PYTHON PROGRAMOZÁS II.

For Ciklusok, Listák és Beépített Függvények

Iterációk, szekvenciák kezelése, a len() méretlekérdezés és a range() számsorozat-generátor

15 diás gyakorlati útmutató

A Tananyag Három Pillére

MODULOK

1. A Lista Típus

Összetett, dinamikus tároló.

0-alapú indexelés és szeletelés.

2. A For Ciklus

Szekvenciák közvetlen bejárása.

Tiszta és hatékony szintaxis.

3. Segédeszközök

Méretlekérdezés a len() függvénnyel.

Tartományok a range() hívással.

A Lista: Összetett Adattípus

ALAPOK

Szögletes Zárójeles Tároló

Több önálló értéket foglal egységbe.

Heterogén elemek tárolására is képes.

Módosítható (mutable) memóriastruktúra.

```
# Lista létrehozása és inicializálása
szamok = [10, 20, 30, 40]
nevek = ["Anna", "Béla", "Cili"]
vegyes = [42, "Python", True, 3.14]

print(szamok) # [10, 20, 30, 40]
```

Elemelérés: Indexelés Szabályai

POZITÍV INDEXEK

Kezdés 0-tól Balról

Az első elem sorszáma mindig 0.

```
lista[0] # Legelső elem
lista[2] # Harmadik elem
```

NEGATÍV INDEXEK

Visszafelé a Végéről

A -1 index a legutolsó adatot adja.

```
lista[-1] # Legutolsó elem
lista[-2] # Utolsó előtti elem
```

A len() Méretlekérdező Függvény

Hány Elem Van Benne?

A 'length' (hossz) rövidítése.

Egész számmal tér vissza futásidőben.

Sztringek és listák méretét is méri.

```
# len() használata különféle adatokon
nevek = ["Dániel", "Éva", "Ferenc"]
print(len(nevek)) # 3

szoveg = "Python"
print(len(szoveg)) # 6 karakter
```

Túlindexelés: IndexError Veszély

IndexError: list index out of range

Egy 3 elemű lista legnagyobb legális indexe 2.

```
lista = [1, 2, 3]
print(lista[3]) # Hiba! Nem létezik
```

A Biztonságos Megoldás

A len(lista) - 1 a megengedett felső határ.

```
utolso_idx = len(lista) - 1
print(lista[utolso_idx]) # Biztonságos
```

A for Ciklus Logikája

```
allatok = ["kutya", "macska", "ló"]

for allat in allatok:
    print(allat)
```

Automatikus Léptetés
Egyesével veszi a tároló minden elemét.

Nincs Manuális Indexelés
Nem kell léptető változót növelni.

