

Дәріс №2 Деректер типтері. Құрылымдық және негізгі деректер типтері. Айнымалылармен жұмыс.

Айнымалылар кез келген бағдарламалау тілінің маңызды бөлігі болып табылады. Айнымалы – компьютер жадындағы мәліметтер сақталатын контейнер. Деректер сақталғаннан кейін оны осы айнымалының атын пайдаланып шақыруға болады. Айнымалылар деректерді сақтауға, пайдалануға және тасымалдауға мүмкіндік береді.

Айнымалылар деректерді ыңғайлы өңдеуге ғана емес, сонымен қатар оны белгілі бір ережелерге сәйкес жұмыс істеуге болатын түрлерге бөлуге мүмкіндік береді. Python тілінде олар басқа бағдарламалау тілдерінде қолданылатын ұғымдарды ұстанады, бірақ олардың түрін анық жарияламау сияқты ерекшеліктер де бар.

Python 3-те айнымалы құру басқа бағдарламалау тілдерінен ерекшеленеді. Оны алдын ала жариялау немесе оның түрін көрсету қажет емес, ол мән тағайындалған кезде жасалады;

Айнымалы мәнді жасау үшін «=» тең белгісі пайдаланылады . Оның сол жағында аты, ал оң жағында - қажетті түрдің мәні жазылады. Мысалы:

```
name="Александр"#Типastr(строка) age = 15 # Типа
int (целое число)
```

Динамикалық теру

Бұл айнымалыларды құру динамикалық терудің арқасында мүмкін болады .

Переменную не нужно объявлять заранее, потому что тип определяется по присвоенному значению, кроме того, при повторном присваивании значения другого типа, не нужно переопределять тип.

Көпше тапсырма Және айырбастау құндылықтар

Python мүмкіндік береді бағдарламашы бірнеше айнымалыға бірдей мәнді тағайындаңыз :

```
a=b=c=5#Все(a,b и c)хранятзначение5
```

қоспағанда Бару, В айырмашылық бастап басқалар тілдер бағдарламалау Python екі айнымалының мәндерін алмасуға мүмкіндік береді -

```
a =5,b=3, c =7
a,b =b,a#Теперь a=3,ab =5
a,b,c=c,a,b#Можнообмениватьзначениянетолькодлядвух
```

Аттар

Python 3-тегі айнымалы атаулар тек әріптер мен сандарды пайдалана алады Және символдар астын сызу. Сандар мүмкін пайдалану В кез келген есімнің басынан басқа бөліктері.

Ал егер синтаксис бағдарламашыға бірнеше шектеулер қойса, онда АТ қауымдастығы айнымалылар үшін «жақсы» атауларды талап етеді :

- Атау мақсатты сипаттауы керек. Кодты бірнеше рет қайта оқуға және өзгертуге болатындықтан, айнымалылардың атаулары оның нені сақтайтынын және не үшін пайдаланылатынын түсіну үшін пайдаланылатындай болуы керек.

- Транслитерацияны қолданудың қажеті жоқ. Егер бағдарламашы айнымалыны жасағысы келсе, сақтау жасы бағдарламашы, Ол міндетті жазу Жок

«возраст» және «жас». Себебі, ІТ саласында ең көп қолданылатын тіл ағылшын тілі, ағылшын тіліндегі атауды бәрі түсінеді, кез келген жағдайда аудармашыны пайдалануға болады.

- Қолайлы ұзындық. Атау тек мәнін көрсетіп қана қоймай, тым ұзын атаулар кодтың көлемін ұлғайтады және оны қабылдауды нашарлатады;

Үшін құру жақсы атаулар пайдаланылады келесі әдістері :

- CamelCase (түйе ісі): бірінші сөз кіші әріптен басталады, ал келесі сөздер бас әріптен басталады. **Python тілінде CamelCase әдетте сынып атаулары үшін пайдаланылады** . Мысалы: WorkersOfFactory .

- Жылан ісі: атау таңба арқылы бөлінген сөздерден тұрады астын сызады "_", әрқайсысы сөз жазу бірге кішкентай хаттар, мысалы, Сәлем Әлем . **IN Python Жылан Іс пайдаланылады атаулар үшін функциялар , модульдер және айнымалылар** . Бұл жазу стилі жақсырақ қабылданады Қалай CamelCase. қоспағанда Бару, Ол Онда бар кейбір вариациялар, қай Python тілінде қолданылмайды: мысалы, «-» (жануарлар түрі) және әрбір сөзді бас әріппен жазу (Сұрыпталған массив-Оф-Аттар).

Резервтелген атаулар (кілт сөздер)

Нельзя назвать переменную именем, которое уже зарезервировано, как ключевое слово. В Python есть около 33 зарезервированных имен: def, True, False, break и так далее.

Түйінді сөздердің толық тізімін Python аудармашысында пәрменді теру арқылы көруге болады: `help("keywords")` .

