Дәріс № 5. Нұсқаулық құрастыру. Басқару операторлары if, while, for.

Салыстыру шамалар

Екі операндты салыстыру үшін Python бағдарламалауында жиі қолданылатын операторлар төмендегі кестеде берілген.

Оператор	Тексеруге болады жағдай
= =	Теңдік
! =	Теңсіздік
>	Көбірек
<	Аздау
> =	Көбірек немесе тең
< =	Аздау немесе тең

Теңдік операторы == екі операндты салыстырады және қайтарады Егер олардың мәндері тең болса, дұрыс, әйтпесе қайтарады Жалған. Сонымен қатар, егер операндтар сандық мәндер болса және олар бірдей болса, онда олар тең, ал символдар болса, олардың ASCII кодтары салыстырылады.

Теңсіздік операторы, !=, керісінше, егер екі операнд те болса, True мәнін қайтарады Жоқ тең қолдану Бұл бірдей ең әдетте, Қалай Және оператор теңдік, әйтпесе False мәнін қайтарады. Теңдік және теңсіздік операторлары екі айнымалы мәндерді салыстыру нәтижелеріне негізделген бағдарламада шартты тармақтауды орындау үшін пайдалы.

Оператордан үлкен, > екі операндты салыстырады және егер біріншінің мәні екіншісінің мәнінен үлкен болса, True мәнін қайтарады және керісінше, егер біріншінің мәні екіншісінің мәніне тең немесе одан кіші болса, False мәнін қайтарады. . «Кіші» операторы бірдей әрекетті жасайды, бірақ бірінші оператор одан кіші болса, True мәнін қайтарады. Бұл екі мәлімдеме жиі циклдегі операциялық есептегіштің мәнін тексеру үшін қолданылады.

Қосу оператор «тең», =, кейін оператор «Көбірек», >, немесе "кем", <, операндтардың мәндері сәйкес келсе, олардың True қайтаруын тудырады.

Болжалды логика

Python бағдарламалауында қолданылатын логикалық операторлар төмендегі кестеде берілген.

Оператор	Операция
және	Логикалық «ЖӘНЕ»
немесе	Логикалық "НЕМЕСЕ"
емес	Логикалық «ЖОҚ»

Логикалық операторлар жұмыс істейді операндтармен, Логикалық мәні

бар, яғни True немесе False немесе True немесе False мәніне түрлендіретін мәндері бар.

Логикалық ЖӘНЕ операторы және, екі операндты бағалайды және егер екі операнд да True болса ғана, ол False мәнін қайтарады; Бұл оператор әдетте шартты түрде қолданылады тармақталу, Қашан бағыт жұмыс бағдарлама екі шартты тексеру арқылы анықталады - егер олардың екеуі де ақиқат болса, бағдарлама белгілі бір бағытта жүреді, әйтпесе - басқа.

Екі операндтың да True болуын талап ететін және операторынан айырмашылығы, логикалық НЕМЕСЕ операторы немесе, екі операндты бағалайды және олардың кем дегенде біреуі True мәнін қайтарса, True мәнін қайтарады. Әйтпесе, немесе операторы False мәнін қайтарады.

Бұл тексерілген екі шарттың біреуі орындалса, белгілі әрекеттерді бағдарламалағанда пайдалы.

Логикалық NOT операторы unary болып табылады және бір операндпен қолданылады. Ол бар болғанына қарама-қарсы мәнді қайтарады операнд. Сонымен, Егер айнымалы a болды мағынасы Рас, Бұл емес a False мәнін қайтарады. Оны, мысалы, a = емес операторы арқылы циклдің дәйекті итерацияларында айнымалының мәнін ауыстыру үшін пайдалануға болады. Бұл циклдің әрбір итерациясында шамды өшіру және қосу сияқты логикалық мән кері өзгеретінін білдіреді.

