Python for AI

Дәуренов Ә.Д

№4 ДӘРІС Python тіліндегі тізімдер және деректер жиындарын өңдеу

Дәрістің мақсаты

Білім алушыларды Python тіліндегі list деректер құрылымымен таныстыру, тізімдерді құру, элементтерге қол жеткізу және өзгерту, тізімдерді цикл арқылы өңдеу, негізгі функцияларды қолдану және өлшеу деректерінің жиындарын талдау тәсілдерін үйрету.

Дәріс соңында білім алушы:

- list ұғымын түсіндіре алады;
- тізім құра алады;
- тізім элементтеріне индекс арқылы қол жеткізе алады;
- тізім элементтерін өзгерте алады;
- append(), remove(), len() әдістері мен функцияларын қолдана алады;

- min(), max(), sum() арқылы деректерді талдай алады;
- орташа мәнді есептей алады;
- for циклі арқылы тізім элементтерін өңдей алады;
- if / elif / else көмегімен тізімдегі мәндерді классификациялай алады;
- берілген шарт бойынша жаңа тізім құра алады.

1. Неліктен тізімдер LIST қажет?

Бірақ нақты көлік жүйесінде бір параметр **бір рет емес, үнемі өлшенеді**. Мысалы, қозғалтқыш температурасы:

88, 92, 95, 101, 106, 99, 87, 103

Оларды жеке айнымалыларға сақтауға болады: t1 = 88, t2 = 92, t3 = 95, t4 = 101

Бірақ жүздеген немесе мыңдаған өлшеу болған жағдайда бұл тәсіл тиімсіз.

Python тілінде осындай деректерді бір жерде сақтау үшін **тізім (list)** қолданылады.

Тізім (list) — бірнеше элементті бір айнымалыда сақтауға мүмкіндік беретін Python деректер құрылымы.

temperature = [88, 92, 95, 101, 106, 99, 87, 103]

Негізгі идея:

Бір параметрдің көптеген мәндері → бір тізім

3. Тізім әртүрлі деректер типтерін сақтай алады

Тізімде бүтін сандар болуы мүмкін: speed = [40, 60, 80, 100]

Нақты сандар: pressure = [2.1, 2.3, 2.4, 2.2]

Мәтін: status = ["NORMAL", "WARNING", "FAULT"]

Логикалық мәндер: sensor = [True, True, False, True]

Python техникалық тұрғыдан әртүрлі типтерді бір тізімде де сақтай алады:
data = [90, 2.5, "NORMAL", True]

Алайда өлшеу деректерін талдау кезінде тізімде әдетте **бірдей типтегі мәндерді сақтау тиімді**.

Тізімдегі әр элементтің өз орны бар. Ол **индекс** деп аталады.

Python тілінде индекстеу **0-ден басталады**.

speed = [40, 60, 80, 100]

print(speed[0]) — 0 индекс мәні 40 қа тең

print(speed[2]) — 2 индекс мәні 80 ге тең

Бірінші элементтің индексі — 0, екінші элементтің индексі — 1.

print(speed[-1]) деп соңғы элементі алуға болады

Тізім элементін өзгерту

Тізім құрылғаннан кейін оның элементтерін өзгертуге болады.

```
battery = [90, 85, 80, 75]

battery[1] = 88

print(battery)
```

[90, 88, 80, 75]

len() — элементтер санын анықтау

len() тізімдегі элементтердің санын анықтайды.

```
speed = [40, 55, 70, 85, 100]

print(len(speed))
```

Нәтиже: 5

append() — жаңа элемент қосу

append() тізімнің соңына жаңа элемент қосады.

```
speed = [40, 60, 80]

speed.append(100)

print(speed)
```

[40, 60, 80, 100]

`remove()` — элементті жою

Белгілі бір мәнді тізімнен жою үшін:

```
speed = [40, 60, 80, 100]
speed.remove(80)
print(speed)
[40, 60, 100]
```

remove() **мән бойынша** жояды.

