**КОМПЬЮТЕРЛІК ОЙЛАУ**

**Компьютерлік ойлау** – бұл мәселелерді компьютерлік алгоритмдер арқылы шешу тәсілі. Бұл процесс нақты өмірдегі мәселелерді құрылымдау, модельдеу және жүйелі түрде шешу үшін қолданылатын әдістер мен тәсілдерді қамтиды. Компьютерлік ойлау тек бағдарламашыларға ғана емес, кез келген салада қолдануға пайдалы.

**Программа** – компьютер түсінетін тілде жазылған командалар мен нұсқаулар жиынтығы. Программа компьютерде белгілі бір тапсырманы орындауға бағытталған нақты команда тізбектерінен және нұсқаулықтардан тұрады. Программа жазу процесін **программалау**, ал программа жазатын адамды **программалаушы** деп атайды. Компьютер түсініп қабылдай алатын тілді **программалау тілі** дейміз. **Компьютерлік программа** – компьютерді басқаруға арналған логикалық құрылымы мен реттілігі бар командалар жиынтығы.

**Программалау тілі** – деректерді белгіленген ережелер бойынша өңдейтін адам мен компьютерді байланыстыратын формальді тіл.

Программалау тілі – жасанды тіл. Ол сөздік қорының аздығымен, жазу ережесінің қатаң сақталуымен ерекшеленеді. Процессор про­граммалау тілінде жазылған программаларды тікелей қабылдамайды. Ол үшін программаларды процессор тіліне аударатын **трансляторлар (аударғыш)** қажет.

Транслятордың 2 түрі бар. Олар:

**1. Компилятор.**

**2. Интерпретатор.**

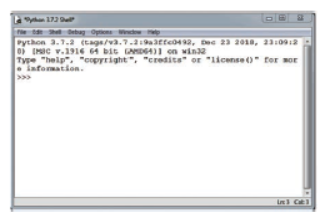
Программалау тілінде жазылған программаны мәшине кодына аудару үшін **компилятор** қолданылады. Мәшине тіліне аударылған программаны процессорда кезең-кезеңімен талдай отырып ретімен орындау үшін **интерпретаторлар** қолданылады. Компилятор мен интерпретатордың екеуі де программа. Компилятор мен интерпре­татордың айырмашылығы бар. Компилятор – программаны мәшине кодына толық аударып алып, орындайды. Интерпретатор – жоғарыда айтылғандай кезең-кезеңімен талдай отырып орындалады.

**Integrated Development Environment (IDE)** (Программа құрудың ин­теграцияланған ортасы) – программалық жасақтама жасауға арналған программалар кешені. IDE жүйесінің бірнеше құрамдас бөлігі бар.

**Python программалау тілін компьютерге орнату**

Тілді компьютерге орнату үшін интернеттегі іздеу серверлерінің іздеу алаңына Python деп жазу жеткілікті. Бұл программалау тілі тегін болғандықтан, оны кез келген сайттан көшіру тегін. Мысалы, Python-ның **3.7.2** нұсқасын *https://www.python.org/downloads/* сайтынан көші­ріп алуға болады. Оны барлық қолданбалы программалар сияқты орна-тады. Программалау тілі компьютерге орнатылған соң, оны іске қосуға болады.

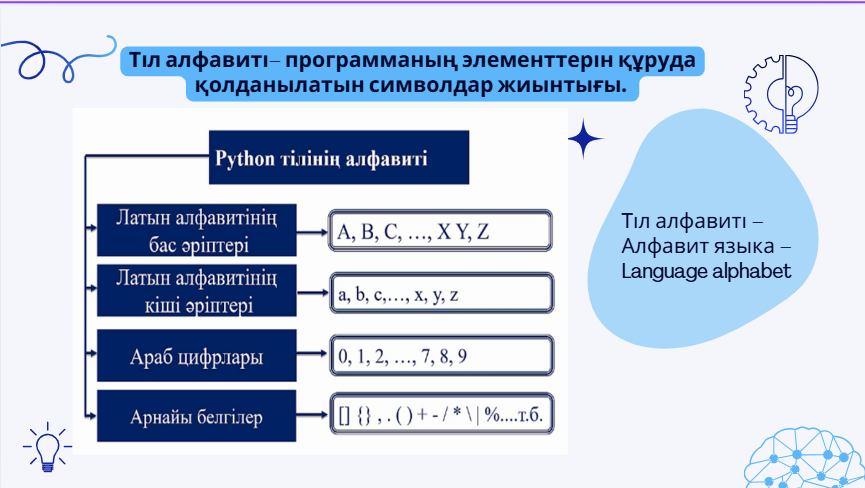
Программаны іске қосу үшін **Пуск – Все программы – Python** (Іске қосу – Барлық программалар – Python) – командалар тізбегін орын­даймыз. Программа іске қосылғанда оның негізгі терезесі пайда болады.



