Жамбыл облысы әкімдігі білім басқармасы

«Білім» кәсіби гуманитарлық-техникалық колледжі

(білім беру ұйымының атауы)

**Оқу сабағының жоспары**

(теориялық немесе өндірістік оқыту)

**Break циклін басқару, нұсқаулығымен жұмыс.**

(сабақ тақырыбы)

**Модуль/Пән атауы** Бағдарламалық кодты қайта өңдеу

**Дайындаған педагог** Нургисаева У.М

**20\_**25**\_ жылғы** «\_\_\_\_» \_\_\_ақпан\_\_\_

**1. Жалпы мәліметтер**

Курс, оқу жылы, топ 3 курс, 3БҚ-22

Сабақ түрі: Кірктірілген

**2. Мақсаты, міндеттері:**

**Оқу:** Білім алушыларға python (пайтон) программалау тілінде Break операторы циклін басқару, қолдану жолдарын ұйымдастыру, программаны өңдеу жұмыснегіздерін үйрету, шартты операторлардың жұмыс істеу принциптерін түсіндіру.

#### Дамыту: Логикалық ойлау қабілеттерін дамыту, есептердің әртүрлі шешімдерін табуға дағдыландыру, программалық кодтың құрылымын дұрыс құру қабілеттерін қалыптастыру.

#### Тәрбиелік: Жауапкершілік пен ұқыптылыққа тәрбиелеу, алгоритмдік ойлауды дамыту, шығармашылық қабілеттерін жетілдіру.

**3. Оқу-жаттығу процесінде білім алушылар меңгеретін күтілетін нәтижелер және кәсіби дағдылар тізбесі:** Python программалау тілінде Break операторы циклін басқару, қолдану жолдарын ұйымдастыру, программаны өңдеу және пайдалану дағдыларын меңгеру. Continue циклін басқару, қолдану жолдарының жұмыснегіздерін үйренеді, код жазады.

**4. Қажетті ресурстар:** ДК немесе ноутбуктер

[***https://baier.kz/programmalau/***](https://baier.kz/programmalau/)

***5. Сабақтың барысы: (90 минут)***

**5.1. Ұйымдастыру кезеңі:** *( 3 мин )*

**5.2. Үй жұмысын жан-жақты тексеру:**

***“Пікірталас ” әдісі*** *(15 минут)*

[*https://quizlet.com/771083301/flashcards*](https://quizlet.com/771083301/flashcards)

**Break циклін басқару, нұсқаулығымен жұмыс.**

Break операторы циклдарда қолданылатын маңызды құрал болып табылады. Оның көмегімен циклдің орындалуын шартқа қарамастан тоқтатуға болады. Break операторын тиімді қолдану арқылы біз циклды басқаруды әлдеқайда икемді ете аламыз.

**2. Циклдер және Break операторының маңызы**

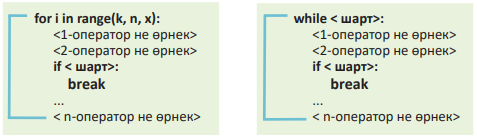
Циклдер көбінесе белгілі бір шартқа байланысты бірнеше рет орындалады. Бірақ кейде циклды тоқтату қажет болады, мысалы, белгілі бір шарт орындалса немесе қажетті нәтиже табылса. Мұндай жағдайларда **break** операторы қолданылады.

**Break** операторы циклдың орындалуын ерте аяқтауға мүмкіндік береді. Яғни, циклдың қалған бөлігін өткізіп, циклден шығуға болады. Бұл өте пайдалы жағдайларда, мысалы, іздеу операциялары немесе қатеге тап болғанда циклды тоқтату кезінде қолдануға ыңғайлы.

**Break (үзілу)** – белгілі бір қойылған шарт орындалуымен ағымдағы цикл жұмысын бірден тоқтатып, программа жұмысын циклден шығарып, әрі қарай жалғастыратын цикл нұсқаулығы.