Егер бірдей мән бірнеше рет кездессе, бірінші кездескен мән жойылады.

`min()` және `max()`

```
temperature = [88, 92, 95, 101, 106, 99]

print(min(temperature))

print(max(temperature))

print("MIN:", min(temperature))

print("MAX:", max(temperature))
```

`sum()` — мәндердің қосындысы

```
speed = [40, 60, 80, 100]
total = sum(speed)
print(total)
```

Орташа мәнді есептеу:

```
speed = [40, 60, 80, 100]

average = sum(speed) / len(speed)

print("Орташа жылдамдық:", average)
```

$$Average = \frac{Sum}{N}$$

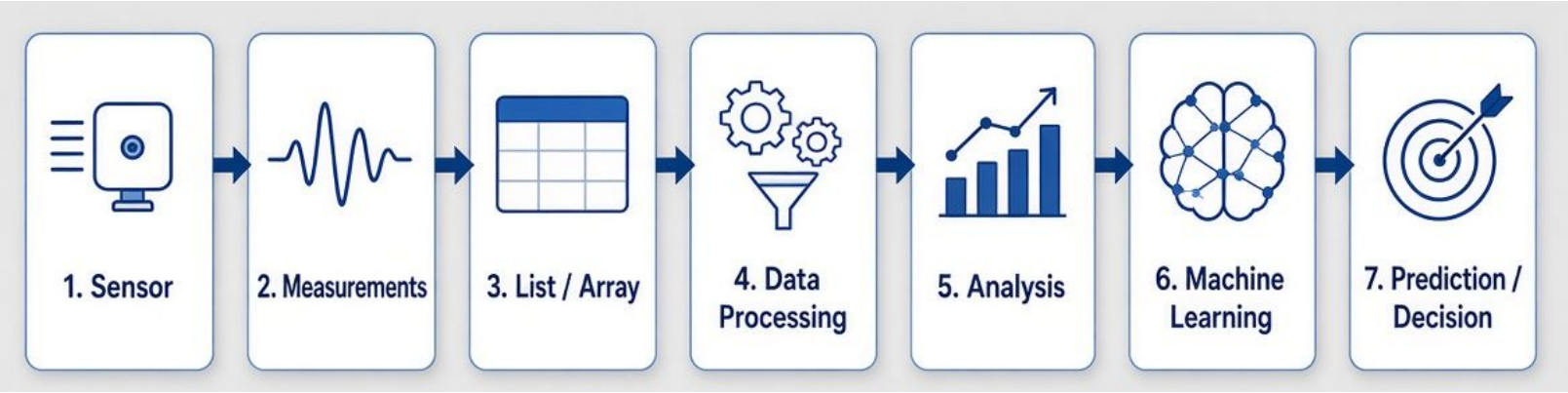
Тізімнің негізгі әдістері

Біз append() және remove() әдістерін көрдік. Тағы бірнеше пайдалы әдіс бар:

Әдіс	Қызметі
`append(x)`	тізім соңына элемент қосады
`insert(i, x)`	көрсетілген орынға элемент қосады
`remove(x)`	көрсетілген мәнді жояды
`pop()`	элементті жояды және қайтарады
`sort()`	тізімді сұрыптайды
`reverse()`	элементтердің ретін кері өзгертеді
`count(x)`	мәннің неше рет кездесетінін анықтайды
`index(x)`	мәннің индексін анықтайды

Неліктен бұл AI және Data Science үшін маңызды?

temperature = 95 – бұл бір өлшеу.
Ал: temperature = [88, 92, 95, 101, 106, 99, 87, 110]
бұл **деректер жиынының қарапайым түрі**. Нақты жүйеде мыңдаған мән болуы мүмкін:



Сондықтан Python тізімдері кейінгі тақырыптарға негіз болады:
list → CSV → NumPy → Pandas → Data Analysis → Machine Learning