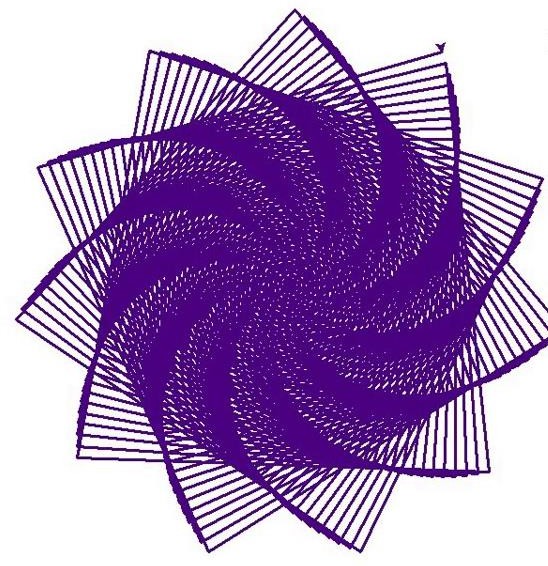
«ПРОГРАММА ҚҰРУДЫҢ КІРІКТІРІЛГЕН ОРТАСЫ ЖӘНЕ PYTHON ТІЛІНДЕ ОҚЫТУ»

*9 сыныпқа арналған таңдау курстың бағдарламасы*

**

# 2025 ж.

1

Авторлық бағдарламада информатика сабақтарын ұйымдастыру негізінде оқушылар қабілетін дамытуға бағытталған технологияларды пайдалану жолдарын қарастырады.

Python программалау тілінің негіздері және жұмыс істеу тәртібі, оның ақпаратты өңдеу мәні, қазіргі ақпараттық технологиялардың даму ісінде осы бағдарламаның рөлі, оқушылар қабілетін дамытуға бағытталған инновациялық технологияларды қолданудың мән-маңызы сабақ жоспарларын жазу барысында қолданылған.

9 сыныпқа қосымша көмек құралы ретінде арналған.

# Мазмұны

3

[Түсінік хат 4](#_TOC_250001)

[Курстың базалық мазмұны 6](#_TOC_250000)

Күнтізбелік-тақырыптық жоспар 13

Қысқа мерзімді жоспар 15

«Python программалау тілінің негіздері» 15

Пайдаланылған әдебиет 137

# Түсінік хат

Қазақстан Республикасында білім беруді және ғылымды дамытудың 2020-2025 жылдарға арналған мемлекеттік бағдарламасында сабақтастығын қамтамасыз ету шеңберінде оқулықтар мен оқу-әдістемелік кешендер әзірлеудің, сараптаудың және басып шығарудың жаңа жүйесі енгізілетін болады. Сондықтан да кез келген мұғалім мектеп бағдарламасы көлемінде ғана білім беріп қоймай, рухани жаңару түріндегі бағдарламалар арқылы баланың жан-жақты дамуын қамтамасыз етуі керек.

"Python программалау тілінің негіздері" тақырыбындағы 9 сыныптарға арналған авторлық таңдау курсының информатика пәні бойынша әзірленген бағдарлама жаңартудың нормативтік-құқықтық, психологиялық-педагогикалық негіздерін, оқыту мен тәрбиелеудің жаңа әдістерін және технологияларын ескере отырып білім берудің басым бағыттарына, білім алушылардың білім алу қажеттіліктеріне жасалған талдау нәтижелеріне сәйкес әзірленді. Қазақстан Республикасының Мемлекеттік білім стандарты мен информатика пәнінің негізіне сүйене отырып бұл бағдарлама оқушылардың жас ерекшелігін ескере отырып құрастырылған. Жоғары сынып оқушылары, соның ішінде 9 сынып оқушылары үшін жасалған оқу бағдарламасы.

Таңдау курстың жұмыс бағдарламасы "Python программалау тілінің негіздері" 9 сыныптар үшін базалық деңгейге арналған оқыту бағдарламасы. Жұмыс бағдарламасында жалпы орта білім беру деңгейінде білім алушылардың жас және психологиялық ерекшеліктері, пәнаралық байланыстар ескеріледі.

**Бағдарламаның өзектілігі:** Python программалау тілінің негіздері бойынша құрылымдық бағдарламалаудың негізгі түсініктерін қалыптастыру, оқушылардың логикасын дамыту

* **Бағдарламаның мақсаты**: білім алушылардың өзін-өзі дамытуға және үздіксіз білім алуға дайындығын қалыптастыру; **Бағдарламаның міндеттері:**
* білім алушылардың өзін-өзі дамытуға және үздіксіз білім алуға дайындығын қалыптастыру;
* білім беру қызметін жүзеге асыратын ұйымның дамытушы білім беру ортасын жобалау және құрастыру;
* білім алушылардың белсенді оқу-танымдық қызметін арттыру;
* білім алушылардың жеке, жас, психологиялық, физиологиялық ерекшеліктері мен денсаулығын ескере отырып, білім беру қызметін құру.
* **Бағдарламаның жаңашылдығы:** цифрландыру, жоғары деңгейлі бағдарламалау тілдері арасында Python тілінің орнын анықтап, тәжірибеге енгізу.

# Күтілетін нәтижелер:

* IDE интерфейсінің мүмкіндіктерін қолданады;
* Python тіліндегі сызықтық алгоритмдерді жазады;
* Мәліметтер типтерін жіктейді;
* Python тілінде ұсынылған бағдарлама құрылымының ерекшеліктерін анықтайды;
* дайын модульдерді пайдалану мүмкіндіктері мен шектеулерін анықтайды;
* операцияның не екенін, операндты және олардың сипаттамаларын талдайды;
* құрылымдық және құрылымданбаған шамалардың іргелі айырмашылықтарын жіктейді;
* Python-ға кіретін математикалық функцияларды қолданады;
* оларға кіруі мүмкін барлық атрибуттардың арифметикалық және логикалық өрнектерінің мысалдарын жаза біледі;

# Білім алушылардың іскерлік, дағдыларға қойылатын талаптар: Оқушылар білуі тиіс:

* берілген форматта мәтіндік файлдарды оқиды жіне жазады;
* Python ортасында негізгі алгоритмдік есептерді шешеді;
* Python-да процедуралар мен функцияларды сипаттау ережелерін және процедуралық қоңырауды құру ережелерін анықтайды;
* ресми, жергілікті және жаһандық айнымалылар арасындағы негізгі айырмашылықтарды анықтайды;
* рәсімдердегі сипаттамалардың қолданылу саласын анықтайды;
* процедура мен функцияны қалыптастырудың негізгі әдістерін меңгереді;

# Бағдарлама құрылымы

Курс бағдарламасы 8 бөлімнен, 34 тақырыптан тұрады. Берілген жоспарда көрсетілген сабақ тақырыптары түгел қамтылған. Әр тақырып бойынша қысқа мерзімді жоспарлар, теориялық мәліметтер, практикалық жұмыстар, тапсырмалар дайындалған.

Курстың маңызы, мазмұны, құрылымы кестемен көрсетілді. Әр тақырып пен бөлімге түсініктеме берілді.

# Курстың базалық мазмұны

|  |  |  |
| --- | --- | --- |
| **№** | **Мазмұны** | **Түсініктеме** |
| 1. | **I бөлім**  Python тілі туралы жалпы ақпарат  **(2 сағат)** | Питон тілінің атқаратын қызметі.  Питон - объектіге бағытталған жоғары деңгейлі, динамикалық программалау тілі, ол интерпретатор арқылы жұмыс істейді. Оның құрамындағы мәліметтер құрылымы мен олардың динамикалық түрде типтелуі қолданбалы программалар жазуға өте  ыңғайлы болып саналады |
| 2. | Python тіліндегі бағдарлама құрылымы.  **(1 сағат)** | Питон нұсқалары (версиялары)   * Питон тілінің кең пайдаланылатын нұсқалары 2.7   … 3.6.   * Кез келген нұсқасын пайдалана аласыз, оны өзіңіз қалауыңызға болады. * Осы нұсқалардың айырмашылығы неде? * Олардың СИНТАКСИСІ және ішкі функциялары әртүрлі болып келеді   Python 2.7.\* Python 3.6.\* Питон нұсқаларын қолдану   * Питон 2.x нұсқалары ескірген, қазіргі қолданыстағысы соңғы 3.х нұсқалары. * Бірақ Питон 3.x нұсқаларының аздаған кемшіліктері бар, кітапханалары үлкейген сайын, сүйемелденуі қиындап барады, Linux және Mac жүйелері әлі де 2.x нұсқаларын пайдаланып келеді * 3.x нұсқалары компьютер жадын өте тиімді пайдаланады |
| 3. | **II бөлім Қолдану белгілері (4 сағат)** | **Шартты өрнектер**  Кейбір операциялар шартты өрнектерді білдіреді. Бұл операциялар екі операндты қабылдап, boolean типіндегі логикалық мәнді қайтарады. Тек екі логикалық мән ғана бар: True (ақиқат) және False (жалған).  **Салыстыру операциялары**  Python келесі салыстыру операцияларын қолдайды:   **==**  Егер екі операнд тең болса, True мәнін әйтпесе False мәнін қайтарады.   * **!=**   Егер екі операнд тең ЕМЕС болса True мәнін қайтарады, әйтпесе False мәнін қайтарады.   * **>** (үлкен)   Егер бірінші операнд екіншісінен үлкен болса True мәнін қайтарады. |

|  |  |  |
| --- | --- | --- |
|  |  | * **<** (кіші)   Егер бірінші операнд екіншісінен кіші болса True мәнін қайтарады.   * **>=** (үлкен не тең)   Егер бірінші оператор екіншісінен үлкен не тең болса True мәнін қайтарады.   * **<=** (кіші немесе тең)   Егер бірінші оператор екіншісінен кіші не тең болса True мәнін қайтарады. |
| 4. | Математикалық функциялар | Python тіліндегі бүтін сандар, басқа бағдарламалау тілдеріне қарағанда ұзақ арифметиканы қолдайды  (бірақ ол жадыдан көп орын талап етеді) |
| 5. | Пернетақтадан деректерді енгізу. | Python-да қандай ортада бағдарламалар құруға және іске қосуға болатындығын анықтағаннан кейін, сіз осы тілдің негіздерін біле бастай аласыз. Ең қызығы, алғашқы қарапайым бағдарламаларды кез-келген күрделі құрылымдар мен бағдарламаның құрамын білместен жазуға болады (Паскальдағыдай).  Бұл сабақта біз Python-да енгізу-шығару операцияларымен, сондай-ақ компьютер экранында арифметикалық амалдардың жылдам шығарылуымен танысамыз. |
| 6. | Сандармен қарапайым әрекеттерге  арналған тапсырмалар | Санау жүйелері. Сандар тек ғана ондық санау жүйесінде ғана емес сонымен қатар, басқа да санау жүйелерінде беріледі. |
| 7. | **III бөлім Шартты ұсыныстар (6 сағат)** | Python тілінде операторлардың орындалу приоритеті бар, яғни бірінші кезекте орындалатыны көбейту және бөлу операторлары орындалады. Ал қосу мен азайту амалдарының приоритеті томен. Егер операцияның проритеті бірдей болса, онда солдан  оңға қарай кезекпен орындалады. |
| 8. | Күрделі шартты өрнектер (and, or, not логикалық амалдары). | Аргументтерінің барлығы **true** болса, **And** функциясы  **true** мәнін қайтарады.  Аргументтерінің кез келгені **true** болса, **Or**  функциясы **true** мәнін қайтарады.  **Not** функциясы аргументі **false** болса **true** мәнін қайтарады; аргументі **true** болса, ол **false** мәнін қайтарады.  Бұл функциялар дәл Excel бағдарламасындағыдай жұмыс істейді. |
| 9. | Шартты оператор. Балама орындау | Операторлар мен оларды қолдану Оператор Аты Түсіндірмесі + Қосу Екі нысанды  қосады. Азайту Екі санның айырмасын табады. Егер |

|  |  |  |
| --- | --- | --- |
|  |  | бірінші сан жазылмаса, онда оны нөлге тең деп есептейді. \* Көбейту Екі санды көбейтеді немесе  жолды берілген сан бойынша қайталайды. |
| 10. | Шартты оператормен есептерді шешудің  мысалдары. | Шартты нұсқауларды жазудың барлық мүмкін нұсқаларын көрсетеді.  Шартты өрнектерді жазудың 4 нұсқасы бар. Оларды мысалдармен қарастырады. |
| 11. | Python тілінде тармақтауды жүзеге асыру. | Көбінесе бағдарламаларда белгілі бір шарт орындалған кезде ғана кодтың бір бөлігін орындау қажет болады. Мысалы, егер пайдаланушы қате деректерді енгізсе, қате туралы хабарды көрсету  керек. |
| 12. | **IV бөлім Циклдер (7 сағат)** | Мен кездестірген әр бағдарламалау тілінде цикл құрылымы бар. Көптеген тілдерде осындай дизайн бар. Python әлемінде циклдердің екі түрі бар:  Цикл үшін While циклі  Мен цикл екіншісіне қарағанда танымал екенін байқадым. Циклдар біз бірнеше рет бір нәрсе  жасауымыз керек болған жағдайда қолданылады. |
| 13. | Цикл операторы while. | Алдын ала берілген шарт бойынша қайталануды жүзеге асыратын бұл оператор While (*оқылуы:* уаил, *аударылуы:*  әзірше) және DO (*оқылуы:* ду, *аударылуы:* орындау) түйінді сөздері қолданылып орындалады. |
| 14. | Циклдарды қолдану мысалдары. | Алдын ала берілген шарт бойынша қайталануды жүзеге асыратын бұл оператор While (*оқылуы:* уаил, *аударылуы:*  әзірше) және DO (*оқылуы:* ду, *аударылуы:* орындау) түйінді сөздері қолданылып орындалады. |
| 15. | For параметрі бар цикл операторы | For циклі бірізділік бойынша Итерация үшін қолданылады (тізім, tuple, dict, set немесе str).  Бұл басқа бағдарламалау тілдерінде for кілт сөзін, басқа объектіге бағытталған бағдарламалау тілдеріндегідей Итератор әдісін қолдануға ұқсас.  For циклі арқылы біз list, tuple, set және т. б. әр элемент үшін әрекеттер жиынтығын орындай аламыз. |
| 16. | Циклдардағы циклдар. | Неліктен Python-да циклдарды қолдану керек? Циклдар артық кодты азайтады. 1-ден 10-ға дейінгі сандарды басып шығаруға болатын сценарийді қарастырыңыз, екі мүмкіндік бар:   * Жұп сандарды басып шығару үшін 10 баспа операторын пайдаланыңыз. * 10 қайталауға арналған цикл ішіндегі бір баспа мәлімдемесі. |

|  |  |  |
| --- | --- | --- |
| 17. | Randrange функциясы. | Параметрлері  start-ауқымның бастапқы нүктесі. Ауқымға қосылады.  Әдепкі мәні 0  stop-диапазонның соңғы нүктесі. Ауқымға қосылады.  Әдепкі мәні 1  қадам-ауқым нүктесінің қадамы. Қадам болады. Қайтару мәні  Бұл функция кездейсоқ элементті берілген ауқымнан  қайтарады. |
| 18. | Random функциясы. | Жұмыстың басында бағдарламаға random модулін импорттау қажет. Осыдан кейін ғана оны толығымен  пайдалануға болады. |
| 19. | **V бөлім Функциялар құру (5 сағат)** | Бұнда атаулы мен анонимді функциялар және def, return және lambda интрукциялары қарастырылады. Сонымен қатар, функцияның міндетті және міндетті емес аргументтері, еркін санды аргументтері бар функциялар қарастырылады.  Атаулы функциялар, def инструкциясы. Python тіліндегі функция аргументтер қабылдайтын және мәнді қайтаратын нысан болып табылады. Әдетте функция def инструкциясы көмегімен анықталады. |
| 20. | Жергілікті және ғаламдық айнымалылар. | Ғаламдық айнымалылар   * Бағдарламаның кез-келген жеріне қол жеткізе алатын айнымалылар ғаламдық айнымалы деп аталады. * Айнымалыларды бағдарламаның басында немесе ғаламдық кілт сөзді қолдану арқылы атауға болады. |
| 21. | Lambda нұсқаулары. Функцияларды қолдана отырып есептерді шешудің мысалдары. | Lambda-бұл Python және басқа бағдарламалау тілдеріндегі анонимді функцияларды шақыруға арналған құрал. Бұл көптеген адамдарға ештеңе айтпайды және оның қалай жұмыс істейтінін түсіндірмейді, сондықтан Мен сізге Lambda өрнектерінің жұмыс механизмін айтамын.  Барлығы өте қарапайым.  Мысалды қарастырайық. Мысалы, шеңбердің ауданын белгілі радиуста есептейтін функцияны жазу керек. |
| 22. | Рекурсивті функциялар. | Рекурсивті функция — бұл өзін-өзі қоздыратын функция.  Рекурсивті функцияны шақыру және оған бүтін санды беру арқылы сіз сол санның факториалын аласыз (n!). Факториалдар туралы қысқаша  Санның факторлық мәні-әр алдыңғы санға 1-ге көбейтілген Сан. |
| 23. | Есептеу | Фибоначчи сандары – әрбір келесі мүшесі алдыңғы |

|  |  |  |
| --- | --- | --- |
|  | факториала. Фибоначчи Сандары. | екі мүшесінің қосындысына тең болатын 1, 1, 2, 3, 5, 8, 13, … қайталама сан тізбегінің (Фибоначчи қатары) элементтері. Фибоначчи сандарының рекурренттік қатынастары  F 0 = 0 , F 1 = 1 , F n = F n − 1 + F n − 2 , n ⩾ 2.  {\displaystyle F\_{0}=0,\qquad F\_{1}=1,\qquad  F\_{n}=F\_{n-1}+F\_{n-2},\quad n\geqslant 2.}  арқылы беріледі. Фибоначчи сандарын 1202 жылы  италиялық математик [Леонардо Пизанский](https://kk.wikipedia.org/wiki/%D0%9B%D0%B5%D0%BE%D0%BD%D0%B0%D1%80%D0%B4%D0%BE_%D0%9F%D0%B8%D0%B7%D0%B0%D0%BD%D1%81%D0%BA%D0%B8%D0%B9) ([Фибоначчи](https://kk.wikipedia.org/wiki/%D0%A4%D0%B8%D0%B1%D0%BE%D0%BD%D0%B0%D1%87%D1%87%D0%B8)) тапқан. |
| 24. | **VI бөлім**  **Жолдар-таңбалар тізбегі (4 сағат)** | Жолдардағы таңбаларға қол жеткізу индекстеу операциясына негізделген-жолға немесе жолға сілтеме жасайтын айнымалының атауынан кейін шаршы жақшада қажетті таңбалардың  позицияларының нөмірлері көрсетіледі. |
| 25. | Оператор in. String модулі. | Синтаксис:  x in s  x not in s  In операторы және not in теріске шығару элементтің дәйектілігін тексереді. X in s өрнегі, егер x s мүшесі болса, ШЫН мәнін алады, әйтпесе жалған болады.  X not in s өрнегі s-де x элементінің болуын жоққа шығарады.  Set/frozenset жиынтықтарының барлық кірістірілген реттілігі мен түрлері Бұл әрекетті қолдайды, сонымен қатар сөздікте берілген кілттің бар-жоғы тексерілетін  сөздік. |
| 26. | Тізбектің барлық түрлеріне арналған (операторлар (жолдар, тізімдер, түйіндер). | Кодта көрсетілген таңбалар жол ретінде түсіндірілуі үшін оларды тырнақшаға орау керек. Мұны істеудің 4 әдісі бар:  – одинарные бағамдар   * Қос тырнақша * үштік бір тырнақша (пішімдеуді сақтай отырып, көп жолды мәтін) * үш есе Қос тырнақша (пішімдеуді сақтай отырып, көп жолды мәтін) |
| 27. | Мәселелерді жолдармен  шешудің мысалдары. | Қосу немесе біріктіру операциясы екі немесе одан да көп жолды бір жолға бүктейді. Дәлелдер тек жолдар бола алады. |
| 28. | **VII бөлім Деректердің күрделі түрлері**  **(6 сағат)** | Тізімдегі мәндерді in операторы арқылы іздеу   * in операторы керекті мән тізімде бар/жоқ екендігін тексереді. Мәнді іздейтін in операторы қосылған өрнек форматы: мән in тізім |

|  |  |  |
| --- | --- | --- |
|  | Тізімдер. Тізім түрі (тізім).  Индекстер. | * Бұл форматтағы мән – ізделетін мән, тізім – ішінен мән ізделетін тізім. Егер мән табылса, өрнек ақиқат,   әйтпесе жалған мәнін береді. |
| 29. | Тізімдерге арналған операторлар. | Тізімдегі мәндерді in операторы арқылы іздеу   * in операторы керекті мән тізімде бар/жоқ екендігін тексереді. Мәнді іздейтін in операторы қосылған өрнек форматы: мән in тізім * Бұл форматтағы мән – ізделетін мән, тізім – ішінен мән ізделетін тізім. Егер мән табылса, өрнек ақиқат, әйтпесе жалған мәнін береді. |
| 30. | Range функциясы. Тізімдер: мәселелерді шешу мысалдары. | Python функциясы range деп қалай жұмыс істейді? Қарапайым сөзбен айтқанда, range () берілген ауқым аясында бірқатар сандарды құруға мүмкіндік береді. Функцияларды қанша дәлелдейтіндігіңізге байланысты, сіз осы сандар қатарының қай жерде басталып, аяқталатынын, сондай-ақ екі санның арасындағы айырмашылық қаншалықты үлкен  болатынын шеше аласыз. |
| 31. | Матрицалар. Python тізімінің генераторлары. | Бұл бөлім жеткілікті дәрежеде ескірген. Қазір модуль numpy деп аталады. Жалпы, Numeric-те болған барлық numpy-да қол жетімді, бірақ аттар сәйкес келмеуі мүмкін. Numeric NumPy/oldnumeric сияқты қол жетімді. Сіздің құжаттама.  Numeric Python - көптеген сандық қосымшалар үшін қажетті көп өлшемді массивтерді есептеу үшін бірнеше модуль. Numeric модулі MatLab, Octave (MATLAB аналогы), APL, J, S+, ID сияқты пакеттер мен жүйелердің Python мүмкіндіктерін енгізеді. Пайдаланушылар Numeric-ті оңай және ыңғайлы табады. Кейбір Python синтаксистік мүмкіндіктері (кесіктерді пайдаланумен байланысты) Numeric үшін  арнайы әзірленген. |
| 32. | Кортеждер. Кортеждерді  тағайындау. | Кортеж — бұл өзгермейтін тізім. Кортежді оны жасағаннан кейін ешқандай жолмен өзгерту мүмкін  емес. |
| 33. | Python тіліндегі жиындар. | Процедуралық бағдарламалау стилі-бұл процедураларды қолдануға негізделген бағдарламалау:  Процедуралық бағдарламалау-бұл императивті тілде бағдарламалау, онда дәйекті орындалған операторларды ішкі бағдарламаларға, яғни кодтың үлкен тұтас бірліктеріне тілдің тетіктерін қолдана отырып жинауға болады.  Процедуралық бағдарламалау-бұл 1940 жылдары Фон Нейман ұсынған дәстүрлі компьютерлік |

|  |  |  |
| --- | --- | --- |
|  |  | архитектураның көрінісі.  Процедуралық бағдарламалаудың теориялық моделі- бұл Тьюринг машинасы деп аталатын дерексіз есептеу жүйесі. |
| 34. | **VIII бөлім Бағдарламалау стилі және бағдарламаларды жөндеу (3 сағат)** | Құрылымдық бағдарламалау туралы толық ақпарат Python құрылымдық бағдарламалау бетінде бар.  Python бағдарламалау стильдері процедуралық, объектіге бағытталған, Функционалды және құрылымдық бағдарламалау стильдері болып табылады, өйткені Python бағдарламалаудың барлық  төрт стилін қолдануға мүмкіндік береді. |
| 35. | Процедуралық бағдарламалау стилі. | Процедуралық бағдарламалау стилі-бұл процедураларды қолдануға негізделген бағдарламалау:  Процедуралық бағдарламалау-бұл императивті тілде бағдарламалау, онда дәйекті орындалған операторларды ішкі бағдарламаларға, яғни кодтың үлкен тұтас бірліктеріне тілдің тетіктерін қолдана  отырып жинауға болады. |
| 36. | "Python тілінде бағдарламалау" курсы бойынша  сынақ | Өткен тақырыптары бойынша толықтай қорытынды сынақ алу. |