**Python-да break**нұсқаулығы цикл жұмысына әсер ететін сыртқы фактор пайда болған кезде циклді бұзуға мүмкіндік береді. **Break**нұсқаулығын**for** параметрлі циклі мен **while**шарт циклінде қолдануға болады . Схемаларда көрсетілгендей, **break-тің** программада жазылуында continue нұсқаулығынан айырмашылығы жоқ. Тек екеуінің атқаратын қызметі екі түрлі, бірі циклден шығаруды қамтамасыз етсе, екіншісі циклдің белгілі бір қадамдарын орындамай өтуге мүмкіндік береді.

**Break**нұсқаулығы циклдің екі түрінде қалай жұмыс істейтінін түсіну үшін бір мысал қарастырайық.



**3. Break операторының синтаксисі**

**Break** операторының қолданылуы өте қарапайым. Ол цикл ішінде қажетті шарт орындалған кезде қолданылады. Мысалы, егер біз белгілі бір мәнді тапсақ немесе басқа шарт орындалса, циклдың орындалуын тоқтатамыз. Оның жалпы синтаксисі келесідей болады:

while шарт:

# Шартқа сай код

if шарт\_орындалса:

break

**4. Break операторының жұмыс принципі**

**Break** операторы циклдың ішіне қойылады. Ол циклдың орындалуын тек сол циклдың шеңберінде тоқтатады. Егер циклда бірнеше деңгей болса (мысалы, ішкі циклдар болса), онда **break** операторы тек сол циклды тоқтатады, ал сыртқы циклдар әлі де орындала береді. Мысалы:

for i in range(5):

for j in range(5):

if j == 3:

break # Ішкі циклды тоқтатады, бірақ сыртқы цикл жалғасады

print(f"i = {i}, j = {j}")

Бұл жағдайда, ішкі цикл j = 3 болғанда тоқтайды, бірақ сыртқы цикл орындала береді.

**5. Break операторының мысалдары**

**Мысал 1: Циклды ерте тоқтату**

Мысалы, біз белгілі бір санды тапсақ, циклды тоқтату керек болар еді. Мысал:

for num in range(1, 10):

if num == 5:

print("5 табылды, цикл тоқтатылады.")

break

print(num)

Бұл кодта цикл 5-ке жеткенде тоқтайды және "5 табылды, цикл тоқтатылады." деген хабарлама шығарылады.

**Мысал 2: Циклдар мен шарттар**

Енді біз бірнеше шарт бойынша циклді басқару мысалын қарастырайық:

for i in range(1, 20):

if i % 2 == 0:

continue # Тек тақ сандарды басып шығарамыз

if i > 15:

break # 15-тен үлкен сандарды қарастыруды тоқтату

print(i)

Бұл кодта тек тақ сандар ғана басылып шығарылады, ал 15-ке жеткенде цикл тоқтайды.

**6. Break операторының қолдану аясы**

**Break** операторының негізгі қолдану аясы келесі жағдайларда болады:

1. **Іздеу операциялары**: Егер біз белгілі бір шартқа сәйкес деректерді іздесек, және сол дерек табылғанда циклды тоқтату қажет болса, break операторы өте пайдалы болады. Мысалы, белгілі бір элементті табу үшін іздеуге арналған циклды қолданғанда.
2. **Қателерді өңдеу**: Цикл ішінде қателік тапқан жағдайда, одан әрі есептеуді жалғастырудың орнына циклды тоқтатуға болады.
3. **Жинақтау немесе есептеу процесі**: Егер сізге белгілі бір шартты орындағаннан кейін нәтиже қажет болса, break операторы оны жүзеге асыру үшін тиімді болады.
4. **Шартқа сай әрекеттерді орындау**: Егер цикл барысында белгілі бір шарт орындалса, сіз тек сол уақыт аралығында әрекет ету үшін циклды тоқтата аласыз.

**7. Break және continue операторларының айырмашылығы**

**Break** және **continue** операторлары ұқсас болып көрінгенімен, олардың функционалдығы әртүрлі. Екеуі де циклдың орындалу тәртібін өзгертеді, бірақ олардың әсері әртүрлі.

* **Break**: Циклдың орындалуын тоқтатады. Циклды толық аяқтап, одан шығуға мүмкіндік береді.
* **Continue**: Тек ағымдағы итерацияны өткізіп, циклды жалғастыруға мүмкіндік береді. Бұл жағдайда циклдың барлық шарттары тексеріледі, бірақ ағымдағы итерация толығымен өткізіледі.