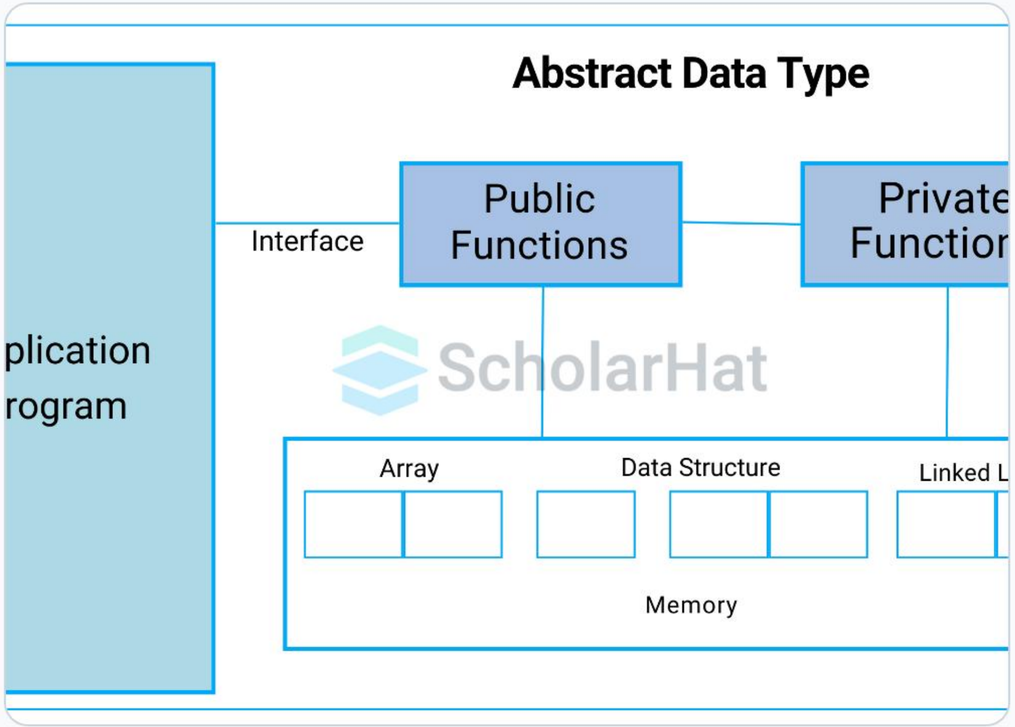
Szekvenciák Bejárása: Sztringek

KARAKTERENKÉNTI FELDOLGOZÁS

A szöveg is iterálható szekvencia

A for ciklus betűről betűre haladva dolgozza fel a szöveget.

```
for betu in "KÓD":
    print(betu) # K, Ó, D külön sorban
```



A range(stop) Alapforma

1 PARAMÉTER

0-tól n-1 Értékig

A felső határ kizáró (exclusive).

Pontosan n darab egész számot generál.

```
# range(5) → 0, 1, 2, 3, 4
for i in range(5):
    print(i) # Kiír: 0, 1, 2, 3, 4

# Az 5 már nem szerepel a futásban!
```

A range(start, stop) Forma

2 PARAMÉTER

Egyedi Kezdőérték

A generálás a megadott start értéktől indul.

```
range(3, 7)
# Tartalom: 3, 4, 5, 6
```

FONTOS KIVÉTEL

Stop Érték Elérése

Ha n-ig szeretnénk menni, n + 1-et kell írni.

```
range(1, 10 + 1)
# Tartalom: 1-től pontosan 10-ig
```

Lépésköz: range(start, stop, step)

```
# Pozitív és negatív lépésköz
# Páros számok 2-től 10-ig:
for x in range(2, 11, 2):
    print(x) # 2, 4, 6, 8, 10

# Visszaszámlálás:
for x in range(5, 0, -1):
    print(x) # 5, 4, 3, 2, 1
```

Ugrások a Sorozatban

Pozitív 'step' esetén növekvő lépésköz.

Visszafelé Iterálás

Negatív 'step' esetén start > stop kötelező.

Gyakori Minta: range(len(lista))

```
nevek = ["Kata", "Gábor", "Zsuzsa"]

for i in range(len(nevek)):
    print(f"{i}. elem: {nevek[i]}")

# Kifejezetten index alapú módosításhoz
```

Mikor Használjuk?

Ha a pozíció sorszáma is fontos.

Lista Elemeinek Írása

Értékadáshoz kötelező az index elérése.

Gyakori Listaműveletek

`.append(elem)`

Hozzáadás Végére

Egy új elemmel bővíti a listát.

`in operátor`

Tartalmazás Vizsgálat

Logikai True/False eredményt ad.

`sum(), min(), max()`

Gyors Összegzés

Közvetlen számítás lista elemein.

Tipikus Kezdő Bakik

range(1, 10) nem 10-ig tart

A felső határ mindig kimarad a ciklusból.

Módosítás Bejárás Közben

A for elem in lista nem írja vissza a változást.

range(len) Túlbonyolítás

Ha nem kell az index, közvetlen bejárás javasolt.

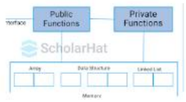


Összefoglalás és Következő Lépések

A for ciklus és a listák kombinációja adja a Python adatkezelésének leghatékonyabb eszköztárát.

Következő témakör: Függvények és Szótárak (Dict)

Image Sources



https://dotnettrickscloud.blob.core.windows.net/article/data%20structures/3720240405182636.com-png-to-webp-converter

Source: www.scholarhat.com