**9 сыныпқа арналған «Python программалау тілінің негіздері» тақырыбындағы факультатив курсқа күнтізбелік-тақырыптық жоспары**

**Жылына 34 сағат, аптасына 1 сағат**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Тақырыптың мазмұны** | **Сағат саны** | **Мерзімі** |
| **I бөлім. Python тілімен танысу: 2 сағат** | | | |
| 1 | Python тілі туралы жалпы ақпарат | 1 |  |
| 2 | Python тіліндегі бағдарлама құрылымы. | 1 |  |
| **II бөлім. Қолдану белгілері: 4 сағат** | | | |
| 3 | Өрнектер. Операциялар. | 1 |  |
| 4 | Математикалық функциялар | 1 |  |
| 5 | Пернетақтадан деректерді енгізу. | 1 |  |
| 6 | Сандармен қарапайым әрекеттерге арналған  тапсырмалар. | 1 |  |
| **III бөлім. Шартты ұсыныстар: 5 сағат** | | | |
| 7 | Логикалық өрнектер мен операторлар. | 1 |  |
| 8 | Күрделі шартты өрнектер (and, or, not логикалық  амалдары). | 1 |  |
| 9 | Шартты оператор. Балама орындау | 1 |  |
| 10 | Шартты оператормен есептерді шешудің мысалдары. | 1 |  |
| 11 | Python тілінде тармақтауды жүзеге асыру. | 1 |  |
| **IV бөлім. Циклдер: 7 сағат** | | | |
| 12 | Цикл ұғымы. Цикл денесі. | 1 |  |
| 13 | Цикл операторы while. | 1 |  |
| 14 | Циклдарды қолдану мысалдары. | 1 |  |
| 15 | For параметрі бар цикл операторы | 1 |  |
| 16 | Циклдардағы циклдар. | 1 |  |
| 17 | Randrange функциясы. | 1 |  |
| 18 | Random функциясы. | 1 |  |
|  | **V бөлім. Функциялар: 5 сағат** |  |  |
| 19 | Функциялар құру | 1 |  |
| 20 | Жергілікті және ғаламдық айнымалылар. | 1 |  |
| 21 | Lambda нұсқаулары. Функцияларды қолдана отырып  есептерді шешудің мысалдары. | 1 |  |
| 22 | Рекурсивті функциялар. | 1 |  |
| 23 | Есептеу факториала. Фибоначчи Сандары. | 1 |  |
| **VI бөлім. Жолдар-таңбалар тізбегі: 4 сағат** | | | |
| 24 | Құрама деректер түрі-жол. Индекс бойынша қатынау. | 1 |  |
| 25 | Оператор in. String модулі. | 1 |  |
| 26 | Тізбектің барлық түрлеріне арналған операторлар  (жолдар, тізімдер, түйіндер). | 1 |  |
| 27 | Мәселелерді жолдармен шешудің мысалдары. | 1 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **VII бөлім Деректердің күрделі түрлері 5 сағат** | | | |
| 28 | Тізімдер. Тізім түрі (тізім). Индекстер. | 1 |  |
| 29 | Тізімдерге арналған операторлар. | 1 |  |
| 30 | Range функциясы. Тізімдер: мәселелерді шешу  мысалдары. | 1 |  |
| 31 | Матрицалар. Python тізімінің генераторлары. | 1 |  |
| 32 | Кортеждер. Кортеждерді тағайындау. Python тіліндегі  жиындар. | 1 |  |
| **Бағдарламалау стилі және бағдарламаларды жөндеу**: **2 сағат** | | | |
| 33 | Бағдарламалау стилі. Процедуралық бағдарламалау  стилі. | 1 |  |
| 34 | "Python тілінде бағдарламалау" курсы бойынша сынақ | 1 |  |

**Сабақтың қысқа мерзімді жоспары**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 1 сабақ** | **Курс атауы:**  «Python программалау тілінің  негіздері» | **Сыныбы: 9**  **Күні: Мұғалім:** | | | |
| **Тақырыбы:** Python тілі туралы жалпы ақпарат | | | | | |
| **Оқу мақсаты:** | Python тілі туралы жалпы ақпаратымен таныстыру | | | | |
| **Сабақ мақсаты** | Оқушылар тақырыпты түсінеді, тапсырманы орындайды, өмірге  білімдерін қолданады. | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқушын ың әрекеті** | **Баға лау** | **Ресурс тар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds05.infourok.ru/uploads/ex/112a/000f5369-09815022/hello_html_m6003f41a.jpg | | Психоло гиялық ахуалға берілген өлеңді дауыстап айтып, тілді жаттықт ыру және жаңа сабаққа назар аудару. | Мұға лімнің ауызша баға лауы | Жұмыс дәптері  Суретт ер Карточ калар Презен тация |
| **Ортасы** | **Мағынаны тану**  Питон тілінің атқаратын қызметі • Питон - объектіге бағытталған жоғары деңгейлі, динамикалық программалау тілі, ол интерпретатор арқылы жұмыс істейді.  Оның құрамындағы мәліметтер құрылымы мен олардың динамикалық түрде типтелуі қолданбалы программалар жазуға өте ыңғайлы болып саналады  Питон тілінің даму жолы   * Питон – жалпы мақсаттағы жоғары деңгейдегі программалау тілі. Ол 1991 жылы пайда болған, оны алғаш рет 1991 ж. жасап шығарған Guido van Rossum (Гидо | | Оқушыл ар сұраққа жауап береді, жұмысты берілген түріне қарай орындай ды | Отш ашу арқы лы бір- бірін баға лау. |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ван Россум) деген маман болатын. 2018 жылдың шілде айында Ван Россум 30 жылдай істеген көшбасшылық қызметінен кетті.  Python тілі неге кең таралған? • Бұл тіл интернетте тегін таратылады: • Кітапханасы өте бай, мүмкіндіктері мол • Питон барлық аймақтарда қолданыс тапты: • Web қосымшаларын жасайды • Мәліметтер базасымен жақсы жұмыс істей алады • Үстелдегі дербес компьютерлерде жеңіл программаланады • Мәліметтерді талдау ісінде де қолданылады • Бейнелер мен суреттерді жақсы өңдейді • Объектіге бағытталған программалауды сүйемелдейді   * Интерактивті түрде жұмыс істеуге де ыңғайлы болып табылады.   **Тапсырма**  Питон тілінің жұмысын бейнежазбадан бақылап, постер қорғау. |  | Бас барм ақ арқы лы өзін- өзі баға лау.  **«Ма дақт ау»** |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://fs00.infourok.ru/images/doc/181/207766/img9.jpg |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 2 сабақ** | **Курс атауы:** «Python  программалау тілінің негіздері» | **Сыныбы: 9**  **Күні: Мұғалім:** | | | |
| **Тақырыбы:** Python тіліндегі бағдарлама құрылымы. | | | | | |
| **Оқу мақсаты:** | Python тіліндегі бағдарлама құрылымы туралы түсіндіру. | | | | |
| **Сабақ мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін өмірде  қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқушы ның әрекеті** | **Бағалау** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/08fb/0000cc4a-d204bcbd/img2.jpg | | Психологиялық  ахуалға берілген өлеңді дауыста п айтып, тілді жаттықт ыру және жаңа сабаққа назар аудару. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану**  Питон нұсқалары (версиялары)   * Питон тілінің кең пайдаланылатын нұсқалары 2.7 … 3.6. * Кез келген нұсқасын пайдалана аласыз, оны өзіңіз қалауыңызға болады. * Осы нұсқалардың айырмашылығы неде? * Олардың СИНТАКСИСІ және ішкі функциялары әртүрлі болып келеді Python 2.7.\* Python 3.6.\*   Питон нұсқаларын қолдану   * Питон 2.x нұсқалары ескірген, қазіргі қолданыстағысы соңғы 3.х нұсқалары. | | Оқушыл ар сұраққа жауап береді, жұмыст ы берілген түріне қарай орындай ды | Отшаш у арқылы бір- бірін бағалау  ÐÐ°ÑÑÐ¸Ð½ÐºÐ¸ Ð¿Ð¾ Ð·Ð°Ð¿ÑÐ¾ÑÑ ÑÐ°Ð»ÑÑ |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | * Бірақ Питон 3.x нұсқаларының аздаған кемшіліктері бар, кітапханалары үлкейген сайын, сүйемелденуі қиындап барады, Linux және Mac жүйелері әлі де   2.x нұсқаларын пайдаланып келеді   * 3.x нұсқалары компьютер жадын өте тиімді пайдаланады   **тапсырма**  келесі сайт мәліметтерімен танысып шығыңдар: [https://wiki.python.org/moin/Python2orPyt](https://wiki.python.org/moin/Python2orPython3) [hon3](https://wiki.python.org/moin/Python2orPython3)  кластер құру. |  | Бас бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  https://ds04.infourok.ru/uploads/ex/0e62/00027f61-35aaaf54/img13.jpg**Рефлексия** |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№ 3 сабақ** | **Курс атауы:** «Python  программалау тілінің негіздері» | **Сыныбы: 9**  **Күні: Мұғалім:** | | | | |
| **Тақырыбы:** Өрнектер. Операциялар. | | | | | | |
| **Оқу мақсаты:** | Питон программасындағы өрнектер мен операциялар туралы түсінік  беру. | | | | | |
| **Сабақ мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін өмірде  қолданады | | | | | |
| **Жоспар** | | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқушын ың әрекеті** | **Бағалау** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/0a8b/001553c2-48fcd72a/img3.jpg | | Психологиялық ахуалға берілген өлеңді дауыстап айтып, тілді жаттықт ыру және жаңа сабаққа назар аудару. | Мұғалім нің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану Шартты өрнектер**  Кейбір операциялар шартты өрнектерді білдіреді. Бұл операциялар екі операндты қабылдап, boolean типіндегі логикалық мәнді қайтарады. Тек екі логикалық мән ғана бар: True (ақиқат) және False (жалған).  **Салыстыру операциялары** Python келесі салыстыру операцияларын қолдайды:   **==** Егер екі операнд тең болса, True мәнін әйтпесе False мәнін қайтарады. | | Оқушыл ар сұраққа жауап береді, жұмысты берілген түріне қарай орындай ды | Отшашу арқылы бір-бірін бағалау.  ÐÐ°ÑÑÐ¸Ð½ÐºÐ¸ Ð¿Ð¾ Ð·Ð°Ð¿ÑÐ¾ÑÑ ÑÐ°Ð»ÑÑ |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | * **!=** Егер екі операнд тең ЕМЕС болса True мәнін қайтарады, әйтпесе False мәнін қайтарады. * **> (үлкен)** Егер бірінші операнд екіншісінен үлкен болса True мәнін қайтарады. * **< (кіші)** Егер бірінші операнд екіншісінен кіші болса True мәнін қайтарады. * **>= (үлкен не тең)** Егер бірінші оператор екіншісінен үлкен не тең болса True мәнін қайтарады. * **<= (кіші немесе тең)** Егер бірінші оператор екіншісінен кіші не тең болса True мәнін қайтарады.   Салыстыру операторларына мысалдар:  a = 5  b = 6  result = 5 == 6 print(result) # False print(a != b) # True print(a > b) # False print(a < b) # True bool1 = True  bool2 = False  print(bool1 == bool2) # False **Логикалық операциялар** Құрама шартты өрнектерді құру үшін логикалық операциялар қолданылады. Python-да келесі логикалық операторлар қолданылады:   * **and** (логикалық көбейту) егер екі өрнек те True мәніне тең болса, True қайтарады.   1. age = 22   2. weight = 58   3. result = age > 21 and weight   4 == 58 |  | Бас бармақ арқылы өзін-өзі бағалау.  **«Мадақ тау»** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | * **not** (логикалық «емес»)   Егер өрнек False-қа тең болса, True мәнін қайтарады  age = 22 isMarried = False  print(not age > 21) # False print(not isMarried) # True Егер бір өрнекте бір мезгілде  бірнеше логикалық операторлар орындалсағ олардың приоритеті әр түрлі екенін ескеру керек.  Алдымен **not** операторы, содан кейін **and** және соңында **or** операторы орындалады.  age = 22 isMarried = False weight = 58  result = weight == 58 or isMarried and not age > 21 # True  print(result)  Мында есептеулер келесідей жүргізіледі:   1. not age > 21 тең False 2. isMarried and False (not age > 21) тең False 3. weight == 58 or False (isMarried and not age > 21) тең True Есептеулердің ретін ауыстыру үшін біз жақшаларды қолдана аламыз:    1. age = 22    2. isMarried = False    3. weight = 58    4. result = (weight == 58 or    5. isMarried) and not age > 21 #   False print(result) |  |  |  |

1

2

3

4

1

2

3

4

5

|  |  |
| --- | --- |
|  | print(result) # True |
|  | age = 22 |
| 1 | weight = 58 |
| 2 | isMarried = False |
| 3 | result = age > 21 and weight |
| 4 | == 58 and isMarried |
| 5 | print(result) # False, так как |
|  | isMarried = False |

|  |  |
| --- | --- |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Тапсырма**  Өрнектерге мысал келтіріп орындау. |  |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://ds04.infourok.ru/uploads/ex/0d0d/00010af4-2268d125/img13.jpg |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 4 сабақ** | **Курс атауы:** «Python  программалау тілінің негіздері» | **Сыныбы: 9 Күні: Мұғалім:** | | | |
| **Тақырыбы:** Математикалық функциялар | | | | | |
| **Оқу мақсаты:** | Математикалық функциялар туралы ақпаратпен таныстыру. | | | | |
| **Сабақ мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін өмірде  қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекет і** | **Бағалау** | **Ресурст ар** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/00e6/0011527d-e94b2917/img2.jpg | Психо логиял ық ахуалғ а берілге н өлеңді дауыст ап айтып, тілді жатты қтыру және жаңа сабаққ а назар аудару. | Мұғалім нің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
|  | **Мағынаны тану**   |  |  | | --- | --- | | Математикалық операциялар | | | x + y | Қосу | | x – y | Айырма | | x \* y | Көбейту | | x / y | Бөлу | | x // y | Бөлгендегі бүтін бөлік | |  |  |  |
| **Ортасы** | 2-кестенің жалғасы   |  |  | | --- | --- | | x % y | Бөлгендегі қалдық | | –x | Санның белгісін  ауыстыру | | abs(x) | Сан модулі | | divmod(x, y) | (x // y, x % y)жұп | | x \*\* y | Дәрежеге шығару | | pow(x, y[, z]) | xy модулі бойынша(егер модулі берілсе) | | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай орынд айды | Отшашу арқылы бір-бірін бағалау.  ÐÐ°ÑÑÐ¸Ð½ÐºÐ¸ Ð¿Ð¾ Ð·Ð°Ð¿ÑÐ¾ÑÑ ÑÐ°Ð»ÑÑ | Бас бармақ арқылы өзін-өзі бағалау  **«Мадақ тау»** |
|  | Тағы бір айта кететіні, Python тіліндегі бүтін сандар, басқа бағдарламалау тілдеріне қарағанда ұзақ арифметиканы қолдайды (бірақ ол жадыдан көп орын талап етеді). Мысалы келесі 22-суретте бағдарламаның бөлігінің листингі көрсетілген.  **>>>** 255 + 34  289  **>>>** 5 \* 2  10  **>>>** 20 / 3  6.666666666666667  **>>>** 20 // 3  6  **>>>** 20 % 3  2  **>>>** 3 \*\* 4  81  **>>>** pow(3, 4)  81  **>>>** pow(3, 4, 27)  0  **>>>** 3 \*\* 150  36998848503512697292470078245  16966441864731003897229738151  84405301748249  Сурет 18 Бүтін сандарға қолданылған |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Сурет 18 Бүтін сандарға қолданылған операциялар  Биттік операциялар. Бүтін сандармен де биттік операцияларды орындауға болады.  Кесте 3  Биттік операциялар тізімі  x | y Биттік немесе  Биттік «алып  x ^ y тастағыш» немесе |  |  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | x & y Биттік *и*  x << n Солға биттік жылжу  x >> y Оңға биттік жылжу  ~x Биттер  инверсиясы  Қосымша әдістер. int.bit\_length() - белгісі мен жетекші нөлдерін алып тастағандағы екілік жүйеде санды беру үшін қажетті биттер саны. Төменде көрсетілгендей:  >>>  >>> n = -37  >>> bin(n) '-0b100101'  >>> n.bit\_length() 6 int.to\_bytes(length, byteorder, \*, signed=False) – осы санды беретін байттар жолын қайтарады. Ол келесі 23- суретте көрсетілді.  >>> (1024).to\_bytes(2, byteorder='big')  b'\x04\x00'  >>> (1024).to\_bytes(10, byteorder='big') b'\x00\x00\x00\x00\x00\x00\x00\x 00\x04\x00'  >>> (-1024).to\_bytes(10, byteorder='big', signed=True) b'\xff\xff\xff\xff\xff\xff\xff\xff\xfc  \x00'  >>> x = 1000  >>> x.to\_bytes((x.bit\_length() //  8) + 1, byteorder='little') b'\xe8\x03'  Сурет 19 Int.to\_bytes әдісі |  |  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Келесі classmethod int.from\_bytes(bytes, byteorder, \*, signed=False) әдісі аталған байттар жолынан сандарды қайтарады.  >>> int.from\_bytes(b'\x00\x10', byteorder='big')  16  >>> int.from\_bytes(b'\x00\x10', byteorder='little')  4096  >>> int.from\_bytes(b'\xfc\x00', byteorder='big', signed=True)  -1024  >>> int.from\_bytes(b'\xfc\x00', byteorder='big', signed=False)  64512  >>> int.from\_bytes([255, 0, 0], byteorder='big')  16711680  Санау жүйелері. Сандар тек ғана ондық санау жүйесінде ғана емес сонымен қатар, басқа да санау жүйелерінде беріледі. Мысалы, компьютерде екілік санау жүйесі қолданылады. 19 саны екілік санау жүйесінде 10011 деп бейнеленеді. Кейбір жағдайда бір санау жүйесінен екіншіге көшу қажет болады. Оны орындауға Python тілі бірнеше функцияларды ұсынады:   * int([object], [санау жүйесінің негізі]) – ондық санау жүйесіндегі бүтінге айналдыру. Бұл жерде үнсіз келісім бойынша ондық санау жүйесі қолданылады, бірақ негізін 2-ден 36-ға дейін таңдай отырып кез келген санау жүйесінде беруге болады. * bin(x) – бүтін санды екілік жолға айналдыру. * hex(х) - бүтін санды он алтылық жолға айналдыру. * oct(х) - бүтін санды сегіздік жолға айналдыру.   Келесі 20-суретте мысалдар көрсетілген.  >>> a = int('19') # жоды санаға айналдыру |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | >>> b = int('19.5') # жол бүтін сан болып табылмайды  Traceback (most recent call last): File "", line 1, in  ValueError: invalid literal for int() with base 10: '19.5'  >>> c = int(19.5) # жылжымалы нүктесі бар санға қолданылғанда бөлшек бөлігін алып тастайды  >>> print(a, c) 19 19  >>> bin(19) '0b10011'  >>> oct(19) '0o23'  >>> hex(19) '0x13'  >>> 0b10011 # сандық тұрақтылықарды ослай да жазуға болады.  19  >>> int('10011', 2)  19  >>> int('0b10011', 2)  19 |  |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://ds04.infourok.ru/uploads/ex/096a/000bdcc1-5f006a63/img23.jpg |  |  |  |