**Мысал**:

for i in range(1, 10):

if i == 5:

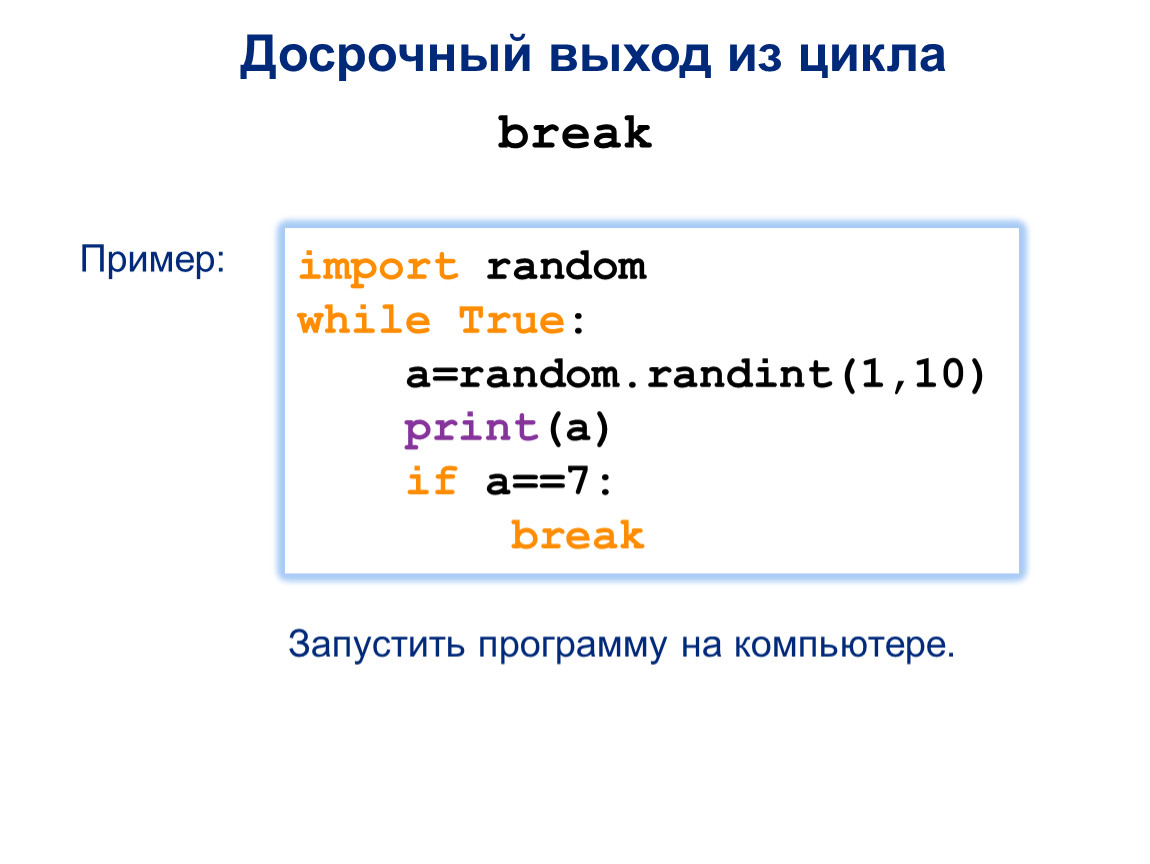
continue # 5-ті өткізіп, келесі санға өтеміз

if i == 8:

break # 8-ге жеткенде циклды тоқтатамыз

print(i)

Бұл жағдайда 5 саны өткізіліп, 8-ге жеткенде цикл тоқтатылады.



**Сабақты қорытындылау**. [**https://learningapps.org/9040640**](https://learningapps.org/9040640)

Қосымшада ашылған сұрақтарға жауап беру. <https://learningapps.org/display?v=pzszrmwtv23>

|  |  |  |
| --- | --- | --- |
| **Білдім** | **Білемін** | **Білгім келеді** |
|  |  |  |

**Рефлексия**

**Үй тапсырмасы –** Зертханалық жұмыс №4 орындау және қорғау