Python for AI

Дәуренов Ә.Д

Дәріс 2 Python тіліндегі шартты конструкциялар және логикалық операторлар

Лекцияның мақсаты

- if, elif, else шартты конструкцияларын,
- салыстыру операторларын және
- and, or, not логикалық операторларын пайдалана отырып,

Python бағдарламаларында шешім қабылдау принциптерін меңгеру.

Лекциядан кейін студент мыналарды түсінуі тиіс:

- бағдарламадағы шарттың не екенін;
- логикалық өрнектің қалай құрылатынын;
- if, elif, else қалай жұмыс істейтінін;
- салыстыру операторларының қалай қолданылатынын;
- бірнеше шартты and, or, not көмегімен қалай біріктіру керектігін;

1. Бағдарламаға шарттар не үшін қажет?

1. Шартсыз бағдарлама

Шартты конструкциялар пайда болғанға дейін бағдарлама командаларды жоғарыдан төмен қарай ретімен орындайды.

Мысалы:

```
speed = 80
distance = 100

print(speed)
print(distance)
```

Айнымалылардың мәндеріне қарамастан, бағдарлама бірдей әрекеттерді орындайды.

2. Нақты жүйелерде

Нақты жүйелер ағымдағы деректерге байланысты **шешім қабылдауы** тиіс.

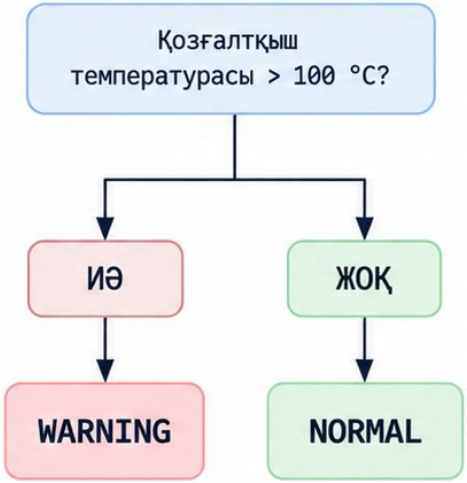
Мысалы, автокөлік жүйесі мыналарды талдай алады:



Автокөліктің жылдамдығы | Қозғалтқыш температурасы | Май қысымы | Отын деңгейі | Кедергіге дейінгі қашықтық

және алынған мәндерге байланысты әртүрлі әрекеттерді орындай алады.

3. Шешім қабылдау мысалы



Осылайша, **шартты конструкция** бағдарламаға белгілі бір шарттың орындалуына байланысты әрі қарайғы әрекетті таңдауға мүмкіндік береді.

2. Логикалық өрнек

Логикалық өрнек дегеніміз не?

Кез келген шартты конструкцияның негізінде логикалықөрнек болады.

Логикалық өрнек — нәтижесі екі мәннің біреуі болатын өрнек:

`True` немесе `False`

Тип `bool`

Python тілінде логикалық мәндер bool типіне жатады.

Мысал 1. Логикалық мәндер

```
print(True)
print(False)
```

Нәтиже:

```
True
False
```

Мысал 2. Салыстыру өрнегі

```
temperature = 105
print(temperature > 100)
```

Нәтиже:

```
True
```

Мысал 3. Тағы бір жағдай

```
temperature = 90
print(temperature > 100)
```

Нәтиже:

```
False
```

Мысал 4. Типті тексеру

```
condition = True
print(type(condition))
```

Нәтиже:

```
<class 'bool'>
```

? Шарт ақиқат (`True`) па, әлде жалған (`False`) ба?

3. Салыстыру операторлары

Шартты құру үшін салыстыру операторлары қолданылады.