|  |  |  |
| --- | --- | --- |
| **№ 5 сабақ** | **Курс атауы:** «Python программалау тілінің  негіздері» | **Сыныбы: 9 Күні:**  **Мұғалім:** |
| **Тақырыбы:** Пернетақтадан деректерді енгізу. | | |
| **Оқу мақсаты:** | Пернетақтадан деректерді енгізу бойынша ақпарат беру | |
| **Сабақ мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін өмірде  қолданады | |
| **Жоспар** | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | **Оқуш ының әрекет і** | **Бағалау** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/0016/00144b65-1141ed41/img1.jpg | Психо логиял ық ахуалғ а берілг ен өлеңді дауыст ап айтып, тілді жатты қтыру және жаңа сабаққ а назар аудару  . | Мұғалім нің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану**  Python-да қандай ортада бағдарламалар құруға және іске қосуға болатындығын анықтағаннан кейін, сіз осы тілдің негіздерін біле бастай аласыз. Ең қызығы, алғашқы қарапайым бағдарламаларды кез-келген күрделі құрылымдар мен бағдарламаның құрамын білместен жазуға болады (Паскальдағыдай). Бұл сабақта біз Python-да енгізу-шығару операцияларымен, сондай-ақ компьютер экранында арифметикалық амалдардың жылдам шығарылуымен танысамыз.  **Пайдаланушы енгізу**  Есіңізде болсын, пайдаланушы енгізгеннің бәрі таңбалар айнымалысы, тіпті егер ол тек сан енгізсе де.  Егер бізге қолданушыдан онымен бірге | Оқуш ылар сұраққ а жауап береді, жұмыс ты берілг ен түріне қарай орынд айды | Отшашу арқылы бір-бірін бағалау.  ÐÐ°ÑÑÐ¸Ð½ÐºÐ¸ Ð¿Ð¾ Ð·Ð°Ð¿ÑÐ¾ÑÑ ÑÐ°Ð»ÑÑ  Бас бармақ арқылы өзін-өзі |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | келесі әрекеттерді орындау үшін нөмір енгізу қажет болса (мысалы, математикалық операциялар үшін), онда біз келесі амалды орындауымыз керек: жол нәтижесін санға түрлендіру.  Біріншіден, пайдаланушы енгізген және басқаннан кейін деректерді енгізеді **Енгізіңіз**, ол енгізген жол туралы ақпарат сандық форматқа өзгертіледі.  Бұл функция бізге көмектеседі. **int ()**. Ол кез-келген деректерді сандық бүтін сан пішіміне түрлендіреді. Бірақ егер сіз кейбір таңбаларды сандық форматқа түрлендіргіңіз келсе, аудармашы сізге қате жібереді.  **Хаттар сандар емес!**  Қарапайым жолмен біз «тілде енгізу және шығару» туралы таныстық **Питон**« **Файлдарды жазу және оқу**  Open () функциясы файл нысанын қайтарады және көп жағдайда екі аргументпен қолданылады: open (*file\_name*, *режимін таңдаңыз*) . Бірінші параметр - файлдың атауы бар жол. Екіншісі - файлды қалай пайдалану керектігін сипаттайтын бірнеше таңбалардан тұратын тағы бір жол.  Параметр мәні *режимін таңдаңыз* егер файл тек оқуға арналған болса, 'r' символы болуы мүмкін, 'w' тек жазу үшін ашық (дәл осындай атаумен бар файл өшіріледі) және 'a' файлы қосу үшін ашық: файлға автоматты түрде жазылған кез келген деректер соңына қосылады. 'r +' файлды оқу мен жазуға ашады. Параметр *режимін таңдаңыз* қосымша: егер ол алынып тасталса, онда ол 'r' -ге тең болады.  **Тапсырма**  Пернетақтадан деректерді енгізу |  | бағалау.  **«Мадақ тау»** |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия** |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | https://ds05.infourok.ru/uploads/ex/0212/000ff419-c6bc4935/hello_html_m62f81942.jpg |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 6 сабақ** | **Курс атауы:** «Python  программалау тілінің негіздері» | **Сыныбы: 9**  **Күні: Мұғалім:** | | | |
| **Тақырыбы:** Сандармен қарапайым әрекеттерге арналған тапсырмалар. | | | | | |
| **Оқу мақсаты:** | Python тіліндегі сандар түрлерін қарастыру. | | | | |
| **Сабақ мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://img.youtube.com/vi/qDkgiOM5Yzs/0.jpg | | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ортасы** | **Мағынаны тану**  Санау жүйелері. Сандар тек ғана ондық санау жүйесінде ғана емес сонымен қатар, басқа да санау жүйелерінде беріледі. Мысалы, компьютерде екілік санау жүйесі қолданылады. 19 саны екілік санау жүйесінде 10011 деп бейнеленеді. Кейбір жағдайда бір санау жүйесінен екіншіге көшу қажет болады. Оны орындауға Python тілі бірнеше функцияларды ұсынады:   * int([object], [санау жүйесінің негізі]) – ондық санау жүйесіндегі бүтінге айналдыру. Бұл жерде үнсіз келісім бойынша ондық санау жүйесі қолданылады, бірақ негізін 2-ден 36-ға дейін таңдай отырып кез келген санау жүйесінде беруге болады. * bin(x) – бүтін санды екілік жолға айналдыру. * hex(х) - бүтін санды он алтылық жолға айналдыру. * oct(х) - бүтін санды сегіздік жолға айналдыру.   Келесі 20-суретте мысалдар көрсетілген. | | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай орында йды | Отшаш у арқылы бір- бірін бағалау  .  Бас бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |  |  |
|  | >>> a = int('19') # жоды санаға айна  >>> b = int('19.5') # жол бүтін сан б Traceback (most recent call last):  File "", line 1, in  ValueError: invalid literal for int() wit  >>> c = int(19.5) # жылжымалы нүк бөлігін алып тастайды  >>> print(a, c) 19 19  >>> bin(19) '0b10011'  >>> oct(19) '0o23'  >>> hex(19) '0x13'  >>> 0b10011 # сандық тұрақтылық | бөлше |
|  |  |

лдыру олып таб

h base 10: тесі бар с

арды осла

ылмайды

'19.5'

анға қолд

й да жазу

анылғанда

ға болады.

19

>>> int('10011', 2)

19

>>> int('0b10011', 2)

19

Сурет 20 Санау жүйесіне арналған мысалдар листингі

Нақты сандар (float). Нақты сандар да бүтін сандардағы сиқты операцияларды қолдайды, бірақ сандарды компьютерде бергендіктен нақты сандар нақты болмауы және қателіктерге соқтыруы мүмкін және оның өзі қателіктерге әкелуі мүмкін.

>>> 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1

+ 0.1 + 0.1 + 0.1 + 0.1

0.9999999999999999

Жоғары нақтылыққа қол жеткізу үшін басқа да объектілерді (Мысалы: Decimal және [Fraction](https://pythonworld.ru/moduli/modul-fractions.html)) қолдануға болады.

Сонымен қатар, нақты типтер ұзақ арифметиканы қолдамайды.

>>>

>>> a = 3 \*\* 1000

>>> a + 0.1

Traceback (most recent call last): File "", line 1, in

OverflowError: int too large to convert to float

Сандармен қарапайым мысалдар:

>>>

>>> c = 150

>>> d = 12.9

>>> c + d 162.9

>>> p = abs(d - c) # сан модулі

>>> print(p) 137.1

>>> round(p) # Дөңгелектеу 137

Қосымша әдістері:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  float.as\_integer\_ratio() - | |  |  |  |
| қатынасы осы санға тең болатын | |
| жұп сандар. | |
|  float.is\_integer() - мән бүтін сан | |
| бола ма. | |
|  float.hex() – float-ты hex-ке | |
| түрлендіреді (он алтылық жүйеде). | |
|  classmethod float.fromhex(s) - он | |
| алтылық жолдағы float. | |
| **>>>** (10.5).hex() | |
| '0x1.5000000000000p+3' | |
| **>>>** | |
| float.fromhex('0x1.5000000000000p+ | |
| 3') | |
| 10.5 | |
| Python тілінде сандармен жұмысқа | |
| қатысты стандартты өрнектерден | |
| басқа бірнеше қажетті модульдер | |
| бар. Келесі [math](https://pythonworld.ru/moduli/modul-math.html) модулі күрделі | |
| математикалық функцияларды | |
| қамтамасыз етеді. Оның | |
| қолданылуы төмендегідей. | |
| >>> import math | |
| >>> math.pi | |
| 3.141592653589793 | |
| >>> math.sqrt(85) | |
| 9.219544457292887 | |
| Келесі random [модулі](https://pythonworld.ru/moduli/modul-random.html) - кездейсоқ | |
| сандар генераторы мен кездейсоқ | |
| функцияны таңдауды жүзеге | |
| асырады. | |
| >>> import random | |
| >>> random.random() | |
| 0.15651968855132303 | |
| Комплексті сандар (complex). | |
| Python тіліне сонымен қатар, | |
| комплекті сандар енгізілген. | |
| 21-суретте комплекс сандардың | |
| программада қолданылуы | |
| көрсетілген. | |
|  | **>>>** x = complex(1, 2) |
| **>>> print**(x) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | (1+2j)  **>>>** y = complex(3, 4)  **>>> print**(y) (3+4j)  **>>>** z = x + y  **>>> print**(x) (1+2j)  **>>> print**(z) (4+6j)  **>>>** z = x \* y  **>>> print**(z) (-5+10j)  **>>>** z = x / y  **>>> print**(z) (0.44+0.08j)  **>>> print**(x.conjugate()) *# түйіндес сан*  (1-2j)  **>>> print**(x.imag) *# жорамал бөлігі*  2.0  **>>> print**(x.real) *# нақты бөлігі*  1.0  **>>> print**(x > y) *# Комплексті санды салыстыруға болмайды* Traceback (most recent call last): File "", line 1, in  TypeError: unorderable types: complex() > complex()  **>>> print**(x == y) *# бірақ теңдікке тексеруге болады*  False  **>>>** abs(3 + 4j) *# комплексті сан модулі*  5.0  **>>>** pow(3 + 4j, 2) *# санның дәрежесін шығару* |  |  |  |
| (-7+24j) |
| **Тапсырма.** Тақырып бойынша өздігінен мысал келтіру. | |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия** | |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | https://ds04.infourok.ru/uploads/ex/0ae2/00026859-899c2fa8/img27.jpg |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 7 сабақ** | **Курс атауы:** «Python  программалау тілінің негіздері» | **Сыныбы: 9 Күні: Мұғалім:** | | | |
| **Тақырыбы:** Логикалық өрнектер мен операторлар. | | | | | |
| **Оқу мақсаты:** | Логикалық өрнектер мен операторлар туралы мәлімет беру | | | | |
| **Сабақ мақсаты** | **Тақырыпты түсінеді, тапсырманы орындайды, алған білімін өмірде қолданады** | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/0d99/000a0c30-180561b4/img0.jpg | | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану**  Python тілінде операторлардың орындалу приоритеті бар, яғни бірінші кезекте орындалатыны көбейту және бөлу операторлары | | Оқушы лар сұраққ а жауап | Отшаш у арқылы бір- бірін |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | орындалады. Ал қосу мен азайту амалдарының приоритеті томен. Егер операцияның проритеті бірдей болса, онда солдан оңға қарай кезекпен орындалады.  Операторлардың орындалу кезегін өзгерту үшін жақшалар қолданылады - бұл ең алдымен орындалатын оператор, ол операндамен бірге жазылады. (3 + 7) \* (6 + 4) Бүл өрнектің мәні 20 болатынын айтқан едік. Егер жақшаны алатын болсақ, нәтиже өзгереді: 3 + 7 \* 6 + 4 Python алдымен 7-ні 6-ға көбейтеді, көбейтіндіге 3 пен 4-ті қосады.  Нәтиже басқаша болып шығады -  49. Операторларды орындау реті және олардың приоритеті (бірінші кезекте орындалатыны) Бірінші өрнек мысалында біз + смиволын пайдаландық және осы символдың жанында қатар тұрған екі санды қосу операциясын орындауға нұсқау бердік. Бұл жағдайда + оператор, бір немесе екі операнда мәніне қарапайым амал орындайтын команда. Бағдарламадағы көптеген сөйлемдер (логикалық жолдар) өрнектен құралған. Өрнекке қарапайым мысал: 2 + 3. Өрнекті оператор мен операндыларға бөлуімізге болады. [Оператор деп](https://topuch.ru/za-merzimdi-jospardi-taraui-derekterdi-sinu-mektep/index.html) [амалдарды](https://topuch.ru/za-merzimdi-jospardi-taraui-derekterdi-sinu-mektep/index.html), ал операнд деп сандарды айтуға болады. Өрнектегі операндылар 2 жэне 3. Мына мысалдарды теріп жазыңыз: » > 2 + 3 5 » > 3 \* 5 15 Жауабы бірден есептеліп отыр. 2+2 деген бұйрыкты өрнек деп атадық. Бағдарламалау тіліндегі өрнек термині - аяқталған іс-әрекетті орындайтын бұйрық. Қандай да бір | береді, жұмыс ты берілге н түріне қарай орында йды | бағалау  .  Бас бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | мэнді есептеу, деректердің андай да бір құрылымын құру, программалық кодты орындау басқару, функцияны немесе әдісті шақыру командасы іс-әрекет деп аталынады. Python тіліндегі кез келген өрнек каретаны кері қайтару жэне жолды ауыстыру символымен аяқталуы керек. Олар программалық кодқа батырмасын басу аркылы қойылады. Тағы да бірнеше өрнек қарастырайық. С] 3\*4+8 3-ті 4-ке көбейтеміз, пайда болған көбейтіндіге 8-ді қосамыз жэне 20-ны аламыз. Көбейту амалы қосу амалынан бұрын орындалады. С] 5/6 5-ті 6-ға бөлеміз және - 0,8333333333333334 мэнін аламыз. Бағдарламадағы жолдың ретіне қарай іс-әрекеттер де рет-ретімен орыңдалады. Бағдарлама кодын жазғанда кері шегіністерге мэн беру аса маңызды болып табылады. Сондықтан, іс-әрекет реттілігіне кіретін операторлардың барлығы да бірдей шегініспен жазылуы тиіс: а = 1 b = 2 а = а + b b = a - b а = а-Ь print (a, b)  **Тапсырма**  Python-да шартты өрнектің минималды дизайнына мысал келтіріңіз.  Шартты өрнектің минималды дизайны тек if операторынан тұрады. Біз практикалық мысал келтіреміз. Сіз пайдаланушыдан 1- ден 100 – ге дейінгі санды енгізуді сұрайсыз делік, содан кейін оның үстіне қандай да бір операция жасаңыз (біздің мысалда сіз дұрыс санды енгізген фразаны басып шығарыңыз: X). Егер басқа сан немесе басқа деректер берілсе, |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | онда бұл мәнді елемеңіз. Шешім функция түрінде ұсынылады.  Мысал-IDE  ---  def is\_correct\_number(number):  if number in range(1, 101):  print (F ' Сіз дұрыс санды енгіздіңіз:: {number}')  # Тесттер is\_correct\_number(4) is\_correct\_number(421) is\_correct\_number(4.11) is\_correct\_number('Dog') Орындалу нәтижесі  ---  Сіз дұрыс санды енгіздіңіз: 4 |  |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://ds04.infourok.ru/uploads/ex/064a/00116b69-030b9b0c/img7.jpg |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 8 сабақ** | **Курс атауы:** «Python  программалау тілінің негіздері» | **Сыныбы: 9 Күні: Мұғалім:** | | | |
| **Тақырыбы:** Күрделі шартты өрнектер (and, or, not логикалық амалдары). | | | | | |
| **Оқу мақсаты:** | Күрделі шартты өрнектер (and, or, not логикалық амалдары) туралы  түсіндіру | | | | |
| **Сабақ мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін өмірде  қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқу шын ың** | **Бағала у** | **Ресурст ар** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **әреке ті** |  |  |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/0fdf/0006edae-8e50c830/img2.jpg | Псих ологи ялық ахуал ға беріл ген өлеңд і дауы стап айты п, тілді жатт ықты ру және жаңа сабақ қа назар аудар у. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану**  Аргументтерінің барлығы **true** болса, **And**  функциясы **true** мәнін қайтарады. Аргументтерінің кез келгені **true** болса, **Or** функциясы **true** мәнін қайтарады.  **Not** функциясы аргументі **false** болса **true** мәнін қайтарады; аргументі **true** болса, ол **false** мәнін қайтарады.  Бұл функциялар дәл Excel бағдарламасындағыдай жұмыс істейді. Сондай-ақ Visual Basic немесе JavaScript синтаксисін пайдаланып, дәл осы әрекеттерді орындау үшін [операторларды](https://docs.microsoft.com/kk-kz/powerapps/maker/canvas-apps/functions/operators) пайдалана аласыз:  **Функция Visual Basic операторы**  **нотациясы нотациясы**  **And( x, y ) x And y** | Оқуш ылар сұрақ қа жауа п беред і, жұмы сты беріл ген түрін е қарай орын дайд | Отшаш у арқылы бір- бірін бағалау  . | **е** |

# ÐÐ°ÑÑÐ¸Ð½ÐºÐ¸ Ð¿Ð¾ Ð·Ð°Ð¿ÑÐ¾ÑÑ ÑÐ°Ð»ÑÑың

**Java нота x &**

**Script оп циясы**

**& y**

**раторын**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Or( x, y ) x Or y**  **Not( x ) Not x**  Бұл функциялар логикалық мәндермен жұмыс істейді. Оларды санға немесе жолға тікелей өткізу мүмкін емес; оның орнына салыстыру немесе тексеру жасау керек.  Мысалы, мына **x** мәні **1**-ден үлкенірек болса, **x > 1** логикалық формуласын есептеу **true** логикалық мәнін береді. **x** мәні **1**-ден азырақ болса, формуланы есептеу **false** мәнін береді.  **Синтаксис**  **And**( *LogicalFormula1*, *LogicalFormula2* [,  *LogicalFormula3*, ... ] )  **Or**( *LogicalFormula1*, *LogicalFormula2* [,  *LogicalFormula3*, ... ] )  **Not**( *LogicalFormula* )   * *LogicalFormula(s)* - Міндетті. Есептейтін және амалдар орындайтын логикалық формулалар.   **Мысалдар**  Бұл бөлімдегі мысалдарда мына глобалдық айнымалылар пайдаланылады:   * **a** = *false* * **b** = *true* * **x** = 10 * **y** = 100 * **s** = "Hello World"   Бағдарламада осы глобалдық айнымалыларды жасау үшін [**Түймешік**](https://docs.microsoft.com/kk-kz/powerapps/maker/canvas-apps/controls/control-button)басқару элементін кірістіріңіз және оның **OnSelect** сипатын мына формулаға орнатыңыз:  Set( a, false ); Set( b, true ); Set( x, 10 ); Set( y, 100 ); Set( s, "Hello World" )  Түймешікті таңдаңыз (Alt пернесін басып тұрып, оны басу арқылы), содан кейін [**Белгі**](https://docs.microsoft.com/kk-kz/powerapps/maker/canvas-apps/controls/control-text-box)басқару элементінің **Мәтін** сипатын келесі кестенің бірінші бағанындағы формулаға орнатыңыз.  **Формула Сипаттама Нәтиж е**  **And( a, b a** және **b** мәндерін *false*  **)** тексереді. | ы | Бас бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |  |

**x || y**

**! x**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
|  | Аргументтердің біреуі — *false*, сондықтан функция *false*  мәнін қайтарады. |  |
| **a And b** | Visual Basic нотациясын пайдаланатын алдыңғы  мысалмен бірдей. | *false* |
| **a && b** | JavaScript нотациясын пайдаланатын  алдыңғы мысалмен бірдей. | *false* |
| **Or( a, b )** | **a** және **b** мәндерін тексереді.  Аргументтердің біреуі — *true*, сондықтан функция *true* мәнін  қайтарады. | *true* |
| **a Or b** | Visual Basic нотациясын пайдаланатын алдыңғы  мысалмен бірдей. | *true* |
| **a || b** | JavaScript нотациясын пайдаланатын  алдыңғы мысалмен бірдей. | *true* |
| **Not( a )** | **a** мәнін тексереді. Аргумент — *false*, сондықтан функция қарама- қарсы нәтижені  қайтарады. | *true* |
| **Not a** | Visual Basic нотациясын пайдаланатын  алдыңғы мысалмен бірдей. | *true* |
| **! a** | JavaScript | *true* |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://ds04.infourok.ru/uploads/ex/06d6/000ec05d-2d82e71d/img10.jpg |  |  |  |

