**Процестер. Тармақталу және циклдар.**

1. **Желілік алгоритм ұғымы. Тармақталатын процесс, циклді процесс.**
2. **Арифметикалық операциялар**
3. **Деректер топтамалары**

Алгоритм – бұл міндетті шешудің қадам-қадамдық әдісі. Ол деректерді өңдеу, есептеулерді және онымен байланысты басқа компьютерлік және математикалық операцияларды жүзеге асыру үшін пайдаланылады.

Алгоритмдердің бірнеше типі бар: желілік, тармақталатын (тармақталу алгоритмі), циклдік.

**Желілік алгоритм** – бұл әрекеттер реттілігі оны орындау кезінде өзгермейтін, яғни әрекеттер қатаң реттілікте және бір рет орындалатын алгоритм типі.

Мысалы: X берілді. Z = Y1 / 2 және Z1 = 1 / Z, Y = X2 + 5 есептеу.

*X енгізіңіз*

*Y = X2 + 5 есептеңіз*

*Z = Y1 / 2* *есептеңіз*

*Z1 = 1 / Z* *есептеңіз*

*Z1 басып шығарыңыз,*

*Соңы.*

**Тармақталу алгоритмі** – бұл кем дегенде бір шартты қамтитын алгоритм. Тармақталу алгоритмі – бұл кейбір шарттарға байланысты өзгеретін қадамдар реттілігі. Екі/бірнеше ықтимал нұсқаның біреуі таңдалады.

Мысалы, функцияны есептеу үшін алгоритмді жасаңыз:

(X + 1, Y>0,)

Z =

(X + Y, Y<0.)

Тармақталатын алгоритмнің сөздік сипаттамасы келесі түрге ие:

*X* *енгізіңіз*

*Егер Y>0 болса, онда Z = X + 1*

*Егер Y<0* *болса*

*Z = X + Y* *басып шығарыңыз*

*Соңы.*

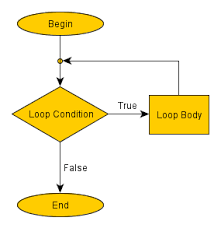
**Циклдік алгоритм**

Көпреттік қайталанатын әрекеттерді белгілеу үшін арнайы циклдік құрылымдар пайдаланылады. Осындай құрылым құрамында белгілі бір әрекеттер реттілігі үшін қайталаулар санын анықтау үшін қажетті шарт бар.

Циклдің негізгі блогында - теле циклда – қажетті есептеулер орындалады. Циклдің көмекші блоктары циклдік процесті ұйымдастырады: деректердің бастапқы мәнін және жаңа мәндерін орнатады, циклдік процестің соңғы жай-күйін тексереді.

Циклдік алгоритм қалаулы нәтижені алу мақсатында әр түрлі деректер бойынша барабар есептеулердің көп санын ықшам сипаттауға мүмкіндік береді. Алдын ала шарты бар циклдік құрылымдар бар (1-сур.) – алдын ала шарты бар циклдар және кейінгі шарты бар циклдік құрылымдар (2-сур.) – кейінгі шарты бар циклдар.

1-сур. Алдын ала шарты бар цикл. Цикл денесі бір рет іске қосыла алмайды.



ложь

правда

Тело цикла

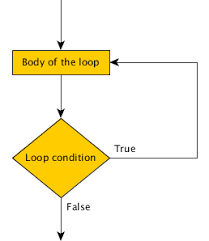
Условие цикла

Конец

Начало

|  |  |
| --- | --- |
| **Рус.** | **Қаз.** |
| Начало | Басы |
| Условие цикла | Цикл шарты |
| правда | ақиқат |
| Тело цикла | Цикл денесі |
| Конең | Соңы |
| ложь | жалған |

2-сур. Кейінгі шарты бар цикл. Цикл денесі кем дегенде бір рет орындалады.



Условие цикла

Тело цикла

ложь

правда

|  |  |
| --- | --- |
| **Рус.** | **Қаз.** |
| Условие цикла | Цикл шарты |
| правда | ақиқат |
| Тело цикла | Цикл денесі |
| ложь | жалған |

Алгоритмнің дұрыс екеніне көз жеткізу үшін, түпкілікті алгоритмнің орындалуын күтпей, оны тестілеу керек. Осы жағдайда алгоритмнің орындалуын тіркеу үшін алгоритм қадамдары, аргументтер, аралық мәндер осы қадамдағы нәтижелер және тексерілетін шарттар көрсетіледі.

