Python for AI

Дәуренов ӘД

Дәріс 5. Деректер мен бағдарламаларды құрылымдауға арналған Python функциялары және сөздіктері

1. Дәрістің мақсаты

Білім алушыларға Python тіліндегі

- функциялар мен
- сөздіктердің негізгі ұғымдарын,
- олардың жұмыс принциптерін
- және бағдарламаларды құрылымдаудағы маңызын түсіндіру.

2. Бағдарламаны құрылымдау дегеніміз не?

Бағдарламаны құрылымдау (Program Structuring) — күрделі бағдарламаны белгілі бір міндеттерді орындайтын жеке логикалық бөліктерге ұйымдастыру процесі.
Қарапайым бағдарламалар бірнеше командадан тұруы мүмкін. Алайда нақты инженерлік жүйелерде көптеген операциялар орындалады.
Мысалы, электрмобильді мониторингтеу бағдарламасы келесі міндеттерді орындай алады:

- Батарея температурасын қабылдау.
- Кернеу мен ток мәндерін өңдеу.
- Параметрлерді рұқсат етілген диапазондармен салыстыру.
- Жүйенің күйін анықтау.
- Нәтижелерді сақтау және көрсету.

Егер барлық операция бір ұзын код ішінде орындалса, бағдарламаны түсіну, өзгерту және тексеру қиындайды.

Модульдік бағдарламалау

Modular Programming

Күрделі бағдарламалар жеке бөліктерге бөлінеді.



Құрылымдалған бағдарламаның жалпы моделі

Әр модуль белгілі бір міндетке жауап береді.

3. Функция дегеніміз не?

Функция (Function) — белгілі бір тапсырманы орындауға арналған, атауы бар және қажет болған жағдайда бірнеше рет шақырылатын бағдарламалық код блогы.
Функциялар бағдарламаны логикалық бөліктерге бөлуге мүмкіндік береді.
Мысалы, автомобиль мониторингі бағдарламасында келесі функциялар болуы мүмкін:

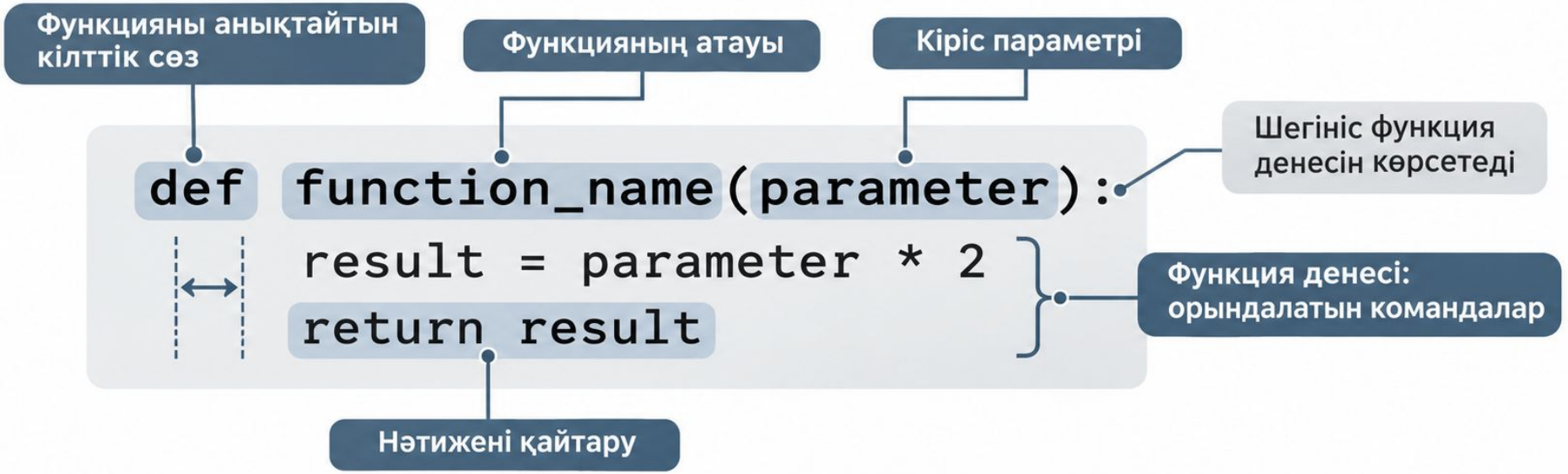
- Температураны тексеру.
- Электр қуатын есептеу.
- Заряд деңгейін бағалау.
- Ескерту хабарламасын қалыптастыру.