|  |  |  |
| --- | --- | --- |
|  | нотациясын пайдаланатын алдыңғы  мысалмен бірдей. |  |
| **Len( s ) < 20**  **And Not I sBlank( s**  **)** | **s** ұзындығы 20-дан азырақ екенін және оның **blank** мәні емес екенін тексереді.  Ұзындық 20-дан азырақ және мән blank емес.  Сондықтан нәтиже  — *true*. | *true* |
| **Or( Len( s ) < 10, x < 100, y < 100 )** | **s** ұзындығы 10-нан азырақ екенін, **x** 100-ден азырақ екенін және **y** 100- ден азырақ екенін тексереді. Бірінші және үшінші аргументтер false, бірақ екіншісі — true. Сондықтан функция *true* мәнін  қайтарады. | *true* |
| **Not IsBlank( s**  **)** | **s** мәні *blank* екенін тексеріп, бұл *false* мәнін қайтарады. **Not** осы нәтиженің қарама-қарсы мәнін, яғни *true*  мәнін қайтарады. | *true* |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 9 сабақ** | **Курс атауы:** «Python  программалау тілінің негіздері» | **Сыныбы: 9**  **Күні: Мұғалім:** | | | |
| **Тақырыбы:** Шартты оператор. Балама орындау | | | | | |
| **Оқу мақсаты:** | Шартты операторла туралы түсінік беру. | | | | |
| **Сабақ мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал** оқушылардың назарын сабаққа аудару Балалар, сабағымызды бастамас бұрын күн нұрына қол созып, алақанды жайып Күнмен амандасып алайық.  .  hello_html_5523f853.jpg | | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану**  Операторлар мен оларды қолдану  Оператор Аты Түсіндірмесі + Қосу Екі нысанды қосады. Азайту Екі санның айырмасын табады. Егер бірінші сан жазылмаса, онда оны нөлге тең деп есептейді. \* Көбейту Екі санды көбейтеді немесе жолды берілген сан бойынша қайталайды.  \*\* Дэрежеге шығару X санын у дәрежесіне шығарады. / Бөлу X | | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай | Отшаш у арқылы бір- бірін бағалау  . |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | санын у санына бөледі. // Бүтінсанды бөлу Бүтінсанды бөлуді іске асырады. % Модуль бойынша бөлу Бөлудің қалдығын көрсетеді. « Солға жылжу Санның битін солға осынша рет жылжытады. Кез келген сан компьютер жадында биттік немесе екілік жүйеде келтіріледі. » Оңға жылжу Санның битін оңға осьгнша позицияға жылжытады. & Битгік ЖӘНЕ Сандарға орьгадалатын битгік ЖӘНЕ операциясы 1 Битгік НЕМЕСЕ Сандарға орындалатын биттік [НЕМЕСЕ операциясы](https://topuch.ru/jaritesu-operaciyasi-3-kezenen-tedi/index.html)  Л Битгік ТЕК ҚАНА НЕМЕСЕ  биттік ТЕК ҚАНА НЕМЕСЕ операциясы  Битгік ЖОҚ X саны үшін биттік ЖОҚ операциясы - (х+1)-ге тең. < Кіші Х-тің у-тен кіші екендігін анықтайды. Салыстыру операторларының барлыгы да True немесе False қайтарады. Бұл сөздер бас әріппен жазылғанына назар аудару керек.   * Үлкен Х-тің у-тен үлкен екендігін анықгайды. <= Кіші немесе тең Х-тің у-тен кіші немесе тең екендігін анықтайды. >= Үлкен немесе тең Х-тің у-тен үлкен немесе тең екендігін анықтайды. —   = Тең Нысандардың тең екенін анықгайды. 1= Тең емес Нысандардың тең емес екенін анықгайды. not Логикалық ЖОҚ Егер х саны True болса, онда False қайтарады. Егер х саны False болса, онда True деп жазады. and Логикалық ЖӘНЕ Егер х саны | орында йды | Бас бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | False болса, х жэне у False деп есептейді. Kepi жағдайда у мэнін жазады. or Логикалық НЕМЕСЕ Егер х саны True болса, онда True деп жазады. Кері жағдайда у мәнін жазады.  Математикалық операцияларды жэне меншіктеуді қысқаша жазуға болады. а = 2;а = а\*3 өрнегін a = 2; a \*= 3 түрінде жазуға болады.  «айнымалы=өрнектің айнымалы мэні» түріндегі өрнектерді  «айнымалы операция=өрнек түрінде жазуға болады. Есептеу тэртібі өзгешерек. Мысалы + 3 \* 4 түрінде өрнек берілсе, көбейту амалының бірінші орындалатыны мектеп курсынан белгілі. Көбейту амальгаың приоритеті қосу амалынан басым. Келесі кестеде Python операторларыньщ приоритет! көрсетілген. Кестенің төменгі жағында орналасқан операторлар ең алдымен есептеледі де, кестенің жоғарғы бөлігіндегілер соңында есептеледі.  Операторлар приоритеті  оператор сипаттамасы lambda лямбда-өрнек or Логикалық  «немесе» and Логикалық «жэне» not X Логикалық «емес» in, not in Тәуелділігін тексеру is, is not Теңцігін тексеру <, <=, >, >=, !=, == салыстыру  J Биттік «немесе» A Битгік «тек қана немесе» & Биттік «жэне» « , » Орын алмастыру +> Қосу жэне азайту Көбейту, бөлу, бүтінсанды болу жэне |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | бөлудің қалдығы +Х, -X Теріс сан, оң сан  х Биттік «емес» \*\* Дәрежеге шығару x.attribute Атрибутқа сілтеме xf индекс) Индекс бойьшша іздеу х [индекс 1  :индекс21 Қырқып алу Г(аргументтер ...) Функцияны шақыру (өрнектер,...) Байланыс немесе кортеж [өрнектер, ...1 Тізім  {кілт:деректер,...} Сөздік Өрнекті есептеу ретін жеңілдету  үшін жақшаны қолдануға болады. Мысалы, 2+(3\*4) дегенді 2+3\*4 деген өрнекті ұғыну оңай.  Мысалды expression.py ретінде сакдаңыз. length = 5 breadth = 2 area  = length \* breadth print(‘Ауданы тең болады’, area) ргЫ('Периметрі тең болады', 2 \* (length + breadth)) Қорытындысы: $ python expression.py Ауданы тең болады 10 Периметрі тең болады 14 Бұл бағдарлама қалай жұмыс істейді?  Тіктөртбұрьпптың ұзындығы мен ені length және breadth айнымалыларында сақгалады.  Оларды периметр мен ауданды есептеу үшін қолданамыз. Есептеу нәтижесі area айнымалысында сақталады жэне print функциясы арқылы экранға шығарылады.  Екіншісінде 2 \* (length + breadth) өрнегінің мәнін print функциясына бірден қоямыз. |  |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://cf.ppt-online.org/files/slide/t/tBHOT7NizbsK1o69xuYVXqgrAnUC5EI4kyRjLa/slide-18.jpg |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 10 сабақ** | **Курс атауы:** «Python  программалау тілінің негіздері» | **Сыныбы: 9**  **Күні: Мұғалім:** | | | |
| **Тақырыбы:** Шартты оператормен есептерді шешудің мысалдары. | | | | | |
| **Оқу мақсаты:** | Шартты оператормен есептерді шешуді үйрету. | | | | |
| **Сабақ мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://fs.znanio.ru/methodology/images/fd/ad/fdad8c13d2673b07ebdd7e678f2585e7d715c044.jpg | | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану Тапсырма**  Шартты нұсқауларды жазудың барлық мүмкін нұсқаларын көрсетіңіз.  Шартты өрнектерді жазудың 4 нұсқасы бар. Оларды мысалдармен қарастырыңыз.  1. Тек if (бұл әдісті біз жоғарыда қарастырдық) | | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай | Отшаш у арқылы бір- бірін бағалау  . |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 2. if / else | орында йды |  |  |
| Бұл жағдайда біз бір жағдайды тексереміз, ал қалғандары үшін жалпы реакция, бір нәтиже береміз. Мысалы, list\_len(lst) функциясы үшін бізге тізім қажет, ал қалған барлық деректер түрлері сәйкес келмейді. Функция берілген тізімнің ұзындығы форматында тізімнің ұзындығын басып шығарады: х. егер кіріске тізім алынбаса, онда жауап аламыз, сізге дәлел ретінде тізім қажет. |  | Бас бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |
| Мысал-IDE  ---  def list\_len(lst):  if isinstance(lst, list):  print (F ' берілген тізімнің ұзындығы: {len (lst)}')  else:  print ('аргумент ретінде тізім қажет') |  |  |
| # Тесттер list\_len([1, 2])  list\_len([]) list\_len(43) |  |  |
| Орындалу нәтижесі  ---  Берілген тізімнің ұзындығы: 2 Берілген тізімнің ұзындығы: 0 Дәлел ретінде тізім қажет |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия** |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | https://fs.znanio.ru/d5aff2/e0/77/fdeb2b6ff2c17c6402414e05a683d98e8c.jpg |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 11 сабақ** | **Курс атауы:** «Python программалау тілінің  негіздері» | **Сыныбы: 9 Күні: Мұғалім:** | | | |
| **Тақырыбы:** Python тілінде тармақтауды жүзеге асыру. | | | | | |
| **Оқу мақсаты:** | Python тілінде тармақтауды жүзеге асыру бойынша түсінік беру. | | | | |
| **Сабақ**  **мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін өмірде  қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әреке ті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/08fb/0000cc4a-d204bcbd/img2.jpg | | Психо логия лық ахуал ға берілг ен өлеңді дауыс тап айтып  , тілді жатты қтыру және жаңа сабақ қа назар аудар у. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану** | | Оқуш | Отшаш |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Көбінесе бағдарламаларда белгілі бір шарт орындалған кезде ғана кодтың бір бөлігін орындау қажет болады.  Мысалы, егер пайдаланушы қате деректерді енгізсе, қате туралы хабарды көрсету керек.  Егер біз бірдеңе жасағымыз келсе, егер қандай‑да бір шарт орындалса, біз if операторын қолдануымыз керек.  Ең оңай жолы-бұл оператордың мысалмен қалай жұмыс істейтінін көрсету. Ол үшін "санды тап"бағдарламасын жазуға тырысайық. Егер сіз жаңадан бастаушы бағдарламашы болсаңыз, оның қалай жұмыс істейтінін сезіну үшін осы бағдарламаны жазып көріңіз.  Компьютер кездейсоқ санды таңдайды, ойыншы болжауға тырысады, бағдарлама қолданушыға оның жауабының дұрыстығын хабарлайды.  кездейсоқ импорттау  Сан = кездейсоқ.randint(1,10)  болжау = int (енгізу ('1-ден 10-ға дейінгі санды енгізіңіз:'))  . егер = = сан болса :  басып шығару ('таптындар!')  If операторының синтаксисі қарапайым  — алдымен сіз if кілт сөзін жазуыңыз керек, содан кейін-шарт, ал жолдың соңында қос нүкте болуы керек.  Шарт іске қосылған кезде орындалатын кодтың барлық жолдары if кілт сөзіне қатысты төрт бос орынмен жазылуы керек. Шегініс арқылы Python бұл жолдардың if-ге қатысты екенін түсінеді және егер шарт дұрыс болса ғана оларды орындайды. Біздің мысалда " біз болжадық!"егер пайдаланушы дұрыс Нөмірді енгізсе ғана орындалады.  ИЯ, Модульдер туралы кейінірек сөйлесетін боламыз, бірақ сіз болжағандай, random.randint кездейсоқ | ылар сұрақ қа жауап береді  ,  жұмы сты берілг ен түріне қарай орынд айды | у арқылы бір- бірін бағалау  .  Бас бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | санды қайтарады, ал import random жолы кездейсоқ сандармен жұмыс істеу үшін модульді біздің бағдарламаға жүктейді. Бұл модульдің тізімін осы бетте көруге болады. Қазір біз шарттар туралы сөйлесіп жатырмыз, сондықтан жалғастырамыз:  import random  num = random.randint(1,10)  guess = int (input ('1-ден 10-ға дейінгі санды енгізіңіз:'))  if guess == num :  print('оны сұраса!') else:  басып шығару ('Кешіріңіз, Сан жасалды', num)  Егер if операторы жұмыс істемесе, кез‑келген кодты орындауға жауапты else операторын қостық. Осы екі оператордың жиынтығын "егер қандай‑да бір шарт дұрыс болса, осындай кодты орындаңыз, әйтпесе басқа кодты орындаңыз"деп оқуға болады. |  |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://ds05.infourok.ru/uploads/ex/0350/000e6eb7-4d87d0ab/hello_html_m37bc1b0f.jpg |  |  |  |

|  |  |  |
| --- | --- | --- |
| **№ 12 сабақ** | **Курс атауы:** «Python программалау тілінің  негіздері» | **Сыныбы: 9 Күні: Мұғалім:** |
| **Тақырыбы:** Цикл ұғымы. Цикл денесі. | | |
| **Оқу мақсаты:** | Цикл ұғымымен таныстыру. | |
| **Сабақ** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін өмірде | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **мақсаты** | қолданады | | | |
| **Жоспар** | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/1162/00073d51-bdf15a21/img0.jpg | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану**  Мен кездестірген әр бағдарламалау тілінде цикл құрылымы бар. Көптеген тілдерде осындай дизайн бар. Python әлемінде циклдердің екі түрі бар: Цикл үшін  While циклі  Мен цикл екіншісіне қарағанда танымал екенін байқадым. Циклдар біз бірнеше рет бір нәрсе жасауымыз керек болған жағдайда қолданылады. Деректер бөлігінде бірнеше операцияны (немесе бірқатар операцияларды) қайта-қайта орындау сирек емес. Мұнда циклдар күшіне енеді. Олардың арқасында бұл мәселені мүмкіндігінше жеңілдетуге болады. Бұл құрылымдардың қалай жұмыс істейтінін егжей-тегжейлі қарастырайық!  Жоғарыда айтылғандай, біз циклды | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай орында йды | Отшаш у арқылы бір- бірін бағалау  .  Бас бармақ |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | бірнеше рет қайталау қажет болған жағдайда қолданамыз. Егер сіз мысалды қарасаңыз, түсіну оңайырақ. Біз кірістірілген Python range функциясын қолданамыз. Range функциясы "n" элементтеріндегі ұзын тізімді жасайды. Python 2 нұсқасында.X xrange деп аталатын тағы бір функция бар, ол сандар генераторы болып табылады және range сияқты ресурстарды қажет етпейді. Бұрын әзірлеушілер Xrange-ді Python 3- те range-ке ауыстырды. Міне мысал: Python  1  басып шығару(range (5)) # жауап: range (0, 5)  Көріп отырғаныңыздай, range Функциясы бүтін санды алып, range нысанын қайтарды. Range функциясы сонымен қатар бастапқы мәнді, соңғы мәнді және қадам мәнін алады. Мұнда тағы екі мысал келтірілген:  Python  1  2  3  4  5  a = range(5, 10) print(a) # range(5, 10)  b = list(range(1, 10, 2))  print(b) # [1, 3, 5, 7, 9]  Қалам мысалда сіз бастапқы және соңғы мәнді бере алатындығыңыз көрсетілген, ал range функциясы бастапқы мәннен бастап соңғы мәнге дейін (бірақ оның ішінде емес) сандарды қайтарады.  Мысалы, 5-10 сұрау кезінде біз 5-9 аламыз. Екінші мысалда range функциясы 1 мен 10 арасындағы әрбір |  | арқылы өзін-өзі бағалау  .  **«Мада қтау»** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | екінші элементті қайтару үшін тізім (тізім) функциясын қалай пайдалану керектігі көрсетілген. Сондықтан ол 1- ден басталады, 2-ді өткізіп жібереді және т.б. Енді сіз оның циклдармен не істейтінін болжайтын шығарсыз? Range функциясын қолдана отырып, циклдің қалай жұмыс істейтінін көрсетудің бір оңай жолы бар! Қарап көрейік:  Python  1  2  for number in range(5): print(number)  **тапсырма**  Мұнда не болды? Мұны түсіну үшін сол жақтан оң жаққа оқимыз. 5 диапазонындағы әр Сан үшін біз санды енгіземіз. Егер біз 5 мәні бар range шақырсақ, 5 элементтің тізімін алатынымызды білеміз. Сондықтан циклден өткен сайын ол элементтердің әрқайсысын шығарады. Жоғарыда көрсетілген Цикл келесіге балама болуы мүмкін:  Python  1  2  for number in [0, 1, 2, 3, 4]: print(number)  Range функциясы нәтижені сәл кішірейтеді. For циклі кез-келген Python итераторын айналып өте алады. Біз оның тізіммен қалай жұмыс істейтінін көрдік. Ол сөздікпен Итерация жасай алатындығын қарастырайық. |  |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия** |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | https://cf.ppt-online.org/files/slide/t/tBHOT7NizbsK1o69xuYVXqgrAnUC5EI4kyRjLa/slide-18.jpg |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 13 сабақ** | **Курс атауы:** «Python  программалау тілінің негіздері» | **Сыныбы: 9 Күні: Мұғалім:** | | | |
| **Тақырыбы:** Цикл операторы while. | | | | | |
| **Оқу мақсаты:** | Таңдау және қайталау операторларын программаны өңдеудің  кіріктірілген ортасыныңда қолдану | | | | |
| **Сабақ мақсаты** | While .. do операторы туралы түсінік береді. Әзір-қайталау командаларын Паскаль опереторында программалауды үйренді, пысықтайды  Программаны ықшам, сапалы жазу дағдысын қалыптастыру. Программаның математикалық моделі туралы түсінік қалыптастырады.  Анық және толық сөйлеу арқылы, мәселені түсінгенін көрсетеді. | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/0110/0010b667-4e6a1c0e/img1.jpg | | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану** | | Оқушы | Отшаш |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **While операторы**.  Алдын ала берілген шарт бойынша қайталануды жүзеге асыратын бұл  оператор While (*оқылуы:* уаил, *ау дарылуы:* әзірше)  және DO (*оқылуы:* ду, *аударылуы:*  орындау) түйінді сөздері қолданылып орындалады.  hello_html_m3e74097f.jpg  Program esep1; uses graphabc; vari,r,k:integer; s:string;  begin lockdrawing; r:=200; k:=0;  for i:=1 to 10 do begin  if i mod 2=0 then setbrushcolor(clred) else setbrushcolor(clwhite); circle(200,200,r-k);  str(i,s); textout(6+k,194,s); k:=k+20;  redraw; end; end.  Шартын алдын ала тексеретін цикл.  **1. While-Do** операторының орындалуы:  *1.* ***While*** *операторы циклге кіру шартынан және цикл денесінен* | лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай орында йды | у арқылы бір- бірін бағалау  .  Бас бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *тұрады.*  *2. Цикл денесін әрбір қайталау алдында шарт тексеріледі. Шарт ақиқат болған жағдайда, цикл*  *денесі орындалады, әйтпесе бағдарлама келесі операторға көшеді.*  *3. Егер шарт алғаш тексерілгенде жалған болса, цикл денесі бір рет те орындалмастан бағдарлама цикл соңына шығады.*  2. **Repeat–until** операторының орындалуы:   1. *шарт кейін тексеріледі, сондықтан серия кем дегенде бір рет атқарылады.* 2. *r****epeat*** *операторында шарт ақиқат (иэ) болғанда қайталау (цикл) аяқталады.* 3. *r****epeat*** *операторында қайталау соңы шартпен аяқталады.*   Енді алдыңғы мысалдағы алғашқы 5 натурал санның қосындысын есептеу бағдарламасын циклдік операторлар көмегімен құрайық.  Ол үшін алгоритмдік тілде, блок схемасын және бағдарламасын жазайық.  **әзір** циклін пайдаланып алгоритмдік тілдегі жазылуын мен жазамын, оқушылар бағдарламалау тіліне аударады (флипчартта, TP бағдарламасында)  **aлг** қосынды **( нәт бүт** S**) басы бүт** х  S:=0; х:=1  **әзір** x<=5 **цб** S:=S+x  х:=х+1  **цс шығару S соңы**  **program** kosyndy; |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **var** x,s**: integer; begin**  s:=0; x:=1;  **while** x<=5 **do begin**  s:=s+x; x:=x+1;  **end;**  **writeln** (‘s=’,s);  **end**.  **дейін** циклін алгоритмдік тілде мен, бағдарламалау тілінде оқушы жазуы керек (флипчартта, ТР бағдарламасында  **aлг** қосынды **(нәт бүт**S**) басы бүт** х  S:=0; х:=1  **қайталау**  S:=S+x  х:=х+1 **дейін** x>5 **шығару** S **соңы**  **program** kosyndy; **var** x,s**: integer; begin**  s:=0; x:=1;  **repeat** s:=s+x; x:=x+1;  **until** x>5; **writeln** (‘s=’,s); **end.**  Алғышартты циклден соңғы шартты циклге ауысып бағдарлама құруда не байқадық?  **program kosyndy; var x,s: integer; begin** s:=0; x:=1;  **while** x<=5 **do begin** s:=s+x; x:=x+1; |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **end; writeln** (‘s=’,s);  **end**.  **program kosyndy; var x,s: integer; begin**  s:=0; x:=1;  **repeat** s:=s+x; x:=x+1;  **until** x>5**; writeln** (‘s=‘,s); **end.**   1. Бірінші бағдарламадағы **while** операторы жазылған жолға екінші бағдарламада **repeat**   операторы жазылады.   1. Бірінші бағдарламадағы цикл денесі **begin … еnd;** операторлық жақшаға алынады, ал   екінші бағдарламада цикл денесін (операторлық жақшаға алу міндетті емес) кейін **until**  операторы жазылады.   1. **әзір** цикліндегі шартқа **дейін** цикліндегі шарт қарама-қарсы. **Жаңа сабақты пысықтау** Оқушыларға жеке нұсқалы, үшдеңг ейлі есептерді компьютерде шешуге таратылу қағаздары беріледі. Оқушылар орындағаннан кейін компьютер арқылы тексеріп, жауабын анықтап,бағалаймыз. **Тапсырма**   1, 3, 5, …, 15 сандары квадраттарының қосындысын есептеу алгоритмін және бағдарламасын құру керек. Соңғы шартты циклді пайдаланып жауабын алу керек. Оқушылар компьютерде өздері орындауы керек.  **program** kosyndy; **var** x,y,s:**integer**; **begin**  x:=1; s:=0; |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **while** x<=15 **do begin** y:=sqr(x); s:=s+y;  x:=x+2;  **end**;  **writeln** ('s=',s);  **end**.  **program** kosyndy; **var** x,y,s:**integer**; **begin**  x:=1; s:=0;  **repeat** y:=sqr(x); s:=s+y; x:=x+2;  **until** x>15; **writeln** ('s=',s); **end**.  **x:=x+2;** өрнегін **inc(x,2)** бүтін санды арттыру процедурасымен ауыстыруға болады. |  |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://im0-tub-ru.yandex.net/i?id=3c4ffb45b2b87f384397c7e17979d7e1-l&ref=rim&n=13&w=960&h=720 |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 14 сабақ** | **Курс атауы:** «Python программалау тілінің  негіздері» | **Сыныбы: 9 Күні:**  **Мұғалім:** | | | |
| **Тақырыбы:** Циклдарды қолдану мысалдары | | | | | |
| **Оқу мақсаты:** | Циклдарды қолдану мысалдарын келтіру | | | | |
| **Сабақ**  **мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/0415/00060b0c-debde792/img4.jpg | | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану**  есеп.  S=22+23+24+25 есептеу бағдарламасы әзір циклімен жазылған, **дейін** цикліне ауыстырып жазуы керек.  **program** kosyndy; **var** x:**integer**; s:**real**;  **begin**  x:=2; s:=0;  **while** x<=5 **do** | | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай | Отшаш у арқылы бір- бірін бағалау  . |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **begin** s:=s+exp(x\*ln(2)); x:=x+1;  **end**;  **writeln** ('s=',s);  **end**.  **program** kosyndy; **var** x:**integer**; s:**real**;  **begin**  x:=2; s:=0;  **repeat** s:=s+exp(x\*ln(2)); x:=x+1;  **until** x>5; **writeln** ('s=',s); **end.** | орында йды | Бас бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://im0-tub-ru.yandex.net/i?id=3c4ffb45b2b87f384397c7e17979d7e1-l&ref=rim&n=13&w=960&h=720 |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 15 сабақ** | **Курс атауы:** «Python программалау тілінің  негіздері» | **Сыныбы: 9 Күні: Мұғалім:** | | | |
| **Тақырыбы:** For параметрі бар цикл операторы | | | | | |
| **Оқу мақсаты:** | For параметрі бар цикл операторымен таныстыру | | | | |
| **Сабақ мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал** | | Психол огиялы | Мұғалі мнің | Жұмыс дәптері |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | https://ds02.infourok.ru/uploads/ex/0417/0001501e-7f4fb15a/img1.jpg | қ | ауызша |  |
| ахуалға | бағалау | Суретте |
| берілге | ы | р |
| н |  | Карточк |
| өлеңді |  | алар |
| дауыст |  | Презент |
| ап |  | ация |
| айтып, |  |  |
| тілді |  |  |
| жаттық |  |  |
| тыру |  |  |
| және |  |  |
| жаңа |  |  |
| сабаққа |  |  |
| назар |  |  |
| аудару. |  |  |
| **Ортасы** | **Мағынаны тану**  For циклі бірізділік бойынша Итерация үшін қолданылады (тізім, tuple, dict, set немесе str).  Бұл басқа бағдарламалау тілдерінде for кілт сөзін, басқа объектіге бағытталған бағдарламалау тілдеріндегідей Итератор әдісін қолдануға ұқсас.  For циклі арқылы біз list, tuple, set және т. б. әр элемент үшін әрекеттер жиынтығын орындай аламыз.  Әр жемісті жеміс тізімінен шығарыңыз:  жемістер = ["алма", "банан", "шие"] for x in fruits:  print(x) Қорытынды:  алма банан шие  For циклі Итерация үшін қосымша айнымалылар жасауды қажет етпейді. | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай орында йды | Отшаш у арқылы бір- бірін бағалау  . |  |
|  |  | Бас бармақ арқылы өзін-өзі бағалау  . |
|  |  | **«Мада қтау»** |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Рефлексия**  https://ds05.infourok.ru/uploads/ex/0b17/00031856-d46df775/hello_html_m17336bfc.jpg |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 16 сабақ** | **Курс атауы:** «Python программалау тілінің  негіздері» | **Сыныбы: 9 Күні: Мұғалім:** | | | |
| **Тақырыбы:** Циклдардағы циклдар. | | | | | |
| **Оқу мақсаты:** | Циклдардағы циклдар туралы түсіндіру. | | | | |
| **Сабақ мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/0fdf/0006edae-8e50c830/img2.jpg | | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану**  Неліктен Python-да циклдарды қолдану керек?  Циклдар артық кодты азайтады. 1- ден 10-ға дейінгі сандарды басып | | Оқушы лар сұраққ а жауап | Отшаш у арқылы бір- бірін |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | шығаруға болатын сценарийді қарастырыңыз, екі мүмкіндік бар:   * Жұп сандарды басып шығару үшін 10 баспа операторын пайдаланыңыз. * 10 қайталауға арналған цикл ішіндегі бір баспа мәлімдемесі. Ілмектерді қолдану дұрысырақ сияқты? Біз кодтар блогын бірнеше рет орындау үшін ғана емес, сонымен қатар Python-да мәліметтер құрылымының элементтерін айналып өту үшін циклдарды қолдана аламыз. мысалдары * 1-ден 10-ға дейінгі сандарды басып шығарыңыз.   i=1 while(i<11):  print(i) i+=1  Кірістірілген циклдер-бұл нағыз бас ауруы. Олар кодтың оқылуын азайтып, оны түсінуді қиындата алады. Мысалы, кірістірілген циклдерді үзу әдетте оңай емес. Біз ішкі циклдің қай жерде үзілгенін, екінші ішкі циклді және т.б. білуіміз керек.  Бақытымызға орай, Python-да itertools модулінен керемет өнім мүмкіндігі бар. Біз оны көптеген кірістірілген циклдерді жазбау үшін қолдана аламыз.  Қарапайым мысалда оның пайдалы екеніне көз жеткізейік:  list\_a = [1, 2020, 70]  list\_b = [2, 4, 7, 2000]  list\_c = [3, 70, 7]  for a in list\_a: | береді, жұмыс ты берілге н түріне қарай орында йды | бағалау  .  Бас бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | for b in list\_b:  for c in list\_c:  if a + b + c == 2077:  print(a, b, c) # 70 2000 7  Көріп отырғанымыздай, үш тізімнен үш санды алу үшін үш кірістірілген цикл қажет, олардың қосындысы 2077. Код өте әдемі емес.  Енді product функциясын қолдануға тырысайық.  from itertools import product list\_a = [1, 2020, 70]  list\_b = [2, 4, 7, 2000]  list\_c = [3, 70, 7]  for a, b, c in |  |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://ds04.infourok.ru/uploads/ex/0695/00137b0c-ed190a52/img20.jpg |  |  |  |