Python-ның программалар құрудың интеграцияланған ортасы (IDE) қабықша (**оболочка; shell**) ретінде берілген. **Python 3.7.2** те­резесінің тақырып жолында «Shell» сөзі – «қабықша» (оболочка) мағынаcын береді. Біз программалау тілімен әрекеттесуді дәл осы қабықша арқылы жүзеге асырамыз. «Қабықша» пайдаланушының графикалық интерфейсі деп те аталады. Программа мәзірінде **File** (Файл)**, Edit** (Түзету), **Shell** (Қабықша), **Debug** (Жөндеу), **Options** (Опциялар), **Windows** (Терезе), **Help** (Көмек) сияқты бөлімдер орналасқан. Программа терезесіндегі **>>>** таңбалары пайдаланушының деректерін енгізуге, программамен жұмыс істеуге мүмкіндік береді.

**ТІЛ АЛФАВИТІ. СИНТАКСИС**

Кез келген тілді үйрену оның алфавитін үйренуден басталады. Әріптерден сөз, сөздерден сөйлем құралатыны сияқты програм­малау тілінің де өз алфавиті бар. Программалау тілін үйренуді бастаған кезде оның командаларын жазу үшін қандай таңбалар қолдану керектігін анықтау керек. Python тілінің алфавиті төмен­дегідей құрылымға ие.



**Идентификаторлар (айнымалылар атауы)** – әріппен басталатын цифрлар мен әріптер тізбегі. Идентификатор латын әріптерінен, төмен­гі сызу таңбасы \_k1 және 0 мен 9 арасындағы цифрлардан құралады. Мысалы, **A, s25, a1b88, tri\_plus\_dv, zzz.** Python-ның әріптер регистрі үлкен және кіші әріптерге ажыратылып берілген. Мысалы, komp1 және KOMP1 – екеуі екі түрлі айнымалыға жатады. Программада қолданыла­тын айнымалыларды **идентификатор** деп атайды.

**Айнымалы** – мәні мен типі бар шама. Программамен жұмыс барысында айнымалылардың мәнін өзгертіп отыруға болады.

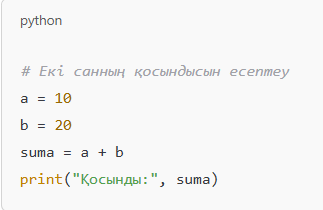
**Қызметтік сөздер.** Кейбір атауларды программалау барысында ай­нымалы ретінде қолдануға болмайды. Мұндай сөздер **қызметтік сөздер** деп аталады. Мысалы: for, if, class, def және т.б.

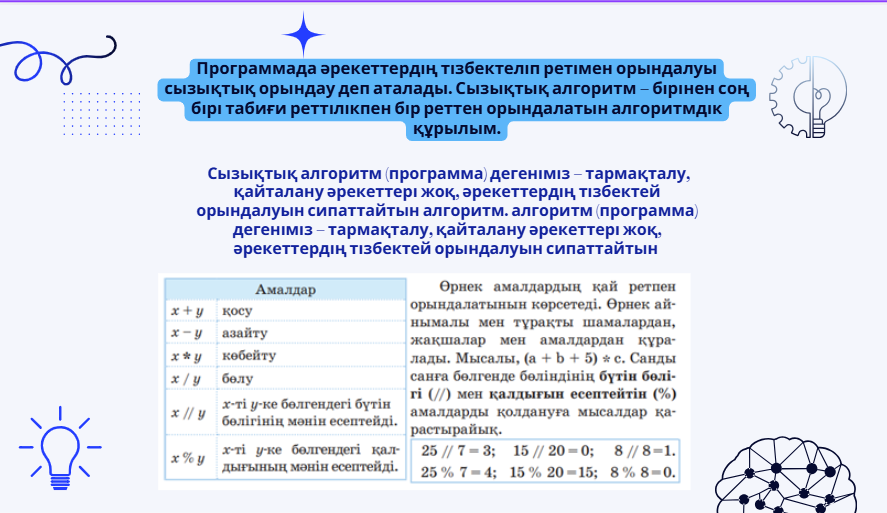


**Python тілінің синтаксисі**

1. Программадағы жолдардың соңына нүкте мен үтір қою міндетті емес.
2. Бірнеше команданы бір жолда нүктелі үтір (;) арқылы жазуға болады.
3. Программа жазу барысында басқа программалау тілдеріндегідей begin..end және {..} жақшалар қолданылмайды. Керек болған жағ­дайда ондай жақшалардың орнына 4 бос орын немесе табуляция қойылады.

Python-да толыққанды программа жасау үшін программа терезесін­де **File (Файл) – New File** командаларын орындап, жаңа программа жазу алаңын дайындап аламыз. Бұл бетке жазылған программа кодын орын­дау үшін **Run – Run Module** командаларын басамыз.