Қадам-қадамдық орындау осы сәтке дейін қанша қадамның орындалғанын және барлық алгоритм қанша қадамда орындалатынын көру мүмкіндігін береді. Алгоритм қадамдарының нөмірленуі алгоритм сипаттамасында және оны орындау процесінде жеңіл бейімделуге мүмкіндік береді. Сонымен қатар, кез келген сәтте алгоритмнің қандай элементі (командасы) орындалатыны айқын. Бұл қателіктерді жоюға көмектеседі.

**Арифметикалық операциялар**

|  |  |  |
| --- | --- | --- |
| **Математикалық таңба** | **Сипаттамасы** | **Мысал** |
| + Қосу | Оператордан екі жақта мәндерді қосады. | a + b = 30 |
| - Азайту | Сол операндтан оң операндты азайтады. | a - b = -10 |
| \* Көбейту | Таңбадан екі жақтағы мәндерді көбейтеді | a \* b = 200 |
| / Бөлу | Сол операндты оң операндқа бөледі | b / a = 2 |
| % Модуль | Сол операндты оң операндқа бөліп, қалдықты қайтарады | b % a = 0 |
| \*\* Дәреже | Таңбалар бойынша экспоненциалдық (дәреже көрсеткіші) есептеуді орындайды | a\*\*b =20 дәрежесіндегі 10 |
| *//* | Толық бөлу – операндтарды бөлу, мұнда нәтиже бөлінді болып табылады, онда ондық нүктеден кейінгі сандарды алып тастайды. Бірақ егер операндтардың бірі теріс болса, нәтиже теріс болып алынады, яғни нөлден жуықталады (теріс шексіздікке қарай) - | 9//2 = 4 және 9.0//2.0 = **4**.**0**, -11//3 = -4. -11.003 =-**4.0** |

Ең кіші басымдығы бар ережеге қатысты ең жоғары басымдығы бар ереже:

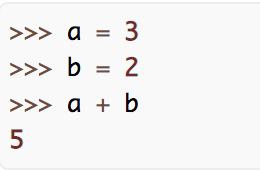
* Дөңгелек жақшалар әрқашан басымдыққа ие
* Экспоненцирлеу (дәрежеге шығару)
* Көбейту, бөлу және қалдық
* Қосу және азайту
* Сол жақтан оңға қарай

Қосу

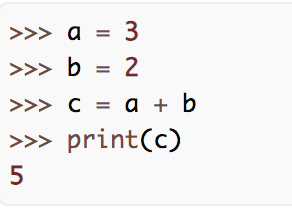
Сіз тікелей сандарды қоса аласыз



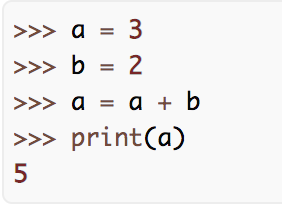
немесе айнымалылар, бірақ айнымалылар алдымен инициалдануы тиіс



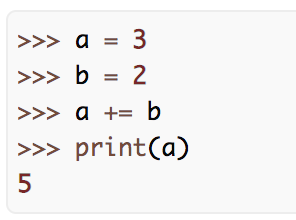
Қосу операциясының нәтижесі басқа айнымалыға берілуі тиіс.



Сіз толық немесе қысқартылған нұсқаны пайдалана аласыз, толық нұсқа

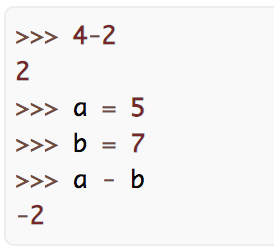


қысқартылған нұсқа

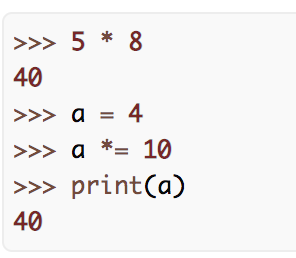


Қосу операциясы үшін қолданылатын барлық жоғарыда аталған нұсқалар барлық келесі операцияларға қатысты қолданылуы мүмкін.