|  |  |  |
| --- | --- | --- |
| **№ 17 сабақ** | **Курс атауы:** «P ython программа лау тілінің  негіздері» | **Сыныбы: 9 Күні: Мұғалім:** |
| **Тақырыбы:** Randrange функциясы. | | |
| **Оқу мақсаты:** | Randrange функциясы туралы түсіндіру | |
| **Сабақ мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған  білімін өмірде қолданады | |
| **Жоспар** | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Сабақтың кезеңдері** | | **Педагогтің әрекеті** | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақ тың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/0110/0010b667-4e6a1c0e/img1.jpg | | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортас ы** | **Мағынаны тану**  Параметрлері  start-ауқымның бастапқы нүктесі. Ауқымға қосылады. Әдепкі мәні 0 stop-диапазонның соңғы нүктесі.  Ауқымға қосылады. Әдепкі мәні 1 қадам-ауқым нүктесінің қадамы. Қадам болады.  Қайтару мәні  Бұл функция кездейсоқ элементті берілген ауқымнан қайтарады.  **Мысал**  Келесі мысал randrange () функциясын қолдануды көрсетеді. #!/usr/bin/python3  import random  # кездейсоқ 1-1000 арасындағы тақ санды таңдайды  print ("randrange(1,1000, 2) : ", random.randrange(1, 1000, 2))  # кездейсоқ 0-999 арасындағы санды | | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай орында йды | Отшаш у арқылы бір- бірін бағалау  .  Бас бармақ арқылы өзін-өзі |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | таңдаңыз  print ("randrange(1000) : ", random.randrange(1000)) **қорытынды**  Жоғарыдағы бағдарламаны іске қосқан кезде ол келесі нәтиже береді: randrange(1,1000, 2) : 456  randrange(1000) : 648 |  | бағалау  .  **«Мада қтау»** |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  «Күн бұлт» әдісімен үш топқа ақынның өлеңіне сүйене отырып **«**Постер» құру арқылы мазмұнын талдау ұсынылады. Әр топқа А4парағында күннің және бұлттың сызбалары беріледі. Адамның мінез- құлқына қатысты пікірлерді өлеңдегі сөз тіркестері немесе сөйлемдерге сүйеніп жазу ұсынылады.  *Саналы адамға қатысты пікірлер* **–**  күнге орналастыру  Балалық шағын бағалайды. Патшаның тағы деп ойлайды.Уақытын текке өткізбейді. Жас кезінде білім, өнер үйренеді. Балалық шақты бағалауға тіл жетпейді. Жастық-алтын деп есептейді. Бағасыз деп ойлайды.  *Санасыз адамға қатысты пікірлер* – бұлтқа жазу  Бос жүреді. Уақытын бағаламайды. Уақытын еркелеп бос өткізеді. |  |  |  |

|  |  |  |
| --- | --- | --- |
| **№ 18 сабақ** | **Курс атауы:** «Python программалау тілінің  негіздері» | **Сыныбы: 9 Күні: Мұғалім:** |
| **Тақырыбы:** Random функциясы. | | |
| **Оқу мақсаты:** | Random функциясы туралы түсіндіру | |
| **Сабақ мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | |
| **Жоспар** | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/0ee2/00048f39-ce0c0421/img3.jpg | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану** | Оқушы | Отшаш |  |
|  | Жұмыстың басында бағдарламаға | лар | у |
|  | random модулін импорттау қажет. | сұраққ | арқылы |
|  | Осыдан кейін ғана оны толығымен | а | бір- |
|  | пайдалануға болады. Random | жауап | бірін |
|  | модулін импорттау операторы | береді, | бағалау |
|  | келесідей: | жұмыс | . |
|  |  | ты |  |
|  | Python | берілге |  |
|  |  | н |  |
|  | 1 | түріне |  |
|  | import random | қарай |  |
|  | Енді қарапайым мысалда random | орында |  |
|  | модулін қолдануды қарастырыңыз: | йды |  |
|  | Python |  |  |
|  | 1 |  |  |
|  | 2  3  4  5  import random |  | Бас бармақ арқылы өзін-өзі |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | басып шығару ("кездейсоқ санды |  | бағалау |  |
| random көмегімен | . |
| шығару.random()") |  |
| print(random.random()) | **«Мада** |
| **Қорытынды:** | **қтау»** |
| Shell |  |
| 1 |  |
| 2 |  |
| Кездейсоқ санды random көмегімен |  |
| шығару.random() |  |
| 0.9461613475266107 |  |
| Көріп отырғаныңыздай, біз |  |
| 0.9461613475266107 алдық. Әрине, |  |
| сізде кездейсоқ басқа сан болады. |  |
| random () - random Модулінің |  |
| негізгі функциясы; |  |
| Random Модулінің барлық дерлік |  |
| функциялары random () негізгі |  |
| функциясына байланысты; |  |
| random () [0.0, 1.0] аралығындағы |  |
| өзгермелі нүктелі келесі кездейсоқ |  |
| санды қайтарады. |  |
| Random Модулінің функцияларын |  |
| талдаудан бұрын оларды |  |
| қолданудың негізгі бағыттарын |  |
| қарастырайық. |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://ds04.infourok.ru/uploads/ex/07d1/00137c48-66bf92d5/img14.jpg |  |  |  |