Функцияның негізгі артықшылықтары:

1. Кодтың қайталануын азайтады.
2. Бағдарламаны түсінуді жеңілдетеді.
3. Жеке бөліктерді тексеруге мүмкіндік береді.
4. Кодты қайта пайдалануды қамтамасыз етеді.
5. Бағдарламаны өзгерту мен кеңейтуді жеңілдетеді.

Функцияның құрылымы

Python тілінде функция **def** кілттік сөзі арқылы анықталады.



Шақыру мысалы



5. Функцияны анықтау және шақыру

Функцияны анықтау мен шақыру — екі бөлек әрекет.
Функцияны анықтау (Function Definition) — функцияның атауын және орындалатын командаларын сипаттау.
Функцияны шақыру (Function Call) — функцияны орындауға жіберу.

Функция бір рет анықталып, әртүрлі мәндермен бірнеше рет шақырылуы мүмкін.

```
def calculate_power(voltage, current):
    power = voltage * current
    return power

result = calculate_power(380, 20)

print(result)
```

6. Параметр мен аргумент ұғымдары

Функциямен жұмыс істеу кезінде параметр (Parameter) және аргумент (Argument) ұғымдарын ажырату маңызды.
Параметр — функция анықтамасында көрсетілген айнымалы.
Аргумент — функция шақырылған кезде оған берілетін нақты мән.

```
def check_temperature(temp):
    return temp > 45

check_temperature(50)
```

Мұнда:

- `temp` — параметр.
- `50` — аргумент.

Бір функция бірнеше параметр қабылдай алады.
Аргументтер позиция бойынша немесе атауы арқылы берілуі мүмкін.

Python функцияларын жіктеу

Анықталуына қарай

Кірістірілген функциялар
(Built-in Functions)

Python тілінде алдын ала анықталған.



- `print()` — экранға шығару
- `len()` — элементтер саны
- `sum()` — мәндер қосындысы
- `max()` — ең үлкен мән
- `min()` — ең кіші мән

Пайдаланушы анықтайтын функциялар
(User-defined Functions)

Бағдарламашы өзі құрады.

```
def battery_status(soc):
    if soc < 20:
        return "LOW"
    return "NORMAL"
```

Мысал: батарея зарядының күйін бағалау.

Қайтарылатын нәтижесіне қарай

Мән қайтаратын функция

```
def double(x):
    return x * 2
```

→ `double(5)` → **10**

Мәнді айқын қайтармайтын функция

```
def show_status():
    print("NORMAL")
```

→ **Экранға:** NORMAL

→ **Қайтарылатын мән:** None

ℹ | Бұл белгілер тәуелсіз: пайдаланушы функциясы мән қайтара алады немесе `None` қайтарады.

return операторының қызметі

return — функцияның орындалуын аяқтап, шақырған бағдарламаға мән қайтаратын оператор.

```
def calculate_average(a, b):
    return (a + b) / 2

average = calculate_average(30, 40)

print(average)
```

print() пен return бірдей емес.

print()	return
Ақпаратты экранға шығарады	Нәтижені функция шақырылған жерге қайтарады
Нәтижені көрсетуге арналған	Нәтижені кейінгі есептеулерде пайдалануға мүмкіндік береді
Функцияның жұмысын міндетті түрде тоқтатпайды	Функцияның орындалуын аяқтайды

Инженерлік есептеулерде функцияның нәтижесін басқа операцияларға беру үшін return жиі қолданылады.

Айнымалылардың қолданылу аймағы



`calculate_power(10) → 3800`

Нәтиже қайтарылады, **power** сыртқа қолжетімді болмайды.

Қажетті мәндерді параметр арқылы беру функцияны тексеруді және қайта пайдалануды жеңілдетеді.

Сөздік дегеніміз не?