Оператор	Мағынасы	Мысал
`>`	үлкен	`speed > 90`
`<`	кіші	`fuel < 10`
`>=`	үлкен немесе тең	`temperature >= 100`
`<=`	кіші немесе тең	`speed <= 90`
`==`	тең	`gear == 1`
`!=`	тең емес	`speed != 0`

Мысал

```
speed = 100
print(speed > 90)
print(speed < 90)
print(speed == 100)
```

Нәтиже

```
True
False
True
```



Салыстыру операторлары екі мәннің арасындағы қатынасты тексеріп, нәтижесінде `True` немесе `False` мәнін қайтарады.

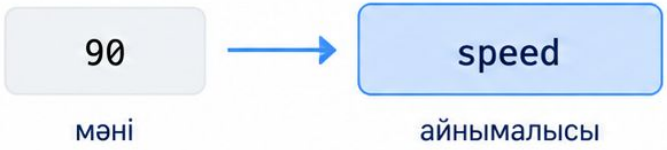
4. = және == арасындағы маңызды айырмашылық

Жаңадан үйренушілер бұл екі операторды жиі шатастырады.

= Мәнді меншіктеу

Оң жақтағы мәнді сол жақтағы айнымалыға сақтайды.

```
speed = 90
```



Бұл мынаны білдіреді:
90 мәнін **speed** айнымалысына сақтау.

== Салыстыру

Екі мәннің тең екенін тексереді және нәтижесі логикалық мән болады (`True` немесе `False`).

```
speed == 90
```

Бұл сұрақ:
speed айнымалысының мәні **90** санына тең бе?

Мысал:

```
speed = 90
print(speed == 90)
```

Нәтиже:

```
True
```

Қысқаша:

- `=` — мәнді меншіктеу (сақтау)
- `==` — салыстыру (теңдігін тексеру)

5. Шартты конструкция `if`

`if` — белгілі бір шарт орындалса, блоктағы командаларды орындайды.

Жалпы құрылымы

```
if шарт:
    командалар
```

Шарттың нәтижесі `True` немесе `False` мәнін қабылдайды.

Мысал

```
temperature = 105

if temperature > 100:
    print("WARNING")
```

Шартты тексеру

```
temperature > 100
```

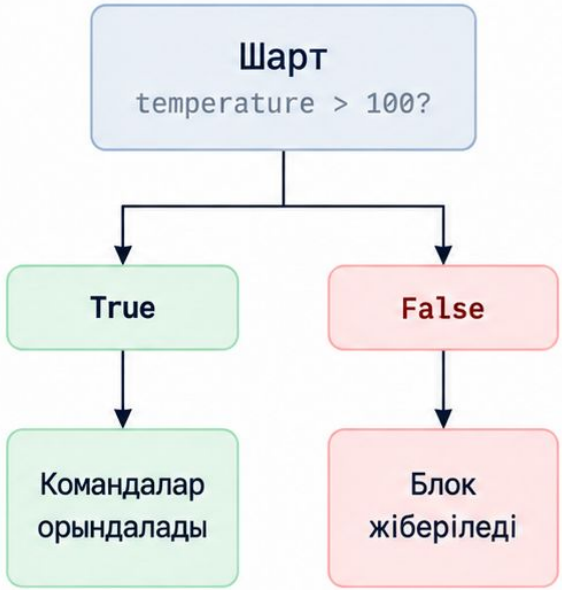
Нәтиже

```
True
```

Орындалатын әрекет

```
print("WARNING")
```

Жұмыс логикасы



7. `if` / `else` конструкциясы

Егер әрекетті тек қана ақиқат шартта емес, жалған шартта да анықтау қажет болса, `else` қолданылады.