Бос айнымалы

Айнымалы болып табылады сілтеме кейбір нысанда ол қамтуы мүмкін емес Ештеңе. Егер бағдарламашы қалайды жасаңыз бос (шартты түрде бос), ол оған None мәнін тағайындайды .

Ешбір мән анықталмаған және жоқ жағдайларда қолданылмайды. Ол кез келген булға, жолға немесе санға баламалы болмаса да, ол да нысан болып табылады.

Қажет болса, айнымалының мазмұнын келесідей тексеруге болады:

```

a=None
ifaisNone:
    print('значениеравноNone')else:
    print('другоезначение')

```

Оны, мысалы, айнымалыдағы ашық файлға көрсеткішті сақтаған кезде қолдануға болады. Ал бұл файлды жапқанда көрсеткішке None мәнін береміз. Болашақта деректерді жазу немесе оқу алдында айнымалы мәннің бос екенін тексеруге болады.

Түрлері деректер В тіл бағдарламалау Python

Тип данных(англ. Data type) - характеристика, определяющая множество значений и набор операций на этих значениях:

1. множестводопустимыхзначений,которыемогутприниматьданные, принадлежащие к этому типу;
2. набор операций, которые можно осуществлять над данными,принадлежащими к этому типу.

Аты	Сипаттама
Жоқ	белгісіз мағынасы айнымалы
бол	Логикалық түрі: True/False
int	бүтін сан түрі
жүзу	Нағыз түрі
кешен	Кешен түрі (Үшін кешен сандар)
көш	Жол түрі
Екілік тізімдер	
байт	Тізім байт
байт жиым	Байт массивтері
жад көрінісі	Арнайы нысандар Үшін қол жеткізу Кімге протокол буфері арқылы ішкі нысан деректері
Жиындар	
орнату	Бір топ
мұздатылған	Өзгермейтін бір топ
Сөздіктер	
дикт	Сөздік

Сандық түрлері

сандарды сақтау үшін пайдаланылады. Барлық сандық типтер өзгермейді, яғни айнымалының мәні өзгерсе, жаңа объект құрылып, ескісі жойылады.

IN Python Сонда бар осындай түрлері сандар, Қалай тұтас, шынайы Және кешен.

Олар анықталды кілт сөздер int, жүзу Және кешен.

- Бүтін сандар – мәндері int типімен берілген сандар. Бұл кез келген ұзындықтағы теріс және оң сандар болуы мүмкін Python аудармашысы оған шектеулер қоймайды;
- Нақты сандар қалқымалы типпен берілген. Оларды қарапайым немесе "e" арқылы көрсетуге болады: 1.252e+5. Себебі Екілік санау жүйесінде кейбір ондық өзгермелі нүктелі сандарды дәл көрсету мүмкін болмағандықтан, есептеулер кезінде қателер болуы мүмкін.
- Күрделі сандар күрделі типпен беріледі. Бұл 3 + 8j түріндегі күрделі сандар, мұндағы “3” – нақты бөлік, “8j” – елестетілген бөлік.

Арифметика әрекеттер

Python бағдарламалауда қолданылатын негізгі операторлар және олардың орындайтын операциялары төмендегі кестеде берілген.

Оператор	Операция
+	Қосу
-	Алу
*	Көбейту
/	Бөлім
%	Бөлім Авторы модуль
//	бүтін сан бөлу
**	Құрылыс В дәрежесі

Қосу, алу, көбейту және бөлу операторлары ешқандай қиындық тудырмауы керек. Олар сіз күткендей жұмыс істейді. Дегенмен, бірнеше операторларды пайдаланған кезде өрнектерді жақшалар арқылы топтастыру ұсынылады - жақша ішіндегі амалдар алдымен орындалады. Мысалы, өрнек

$$a = b * c - g \% e / f$$

есептеу реті бойынша толық түсінікті болмауы мүмкін. Бірақ оны келесі формада жазу арқылы түсіндіруге болады:

$$a = (b * c) - (g \% e) / f$$

% (модульді бөлу) операторы бір санды екінші санға бөліп, бөлімнің қалған бөлігін қайтарады. Бұл санның жұп немесе тақ екенін анықтау үшін өте пайдалы.

// (бүтін бөлу) операторы қалыпты бөлу , / сияқты жұмыс істейді , бірақ ондық бөлшектен кейін нәтижені алып тастайды.

** (көрсеткіш) операторы бірінші операндты екінші операндтың дәрежесіне көтереді.

Ұсынылатын әдебиеттер тізімі:

1. Бөрібаев Б. Алгоритмдеу, мәліметтер құрылымы және программалау тілдері. оқулық Алматы: Қазақ университеті, 2012
2. Султанова Б.К., Томилова Н.И. Деректер құрылымдары және бағдарламаларды әзірлеу процестері. Оқу құралы. Қарағанды: ҚарМТУ, 2015
3. Көксеген Ә.Е., Сейфуллина Ә.О. Алгоритмдеу және программалау тілдері Оқулық Алматы: ЖШС РПБК "Дәуір", 2011