**Дерек** – өңделмеген әртүрлі фактілер, цифрлар. Өңдеудің нәтиже­сінде белгілі бір мағынаға ие болған дерек ақпаратқа жатады. Төменгі сыныптарда ақпаратты ұсынудың мәтіндік, сандық, дыбыстық және графикалық түрлерімен танысқанбыз. Программалау тілдерінің көме­гімен деректерді өңдеу үшін оларды типтерге жіктейді. Сандық ақпарат сандық деректерден құрылады. Математикада сандар осінде орналасқан сандарды натурал, бүтін, нақты және т.б. түрлерге жіктейді. Ал мәтіндік ақпарат – символдық және жолдық деректерден құралады. Деректердің жіктелуі оларды компьютерде өңдеу кезінде маңызды рөл атқарады.

Тұрақты, айнымалы және өрнектер қабылдай алатын мәндерді, олар- мен орындауға болатын амалдардың жиынтығын **деректер типі** деп атайды.

**СЫЗЫҚТЫҚ АЛГОРИТМДЕРДІ ПРОГРАММАЛАУ**

Программалау барысында қолданылатын негізгі үш алгоритм түрі бар. Оларға сызықтық, тармақталу және циклдік алгоритмдер жатады. Олардың арасындағы ең қарапайымы – сызықтық алгоритм.

Сызықтық алгоритмдердің командаларын бір реттен тізбекпен орындай отырып, нәтиже аламыз.

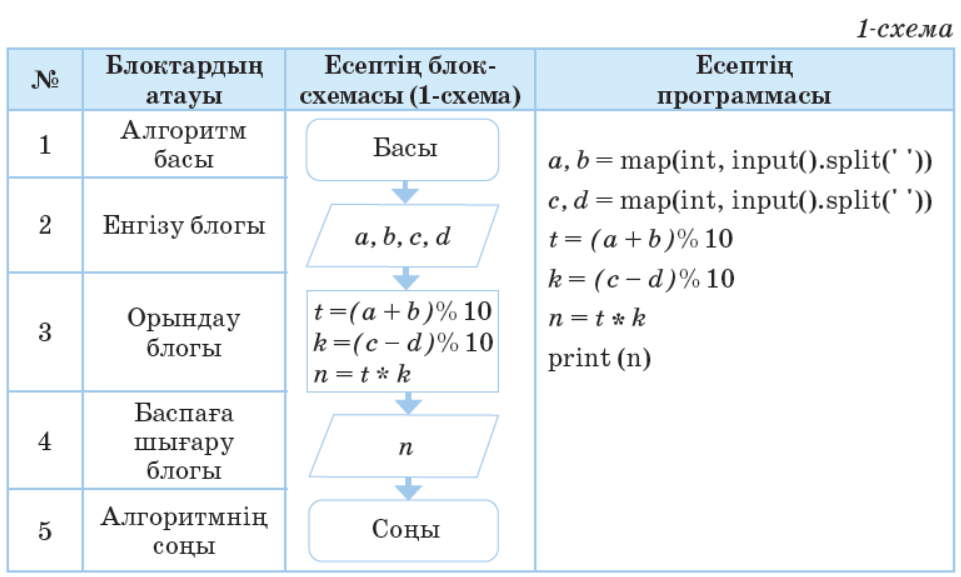
Қалған екі алгоритм түрімен жоғарғы сыныптарда танысамыз. Өт­кен тақырыптарда сызықтық алгоритмге сипаттама берілген. Жоғарыда берілген сипаттаманы сызықтық алгоритмді сипаттаудың тағы бір түрі­не жатқызуға болады.

**Сызықтық құрылымдағы программаны құру қадамдары**

1. Программада қолданылатын негізгі деректер мен олардың типте­рін анықтау. Айнымалыларға атау беру.
2. Нәтиже ретінде нені шығару керек және оның типі қандай болу ке­ректігін анықтау. Нәтижеге шығаратын айнымалыға атау беру.
3. Деректерді енгізетін, есептейтін және нәтижені экранға беретін алгоритмдерді дайындау.
4. Құрастырған программаның жұмысын тексеру мақсатында, де­ректердің түрлі мәндерін енгізіп, программаны сынақтан өткізу (тестілеу).

**Сызықтық программаның құрылымы**

**Мысалы:** *a, b, c* және *d* бүтін сандары берілген. *а* және *b* сан- дарының қосындысы мен *с* және *d* сандары айырмасының соңғы цифр- ларының көбейтіндісін табатын программа құрастыр *(1-схема)*.



**1-практикалық тапсырма**

*S* теңгені құны 1, 5, 10, 50 және 100 теңгелік монеталар көмегімен ең аз нұсқада қалай төлеуге болады? Тапсырманың блок-схемасын сызып, программасын құрастыр. *S* (0 < S < 100000) – бүтін сан. Тапсырманың блок-схемасы *(2-схема)* мен программа коды берілген *(1-код)*.