Жалпы құрылымы

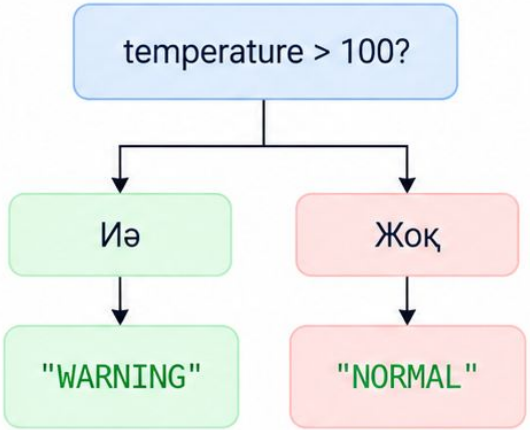
```
if шарт:
    әрекет_1
else:
    әрекет_2
```

Мысал

```
temperature = 90

if temperature > 100:
    print("WARNING")
else:
    print("NORMAL")
```

Жұмыс логикасы



Программа шарттың нәтижесіне қарай **екі тармақтың бірін** міндетті түрде орыңаайды.

8. `if` / `elif` / `else` конструкциясы

Бірнеше шартты тексеру қажет болғанда, `if` / `elif` / `else` қолданылады.

Мәселе

Қозғалтқыштың температурасын үш диапазонға бөлеміз:

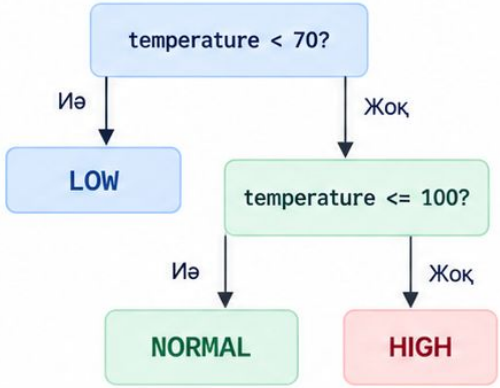
< 70 °C	LOW
70 – 100 °C	NORMAL
> 100 °C	HIGH

Python коды

```
temperature = float(input("Температураны
                          енгізіңіз: "))

if temperature < 70:
    print("LOW")
elif temperature <= 100:
    print("NORMAL")
else:
    print("HIGH")
```

Жұмыс логикасы



Мысал

```
temperature = 85

if temperature < 70:
    print("LOW")
elif temperature <= 100:
    print("NORMAL")
else:
    print("HIGH")
```

Нәтиже

```
NORMAL
```

Шарттар жоғарыдан төмен қарай тексеріледі. Бірінші ақиқат шарт орындалғаннан кейін, қалған шарттар орындалмайды.

9. Неліктен `if` / `elif` реті маңызды?

Шарттар жоғарыдан төмен қарай тексеріледі. Алғашқы `True` болған шарт орындалады, қалғандары тексерілмейді.

Мысал (қате логика)

```
temperature = 105

if temperature > 70:
    print("NORMAL")
elif temperature > 100:
    print("HIGH")
```

Қалай орындалады?

1. `temperature = 105`
2. Бірінші шарт тексеріледі:
   `temperature > 70`
   `105 > 70` → `True`
3. Алғашқы шарт `True` болғандықтан, `print("NORMAL")` орындалады.
4. Екінші шарт (`elif`) тексерілмейді.

Нәтиже

NORMAL

Нақты мәндермен тексерейік

`temperature > 70`	→	`True`
`temperature > 100`	→	`False`

! Бағдарлама бірінші шынайы шарт орындалғаннан кейін қалған шарттарды тексермейді.

Бірнеше шартпен жұмыс істегенде, олардың ретін дұрыс анықтау өте маңызды.

10. Диапазонды тексеру

Көбінесе параметр белгілі бір диапазонның ішінде екенін анықтау қажет.

Диапазон мысалы

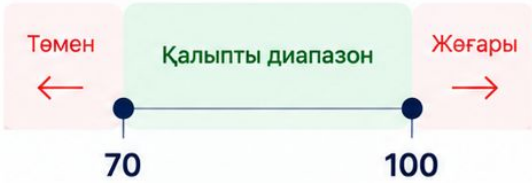
$$70 \leq temperature \leq 100$$

Бұл — temperature мәні 70 мен 100 қоса алғанда, осы аралықта деген сөз.