Шартты оператор ЕГЕР

Әрбір if мәлімдемесінде орталық орынды нәтижесі логикалық ақиқат (Шын) немесе логикалық болатын өрнек алады. жалған (жалған); бұл өрнек шарт деп аталады. Нәтижеге байланысты Python тексерулері шешеді, код орындалу керек егер команда. Шарттың нәтижесі True болса, Python if операторынан кейінгі кодты орындайды. Егер нәтиже False болса, Python if пәрменінен кейінгі кодты елемейді.

Қарапайым командалар егер

if командасының қарапайым түрі бір шарт пен бір әрекеттен тұрады:

```
шарт
   болса:
   әрекет
```

Бірінші жол шартты қамтиды, ал шегініс блокта кез келген дерлік әрекет бар. Шарт шын болса, Python орындалады if операторынан кейінгі блоктағы код, ал егер жалған болса, бұл код еленбейді.

if операторындағы шегініс for циклдеріндегі сияқты рөл атқарады. Егер шарт ақиқат болса, онда if операторынан кейінгі барлық шегініс жолдар орындалады, ал егер жалған болса, шегінген блоктың барлығы еленбейді.

Көбінесе бағдарлама шарт ақиқат болса бір әрекетті, ал жалған болса басқа әрекетті орындауы керек. if-else синтаксисі арқылы бұл мүмкін. if-else блогы негізінен if операторына ұқсас, бірақ else бөлімі сынақ сәтсіз болғанда орындалатын әрекетті немесе әрекеттер жинағын көрсетеді.

Тізбектер if-elif- басқа

Көбінесе бағдарлама екіден көп ықтимал жағдайларды тексеруі керек; Python мұндай жағдайлар үшін if-elif-else синтаксисін қамтамасыз етеді. Python if-elif-else тізбегінде тек бір блокты орындайды. Барлық шарттар олардың біреуі шынайы нәтиже бергенше ретімен тексеріледі. Содан кейін осы шарттан кейінгі код орындалады және Python барлық басқа тексерулерді өткізіп жібереді.

Көптеген нақты өмірлік жағдайларда екіден көп нәтиже болуы мүмкін.

Әртүрлі жас топтары үшін әртүрлі кіру ақысын алатын ойын-сауық саябағын елестетіп көріңіз:

- Үшін келушілер астында 4 жылдар Кіру Тегін.
- Үшін келушілер бастап 4 бұрын 18 жылдар билет шығындар $25.
- Үшін келушілер бастап 18 жылдар Және егде билет шығындар $40.

1-тармақтағы if шарты келушінің жасы 4 жастан аз екенін тексереді. Шарт дұрыс болса, бағдарлама сәйкес хабарламаны көрсетеді және Python қалған тексерулерді өткізіп жібереді. 2-нүктедегі elif сызығы шын мәнінде басқа, егер алдыңғы сынақ сәтсіз болса ғана орындалатын сынақ. Тізбектің осы нүктесінде біз келушінің кем дегенде 4 жаста екенін білеміз, өйткені бірінші шарт жалған болды. Егер келушінің жасы 18-ге толмаған болса, бағдарлама сәйкес хабарламаны көрсетеді және Python else блогын өткізіп жібереді. Егер if және elif шарттары жалған болса, Python блоктағы кодты орындайды басқа 3-тармақта.

Сериялар блоктар элиф

Кодта қалағаныңызша көп elif блоктары болуы мүмкін. Мысалы, егер тақырыптық саябақ егде жастағы пайдаланушылар үшін арнайы жеңілдікті енгізсе, жеңілдік ағымдағы меценатқа қолданылатынын анықтау үшін кодыңызға басқа чек қосуға болады. Кодтың көпшілігі өзгеріссіз қалады. 1-ші нүктедегі екінші elif блогы енді келушінің 65 жастан төмен екеніне көз жеткізіп, билеттің толық құнын 40 доллардан алады. Есіңізде болсын, мән else блогында тағайындалған 2, міндетті болуы ауыстырылды қосулы $20, Сондықтан Не бұрын бұл блок Тек 65 жастан асқан келушілер ғана қатыса алады.