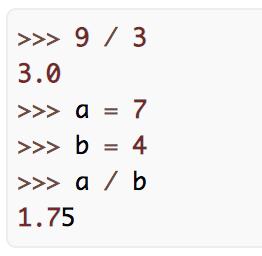
Азайту



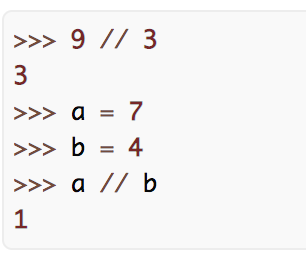
Көбейту



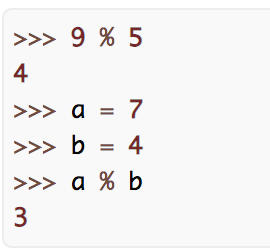
Бөлу



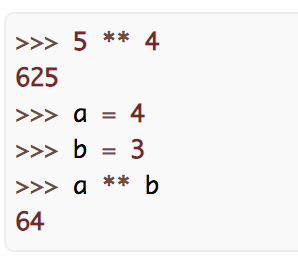
Бөлуден бүтін бөлікті алу



Бөлуден қалдықты алу.



Дәрежеге шығару

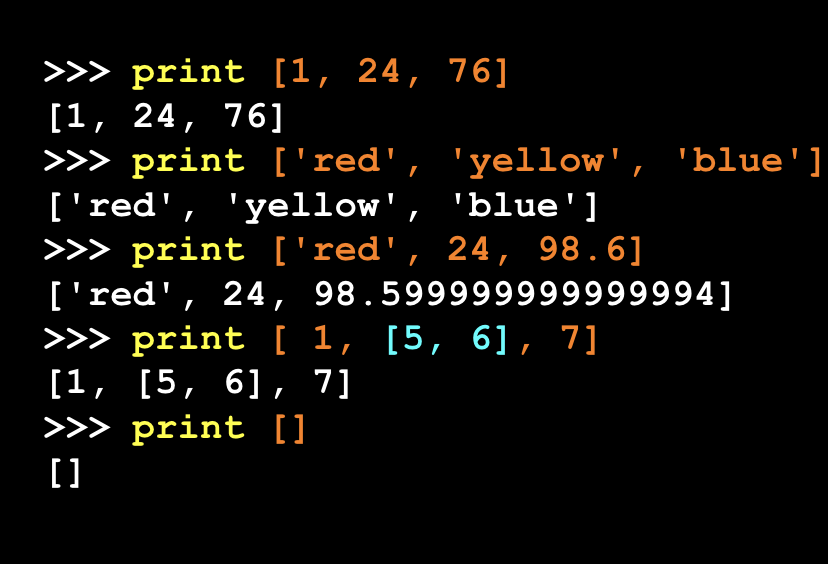


**Топтамалар**

Топтамалар бізге көп мәндерді бір «айнымалыға» орналастыруға мүмкіндік береді.

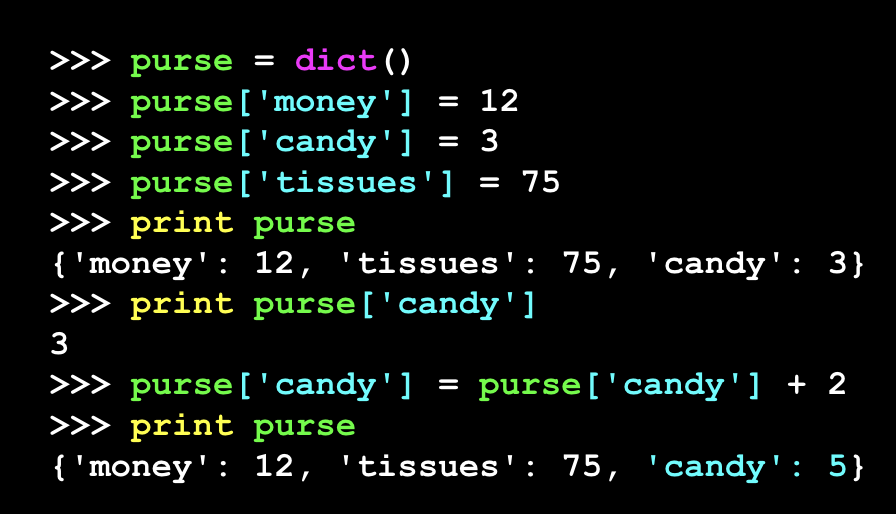
Топтаманың артықшылығы: біз мәндердің барлық жиынын бір қолайлы пакетте сақтай аламыз.