Python кодында

```
if 70 <= temperature <= 100:
    print("NORMAL")
```

Жұмыс логикасы



Нақты мысал

```
pressure = 2.3

if 2.0 <= pressure <= 2.5:
    print("Қысым рұқсат етілген диапазонда")
```

Нәтиже

Қысым рұқсат етілген диапазонда

Қолдану саллары

- Температура
- Қысым
- Жылдамдық
- Кернеу
- Заряд деңгейі
- Қозғалтқыш айналымы

Мұндай тексерулер нақты техникалық шектерді (условные учебные пороги) қолдану үшін өте ыңғайлы.

11. Неліктен логикалық операторлар қажет?

Көбінесе бір ғана параметрді тексеру жеткіліксіз. Нақты жағдайларда бірнеше шартты бір уақытта тексеру керек.

Бір параметрді тексеру

Мысалы:

```
temperature > 100
```

Бұл тек бір шартты тексереді.

Нақты жүйелерде бірнеше параметр бар

- Қозғалтқыштың температурасы
- Жанармай деңгейі
- Май қысымы
- Кедергіге дейінгі қашықтық
- Жылдамдық
- Қауіпсіздік белдігінің жағдайы

Бірнеше шартты тексеру үшін

Логикалық операторлар қолданылады:

and — барлық шарттар дұрыс болуы керек.

or — кем дегенде бір шарт дұрыс болуы керек.

not — шартты терістеу (дұрыс болса — қате, қате болса — дұрыс).

Мысал

```
temperature = 95
oil_pressure = 3.2

if temperature <= 100 and oil_pressure >= 3.0:
    print("Жұмыс жағдайы қалыпты")
else:
    print("Тексеру қажет")
```

Логикалық операторлар күрделі жағдайларды тексеруді оңай және түсінікті етеді.

12. Логикалық оператор `and`

`and` — екі шарт та ақиқат болғанда ғана `True` мәнін қайтарады.

Мысал

```
temperature = 90
oil_pressure = 2.5

if temperature <= 100 and oil_pressure >= 2:
    print("NORMAL")
else:
    print("CHECK")
```

Шындық кестесі (and)

Шарт 1	Шарт 2	Нәтиже
True	True	True
True	False	False
False	True	False
False	False	False

Қалай жұмыс істейді?

`temperature <= 100` (True) **and** `oil_pressure >= 2` (True) = **True**

Ебінде сақта!



and барлық біріктірілген шарттардың орындалуын талап етеді.

13. Логикалық оператор `or`

or — кем дегенде бір шарт ақиқат болса **True** мәнін қайтарады (ЛИБО).

Нені білдіреді?

A **or** **B**

Кем дегенде бір шарт ақиқат болса, нәтиже **True** болады.

Мысал жағдай

- Температура тым жоғары (`temperature > 100`)
- Май қысымы төмен (`oil_pressure < 2`)

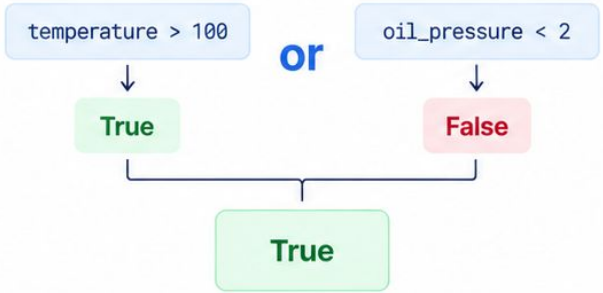
Осы жағдайлардың кем дегенде біреуі орындалса — **WARNING** көрсетіледі.

Python коды

```
temperature = 105
oil_pressure = 2.5

if temperature > 100 or oil_pressure < 2:
    print("WARNING")
else:
    print("NORMAL")
```

Қалай жұмыс істейді?



Шындық кестесі (or)

Шарт 1	Шарт 2	Нәтиже
True	True	True
True	False	True
False	True	True
False	False	False

Есінде сақта!

or — біріктірілген шарттардың кем дегенде біреуінің орындалуын талап етеді.