Сөздік (Dictionary) — Python тіліндегі деректерді кілт және мән жұптары түрінде сақтайтын құрылым.

Сөздіктің әр элементі екі бөліктен тұрады:

- Кілт (Key) — мәнді анықтайтын бірегей идентификатор.
- Мән (Value) — кілтпен байланыстырылған ақпарат.

```
vehicle = {
    "model": "EV",
    "temperature": 35,
    "voltage": 380,
    "soc": 80
}
```

Бұл сөздік бір электрмобиль туралы бірнеше параметрді сақтайды.

Сөздіктердің негізгі ерекшеліктері

Python сөздіктерінің маңызды қасиеттері:

1. Деректер кілт–мән жұптары арқылы сақталады.
2. Әр кілт бірегей болуы керек.
3. Мәндер әртүрлі деректер типінде болуы мүмкін.
4. Сөздіктің мазмұнын өзгертуге болады.
5. Қазіргі Python нұсқаларында элементтердің енгізілу реті сақталады.
6. Кілттер хэштелетін типтерден болуы керек.

Мысалы, кілт ретінде жол немесе бүтін сан қолдануға болады.
Сөздіктің мәні сан, мәтін, тізім немесе басқа сөздік болуы мүмкін.

Тізім мен сөздіктің айырмашылығы

Тізім (List)	Сөздік (Dictionary)
Элементтер ретпен сақталады	Кілт–мән жұптары сақталады
Индекс арқылы қол жеткізіледі	Кілт арқылы қол жеткізіледі
Бір типтегі немесе әртүрлі мәндерді сақтайды	Әртүрлі типтегі мәндерді сақтайды
Бір параметрдің өлшеулеріне ыңғайлы	Бір объектінің бірнеше сипаттамасына ыңғайлы

Python тіліндегі тізім мен сөздік бірнеше мәнді сақтау үшін қолданылады, бірақ олардың ұйымдастырылуы әртүрлі. Мысалы, бес температура өлшеуін сақтау үшін тізім ыңғайлы.
Ал бір автомобильдің моделі, температурасы, кернеуі және заряд деңгейін сақтау үшін сөздік ыңғайлы.

Сөздіктердің құрылымдық мүмкіндіктері

Сөздіктер бір ғана объект туралы ақпаратты емес, күрделі құрылымдалған деректерді де сақтай алады.
Мысалы, бір сөздіктің ішінде басқа сөздік орналасуы мүмкін.
Бұл кірістірілген сөздік (Nested Dictionary) деп аталады.

```
vehicle = {
    "battery": {
        "voltage": 380,
        "soc": 80
    },
    "motor": {
        "temperature": 45,
        "rpm": 2500
    }
}
```

Мұндай құрылым күрделі жүйелердің құрамдас бөліктерін логикалық түрде ұйымдастыруға мүмкіндік береді.

Қорытынды

Функциялар мен сөздіктер — Python тілінде бағдарламаларды құрылымдаудың маңызды құралдары.

Функциялар бағдарламаны жеке логикалық бөліктерге бөліп, кодтың қайталануын азайтады және қайта пайдалануға мүмкіндік береді.

Сөздіктер деректерді кілт–мән жұптары түрінде сақтап, күрделі объектілер туралы ақпаратты түсінікті түрде ұйымдастырады.

Инженерлік бағдарламалар мен жасанды интеллект жүйелерінде бұл екі құрал жиі бірге қолданылады.

Бақылау сұрақтары

1. Бағдарламаны құрылымдау дегеніміз не?
2. Модульдік бағдарламалаудың негізгі мақсаты қандай?
3. Функция дегеніміз не?
4. Функцияны анықтау мен шақырудың айырмашылығы қандай?
5. Параметр мен аргументтің айырмашылығы неде?
6. return операторы қандай қызмет атқарады?
7. Жергілікті және ғаламдық айнымалылар дегеніміз не?
8. Сөздік қандай деректер құрылымы?
9. Сөздіктегі кілт пен мәннің айырмашылығы қандай?
10. Тізім мен сөздіктің айырмашылығы неде?
11. Кірістірілген сөздік дегеніміз не?
12. Функциялар мен сөздіктер инженерлік бағдарламаларда қалай бірге қолданылады?