* Тізім константалары квадрат жақшаларға алынады, ал тізімдегі элементтер үтірлермен бөлінеді.
* Тізім элементі Python кез келген объектісі болуы мүмкін - тіпті басқа тізім де.
* Тізім бос болуы мүмкін

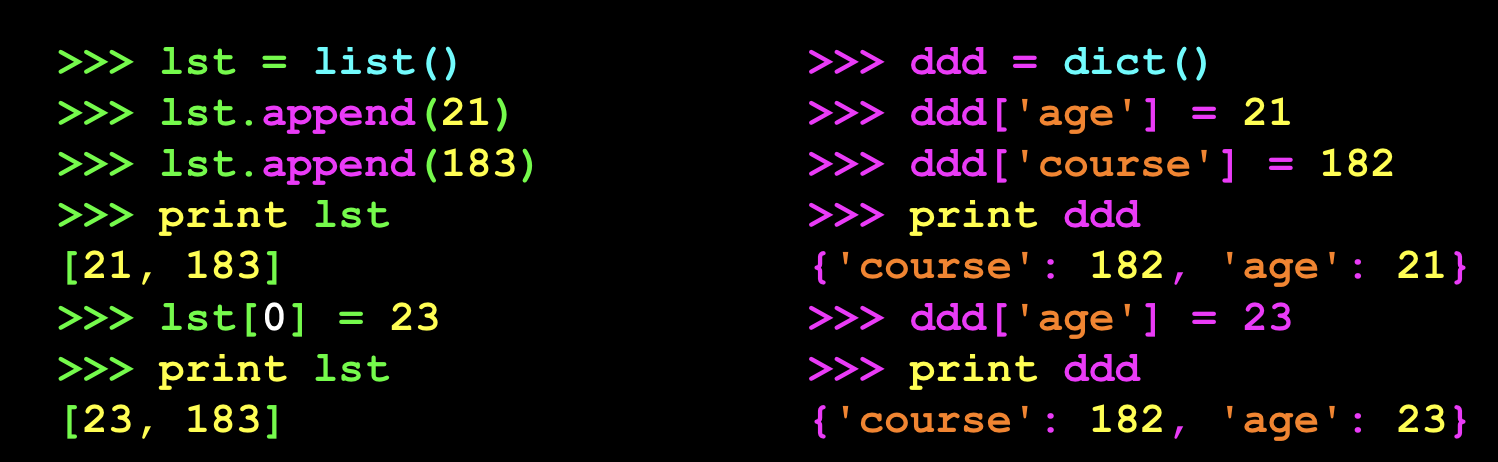


Сөздіктер

* Тізімдер элементтерін индекстеу тізімдегі олардың позициясын ескере отырып жүргізіледі
* Сөздіктерде, әйел сөмкелерінде сияқты, тәртіп жоқ
* Сондықтан біз сөздікке орналастыратын элементтер «іздеу тегі» көмегімен индекстеледі.

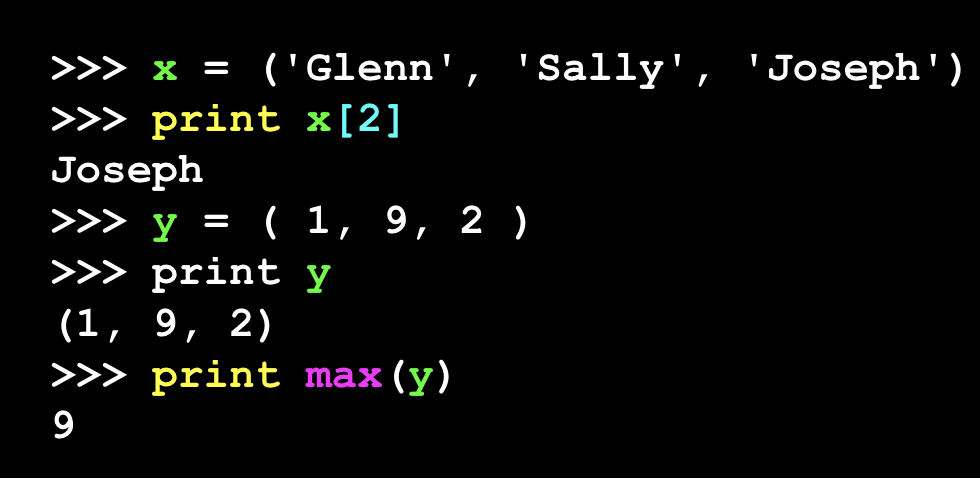


Мәндерді іздеу үшін олар сандар орнына кілттерді пайдаланатынын қоспағанда, сөздіктер тізімдерге ұқсас



Кортеждер

Кортеждер – бұл көбінесе тізім ретінде әрекет ететін тізбектің басқа түрі – олардың құрамында ( ) бастап индекстелетін элементтер бар.



Тізімге қарағанда, кортежді жасағаннан кейін сіз оның құрамын өзгерте алмайсыз – жол сияқты.