14. and және or арасындағы айырмашылық

Екі оператор да бірнеше шартты тексеруге қолданылады, бірақ олардың логикасы әртүрлі.

and — барлық шарттар орындалуы керек

and тек екі шарт та ақиқат болғанда ғана **True** қайтарады.

Мысал:

```
speed = 100
distance = 20

if speed > 90 and distance < 30:
    print("DANGER")
```

Бұл нені білдіреді?

Жылдамдық 90-нан жоғары **ЖӘНЕ** бір уақытта қашықтық 30-дан аз.

Шындық кестесі (and)

Шарт 1	Шарт 2	Нәтиже
True	True	True
True	False	False
False	True	False
False	False	False

or — кем дегенде бір шарт орындалуы керек

or кем дегенде бір шарт ақиқат болса **True** қайтарады.

Мысал:

```
speed = 100
distance = 20

if speed > 90 or distance < 30:
    print("WARNING")
```

Бұл нені білдіреді?

Жылдамдық 90-нан жоғары **НЕМЕСЕ** қашықтық 30-дан аз, немесе екеуі де.

Шындық кестесі (or)

Шарт 1	Шарт 2	Нәтиже
True	True	True
True	False	True
False	True	True
False	False	False



and — қатаң шарт, **or** — икемді шарт. Қажетті логикаға сәйкес операторды таңдау маңызды.

15. Логикалық оператор `not`

`not` — логикалық мәнді терістейді (НЕ). Ол `True` мәнін `False`-қа, ал `False` мәнін `True`-ға өзгертеді.

Қалай жұмыс істейді?

not X

X — логикалық мән (True немесе False).
not оны терістейді.

not — шартты тексереді, бірақ мәнін қарама-қарсы етеді.

Шындық кестесі (not)

Бастапқы мән	not нәтижесі
True	False
False	True

Мысал 1

```
engine_running = True

print(not engine_running)
```

Нәтиже:

```
False
```

Мысал 2

```
seat_belt = False

if not seat_belt:
    print("Қауіпсіздік белдігі тағылмаған")
```

Мысал 3

```
not True    # → False

not False   # → True
```

`not` жиі `if` шарттарында қолданылады, мысалы: «егер емес ...» («если не ...»).

16. Бірнеше логикалық оператор

Шарттарды біріктіруге болады. Мұнда **and** , **or** және **жақшалар** қолданылы мүмкін.

Python коды

```
if speed > 0 and (temperature > 100 or oil_pressure < 2):
    print("WARNING")
```

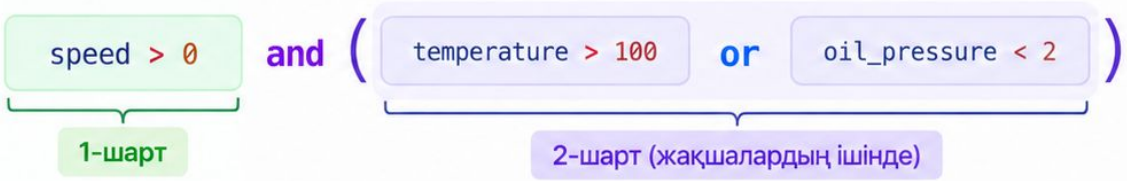
Нәтиже

Шарт орындалса, программа

WARNING

жолын шығарады.

Шарттың құрылымы



Мағынасы



Автомобиль қозгалады
ЖӘНЕ
(температура жоғары
НЕМЕСЕ май қысымы төмен).

Қолданылатын операторлар

and
барлық шарттар орындалуы керек.

or
кем дегенде бір шарт орындалуы керек.

( )
жақшалар орындалу ретін анықтайды.

Кеңес



Күрделі шарттарда жақшаларды қолдану кодтың оқылымдылығын жақсартады және қателіктерден сақтайды.