|  |  |  |
| --- | --- | --- |
| **№ 19 сабақ** | **Курс атауы:** «Python программалау тілінің  негіздері» | **Сыныбы: 9 Күні: Мұғалім:** |
| **Тақырыбы:** Функциялар құру | | |
| **Оқу мақсаты:** | Python бағдарламалау тілінде функциялар және файлдармен  жұмыс жасауды *қарастыру.* | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Сабақ мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | |
| **Жоспар** | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/0bcd/0001cd04-dd239b8d/img0.jpg | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану**  Бұнда атаулы мен анонимді функциялар және def, return және lambda интрукциялары қарастырылады. Сонымен қатар, функцияның міндетті және міндетті емес аргументтері, еркін санды аргументтері бар функциялар қарастырылады.  Атаулы функциялар, def инструкциясы. Python тіліндегі функция аргументтер қабылдайтын және мәнді қайтаратын нысан болып табылады. Әдетте функция def инструкциясы көмегімен анықталады.  Қарапайым функцияны анықтайық: def add(x, y):  return x + y | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай орында йды | Отшаш у арқылы бір- бірін бағалау  .  Бас |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | return Инструкция мәнді қайтаруды талап етеді. Ал аталған жағдайда функция x пен y қосындысын қайтарады. Енді оны шақыруға болады  >>> add(1, 10)  11  >>> add('abc', 'def') 'abcdef'  Функция кез-келген күрделілікте болуы және барлық нысандарды (тізімдер, кортеждерді және тіпті функцияларды) қайтаруы мүмкін:  >>> def newfunc(n):  ... def myfunc(x):  ... return x + n  ... return myfunc  ...  >>> new = newfunc(100) *# new - бұл функция*  >>> new(200) 300  Функция return инструкциясымен аяқталмауы да мүмкін, бірақ соның өзінде функция [None](https://pythonworld.ru/tipy-dannyx-v-python/none.html) мәнін қайтарады:  >>> def func():  ... pass  ...  >>> print(func()) None  Функция аргументтері. Функция кез келген сандағы аргументтерді қабылдауы мүмкін немесе мүлдем қабылдамауы да мүмкін. Сондай- ақ, аргументтер саны кез келген, позициялық және атаулы аргументтері бар функциялар, міндетті және міндетті емес функциялар кең таралған.  >>> def func(a, b, c=2): # c – мінддетті емес аргумент  ... return a + b + c  ... |  | бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | >>> func(1, 2) # a = 1, b = 2, c = 2  (үнсіз жағдайда) 5  >>> func(1, 2, 3) # a = 1, b = 2, c = 3  6  >>> func(a=1, b=3) # a = 1, b = 3, c  = 2  6  >>> func(a=3, c=6) # a = 3, c = 6, b анықталмаған  Traceback (most recent call last):  File "", line 1, in func(a=3, c=6)  TypeError: func() takes at least 2 arguments (2 given)  Функция позициялық аргументтердің айнымалы санын қабылдауы мүмкін, ондай жағдайда оның атауының алдына \* қойылады:  >>> def func(\*args):  ... return args  ...  >>> func(1, 2, 3, 'abc')  (1, 2, 3, 'abc')  >>> func() ()  >>> func(1) (1,)  Мысалдан көргендей, функцияның барлық берілген аргументтерінің ішінен args – бұл [кортеж](https://pythonworld.ru/tipy-dannyx-v-python/kortezhi-tuple.html), сонымен қатар айнымалының өзімен де кортежбен сияқты жұмыс істеуге болады.  Функция атаулы аргументтердің кез келген санын қабылдай алады, онда атудың алдына \*\* белгісі қойылады:  >>> def func(\*\*kwargs):  ... return kwargs  ...  >>> func(a=1, b=2, c=3)  {'a': 1, 'c': 3, 'b': 2} |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | >>> func()  {}  >>> func(a='python')  {'a': 'python'}  Kwargs айнымалысында барлық әрекеттерді атқаруға болатын сөздік сақталады.  Анонимді функциялар, lambda инструкциясы. Анонимді функциялар тек бір ғана өрнекті қамтығандықтан тез орындалады. Анонимді функциялар lambda инструкциясының көмегімен құрылады. Сонымен қатар, оларды def func() инструкциясындағы сияқты міндетті түрде айнымалыларға меншіктеудің қажеті жоқ:  >>> func = lambda x, y: x + y  >>> func(1, 2)  3  >>> func('a', 'b') 'ab'  >>> (lambda x, y: x + y)(1, 2) 3  >>> (lambda x, y: x + y)('a', 'b') 'ab'  lambda функциясы басқа функцияларға қарағанда return инструкциясын қажет етпейді, ал басқа жағынан еш айырмашылығы жоқ:  >>>  >>> func = lambda \*args: args  >>> func(1, 2, 3, 4)  (1, 2, 3, 4)  Файлдар. Файлдармен жұмыс. Төменде python тілінің файлдармен әрекет жасауға арналған ендірілген құралдары қарастырылады: ашу / жабу, оқу және жазу.  Файлмен жұмыс жасау үшін алдымен оны ашып алу керек. Ол үшін open ендірілген функциясы |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | қолданылады:  f = open('text.txt', 'r')  Open функциясының көптеген өлшемдері бар, бірақ олардың ішінде ең қажетті 3 аргументі бар. Олардың біріншісі файлдың аты, файлға баратын жолдың өзі салыстырмалы немесе абсолютті болуы мүмкін. Екінші аргумент –ол файлды ашу режімі.  **Тапсыра**  Файлға жазу*.* Файлға келесі тізімді жазуды орындап көрейік:  >>> l = [str(i)+str(i-1) for i in range(20)]  >>> l  ['0-1', '10', '21', '32', '43', '54', '65', '76',  '87', '98', '109', '1110', '1211', '1312',  '1413', '1514', '1615', '1716', '1817',  '1918']  Файлды жазу үшін ашу:  >>> f = open('text.txt', 'w')  Файлға жазу write әдісі арқылы жүзеге асырылады. Ол келесі 26- суретте көрсетілген:  >>> for index in l:  ... f.write(index + '\n')  ... 4  3  3  3  3 |  |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://doc4web.ru/uploads/files/253/254800/hello_html_246081cf.jpg |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 20 сабақ** | **Курс атауы:** «Python программалау тілінің  негіздері» | **Сыныбы: 9 Күні:**  **Мұғалім:** | | | |
| **Тақырыбы:** Жергілікті және ғаламдық айнымалылар. | | | | | |
| **Оқу мақсаты:** | Жергілікті және ғаламдық айнымалылар туралы ақпарат беру | | | | |
| **Сабақ**  **мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/0c6c/0011718b-fdaec27a/img1.jpg | | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану**  Ғаламдық айнымалылар   * Бағдарламаның кез-келген жеріне қол жеткізе алатын айнымалылар ғаламдық айнымалы деп аталады. * Айнымалыларды бағдарламаның басында немесе ғаламдық кілт сөзді қолдану арқылы атауға болады.   1. g\_v = «ғаламдық айнымалы» def m (): print (g\_v) m () print (g\_v) - | | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай | Отшаш у арқылы бір- бірін бағалау  . |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | - def m1 ():  ғаламдық g\_v g\_v = 12m1 () # өзгермелі m1 ішінде глобалды анықталған, сондықтан оны шақырыңыз .. басып шығару (g\_v) Жергілікті айнымалы -   * Анықталған жерде ғана қол жеткізуге болатын айнымалы.   1. a = 'ғаламдық айнымалы m (): a =' жергілікті айнымалы'm () басып шығару (a)  Статикалық айнымалы  & дананың айнымалы мәні | орында йды | Бас бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://ds05.infourok.ru/uploads/ex/1377/00108e36-6373b5df/hello_html_m1bc64632.png |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 21 сабақ** | **Курс атауы:** «Python программалау тілінің  негіздері» | **Сыныбы: 9 Күні: Мұғалім:** | | | |
| **Тақырыбы:** Lambda нұсқаулары. Функцияларды қолдана отырып есептерді  шешудің мысалдары. | | | | | |
| **Оқу мақсаты:** | Lambda нұсқаулары. Функцияларды қолдана отырып есептерді  шешудің мысалдарын келтіру. | | | | |
| **Сабақ мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал** | | Психол огиялы қ ахуалға берілге | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | https://ds04.infourok.ru/uploads/ex/05e9/000e9860-98f6b745/img2.jpg | н |  | Карточк |
| өлеңді | алар |
| дауыст | Презент |
| ап | ация |
| айтып, |  |
| тілді |  |
| жаттық |  |
| тыру |  |
| және |  |
| жаңа |  |
| сабаққа |  |
| назар |  |
| аудару. |  |
| **Ортасы** | Мағынаны тану  Lambda-бұл Python және басқа бағдарламалау тілдеріндегі анонимді функцияларды шақыруға арналған құрал. Бұл көптеген адамдарға ештеңе айтпайды және оның қалай жұмыс істейтінін түсіндірмейді, сондықтан Мен сізге Lambda өрнектерінің жұмыс механизмін айтамын.  Барлығы өте қарапайым. Мысалды қарастырайық. Мысалы, шеңбердің ауданын белгілі радиуста есептейтін функцияны жазу керек. | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай орында йды | Отшаш у арқылы бір- бірін бағалау  . |  |
|  | Шеңбер аймағының формуласы S = pi\*(r\*\*2)  онда  S-шеңбердің ауданы pi-біз Math стандартты  кітапханасынан алатын 3.14-ке тең математикалық тұрақты  R-шеңбердің радиусы-біз функциямызға беретін жалғыз айнымалы  біз мұның бәрін python-да жасаймыз:  import math #math кітапханасын қосыңыз |  | Бас бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | pi\_const = round(math.pi, 2) # Pi-ді үтірден кейінгі екінші белгіге дөңгелектеңіз, әйтпесе ол 3.141592653589793 сияқты болады және бұл бізге ыңғайсыз болады  #**тапсырма**  Біз әдеттегі жазба нұсқасында шеңбердің ауданын берілген радиус бойынша есептейтін функцияны жазамыз  def area\_of\_circle\_simple(radius): return pi\_const\*(radius\*\*2)  print(area\_of\_circle\_simple(5)) print(area\_of\_circle\_simple(12)) print(area\_of\_circle\_simple(26))  >>>78.5  >>>452.16  >>>2122.64  Бұл жаман емес сияқты, бірақ егер сіз оны lambda арқылы жазсаңыз, бәрі салқын болып көрінуі мүмкін: import math #math кітапханасын қосыңыз  pi\_const = round(math.pi, 2) # Pi екінші белгіге дейін дөңгелектеңіз # үтірден кейін ол басқаша көрінеді # 3.141592653589793 сияқты және бұл бізге ыңғайсыз болады print((lambda radius: pi\_const\*(radius\*\*2))(5)) print((lambda radius: pi\_const\*(radius\*\*2))(12)) print((lambda radius: pi\_const\*(radius\*\*2))(26))  >>>78.5  >>>452.16  >>>2122.64  Түсінікті болу үшін функцияның |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | анонимді шақыруы оны жоғарыда келтірілген мысалдағыдай еш жерде жарияламай-ақ қолданғаныңызды білдіреді.  Ламбда функциясы келесі принцип бойынша жұмыс істейді  басып шығару ((lambda үтір арқылы келтірілген дәлелдер: олармен не істеліп жатыр) (дәлелдерді беру))  >> > біз жоғарыдағы жолдың қос нүктесінен кейін нәтиже аламыз  Екі кіріс дәлелдері бар мысалды қарастырыңыз. Мысалы, конустың көлемін келесі формула бойынша есептеу керек:  V = (height\*pi\_const\*(radius\*\*2))/3 |  |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://ds04.infourok.ru/uploads/ex/1007/0013e626-c54eeffe/img25.jpg |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 22 сабақ** | **Курс атауы:** «Python программалау тілінің  негіздері» | **Сыныбы: 9 Күні:**  **Мұғалім:** | | | |
| **Тақырыбы:** Рекурсивті функциялар. | | | | | |
| **Оқу мақсаты:** | Рекурсивті функциялар туралы түсінік қалыптастыру. | | | | |
| **Сабақ**  **мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  **«Сиқырлы айналу» жылулық шеңбері. «Мен және аспан»** Педагог балалардың назарын аспанның әдемілігіне аударады.Балалар аспандағы бұлттарды түрлі заттарға ұқсатып айтады;үлпілдік ақша бұлттаға,жауар бұлттай түнерген аспанға,мейірімді жылы күнге,ай мен жұлдыздарға,қарлығашқа,бозторға йға,шымшыққа,бүркітке т.б.айналады. «Жалаулар» арқылы топқа бөлінеді. Оқушыларды қалаған жалауларын алып, 1,2,3 топтарға бөлініп отырады. | | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану**  Рекурсивті функция — бұл өзін-өзі қоздыратын функция.  Қарапайым мысал ретінде келесі кодты қарастырыңыз:  Көшіруге  def factorial\_recursive(n): if n == 1:  return n | | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне | Отшаш у арқылы бір- бірін бағалау  . |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | else:  return n\*factorial\_recursive(n-1)  Мұнда рекурсивті функцияны шақыру және оған бүтін санды беру арқылы сіз сол санның факториалын аласыз (n!).  Факториалдар туралы қысқаша Санның факторлық мәні-әр алдыңғы санға 1-ге көбейтілген Сан.  Мысалы, 7 санының факторы: 7! = 7\*6\*5\*4\*3\*2\*1 = 5040  Функцияның көмегімен санның факторлық мәнін шығаруға болады:  Көшіруге num = 3  басып шығару(f"факторлық {num} бұл {factorial\_recursive (num)}") Бұл функция "Факториал 3-6" деп көрсетеді. Осы рекурсивті функцияны тағы бір рет қарастырыңыз:  def factorial\_recursive(n):  ...  Кәдімгі функцияға ұқсас, рекурсивті атау def-тен кейін, ал жақшада N параметрі көрсетілген: def factorial\_recursive(n):  if n == 1:  return n else:  return n\*factorial\_recursive(n-1) Шартты құрылымның арқасында N айнымалы мәні 1-ге тең болған жағдайда ғана оралады. Мұны аяқтау шарты деп те атайды.  Рекурсия шарттарды қанағаттандыру сәтінде тоқтайды. | қарай орында йды | Бас бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | def factorial\_recursive(n):  if n == 1:  return n else:  return n\*factorial\_recursive(n-1) Жоғарыдағы кодта рекурсияның фрагменті көрсетілген. Шартты құрылымның else блогында N көбейтіндісі және сол функцияның мәні N-1 параметрімен қайтарылады.  Бұл рекурсия. Біздің мысалда ол осылай жұмыс істеді:  3 \* (3-1) \* ((3-1)-1) # 3-1-1 1-ге тең  болғандықтан, рекурсия тоқтады Бөлшектер жұмыс рекурсивной функциялары  Мұның қалай жұмыс істейтінін жақсы түсіну үшін 3 параметрімен функцияны орындау процесін кезеңдерге бөлеміз.  Мұны істеу үшін төменде нақты сандары бар әр дананы ұсынамыз. Бұл аргумент ретінде 3 мәні бар бір функцияны шақырған кезде не болатынын "бақылауға" көмектеседі:  Көшіруге  # Бірінші қоңырау factorial\_recursive(3):  if 3 == 1:  return 3 else:  return 3\*factorial\_recursive(3-1)  # Екінші қоңырау factorial\_recursive(2):  if 2 == 1:  return 2 else:  return 2\*factorial\_recursive(2-1)  # Үшінші қоңырау |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | factorial\_recursive(1):  if 1 == 1:  return 1 else:  return 1\*factorial\_recursive(1-1) Рекурсивті функция 3\*factorial\_recursive(3-1) өрнегінің жауабын білмейді, сондықтан ол стекке тағы бір қоңырау қосады. |  |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://ds04.infourok.ru/uploads/ex/0fb3/0002943a-3660addc/1/img16.jpg |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 23 сабақ** | **Курс атауы:** «Python программалау тілінің  негіздері» | **Сыныбы: 9 Күні:**  **Мұғалім:** | | | |
| **Тақырыбы:** Есептеу факториалы. Фибоначчи Сандары. | | | | | |
| **Оқу мақсаты:** | Есептеу факториалы. Фибоначчи Сандары туралы ақпарат беру. | | | | |
| **Сабақ мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/0651/00172448-ae79a9ba/img2.jpg | | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. |  |  |
| **Ортасы** | **Мағынаны тану**  Фибоначчи сандары – әрбір келесі мүшесі алдыңғы екі мүшесінің қосындысына тең болатын 1, 1, 2,  3, 5, 8, 13, … қайталама сан тізбегінің (Фибоначчи қатары) элементтері. Фибоначчи сандарының рекурренттік қатынастары  F 0 = 0 , F 1 = 1 , F n = F n − 1 + F n  − 2 , n ⩾ 2. {\displaystyle F\_{0}=0,\qquad F\_{1}=1,\qquad F\_{n}=F\_{n-1}+F\_{n-2},\quad  n\geqslant 2.}  арқылы беріледі. Фибоначчи сандарын 1202 жылы италиялық математик [Леонардо Пизанский](https://kk.wikipedia.org/wiki/%D0%9B%D0%B5%D0%BE%D0%BD%D0%B0%D1%80%D0%B4%D0%BE_%D0%9F%D0%B8%D0%B7%D0%B0%D0%BD%D1%81%D0%BA%D0%B8%D0%B9) ([Фибоначчи](https://kk.wikipedia.org/wiki/%D0%A4%D0%B8%D0%B1%D0%BE%D0%BD%D0%B0%D1%87%D1%87%D0%B8)) тапқан.  Бине формуласы[[өңдеу](https://kk.wikipedia.org/w/index.php?title=%D0%A4%D0%B8%D0%B1%D0%BE%D0%BD%D0%B0%D1%87%D1%87%D0%B8_%D1%81%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%8B&action=edit&section=1)]  [Бине](https://kk.wikipedia.org/w/index.php?title=%D0%91%D0%B8%D0%BD%D0%B5%2C_%D0%96%D0%B0%D0%BA_%D0%A4%D0%B8%D0%BB%D0%B8%D0%BF%D0%BF_%D0%9C%D0%B0%D1%80%D0%B8&action=edit&redlink=1) формуласы F n {\displaystyle  F\_{n}} мүшелерін nге қатысты функция ретінде өрнектейді:  F n = ( 1 + 5 2 ) n − ( 1 − 5 2 ) n 5 = φ  n − ( − φ ) − n φ − ( − φ ) − 1 = φ n − ( − φ ) − n 2 φ − 1 {\displaystyle F\_{n}={\frac {\left({\frac {1+{\sqrt  {5}}}{2}}\right)^{n}-\left({\frac {1-  {\sqrt {5}}}{2}}\right)^{n}}{\sqrt  {5}}}={\frac {\varphi ^{n}-(-\varphi  )^{-n}}{\varphi -(-\varphi )^{-  1}}}={\frac {\varphi ^{n}-(-\varphi  )^{-n}}{2\varphi -1}}} , | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай орында йды | Отшаш у арқылы бір- бірін бағалау  .  Бас бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | мұндағы φ = 1 + 5 2 {\displaystyle  \varphi ={\frac {1+{\sqrt {5}}}{2}}}  — [алтын қима](https://kk.wikipedia.org/wiki/%D0%90%D0%BB%D1%82%D1%8B%D0%BD_%D2%9B%D0%B8%D0%BC%D0%B0). Сонымен қатар  φ {\displaystyle \varphi \,\!} мен ( − φ ) − 1 = 1 − φ {\displaystyle (-  \varphi )^{-1}=1-\varphi \,\!} сипаттауыш x 2 − x − 1 = 0  {\displaystyle x^{2}-x-1=0\,\!} теңдеуінің түбірлері болып табылады.  Бине формуласы бойынша кез келген n ⩾ 0 {\displaystyle  n\geqslant 0} үшін, F n  {\displaystyle F\_{n}} φ n 5  {\displaystyle {\frac {\varphi  ^{n}}{\sqrt {5}}}\,} санына ең жақын [бүтін сан](https://kk.wikipedia.org/wiki/%D0%91%D2%AF%D1%82%D1%96%D0%BD_%D1%81%D0%B0%D0%BD) болып табылады, яғни F n = ⌊ φ n 5 ⌉ {\displaystyle F\_{n}=\left\lfloor {\frac {\varphi  ^{n}}{\sqrt {5}}}\right\rceil } . Жеке түрде, n → ∞ {\displaystyle  n\to \infty } болғанда F n ∼ φ n 5  {\displaystyle F\_{n}\sim {\frac  {\varphi ^{n}}{\sqrt {5}}}} [асимптотика](https://kk.wikipedia.org/w/index.php?title=%D0%90%D1%81%D0%B8%D0%BC%D0%BF%D1%82%D0%BE%D1%82%D0%B8%D0%BA%D0%B0&action=edit&redlink=1) орындалады.  Бине формуласы [аналитикалық](https://kk.wikipedia.org/w/index.php?title=%D0%90%D0%BD%D0%B0%D0%BB%D0%B8%D1%82%D0%B8%D0%BA%D0%B0%D0%BB%D1%8B%D2%9B_%D0%B6%D0%B0%D0%BB%D2%93%D0%B0%D1%81%D1%8B&action=edit&redlink=1) [келесі түрде жалғастыруға болады](https://kk.wikipedia.org/w/index.php?title=%D0%90%D0%BD%D0%B0%D0%BB%D0%B8%D1%82%D0%B8%D0%BA%D0%B0%D0%BB%D1%8B%D2%9B_%D0%B6%D0%B0%D0%BB%D2%93%D0%B0%D1%81%D1%8B&action=edit&redlink=1): F z = 1 5 ( φ z − cos ⁡ π z φ z )  {\displaystyle F\_{z}={\frac {1}{\sqrt  {5}}}\left(\varphi ^{z}-{\frac {\cos  {\pi z}}{\varphi ^{z}}}\right)} Ал F z + 2 = F z + 1 + F z  {\displaystyle  F\_{z+2}=F\_{z+1}+F\_{z}}  теңдігі кез келген [комплекс сан](https://kk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%BB%D0%B5%D0%BA%D1%81_%D1%81%D0%B0%D0%BD) z үшін орындалады. |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Теңдіктер[[өңдеу](https://kk.wikipedia.org/w/index.php?title=%D0%A4%D0%B8%D0%B1%D0%BE%D0%BD%D0%B0%D1%87%D1%87%D0%B8_%D1%81%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%8B&action=edit&section=2)]  F 1 + F 2 + F 3 + ⋯ + F n = F n + 2 −  1 {\displaystyle F\_{1}+F\_{2}+F\_{3}+\dots  +F\_{n}=F\_{n+2}-1}  F 1 + F 3 + F 5 + ⋯ + F 2 n − 1 = F 2  n {\displaystyle F\_{1}+F\_{3}+F\_{5}+\dots +F\_{2n-  1}=F\_{2n}}  F 2 + F 4 + F 6 + ⋯ + F 2 n = F 2 n +  1 − 1 {\displaystyle F\_{2}+F\_{4}+F\_{6}+\dots  +F\_{2n}=F\_{2n+1}-1}  F n + 1 F n + 2 − F n F n + 3 = ( − 1 )  n {\displaystyle F\_{n+1}F\_{n+2}^{}-  F\_{n}F\_{n+3}=(-1)^{n}}  F 1 2 + F 2 2 + F 3 2 + ⋯ + F n 2 = F  n F n + 1 {\displaystyle F\_{1}^{2}+F\_{2}^{2}+F\_{3}^{2}+\  dots +F\_{n}^{2}=F\_{n}F\_{n+1}}  F n 2 + F n + 1 2 = F 2 n + 1  {\displaystyle F\_{n}^{2}+F\_{n+1}^{2}=F\_{2n+1}  }  F 2 n = F n + 1 2 − F n − 1 2  {\displaystyle F\_{2n}=F\_{n+1}^{2}-  F\_{n-1}^{2}}  F 3 n = F n + 1 3 + F n 3 − F n − 1 3  {\displaystyle F\_{3n}=F\_{n+1}^{3}+F\_{n}^{3}-  F\_{n-1}^{3}}  F 5 n = 25 F n 5 + 25 ( − 1 ) n F n 3 +  5 F n {\displaystyle F\_{5n}=25F\_{n}^{5}+25(-  1)^{n}F\_{n}^{3}+5F\_{n}} Жалпы формулалар: |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | F n + m = F n − 1 F m + F n F m + 1  = F n + 1 F m + 1 − F n − 1 F m − 1  {\displaystyle F\_{n+m}^{}=F\_{n- 1}F\_{m}+F\_{n}F\_{m+1}=F\_{n+1}  F\_{m+1}-F\_{n-1}F\_{m-1}}  F ( k + 1 ) n = F n − 1 F k n + F n F k n + 1 {\displaystyle F\_{(k+1)n}^{}=F\_{n-  1}F\_{kn}+F\_{n}F\_{kn+1}}  F n = F l F n − l + 1 + F l − 1 F n − l  {\displaystyle F\_{n}^{}=F\_{l}F\_{n-  l+1}+F\_{l-1}F\_{n-l}} Фибоначчи сандары [континуанталар](https://kk.wikipedia.org/w/index.php?title=%D0%9A%D0%BE%D0%BD%D1%82%D0%B8%D0%BD%D1%83%D0%B0%D0%BD%D1%82%D0%B0&action=edit&redlink=1) мәндері ретінде бірліктер жиынында өрнектеле алады: F n + 1 = K n ( 1 , … , 1 )  {\displaystyle  F\_{n+1}=K\_{n}(1,\dots ,1)} , сол дегеніміз  F n + 1 = det ( 1 1 0 ⋯ 0 − 1 1 1 ⋱ ⋮ 0  − 1 ⋱ ⋱ 0 ⋮ ⋱ ⋱ ⋱ 1 0 ⋯ 0 − 1 1 )  {\displaystyle F\_{n+1}=\det  {\begin{pmatrix}1&1&0&\cdots &0\\-1&1&1&\ddots &\vdots \\0&- 1&\ddots &\ddots &0\\\vdots &\ddots &\ddots &\ddots &1\\0&\cdots &0&-  1&1\end{pmatrix}}} , сонымен қатар F n + 1 = det ( 1 i 0 ⋯ 0 i 1 i ⋱  ⋮ 0 i ⋱ ⋱ 0 ⋮ ⋱ ⋱ ⋱ i 0 ⋯ 0 i 1 )  {\displaystyle \ F\_{n+1}=\det  {\begin{pmatrix}1&i&0&\cdots &0\\i&1&i&\ddots &\vdots  \\0&i&\ddots &\ddots &0\\\vdots &\ddots &\ddots &\ddots &i\\0&\cdots  &0&i&1\end{pmatrix}}} , мұндағы матрицалар өлшемі n × n  {\displaystyle n\times n} , i — [жалған бірлік](https://kk.wikipedia.org/w/index.php?title=%D0%96%D0%B0%D0%BB%D2%93%D0%B0%D0%BD_%D0%B1%D1%96%D1%80%D0%BB%D1%96%D0%BA&action=edit&redlink=1). |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Фибоначчи сандарын [Чебышев](https://kk.wikipedia.org/w/index.php?title=%D0%A7%D0%B5%D0%B1%D1%8B%D1%88%D0%B5%D0%B2_%D0%BA%D3%A9%D0%BF%D0%BC%D2%AF%D1%88%D0%B5%D0%BB%D1%96%D0%BA%D1%82%D0%B5%D1%80%D1%96&action=edit&redlink=1) [көпмүшеліктерімен](https://kk.wikipedia.org/w/index.php?title=%D0%A7%D0%B5%D0%B1%D1%8B%D1%88%D0%B5%D0%B2_%D0%BA%D3%A9%D0%BF%D0%BC%D2%AF%D1%88%D0%B5%D0%BB%D1%96%D0%BA%D1%82%D0%B5%D1%80%D1%96&action=edit&redlink=1) өрнектеуге болады:  F n + 1 = ( − i ) n U n ( − i 2 ) = ( − i )  n T n ( − i ) , {\displaystyle F\_{n+1}=(-i)^{n}U\_{n}\left({\frac {-  i}{2}}\right)=(-i)^{n}T\_{n}(-i),}  F 2 n + 2 = U n ( 3 2 ) = T n ( 3 ) .  {\displaystyle F\_{2n+2}=U\_{n}\left({\frac  {3}{2}}\right)=T\_{n}(3).} |  |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://ds05.infourok.ru/uploads/ex/0b54/0013e173-bc9648df/hello_html_m3997c853.jpg |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 24 сабақ** | **Курс атауы:** «Python программалау тілінің  негіздері» | **Сыныбы: 9 Күні:**  **Мұғалім:** | | | |
| **Тақырыбы:** Құрама деректер түрі-жол. Индекс бойынша қатынау. | | | | | |
| **Оқу мақсаты:** | Құрама деректер түрі-жол. Индекс бойынша қатынау бойынша  түсіндіру | | | | |
| **Сабақ**  **мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/0331/00072f20-267d5c90/img2.jpg | | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану**  Жолдардағы таңбаларға қол жеткізу индекстеу операциясына негізделген-жолға немесе жолға сілтеме жасайтын айнымалының атауынан кейін шаршы жақшада қажетті таңбалардың позицияларының нөмірлері көрсетіледі.  Сондай-ақ, бұл қол жетімділік ығысуға, яғни жолдың сол немесе оң жақ шетінен таңбалардың қашықтығына негізделгенін түсіну | | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай орында | Отшаш у арқылы бір- бірін бағалау  . |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | керек. Бұл қашықтық бүтін сандармен өлшенеді және шын мәнінде жолдағы таңбалардың позициясының нөмірін – олардың индексін анықтайды. Ұстау "бүтін сандармен өлшенеді" деген тіркес болып табылады, яғни индекс оң да, теріс те болуы мүмкін: оң индекстер – сол жақ жиектен, ал теріс – оң жақтан санау. Сонымен қатар, сол жақ жиектен таңбаларды санау басталады  0  0  , ал оң жақтан басталады  − 1  −1  (минус бірлік).  Қарапайым жағдайда сіз бір ерікті таңбаны шығара аласыз:  >>> 'string'[0] 's'  >>>  >>> 'string'[5] 'g'  >>>  >>>  >>> s = 'STRING'  >>>  >>> s[0] 'S'  >>>  >>> s[5] 'G'  Сондай – ақ, біз кесіп аламыз- бастапқы жолдың ішіндегі таңбалар тізбегі, оны кейде ішкі жол деп те атайды:  >>> s = 'AAA\*\*\*BBB^^^'  >>>  >>> s[0:3] 'AAA'  >>> | йды | Бас бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | >>> s[3:6]  '\*\*\*'  >>>  >>> s[9:12]  '^^^'  Сондай-ақ, біз таңбаларды жолдан немесе кесуден көрсетілген қадаммен шығара аламыз:  >>> s = 'a0-b1-c2-d3-e4-f5-g6'  >>>  >>> s[::3]  'abcdefg'  >>>  >>> s[1::3] '0123456'  >>>  >>> s[2::3] ' '  >>>  >>>  >>> s[3:12:3]  'bcd'  >>> s[9:1:-3]  'dcb'  Көріп отырғаныңыздай, бір жағынан бәрі қарапайым болып көрінеді, бірақ екінші жағынан, әсіресе соңғы мысалдарға қарап, әлі де қиындықтар бар екені белгілі болады. Рет-ретімен анықтайық.  Таңбаларды шығару-S [i]  Жолдан бір таңбаны шығару-бұл ең оңай жағдай. Жолды қайтадан қарастырайық:  >>> s = 'STRING'  'S', 'R' және 'G' таңбаларын алу үшін біз келесі қарапайым әрекеттерді орындай аламыз:  >>> s[0], s[2], s[5]  ('S', 'R', 'G')  Бірақ біз индекстер теріс болуы мүмкін екенін білеміз, яғни. жолдың соңынан есептеледі, сол |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | таңбаларды алу үшін біз осындай әрекеттерді орындай аламыз:  >>> s[-6], s[-4], s[-1]  ('S', 'R', 'G')  Теріс және оң таңбалар индекстері жолдың ұзындығы арқылы көрсетілуі мүмкін:  >>> s[4], s[-2]  ('N', 'N')  >>>  >>> s[len(s) - 2], s[4 - len(s)] ('N', 'N')  Жол таңбаларына индекстерді визуалды түрде көрсету келесідей:  https://skr.sh/i/060921/ha9rnwqA.jpg?download=1&name=Скриншот%2006-09-2021%2014:18:10.jpg |  |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://ds04.infourok.ru/uploads/ex/0bad/00004644-b4c59e35/img41.jpg |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 25 сабақ** | **Курс атауы:** «Python программалау тілінің  негіздері» | **Сыныбы: 9 Күні:**  **Мұғалім:** | | | |
| **Тақырыбы:** Оператор in. String модулі. | | | | | |
| **Оқу мақсаты:** | Оператор in. String модулі туралы түсінік беру | | | | |
| **Сабақ**  **мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds05.infourok.ru/uploads/ex/112a/000f5369-09815022/hello_html_m6003f41a.jpg | | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану**  Синтаксис:  x in s  x not in s  In операторы және not in теріске шығару элементтің дәйектілігін тексереді. X in s өрнегі, егер x s мүшесі болса, ШЫН мәнін алады, әйтпесе жалған болады.  X not in s өрнегі s-де x элементінің болуын жоққа шығарады.  Set/frozenset жиынтықтарының барлық кірістірілген реттілігі мен | | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай орында | Отшаш у арқылы бір- бірін бағалау  . |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | түрлері Бұл әрекетті қолдайды, сонымен қатар сөздікте берілген кілттің бар-жоғы тексерілетін сөздік.  Тізім, tuple, set, frozenset, dict немесе collections сияқты контейнер түрлері үшін.deque, өрнек X in y қоңырауға тең кез келген (x is e or x == e for e in Y) Жол және Байт түрлері үшін x in y мәні бар, егер x Y ішкі жолы болса ғана. балама тест y. find (x)!= -1.  Бос жолдар әрқашан кез-келген басқа жолдың ішкі жолы болып саналады, сондықтан " in 'abc' true қайтарады.  Contains () әдісі бар пайдаланушы анықтаған сыныптар үшін x in y өрнегі шын мәнін қайтарады, егер y. contains \_ (x) шын мәнін қайтарса, әйтпесе жалған болады.  Contains () әдісін анықтамайтын, бірақ iter () әдісін анықтайтын пайдаланушы анықтаған сыныптар үшін X in y өрнегі y итерациясында z мәні пайда болса және X is z or X == z өрнегі шын болса, шын мәні болады.  Соңында, ескі стильдегі Итерация протоколы қолданылады: егер класс getitem () әдісін анықтаса, X in Y өрнегі теріс емес бүтін сан индексі болған кезде ғана қайтарылады, мысалы x is y[i] or x  == y[i] және кіші бүтін сандардың ешқайсысы IndexError-дан ерекшеленбейді.  Not in операторы кері шындық in мәніне ие.  **Тапсырма Бірлесіп орындау**  In қолдану мысалдары, элементтің | йды | Бас бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | болуын тексеру:  Тізімге кіру / элементтің болуы, көп  >>> x = [1, 2, 3, 4, 5, 6, 7, 8]  >>> 5 in x # True  >>> 5 not in x # False  >>> 0 in x # False  >>> 0 not in x # True  Кіру / жолда ішкі жолдың болуы   * > > x = 'бас тартуды қайтарады элементтің болуы' * > > 'теріске шығару' in x # True * > > 'болуы' not in X # False   >>> x = 'абракадабра'   * > > 'B' in X   # True   * > > 'B' not in X # False * > > 'n' in x # False * > > 'у' not in X # True   Сөздікке кіру / кілттің болуы  >>> x = {"бір": 1, "екі":2, "үш":3, "төрт": 4}   * > > x-тегі" бір" # Шындық * > > "бір" x емес # Өтірік * > > x-тегі "бес" # Өтірік * > > "бес" x емес |  |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия** |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | https://ds04.infourok.ru/uploads/ex/07d1/00137c48-66bf92d5/img14.jpg |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№ 26 сабақ** | **Курс атауы:** «Python программалау тілінің  негіздері» | **Сыныбы: 9 Күні: Мұғалім:** | | | | | | |
| **Тақырыбы:** Тізбектің барлық түрлеріне арналған операторлар (жолдар,  тізімдер, түйіндер). Мәселелерді жолдармен шешудің мысалдары. | | | | | | | | |
| **Оқу мақсаты:** | Жолдар, тізімдер, түйіндер бойынша есептерді орындау жолын  түсіндіру. | | | | | | | |
| **Сабақ мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | | | | | |
| **Жоспар** | | | | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | | **Бағала у** | | **Ресурст ар** | |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds05.infourok.ru/uploads/ex/1325/000cbd54-a55addd2/hello_html_2fcf0102.png | | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. | | Мұғалі мнің ауызша бағалау ы | | Жұмыс дәптері  Суретте р Карточк алар Презент ация | |
| **Ортасы** | **Мағынаны тану**  кодта көрсетілген таңбалар жол ретінде түсіндірілуі үшін оларды тырнақшаға | | | Оқушы лар сұраққ | | Отша шу арқыл | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | орау керек. Мұны істеудің 4 әдісі бар:  – одинарные бағамдар   * Қос тырнақша * үштік бір тырнақша (пішімдеуді сақтай отырып, көп жолды мәтін) * үш есе Қос тырнақша (пішімдеуді сақтай отырып, көп жолды мәтін)   Мысал-Интерактивті режим  >>> 'Одинарные бағамдар' 'Бір тырнақша'  >>> "Қос бағамдар" Қос тырнақша   * > > "'Үштік одинарные тырнақша"'   'Үштік \N Бір \пкавычки'  >>> """  Үштік Қос  бағамдар"""  '\N үш еселік \пдвойлық \n\пкавычки'  Қосу немесе біріктіру операциясы екі немесе одан да көп жолды бір жолға бүктейді. Дәлелдер тек жолдар бола алады.  Мысал-Интерактивті режим  >>> 'Hello ' + 'Py' 'Hello Py'  >>> 'Hello ' + 4  TypeError: can only concatenate str (not "int") to str  Жолдарға қолданылатын көбейту әрекеті оны бірнеше рет қайталауды білдіреді (параметр ретінде берілген санның мәніне сәйкес). Егер сіз теріс санды немесе нөлді берсеңіз, онда бос жол оралады).  Мысал-Интерактивті режим | а жауап береді, жұмыс ты берілге н түріне қарай орында йды | ы бір- бірін бағала у.  Бас барма қ арқыл ы өзін- өзі  бағала у.  **«Мад ақтау**  **»** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | >>> 'Hello ' \* 4  'Hello Hello Hello Hello '  >>> 'Hello ' \* -4 |  |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  «Бес саусақ» тәсілі Оқушылар параққа алақандарын қойып, оны қаламмен айналдырып өтеді. Осыдан соң олар мынадай сұрақтарға жауап беріп, салынған саусақтарына жазады. Бас бармақ – басты мәселе. Бүгін сабақта ең құнды мәселе қандай болды? Балалы үйрек- бірлесу. |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 27 сабақ** | **Курс атауы:** «Python  программалау тілінің негіздері» | **Сыныбы: 9 Күні: Мұғалім:** | | | |
| **Тақырыбы:** Тізімдер. Тізім түрі (тізім). Индекстер. Тізімдерге арналған  операторлар. | | | | | |
| **Оқу мақсаты:** | Тізімдерге арналған операторлар туралы түсіндіру. | | | | |
| **Сабақ мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/08fb/0000cc4a-d204bcbd/img2.jpg | | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | жаңа сабаққа назар аудару. |  |  |
| **Ортасы** | **Мағынаны тану**  Тізімдегі мәндерді in операторы арқылы іздеу • in операторы керекті мән тізімде бар/жоқ екендігін тексереді. Мәнді іздейтін in операторы қосылған өрнек форматы: мән in тізім • Бұл форматтағы мән – ізделетін мән, тізім – ішінен мән ізделетін тізім. Егер мән табылса, өрнек ақиқат, әйтпесе жалған мәнін береді.  Келесі программада осыған мысал келтірілген: # тізімге in операторын қолдану. def main() : # заттар нөмірінің тізімін жасау. prod\_nums  = ['V475', 'F987', 'Q143', 'R688'] #  Керекті заттың нөмірін алу. search  = input('Зат нөмірін енгізіңіз: ') # Тізімде зат нөмірі бар екенін анықтау. if search in prod\_nums: print(search, 'тізімнен табылды.') else: print(search, 'тізімнен табылмады.') main () # Басты функцияны шақыру.  **Сұраққа жауап берейік**  7.14. Төмендегі код фрагментінің нәтижесі қандай? names = ['Жамал', 'Жора', 'Дулат' , 'Денис'] if 'Дулат' not in names: print('Не могу найти Дулат табылмады. ') else: print("Ceмья Дулаттың отбасы:") print(names) | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай орында йды | Отшаш у арқылы бір- бірін бағалау  .  Бас бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия** |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | https://ds04.infourok.ru/uploads/ex/04eb/0012064a-1660c0ce/img16.jpg |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 28 сабақ** | **Курс атауы:** «Python программалау тілінің  негіздері» | **Сыныбы: 9 Күні:**  **Мұғалім:** | | | |
| **Тақырыбы:** Range функциясы. Тізімдер: мәселелерді шешу мысалдары. | | | | | |
| **Оқу мақсаты:** | Тізімдер: мәселелерді шешу мысалдар келтіру. | | | | |
| **Сабақ**  **мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/10c8/0007d8f7-87919f29/img1.jpg | | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану**  Сонымен, Python функциясы range деп қалай жұмыс істейді?  Қарапайым сөзбен айтқанда, range () берілген ауқым аясында бірқатар сандарды құруға мүмкіндік береді. Функцияларды қанша дәлелдейтіндігіңізге байланысты, сіз осы сандар қатарының қай жерде басталып, аяқталатынын, сондай-ақ екі санның арасындағы айырмашылық қаншалықты үлкен болатынын шеше аласыз. | | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай орында | Отшаш у арқылы бір- бірін бағалау  . |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Міне, іс-әрекеттегі range() шағын мысалы:  Python 1  2  3  for i in range(3, 16, 3): quotient = i / 3  басып шығару (f " {i} 3-ке бөлінеді, нәтиже {Int (quotient)}.") Бұл циклде сіз үшке көбейтілген сандар қатарын жасай аласыз, сондықтан олардың әрқайсысын жеке енгізудің қажеті жоқ.  Назар аударыңыз: Бұл мысалда range () дұрыс қолданылуы көрсетілгенімен, оны циклдарда қолдану үшін жиі алып келеді. Мысалы, range() келесі қолданылуын Питоникалық деп атауға болмайды (бұл жаман мысал):  captains = ['Janeway', 'Picard', 'Sisko']  for i in range(len(captains)): print(captains[i])  range () қайталанатын сандарды құру үшін өте жақсы, бірақ егер сіз in операторымен байланыстырылуы мүмкін деректерді сұрыптау қажет болса, бұл ең жақсы таңдау емес.  Range () қоңырау шалудың үш әдісі бар:  range (тоқтату) бір дәлел алады range (бастау, тоқтату) екі дәлел алады  range (бастау, тоқтату, қадам) үш дәлел алады  Бір аргументпен range() шақыру арқылы сіз 0-ден басталып, әр санға дейінгі сандарды қосасыз, | йды | Бас бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | бірақ сіз соңғы (тоқтату) деп белгілеген санды қоспайсыз. Бұл іс жүзінде қалай көрінеді:  Python 1  2  for i in range(3): print(i) |  |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://cf2.ppt-online.org/files2/slide/z/Zt9aedfocnXh15jxGsiTlKYNFV0AEIORyBrgvCq7b/slide-6.jpg |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 29 сабақ** | **Курс атауы:** «Python программалау тілінің  негіздері» | **Сыныбы: 9 Күні: Мұғалім:** | | | |
| **Тақырыбы:** Матрицалар. Python тізімінің генераторлары. | | | | | |
| **Оқу мақсаты:** | Сандық алгоритмдер мен матрицалық есептеулерді *қарастыру.* | | | | |
| **Сабақ мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds05.infourok.ru/uploads/ex/12f8/00057daf-0ce8523e/img2.jpg | | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | тілді жаттық тыру және жаңа сабаққа назар аудару. |  |  |
| **Ортасы** | **Мағынаны тану**  Бұл бөлім жеткілікті дәрежеде ескірген. Қазір модуль numpy деп аталады. Жалпы, Numeric-те болған барлық numpy-да қол жетімді, бірақ аттар сәйкес келмеуі мүмкін. Numeric NumPy/oldnumeric сияқты қол жетімді. Сіздің құжаттама.  Numeric Python - көптеген сандық қосымшалар үшін қажетті көп өлшемді массивтерді есептеу үшін бірнеше модуль. Numeric модулі MatLab, Octave (MATLAB аналогы), APL, J, S+, ID сияқты пакеттер мен жүйелердің Python мүмкіндіктерін енгізеді. Пайдаланушылар Numeric-ті оңай және ыңғайлы табады. Кейбір Python синтаксистік мүмкіндіктері (кесіктерді пайдаланумен байланысты) Numeric үшін арнайы әзірленген.  Numeric Python үшін құралы бар:   * матрицалық есептеулер; * FFT Фурье жылдам түрлендіру; * жетіспейтін ma   эксперименталдық деректермен жұмыс істеу;   * RNG статистикалық модельдеу; * MatLab бағдарламасының негізгі функцияларының эмуляциясы.   1. Numeric Модулі.  Numeric модулі толымды түрді | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай орында йды | Отшаш у арқылы бір- бірін бағалау  .  Бас бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | анықтайды-массив және массив операцияларына арналған функциялардың көп санын қамтиды. Массив-бұл индекстер бойынша қолжетімді біртекті элементтер жиынтығы. Numeric Модулінің массивтері көп өлшемді, яғни бір мөлшерден артық болуы мүмкін.  2. Массивті құру.  Массивті жасау үшін, массивтің мазмұнын (ішкі тізімдер түрінде) және түрін көрсете отырып, array () функциясын пайдалануға болады. Array() функциясы дәлел - массив болса, көшірмесін жасайды. As array() функциясы ұқсас жұмыс істейді, бірақ оның аргументі массиві болған кезде жаңа массив жасамайды:  >>> from Numeric import \*  >>> print array([[1, 2], [3, 4], [5, 6]])  [[1 2]  [3 4]  [5 6]]  >>> print array([[1, 2, 3], [4, 5, 6]], float)  [[ 1. 2. 3.]  [ 4. 5. 6.]]  >>> print array([78, 85, 77, 69, 82,  73, 67], 'c')  [N U M E R I C]  Массив элементтері ретінде келесі түрлерді пайдалануға болады: Int8- Int32, Unsigned Int 8-Unsigned Int32, Float 8-Float64, Complex8- Complex64 және PyObject. 8, 16, 32 және 64 сандар шаманы сақтауға арналған биттердің санын көрсетеді. Int, Unsigned Integer, Float және Complex түрлері осы платформада қабылданған |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | мәндерге сәйкес келеді . Массивте сондай-ақ еркін нысандарға сілтемелерді сақтауға болады .  Әрбір ось бойынша массив мөлшері мен ұзындығы массив формасы (shape) деп аталады . Массив пішініне қол жеткізу shape атрибуты арқылы жүзеге асырылады:  >>> from Numeric import \*  >>> a = array(range(15), int)  >>> print a.shape (15,)  >>> print a  [ 0 1 2 3 4 5 6 7 8 9 10 11 12 13  14]  >>> a.shape = (3, 5)  >>> print a.shape (3, 5)  >>> print a  [[ 0 1 2 3 4]  [ 5 6 7 8 9]  [10 11 12 13 14]] |  |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://ds04.infourok.ru/uploads/ex/0fb3/0002943a-3660addc/1/img16.jpg |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 30 сабақ** | **Курс атауы:** «Python программалау тілінің  негіздері» | **Сыныбы: 9 Күні: Мұғалім:** | | | |
| **Тақырыбы:** Кортеждер. Кортеждерді тағайындау. | | | | | |
| **Оқу мақсаты:** | Кортеждерді тағайындау | | | | |
| **Сабақ мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың** | **Педагогтің әрекеті** | | **Оқуш** | **Бағала** | **Ресурст** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **кезеңдері** |  | **ының әрекеті** | **у** | **ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/0c1f/000ed92e-a324db43/img0.jpg | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану Python-да кортеждер**  Кортеж — бұл өзгермейтін тізім. Кортежді оны жасағаннан кейін ешқандай жолмен өзгерту мүмкін емес.  >>> a\_tuple = ("a", "b", "mpilgrim",  "z", "example") (1)  >>> a\_tuple  ('a', 'b', 'mpilgrim', 'z', 'example')  >>> a\_tuple[0] (2)  'a'  >>> a\_tuple[-1] (3)  'example'  >>> a\_tuple[1:3] (4)  ('b', 'mpilgrim')  1. Кортеж тізім сияқты анықталады, тек элементтер | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай орында йды | Отшаш у арқылы бір- бірін бағалау  .  Бас бармақ арқылы өзін-өзі бағалау |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | жиынтығы төртбұрышты емес, жақшада болады .   1. Кортеж элементтері тізімдегідей белгілі бір ретпен берілген. Кортеж элементтері тізімнің элементтері сияқты нөлден индекстеледі, осылайша бос емес кортеждің бірінші элементі — бұл әрқашан a\_tuple[0]. 2. Индекстің теріс мәндері тізімдегідей кортеждің соңынан есептеледі. Соңғы элемент -1 индексі бар. 3. Кортеждің кескінін жасау ("slicing") тізім кескінін жасауға ұқсас. Тізім кесілгенде, жаңа тізім пайда болады; кортеж кесілгенде, жаңа кортеж пайда болады. Кортеждер мен тізімдер арасындағы басты айырмашылық- кортеждерді өзгерту мүмкін емес. Іс жүзінде бұл олардың өзгеруіне мүмкіндік беретін әдістері жоқ дегенді білдіреді. Тізімдерде append (), extend (), insert (), remove () және pop () сияқты әдістер бар. Кортеждерде бұл әдістердің ешқайсысы жоқ. Сіз кортежден кесуге болады (өйткені бұл жағдайда жаңа кортеж жасалады).   # алдыңғы мысалды жалғастыру  >>> a\_tuple ('a', 'b', 'mpilgrim', 'z', 'example')  >>> a\_tuple.append("new") (1)  Traceback (innermost last):  File "< interactive input >", line 1, in  ?  AttributeError: 'tuple' object has no attribute 'append'  >>> a\_tuple.remove("z") (2)  Traceback (innermost last): File "< |  | .  **«Мада қтау»** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | interactive input >", line 1, in ? AttributeError:  'tuple' object has no attribute 'remove'  >>> a\_tuple.index("example") (3)  4  >>> "z" in a\_tuple (4) True  *Қабық хабарламаларын аудару:*  Стекті жылжыту (сыртқы және ішкі):  Файл " < интерактивті енгізу>", 1- жол, орын ?  AttributeError: объектіде'tuple' атрибуты жоқ '<атрибут>'   1. Сіз элементтерді кортежге қоса алмайсыз. Кортеждерде append() немесе extend әдістері жоқ(). 2. Сіз элементтерді кортежден алып тастай алмайсыз. Кортеждерде remove() немесе pop әдістері жоқ(). 3. Сіз элементтерді кортежден іздей аласыз, өйткені ол кортежді өзгертпейді. 4. Жәшікте элементтің болуын тексеру үшін in операторын пайдалануға болады.   Сонымен, қайда ыңғайлы болуы мүмкін?   * + Кейбір жағдайларда тізімдерге қарағанда тезірек. Бірақ мұндай оңтайландыру әр жағдайда қосымша зерттеулерді қажет етеді.   + Егер сізде өзгермейтін "жазбадан қорғалған" деректер болса, кортеждер кодты қауіпсіз етеді.   + Кейбір кортеждерді жиын элементтері және сөздік кілттері |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ретінде пайдалануға болады (атап айтқанда, жолдар, сандар және басқа да кортеждер сияқты өзгермейтін мағыналары бар кортеждер). Тізімдерді ешқашан сөздік кілт ретінде пайдалануға болмайды, өйткені тізімдер & nbsp;  - өзгертілетін Нысандар. Кортеждерді тізімдерге және керісінше түрлендіруге болады. Кірістірілген tuple () функциясы тізімді қабылдайды және оның барлық элементтерінен кортежді қайтарады, list () функциясы кортежді қабылдайды және тізімді қайтарады. Шын мәнінде, tuple() тізімді қатырады, ал list () кортежді ерітеді.  ***Логикалық контекстегі кортеждер***  Сіз логикалық контексте, мысалы, if операторында қолдана аласыз.  >>> def is\_it\_true(anything):  … if anything:  … print("ия, бұл шындық")  … else:  … print("жоқ, бұл өтірік")  …  >>> is\_it\_true(()) (1) жоқ, бұл өтірік  >>> is\_it\_true(('a', 'b')) (2) ия, бұл шындық  >>> is\_it\_true((False,)) (3) ия, бұл шындық  >>> type((False)) (4)  < class 'bool'>  >>> type((False,))  < class 'tuple'>   1. Логикалық тұрғыда бос кортеж өтірік болып табылады. 2. Кем дегенде бір элементтен тұратын кез — келген кортеж- |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | шындық.   1. Кем дегенде бір элементтен тұратын кез — келген кортеж- шындық. Элементтердің мәндері маңызды емес. Бірақ бұл үтір не істейді? 2. Бір элементтің түйінін жасау үшін одан кейін үтір қою керек. Үтір болмаса, Python сіз жай ғана басқа жақшаларды қосқаныңызды болжайды, бұл ештеңе жасамайды, сонымен қатар сызықша жасамайды.   **Тапсырма**  Python-да бірнеше айнымалыларға мәнді бірден тағайындау үшін кортеждерді қолдану |  |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://ds05.infourok.ru/uploads/ex/0b54/0013e173-bc9648df/hello_html_m3997c853.jpg |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 31 сабақ** | **Курс атауы:** «Python  программалау тілінің негіздері» | **Сыныбы: 9 Күні: Мұғалім:** | | | |
| **Тақырыбы:** Python тіліндегі жиындар. | | | | | |
| **Оқу мақсаты:** | Python тіліндегі жиындар туралы ақпарат беру. | | | | |
| **Сабақ**  **мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал** | | Психол огиялы | Мұғалі мнің | Жұмыс дәптері |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | https://ds04.infourok.ru/uploads/ex/00e6/0011527d-e94b2917/img2.jpg | қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. | ауызша бағалау ы | Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану**  Жиындар (set) элементтер жиынын көрсетеді. Жиындарды анықтау үшін ішінде элементтері тізбектелген жүйелі жақша қолданылады.  users = {"Tom","Bob","Alice", "Tom"}  print(users) # {"Tom","Bob","Alice"} Мұнда print функциясы Tom элементін бір рет қана шығарды, себебі жиын тек қайталанбайтын **уникалды** мәндерден тұрады.  Жиындарды анықтау үшін **set()** функциясы да қолданылады, онда элементтердің тізімі не кортежі беріледі:  users3 = set(["Mike", "Bill", "Ted"])  **set** функциясын бол жиын құруда қолданған ыңғайлы:  users = set()  Жиын ұзындығын алу үшін **len()** кіріктірілген функциясы қолданылады. users = {"Tom","Bob","Alice"} print(len(users)} # 3  **Элементтерді қосу**  Жалғыз элементтерді қосу үшін **add()**  функциясы қолданылады. users = set() users.add("Sam") | | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай орында йды | Отшашу арқылы бір-бірін бағалау.  ÐÐ°ÑÑÐ¸Ð½ÐºÐ¸ Ð¿Ð¾ Ð·Ð°Ð¿ÑÐ¾ÑÑ ÑÐ°Ð»ÑÑ  Бас бармақ арқылы өзін-өзі бағалау.  **«Мадақ тау»** |

