PYTHON PROGRAMOZÁS ALAPJAI

A while Ciklus Működése és Alkalmazása

Feltételvezérelt ismétlés, léptető számlálók és felhasználói adatbekérés kezelése

Gyakorlati kódpéldák 10. osztályosoknak

ALAPFOGALMAK

A Feltételes Ismétlés Logikája

Elöltesztelő ciklus

A feltétel ellenőrzése minden egyes lépés előtt lefut.

Igaz értéknél ismétel

A ciklusmag mindaddig újra lefut, amíg a feltétel True.

Hamis értéknél kilép

Amint False a logikai érték, a program továbblép.

```
# Általános szintaxis
while :
    # Ciklusmag utasításai
    utasitas_1()
    utasitas_2()

print("Ciklus befejezve")
```

SZÁMLÁLÓ MINTÁZAT

A Léptető Változó Szerepe

```
# Léptetés 1-től 5-ig
szamlalo = 1

while szamlalo <= 5:
    print(f"Aktuális érték: {szamlalo}")
    szamlalo += 1  # Léptetés kötelező!

print("Kész a számlálás!")
```

1. Kezdőérték beállítás

A számláló létrehozása a ciklus előtt.

2. Megállási feltétel

Határérték vizsgálata minden körben.

3. Érték növelése

A lépésköz hozzáadása a ciklusmagon belül.

GYAKORLATI PÉLDA

Adatellenőrzés: Helyes Jelszó Bekérése

```
# Jelszó ellenőrzése ciklussal
titok = "python123"
tipp = input("Add meg a jelszót: ")

while tipp != titok:
    print("Hibás jelszó! Próbáld újra.")
    tipp = input("Add meg a jelszót: ")

print("Sikeres belépés!")
```

Ismeretlen lépésszám

Előre nem tudjuk, a felhasználó hány próbálkozásból találja el.

Folyamatos felülírás

A tipp változó minden rontás után új értéket kap.

GYAKORI HIBÁK

A Végtelen Ciklus Veszélye

Elfelejtett léptetés

A feltételben szereplő változó értéke nem változik, örökké True marad.

Túlterhelés

A program lefagy és 100%-os processzorterhelést idéz elő.

Kényszerített kilépés

A terminálban a Ctrl + C billentyűparanccsal állítható le azonnal.

CIKLUSVEZÉRLÉS

A break és a continue Utasítások

break — Megszakítás

Azonnal kilép a ciklusból, a feltételtől függetlenül.

```
while True:
    valasz = input("Kilépsz? ")
    if valasz == "igen":
        break
```

continue — Átugrás

Átugorja a ciklusmag hátralévő részét a következő körre.

```
szam = 0
while szam < 6:
    szam += 1
    if szam % 2 == 0:
        continue
    print(szam)
```

HALADÓ TECHNIKA

Állapotvezérlés Logikai Zászlóval

A logikai flag (zászló) egy olyan változó, amely a program futási állapotát jelzi.

Tisztább kódstruktúra

Nem kell végtelen ciklusokat és break hívásokat halmozni.

Több feltétel összevonása

Bármelyik esemény False-ra billentheti az állapotot.

```
aktiv = True

while aktiv:
    adat = input("Adj meg egy szót ('stop' = kilépés): ")
    if adat == "stop":
        aktiv = False
    else:
        print(f"Beírt adat: {adat}")

print("A feldolgozás véget ért.")
```

DÖNTÉSI SEGÉDLET

Mikor Melyik Ciklust Használjuk?

for Ciklus

Ismert lépésszám esetén

- Listák elemeinek sorfolytonos bejárása
- Fix lépésszám (range számsorozat)
- Karakterek olvasása szövegből

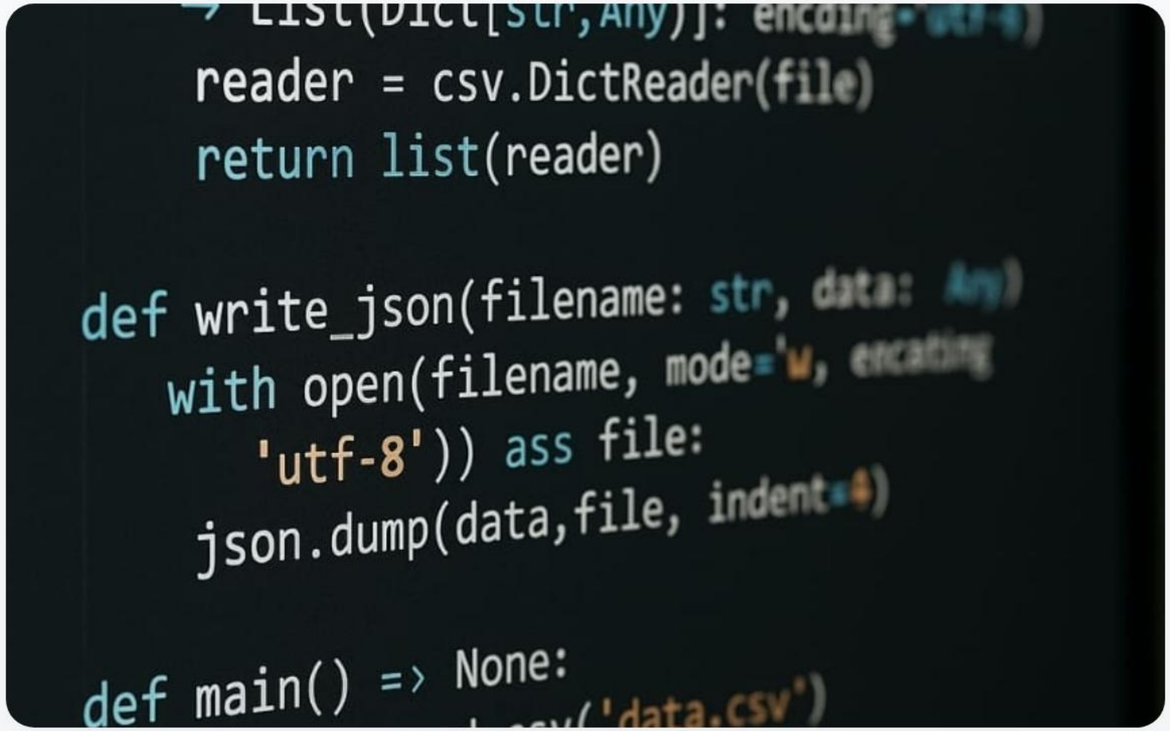
while Ciklus

Ismeretlen lépésszám esetén

- Felhasználói bemenet ellenőrzése
- Külső feltétel vagy esemény bekövetkezése
- Játékciklusok és menürendszerek futtatása

JÓ GYAKORLATOK

Biztonságos Kódolási Aranyszabályok



Garantált termináció

Mindig győződj meg róla, hogy a feltétel el tudja érni a False állapotot.

Beszédes változónevek

Az i helyett használd a szamlalo, probalkozasok kifejezéseket.

Védelem rossz bevitelnél

A ciklus törzsében tájékoztasd a felhasználót a hibáról.



Gyakorold a while Ciklust!

A while ciklus az interaktív alkalmazások, számkitaláló játékok és bemeneti ellenőrzések elengedhetetlen eszköze.

#Kérdések & Válaszok

#Gyakorló Feladatok

Image Sources



https://miro.medium.com/1*qH5euOTdLyg4NBY4ZqrKqQ.jpeg

Source: python.plainenglish.io