Болмауы блок басқа

Python Жоқ талап етеді дейін шынжыр егер-элиф аяқталатыны сөзсіз блок басқа. 1-тармақтағы elif блогы келушінің жасы 65 немесе одан жоғары болса, $20 бағасын белгілейді; Мұндай кодтың мағынасы жалпы else блогына қарағанда түсінікті. Бұл өзгерту арқылы әрбір блок белгілі бір шарт ақиқат болса ғана орындалуы мүмкін. else блогы «әмбебап» болып табылады: ол сәйкес келмейтін барлық шарттарды өңдейді бір нақты if немесе elif тексеруі аясында және жарамсыз немесе тіпті зиянды деректер кейде осы санатқа жатады. Егер сізде болса финал нақты жағдайы, жақсырақ кейінгі elif блогын пайдаланыңыз және else блогын өткізбеңіз. Осылайша сіздің кодыңыз тек дұрыс жағдайларда жұмыс істейтініне сенімді бола аласыз.

Циклдер

Цикл – автоматты түрде қайталанатын программадағы код бөлігі. Цикл ішіндегі нұсқаулардың бір толық орындалуы итерация немесе өту деп аталады. Цикл өлшемі цикл ішінде жасалған сынақ шартымен басқарылады. Цикл сынақ өрнегі True болғанша жалғасады және ол False болатын нүктеде аяқталады.

Цикл кезінде

Python бағдарламалауында циклдер while кілт сөзі арқылы жасалады. Одан кейін сынақ өрнегі, содан кейін қос нүкте белгісі қойылады. Төменде өрнек сәтті тексерілгенде орындалатын нұсқаулар болуы керек. Әрбір жолда while кілт сөзі бар жолдан бірдей шегініс болуы керек. Бұл блокта тексеру өрнегі мәнін өзгертетін нұсқау болуы керек - әйтпесе цикл шексіз болады.

Циклдерді бір-бірінің ішіне салуға болады және сыртқы циклдің әрбір итерациясында ішкі циклдің барлық итерациялары әлдеқашан аяқталды. Ыңғайлы болу үшін әрбір цикл анықтамасының алдында «есептегіш» деп аталатын айнымалы мәнді инициализациялауға болады, оған бастапқы мән беріп, оны әрбір иерацияда көбейтуге, сондай-ақ бұл айнымалы мәнді сынақ өрнегіне қосуға болады. Өрнекті тексеру сәтсіз аяқталса, цикл аяқталады.

```
n=3
m=4
factorial=1
i=1

whilei<=n:j=1
    whilej<=m:
        print(i,j)j
        +=1
    i+=1
```

Цикл үшін

Python кез келген берілген тізімнің элементтерін қайталау үшін for in операторын пайдаланады. in кейін тізімнің аты. Нұсқау қос нүктемен аяқталуы керек, одан кейін циклдің әрбір итерациясында орындалатын нұсқаулар келеді. Мысалы:

үшін элемент жылы тізім атауы :

әр итерациядағы орындалатын-нұсқаулар

МЕН көмегімен нұсқаулар үшін жылы мүмкін айналма жол В цикл элементтері тізім, кортеж, жиын және сөздік пернелері. Жол таңбалар тізімінен басқа ештеңе болмағандықтан, жолдағы әрбір таңбаны for in операторы арқылы циклде қайталауға болады.

for циклі арқылы кез келген тізімнің элементтерін немесе жолдағы таңбаларды олар пайда болу ретімен қайталауға болады,

бірақ циклдің қайталану санын, тоқтату шартын немесе итерация қадамының өлшемін нақты көрсете алмайсыз. . Дегенмен, diapazon() to тіл функциясын пайдалана аласыз жасау кейінгі реттілік сандар, пайдаланылады Үшін итерациялар. Бұл функция нөлден басталып, жақшадағы санмен аяқталатын, оны қоспай-ақ тізбекті жасайды. Мысалы, range(5) 0,1,2,3,4 тізбегін жасайды.

Мысалы:

```
s=0
үшін мен
   жылы
   диапазон(5
   ): басып
   шығару(s+i
   )
```