|  |  |  |  |
| --- | --- | --- | --- |
|  | print(users)  **Элементтерді жою**  Бір элементті жою үшін remove() әдісі қолданылады, оның ішінде жойылатын элемент көрсетіледі. Бірақ егер ондай элемент жиында болмаса, қате генерацияланады. Сондықтан жоюдан бұрын осы элементтің бар-жоқтығын **in** операторы көмегімен тексеріп алу керек:  users = {"Tom", "Bob", "Alice"} user = "Tom"  if user in users: users.remove(user)  print(users) # {"Bob", "Alice"}  Жою үшін discard() әдісін де қолдануға болады, егер эойылатын элемент жиында жоқ болса, бұл әдіс ерекше жағдай тудырмайды.  user = "Tim" users.discard(user)  Барлық элементтерді жою үшін clear() әдісі қолданылады.  users.clear()  **Жиындарға қолданылатын операциялар copy()** әдісінің көмегімен бір жиынның мазмұнын екіншісіне көшіруге болады: users = {"Tom","Bob","Alice"}  users3 = users.copy()  union() әдісі екі жиынды біріктіріп, жаңа жиын қайтарады:  users = {"Tom","Bob","Alice"}  users2 = {"Sam","Kate", "Bob"} users3 = users.union(users2)  print(users3) # {"Bob", "Alice", "Sam", "Kate", "Tom"}  Жиындардың қиылысуы екі жиында да бар элементтерді алуға мүмкіндік береді. intersection() әдісі жиындар қиылысуы операциясын орындап, нәтижесінде жаңа жиын қайтарады.  users = {"Tom","Bob","Alice"}  users2 = {"Sam","Kate", "Bob"} users3 = users.intersection(users2) print(users3) # {"Bob"} |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **intersection** әдісінің орнына логикалық көбейту операциясын да қолдануға болар еді.  users = {"Tom","Bob","Alice"}  users2 = {"Sam","Kate", "Bob"} print(users & users2) # {"Bob"}  Тағы бір операция жиындар айырымы - бірінші жиында бар, бірақ екінші жиында жоқ элементтерді қайтарады. Жиындар айырымын алу үшін **difference** әдісін немесе азайту операциясын орындауға болады: users = {"Tom","Bob","Alice"}  users2 = {"Sam","Kate", "Bob"} users3 = users.difference(users2) print(users3) # {"Tom", "Alice"}  print(users - users2) # {"Tom", "Alice"} | |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://ds04.infourok.ru/uploads/ex/0bad/00004644-b4c59e35/img41.jpg |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 32 сабақ** | **Курс атауы:** «Python программалау тілінің  негіздері» | **Сыныбы: 9 Күні:**  **Мұғалім:** | | | |
| **Тақырыбы:** Бағдарламалау стилі. | | | | | |
| **Оқу мақсаты:** | Питонның бағдарламалау стилі бойынша түсіндіру. | | | | |
| **Сабақ**  **мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/1281/00004d18-f880e7ca/img2.jpg | | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану** Құрылымдық бағдарламалау туралы толық ақпарат Python құрылымдық бағдарламалау бетінде бар.  Python бағдарламалау стильдері процедуралық, объектіге бағытталған, Функционалды және құрылымдық бағдарламалау стильдері болып табылады, өйткені Python бағдарламалаудың барлық төрт стилін қолдануға мүмкіндік береді. | | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай орында | Отшаш у арқылы бір- бірін бағалау  . |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Мен императивті және функционалды бағдарламалау стильдерінің арасындағы айырмашылықты нақты көрсететін ақпарат беремін:  Императивті бағдарламалар белгілі бір әрекетті орындау үшін қадамдар тізбегін, ал функционалды бағдарламаларды көбінесе қадамдардың нақты тізбегін көрсетпестен функциялардың орналасуы мен құрамына бағыттауға бейім.  Бір мәселенің екі шешімінің қарапайым мысалы (бірдей Python тілі қолданылады) мұны суреттейді.  # императивті стиль  мақсат = [] # бос тізімді жасаңыз source\_list үшін: # әр элемент үшін # бастапқы тізім  'trans1 = G (item) # G()функциясын қолданыңыз  'trans2 = F (trans1) # F () функциясын қолданыңыз  ‘ target.append (trans2) # тізімге түрлендірілген элементті қосыңыз Функционалды нұсқа басқаша көрінеді:  # функционалды стиль # FP тілдерінде көбінесе  кірістірілген compose () функциясы бар  compose2 = lambda A, B: lambda x: A(B(x))  target = map(compose2(F, G), source\_list)  **Тапсырма** Бағдарламалаудың негізгі модельдері бойынша ақпарат тауып. Постер құру.  Императивті бағдарламалау Декларативті бағдарламалау | йды | Бас бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Құрылымдық бағдарламалау Функционалдық бағдарламалау Логикалық бағдарламалау Объектіге бағытталған бағдарламалау  Сыныптарға негізделген бағдарламалау Прототиптерге негізделген бағдарламалау  Тақырыпқа бағытталған бағдарламалау |  |  |  |
| **Соңы** | **Қорытынды**  Сабақты қорытындылау  **Рефлексия**  https://ds04.infourok.ru/uploads/ex/0ae2/00026859-899c2fa8/img27.jpg |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 33 сабақ** | **Курс атауы:** «Python  программалау тілінің негіздері» | **Сыныбы: 9 Күні: Мұғалім:** | | | |
| **Тақырыбы:** Процедуралық бағдарламалау стилі. | | | | | |
| **Оқу мақсаты:** | Процедуралық бағдарламалау стилі бойынша түсінік  қалыптастыру. | | | | |
| **Сабақ**  **мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/0da3/00048dfa-e61b43b1/img2.jpg | | Психол огиялы қ ахуалға берілге н өлеңді | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. |  | Презент ация |
| **Ортасы** | **Мағынаны тану**  Процедуралық бағдарламалау стилі-бұл процедураларды қолдануға негізделген бағдарламалау:  Процедуралық бағдарламалау-бұл императивті тілде бағдарламалау, онда дәйекті орындалған операторларды ішкі бағдарламаларға, яғни кодтың үлкен тұтас бірліктеріне тілдің тетіктерін қолдана отырып жинауға болады.  Процедуралық бағдарламалау-бұл 1940 жылдары Фон Нейман ұсынған дәстүрлі компьютерлік архитектураның көрінісі. Процедуралық бағдарламалаудың теориялық моделі-бұл Тьюринг машинасы деп аталатын дерексіз есептеу жүйесі.  Негізгі мәліметтер  Бағдарламаның орындалуы жадтың бастапқы күйін, яғни бастапқы деректердің мәндерін соңғы, яғни нәтижелерге түрлендіру үшін операторлардың дәйекті орындалуына дейін азаяды.  Осылайша, бағдарламашының көзқарасы бойынша бағдарлама мен жад бар, біріншісі соңғысының мазмұнын дәйекті түрде жаңартады.  Процедуралық бағдарламалау тілі бағдарламашыға мәселені шешу процесінде әр қадамды анықтауға мүмкіндік береді.  Мұндай бағдарламалау тілдерінің ерекшелігі-тапсырмалар қадамдарға бөлініп, кезең-кезеңімен шешіледі. | | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай орында йды | Отшашу арқылы бір-бірін бағалау.  ÐÐ°ÑÑÐ¸Ð½ÐºÐ¸ Ð¿Ð¾ Ð·Ð°Ð¿ÑÐ¾ÑÑ ÑÐ°Ð»ÑÑ  Бас бармақ арқылы өзін-өзі бағалау.  **«Мадақ тау»** |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Процедуралық тілді қолдана отырып, бағдарламашы алгоритмдік қадамдар тізбегін орындау үшін тілдік құрылымдарды анықтайды.  **Тапсырма**  Тақырып бойынша кластер құру |  |  |
| **Соңы** | https://ds04.infourok.ru/uploads/ex/0d0d/00010af4-2268d125/img13.jpg**Қорытынды**  Сабақты қорытындылау  **Рефлексия** |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ 36 сабақ** | **Курс атауы:** «Python программалау тілінің  негіздері» | **Сыныбы: 9 Күні:**  **Мұғалім:** | | | |
| **Тақырыбы:** "Python тілінде бағдарламалау" курсы бойынша сынақ | | | | | |
| **Оқу мақсаты:** | Оқушылардың курс бойынша білімін қорытындылау. | | | | |
| **Сабақ**  **мақсаты** | Тақырыпты түсінеді, тапсырманы орындайды, алған білімін  өмірде қолданады | | | | |
| **Жоспар** | | | | | |
| **Сабақтың кезеңдері** | **Педагогтің әрекеті** | | **Оқуш ының әрекеті** | **Бағала у** | **Ресурст ар** |
| **Сабақтың басы** | **Психологиялық ахуал**  https://ds04.infourok.ru/uploads/ex/0da3/00048dfa-e61b43b1/img2.jpg | | Психол огиялы қ ахуалға берілге н өлеңді дауыст ап айтып, тілді жаттық тыру және жаңа сабаққа назар аудару. | Мұғалі мнің ауызша бағалау ы | Жұмыс дәптері  Суретте р Карточк алар Презент ация |
| **Ортасы** | **Мағынаны тану** Тіктөртбұрыштан x жағы бар шаршы кесіліп, суретте көрсетілгендей фигура пайда болды. Y сандарының қосындысы қандай? | | Оқушы лар сұраққ а жауап береді, жұмыс ты берілге н түріне қарай | Отшаш у арқылы бір- бірін бағалау  . |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | https://skr.sh/i/060921/2AoajW0m.jpg?download=1&name=Скриншот%2006-09-2021%2015:01:36.jpg  Мумба-юмба тайпасының алфавитінде 32 әріп бар. Бұл тайпаның тіліндегі кез-келген сөз бес әріптен тұрады және оңнан солға және солдан оңға қарай бірдей оқылуы керек, сөздің алғашқы екі әрпі әр түрлі болады, ал үшіншісі бесіншіге сәйкес келеді.  Бұл тілдегі сөздердің максималды саны қандай?  атя орамға үш түрлі түсті жұқа көлденең сақиналарды жабыстырды. Егер сіз сұр сақиналарға кессеңіз, сіз 25 дана орам аласыз, егер таңқурай үшін — 47 дана, ал жасыл болса — 31 дана.  Егер сіз барлық үш түсті сақиналарға кесіп тастасаңыз, қанша орама шығады?  Ескертпелер  Есіңізде болсын, бір түсті сақинаны екіншісінің сақинасына жабыстыру мүмкін емес.  Суретте A, B, C және D қалаларының орналасуы және олардың арасындағы қашықтық көрсетілген. Турист в қаласынан шығып, қалған қалаларға барады, әр уақытта болады. | орында йды | Бас бармақ арқылы өзін-өзі бағалау  .  **«Мада қтау»** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Егер ол сол қалада өз жолын аяқтағысы келсе, маршруттың ең аз ұзындығы қандай?  https://skr.sh/i/060921/BMIuDbEs.jpg?download=1&name=Скриншот%2006-09-2021%2015:02:52.jpg  Жалпы салмағы 432 грамм болатын фигуралар салмақсыз жіптер мен жолақтардың көмегімен суретте көрсетілген құрылымға жиналды.  Бұл оның барлық бөлігінде орналасқан тепе-теңдік. Ромбтың салмағы қанша?  https://skr.sh/i/060921/dsXibqCM.jpg?download=1&name=Скриншот%2006-09-2021%2015:03:38.jpg  Адамдар қалаларға көшеді, бір жылда адамдар саны екі есе артады.  Егер адамдар бүкіл қаланы 12 жыл ішінде қоныстандырса, онда қаланың жартысын алу үшін қанша жыл қажет болады?  Сөз жұптарын салыстырыңыз. Олардың ішінде қаншасы бірдей?  https://skr.sh/i/060921/2kaN9cnZ.jpg?download=1&name=Скриншот%2006-09-2021%2015:04:49.jpg |  |  |  |
| **Соңы** | **Қорытынды** |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Сабақты қорытындылау  **Рефлексия**  https://ds04.infourok.ru/uploads/ex/0ae2/00026859-899c2fa8/img27.jpg |  |  |  |

