**Файлдармен жұмыс**

*Файл* – сыртқы есте сақтау құрылғыларында (магниттік немесе flash жадыларда) жазылған және мәліметтерді өңдеу, тасымалдау кездерінде біртұтас күйде қарастырылатын мәліметтер жиыны.

Файлдармен жұмыс істеу үшін оларды, алдымен, ашу керек, яғни файл туралы мәліметті – атын, адресін программаға белгілі ету қажет.

файл\_айнымалысы = open(файл аты [, access\_mode][, buffering])

**file\_name** – жолымен (орналасу орнымен) немесе жолсыз жазылған файл аты

**access\_mode** – файлды ашу түрі, файл қандай жұмыс үшін ашылу керек екенін көрсетеді (оқу, жазу және жалғастырып жазу т.с.с.).

**buffering** – буффер көлемі, 0 – ге тең болса буффер қолданылмайды, теріс сан болса буфер көлемін оперциялық жүйе белгілейді.

**Файлды ашу түрлері (access\_mode): r –** оқу үшін.

**w -** жазу үшін. Жаңа файл құрылады. Егер тиісті орында ондай файл бар болса, ол қайтадан құрылады, яғни бұрынғы файл өшіріледі.

**a – жалғастырып жазу үшін.**

**b –** екілік файл белгісі, егер **b** көрсетілмесе файл мәтіндік есептеледі.

**a+, w+ -** оқу және жазу үшін.

Файл ашу белгілері жазылмаса, онда екілік файлды оқу үшін ашылды деп есептеледі (rb).

file айнымалысының қасиеттері

**file.closed** – True егер файл жабылса, әйтпесе False

**file.mode** – файлдың ашылу түрі

**file.name** – файлдың аты

Файлды оқу немесе оған деректер енгізу үшін, алдымен оны ашу керек. Жұмыс аяқталған соң файлға қатысты ресурстарды босату үшін, оны жабамыз.

Python-да файлдармен жүргізілетін амалдар төмендегідей ретпен орындалады:

1. Файлды ашу.

2. Оқу немесе жазу.

3. Файлды жабу.

Python-да файлдардың екі типін қарастыруға болады.

1. Белгісіз ұзындықтағы мәтіндік файлдар. Бұл файлдарда мәтіндер сақталады.

2. Екілік (бинарлық) файлдар. Бұл типтегі файлдарда суреттер, дыбыстар, бейнелердің (видео) екілік кодтары сақталады.

**Файлды жасау мен ашу**

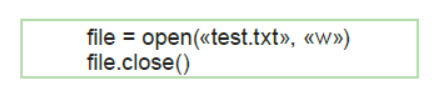
Python-да файл жасауды қарастырайық. Мысалы, мәтіндік файл жасау керек болса, онда оны екі түрлі жолмен жасайды:

1. Төменгі сыныптарда үйренген әдіспен (тінтуірдің көмегімен

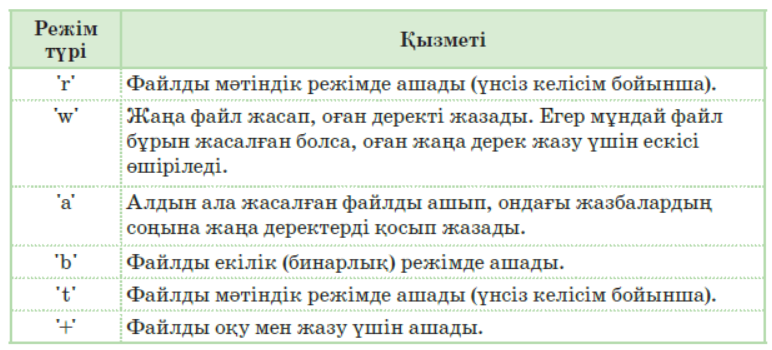
контекстік мәзір арқылы... Python орналасқан бумаға **\*.txt** типіндегі файлды жасайды.

2. Python-дағы **open ()** функциясының көмегімен автоматты түрде жасауға болады. Ол үшін **open ()** функциясында файлдың атауы мен файлды өңдеу параметрін көрсету керек.

Төмендегі программа кодында **file** айнымалысы жаңа мәтіндік файлға сілтеме жасалғанын көрсетіп тұр. Программаны іске қосқанда осы программа сақталған бумада **test.txt** мәтіндік файлы пайда болады.



Егер **test.txt** файлы бумада алдын ала жасалған болса, онда программа жаңа файл жасамайды, бұрынғы жасалғанымен жұмыс істей береді. **Open ()** функциясының екі маңызды параметрі бар. Оларға файл атауы мен функцияның жұмыс режімі кіреді



Деректерді файлдан оқыту үшін, **read ()** әдісін пайдаланамыз. Жолдық командаларға **read ()** және **write ()** командалары жатады. Сол үшін қосынды мен көбейтіндінің мәндерін **str** әдісі арқылы жолдық шамаға айналдырып, файлға жазамыз. Нәтижелерді жеке- жеке қатарға жазуда **'\n'** әдісін қолданамыз.

**Файлды жабу**

Файлмен жұмыс (оқу/жазу) аяқталған соң, файл жабылу керек. Файл close() әдісімен жабылады. файл\_айнымалысы.close();

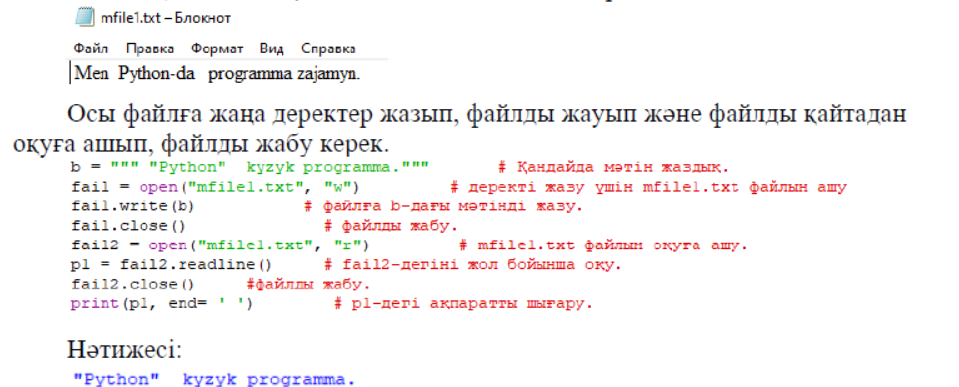
Егер файл келесі түрде ашылса, ол міндетті түрде жабылады.

with open(файл\_аты, ‘w’) as файл\_айнымалысы:

файлмен жұмыс операторлары

**Файлға жазу**

Файлға деректі жазу үшін **write() әдісі** қолданылады. Файлға бірнеше жолды жазу үшін writelines() әдісі қолданылады.Бұл әдіс жол соңы белгілерін файлға жазбайды. Мысалы, mfile1.txt файлына төмендегідей ақпарат жазылған.



**Файл көрсеткіші.**

Файлмен жұмыс кезінде деректі оқу немесе жазу орнын анықтаушы файл көрсеткіші болады.

**Файл көрсеткішінің әдістері**.

Файл көрсеткішінің мәнін *tell()* әдісі арқылы білуге болады*.*

Файл көрсеткішінің орнын *seek(offset[, from])* әдісі арқылы белгілеуге болады*.* offset аргументі файл көрсеткішінің неше байтқа ауысу керек екенін анықтайды.

**Практикалық жұмыс.**

**№1.** Файлды жол бойынша оқу.

