefut. –
16. A ciklusok nem lehetnek egymásba ágyazva. –
17. A „for” ciklus fix elemszámú ismétléshez ideális. –
18. A ciklusváltozók elnevezésénél mindegy, hogy ismétlődnek-e. –
19. A beágyazott ciklus gyakori mátrixkezelésnél. –
20. A „range(5)” öt értéket generál: 0-tól 4-ig. –

---

5. Tesztsor (Listák, sztringek):

1. A lista egy rendezett, módosítható adattípus. –
2. A string (str) módosítható karakterlánc. –
3. A lista indexelhető és szeletelhető. –
4. A lista bármilyen adattípust tartalmazhat. –
5. A sztring elemét közvetlenül nem lehet átírni. –
6. A „append()” függvény új elemet ad a lista végéhez. –
7. A „sort()” függvény csökkenő sorrendbe állít. –
8. A „len()” függvény nem működik listákra. –
9. A „clear()” eltávolít minden elemet a listából. –
10. A „copy()” új példányt hoz létre a listából. –
11. A „in” operátor megmondja, hogy szerepel-e egy érték a listában. –
12. A „list comprehension” nem támogatott Pythonban. –
13. A sztringből listát a „list()” függvénnyel készíthetünk. –
14. A „join()” a karakterek listájából készít sztringet. –
15. A string „+” operátorral nem bővíthető. –
16. A listák és sztringek is ciklussal bejárhatók. –
17. A sztring csak karaktereket tartalmazhat. –
18. A lista csak sztringeket tartalmazhat. –
19. A lista „del” művelettel is módosítható. –
20. A sztring és lista közti konverzió nem lehetséges. –