**Тест тапсырмалары**

1. Программада жаңа терезе ашу үшін орындалатын командалар: А) File →Save.

B) File → Save As. С) File → Close.

D) File → Exit.

Е) File → New File.

1. Python тілінің кеңейтілімі –

А) .py. B) .docх. С) .jpеg. D) .pptх. Е) .exe.

1. Python тілінің программасын іске қосып, тексеру үшін қолданылатын команда –

А) File → New File

B) Run → Run Module С) Edit →Undo

D) Help →About IDLE Е) Window → Untitled

1. Python ті лін де қолданылатын әліп би –

А) Ағылшын әліпбиі. B) Орыс әліпбиі. С) Латын әліпбиі. D) Қазақ әліпбиі.

Е) Грек әліпбиі.

1. Тіл синтаксисі дегеніміз –

А) Программа жазу барысында сақталуға тиісті тілдің ережелері.

B) Аудару, түсінікті тілге аудару деген ұғымды білдіреді.

С) Пернетақтадан мәліметтерді енгізу үшін кірістірілген функция.

D) Программаның элементтерін құруда қолдануға болатын символдар жиынтығы.

Е) Қабылдай алатын мәндерінің және олармен орындауға болатын амалдардың жиынын анықтау.

1. Транслятор қандай ұғымды білдіреді? А) Түсініктемелер (комментарий).

B) Программаны іске қосу. С) Программа элементтері.

D) Аудару, түсінікті тілге аудару. Е) Кірістірілген функция.

1. іnt сөзінің мағынасы:

А) Нақты сан.

B) Айнымалы сан. С) Бүтін сан.

D) Қалдық сан. Е) Жай сан.

1. Пернетақтадан мәліметтерді енгізу үшін кірістірілген функция: А) boolean.

B) print. С) integer.

D) float.

Е) input

1. Мәліметтер типтері дегеніміз –

А) Программа жазу барысында сақталуға тиісті тілдің ережелері.

B) Аудару, түсінікті тілге аудару деген ұғымды білдіреді.

С) Пернетақтадан мәліметтерді енгізу үшін кірістірілген функция.

D) Программаның элементтерін құруда қолдануға болатын символдар жиынтығы.

Е) Қабылдай алатын мәндерінің және олармен орындауға болтын амалдардың жынын анықтау.

1. Қалдықты табу қандай белгімен белгіленеді?

А) //. B) %. С) \*\*. D) ++. Е) –.

1. Бүтін санды табу қандай белгімен белгіленеді?

А) //. B) %. С) \*\*. D) ++. Е) –.

1. and, or, not сөздерінің мағынасы:

А) Егер, онда, әйтпесе.

B) Және, немесе, емес. С) Және, онда, егер.

D) Емес, онда, және. Е) Немесе, егер, емес.

# Компьютерде биттер тізбегі түрінде сақталған деректер жинағы –

А) Файл В) Бума С) Құжат

D) Программа Е) Каталог

# Python программалау тіліндегі файлдың түрлері:

А) Мәтіндік В) Бинарлық С) Құжаттық

D) Бумалық

Е) Кірістірілген

# Сәйкестендіріңдер.

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | r | A) | Жазу және оқу үшін |
| 2. | r+ | B) | Жазу және оқу үшін (бинарлық) |
| 3. | a+ | C) | Тек қана жазу үшін |
| 4. | wb+ | D) | Тек қана оқу үшін |
| 5. | w | E) | Жаңа мазмұн қосу үшін ашады |

1. **Бос орындарды толтырыңдар.**

Әрекетті орындау - ..., логикалық өрнек - ..., фигурасымен бейнеленеді.

# Логикалық өрнек ақиқат болған кезде бір нұсқау орындалады, ал жалған болған кезде басқа нұсқау орындалады. Бұл –

А) if-else құрылымы В) if құрылымы

С) if-elif-else құрылымы

# Бос орындарды толтырыңыздар.

if логикалық\_өрнек құрылымы – шартты оператордың ..., ал өрнектер орналасқан жолдар – шартты оператордың ... деп аталады.

# Логикалық амалдар көмегімен біріктірілген екі немесе бірнеше қарапайым қатынастар –

А) Күрделі шарттар В) Логикалық қосу

С) Логикалық көбейту

D) Логикалық терістеу Е) True, False

# Сәйкестендіріңіз

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | ЖӘНЕ | А) | логикалық қосу |
| 2. | НЕМЕСЕ | В) | логикалық терістеу |
| 3. | ЕМЕС | С) | логикалық көбейту |

1. **Екі шарттың да бір мезетте орындалуын талап ететін логикалық амал:**

А) ЖӘНЕ В) НЕМЕСЕ С) ЕМЕС

# Екі шарттың кем дегенде біреуінің орындалуын талап ететін логикалық амал –

А) НЕМЕСЕ В) ЖӘНЕ С) ЕМЕС

# While цикл операторының дұрыс жазылуын көрсетіңіздер.

А) while (өрнек):

{

цикл денесі

}

В) while (өрнек);

{

цикл денесі

}

С) while (өрнек): do

{

цикл денесі

}

D) while (өрнек): begin

цикл денесі end

# For цикл операторының дұрыс жазылуын көрсетіңіздер.

А) for <цикл параметрі > ; < циклдің қайталану шарты > ; < цикл қадамы >

{ цикл денесі}

1. for <цикл параметрі > ; <тізбек атауы немесе range ( ) функциясы > ;

{

цикл денесі }

1. for <цикл параметрі > in <тізбек атауы немесе range ( ) функциясы> :

{

цикл денесі

}

1. for (цикл параметрі > in <тізбек атауы немесе range ( ) функциясы>) :

{

цикл денесі

}

# Итерация дегеніміз не?

А) Бір реттік орындау

В) Бірнеше рет орындауды қажет ететін код тізбегі С) Алдын ала шартты тексеру

D) Шартты оператор

# While ( x < 2 ) циклінің денесі қай уақытқа дейін орындалады?

* 1. x екіден кіші немесе оған тең болғанша
  2. x екіден кіші болғанда
  3. x екіден үлкен болғанша
  4. x екіге тең болғанша

# Python – ның цикл операторына жатпайтын оператор:

* 1. for to do
  2. while
  3. repeat … until
  4. do … while

# for цикл операторының дұрыс жазылуын көрсетіңдер.

* 1. for i=1 to 10 do
  2. for i := 1; i<10; i++
  3. for i in 10
  4. for i in range (1, 10)

# Трассировка дегеніміз не?

* 1. Процессор жұмысының моделі
  2. Электронды кесте
  3. Алгоритмді сипаттаудың графикалық әдісі
  4. Алгоритмді орындау

# Берілген программа фрагментінің орындалу нәтижесінде экранға не шығады?

i = 0

while i < 3 :

print ( i ) i=i+1

print( i + 1 )

1. 0 1 2 4
2. 0 1 2 3 4 5
3. Error
4. 1 0 2 4 3 5

# Программа фрагментінің орындалу нәтижесінде экранға не шығады?

i = 0

while i < 3 :

print ( i ) i += 1

else:

print (0)

1. 0 1 2 3 0
2. 0 1 2 0
3. 0 1 2
4. 0

# Программа фрагментінің орындалу нәтижесінде экранға не шығады?

i = 0

while i < 5 :

print ( i ) i += 1

if i == 3 : break

else:

print (0)

1. 0 1 2 0
2. 0 1 2
3. 0
4. error

# Программа фрагментінің орындалу нәтижесінде экранға не шығады?

for i in [1, 0] : print (i+1)

* 1. 2

1

* 1. [2, 1]
  2. [2, 0]
  3. 2

0

# Қай операторды циклді тоқтатуға пайдаланады?

* 1. break
  2. return
  3. continue
  4. else

# Программа фрагментінің орындалу нәтижесінде экранға не шығады?

i = sum = 0

while i < 4 :

sum += i i = i+1

print (sum)

1. 0
2. 10
3. 40
4. 4

# Программа фрагментінің орындалу нәтижесінде экранға не шығады?

for i in range (1, 6, 1) : print (i, end=’ , ‘)

* 1. 1, 2, 3, 4, 5,
  2. 1, 2, 3, 4, 5, 6
  3. 1, 6, 1
  4. 1

# Қолдан жасалған трассировка деген не?

* 1. Процессор жұмысын модельдеу, алгоритмнің әрбір командасын орындай отырып, командалардың орындалу нәтижесін трассировка кестесіне енгізу
  2. Қойылған мақсатқа жету үшін белгілі іс-әрекеттер тізбегін орындау
  3. Алгоритм құруға арналған таңбалар мен ережелер жүйесі
  4. Орындаушыға берілетін түсінік және нақты командалар

# Қай жазбада массивті жариялау көрсетілген?

* 1. var A: array [1 . . 10] of integer.
  2. for i in range (10) : if A[ i ]<min: min=A[ i ] .
  3. S=0 for i in range (10) : S+=array[ i ] .
  4. for i in range (10) : A [ i ]=random.randint (1, 100) .
  5. a = arr. array (‘i’) .

# Қай жазбада массив элементтерін енгізу көрсетілген?

* 1. a = arr. array (‘i’) .
  2. for i in range (10) : if A[ i ]<min: min=A[ i ] .
  3. S=0 for i in range (10) : S+=array[ i ] .
  4. for i in range (10) : A [ i ]=random.randint (1, 100) .
  5. var A: array [1 . . 10] of integer.

# Қай жазбада массив элементтерін шығару көрсетілген?

* 1. for i in range (10) : print ( A [ i ] ) .
  2. for i in range (10) : if A[ i ]<min: min=A[ i ] .
  3. S=0 for i in range (10) : S+=array[ i ] .
  4. for i in range (10) : A [ i ]=random.randint (1, 100) .
  5. var A: array [1 . . 10] of integer.

# Қай жазбада оң сандар көрсетілген?

* 1. B [ i ] %2! = 0 .
  2. B [ i ] %2==0 .
  3. B [ i ] >0 .
  4. B [ i ]<0 .
  5. B [ i ]<>0 .

# Қай жазбада теріс сандар көрсетілген?

* 1. B [ i ] >0 .
  2. B [ i ] %2==0 .
  3. B [ i ] %2! = 0 .
  4. B [ i ]<0 .
  5. B [ i ]<>0 .

# Қай жазбада жұп сандар көрсетілген?

* 1. B [ i ] >0 .
  2. B [ i ] %2==0 .
  3. B [ i ] %2! = 0 .
  4. B [ i ]<0 .
  5. B [ i ]=0 .

# Қай жазбада тақ сандар көрсетілген?

* 1. B [ i ] %2! = 0 .
  2. B [ i ] >0 .
  3. B [ i ] %2==0 .
  4. B [ i ]<0 .
  5. B [ i ]=0 .

# Массивтің 4-элементі 7-ге тең болуы қай жазбаға сәйкес?

* 1. B [ 3 ] =7 .
  2. B [ 4 ] =7 .
  3. B [ 7 ] =4 .
  4. B [ 7 ] =3 .
  5. B [ i ] =7.

# Массивтің әрбір элементінің екі есеге өсуі қай жазбада дұрыс жазылған?

* 1. for i in range (n) : A [ 2\*i ] =A [ i ] .
  2. for i in range (n) : A [ i ] =A [ i ] /2 .
  3. for i in range (n) : A [ i ] =A [ i ] +2 .
  4. for for i in range (1, n) : A [ i ] =A [ i ] +2 .
  5. for i in range (n) : A [ i ] =A [ i ] \* 2 .

# Массивтің әрбір элементінің екі есеге азаюы қай жазбаға сәйкес?

* 1. for i in range (1, n) : A [ i ] =A [ i ] +2 .
  2. for i in range (n) : A [ i ] =A [ i ] \* 2 .
  3. for i in range (n) : A [ i ] =A [ i ] +2 .
  4. for i in range (n) : A [ i ] =A [ i ] /2 .
  5. for i in range (n) : A [ 2\*i ] =A [ i ] .

# Массив элементтерінің көбейтіндісін тап.

* 1. P=1 for i in range ( 0, n ) : P\*=A [ i ] .
  2. P = 0 for i in range ( 0, n ) P\*=A [ i ] .
  3. P = 1 for i in range ( 1, n ) P\*=A [ i ] .
  4. P = 0 for i in range ( 1, n ) P\*=A [ i ] .
  5. P = n for i in range ( 0, n ) P\*=A [ i ] .

# Массивтің теріс элементтерінің қосындысын тап.

* 1. S=0 for i in range (0, n) : if A [ i ]<0 : S+=A [ i ] .
  2. S=0 for i in range (0, n) : if A [ i ]>0 : S+=A [ i ] .
  3. S=1 for i in range (0, n) : if A [ i ]<0 : S+=A [ i ] .
  4. S=0 for i in range (1, n) : if A [ i ]<0 : S+=A [ i ] .
  5. S=0 for i in range (0, n) : if A [ i ]<>0 : S+=A [ i ] .

# Массивтің оң элементтерінің көбейтіндісін тап.

* 1. P=1 for i in range (0, n) : if A [ i ]>0 : P\*=A [ i ] .
  2. P=1 for i in range (0, n) : if A [ i ]<0 : P\*=A [ i ] .
  3. P=0 for i in range (0, n) : if A [ i ]<0 : P\*=A [ i ] .
  4. P=1 for i in range (1, n) : if A [ i ]>0 : P\*=A [ i ] .
  5. P=1 for i in range (0, n) : if A [ i ]<>0 : P\*=A [ i ] .

# Массивтің жұп элементтерінің қосындысын тап.

* 1. S=0 for i in range (0, n) : if A [ i ] %2 == 0 : S+=A [ i ] .
  2. S=0 for i in range (0, n) : if A [ i ] %2 = 0 : S+=A [ i ] .
  3. S=1 for i in range (0, n) : if A [ i ] %2 == 0 : S+=A [ i ] .
  4. S=0 for i in range (1, n) : if A [ i ] %2 == 0 : S+=A [ i ] .
  5. S=0 for i in range (0, n) : if A [ 2\*i ] %2 == 0 : S+=A [ i ] .

# Массивтің тақ элементтерінің көбейтіндісін тап.

* 1. P=1 for i in range (0, n) : if A [ i ] %2 ! = 0 : P\*=A [ i ] .
  2. P=1 for i in range (0, n) : if A [ i ] %2 ! == 0 : P\*=A [ i ] .
  3. P=0 for i in range (0, n) : if A [ i ] %2 ! = 0 : P\*=A [ i ] .
  4. P=1 for i in range (0, n) : if A [ i ] %2 == 0 : P\*=A [ i ] .
  5. P=1 for i in range (0, n) : if A [ i ] %2 < >0 : P\*=A [ i ] .

# Дыбыстарды жүктеу және ойнату:

* 1. pygame.mixer.
  2. pygame.cdrom.
  3. pygame.image.
  4. pygame.display.
  5. pygame.key.

# Фигураларды, сызықтарды және нүктелерді сызу:

* 1. pygame.draw.
  2. pygame.cdrom.
  3. pygame.image.
  4. pygame.display.
  5. pygame.key.

# Сыртқы оқиғаларды басқару:

* 1. pygame.event.
  2. pygame.cdrom.
  3. pygame.image.
  4. pygame.display.
  5. pygame.key.

# Суреттерді өзгерту және жылжыту:

* 1. pygame.transform.
  2. pygame.cdrom.
  3. pygame.image.
  4. pygame.display.
  5. pygame.key.

# Уақыт пен кадр жиілігін басқару:

* 1. pygame.time.
  2. pygame.cdrom.
  3. pygame.image.
  4. pygame.display.
  5. pygame.key.

# pygame.rect –

* 1. төртбұрышты аймақтарды басқару.
  2. меңзер бейнелерін жүктеу.
  3. пернетақтада басқан пернені оқу.
  4. суреттер мен экранды басқару.
  5. фигураларды, сызықтарды және нүктелерді сызу.

# pygame.surface –

* 1. суреттер мен экранды басқару.
  2. меңзер бейнелерін жүктеу.
  3. пернетақтада басқан пернені оқу.
  4. суреттер мен экранды басқару.
  5. фигураларды, сызықтарды және нүктелерді сызу.

# Бейнені көрсету үшін терезені немесе экранды инициализациялау:

* 1. pygame.display.set\_mode.
  2. pygame.display.get\_surface.
  3. pygame.display.set\_caption.
  4. pygame.display.get\_caption.
  5. pygame.display.quit.

# Ағымдағы уақытта орнатылған дисплей бетінен сілтеме алу:

* 1. pygame.display.set\_mode.
  2. pygame.display.get\_surface.
  3. pygame.display.set\_caption.
  4. pygame.display.get\_caption.
  5. pygame.display.quit.

# Толық экран мен терезе дисплей арасындағы ауысу:

* 1. pygame.display.toggle\_fullscreen.
  2. pygame.display.get\_surface.
  3. pygame.display.set\_caption.
  4. pygame.display.get\_caption.
  5. pygame.display.quit.

# Терезе атауын тағайындау:

* 1. pygame.display.toggle\_fullscreen.
  2. pygame.display.get\_surface.
  3. pygame.display.set\_caption.
  4. pygame.display.get\_caption.
  5. pygame.display.quit.

# pygame.display.get\_caption –

* 1. дисплей модулін баптау.
  2. дисплей бетін толығымен жаңарту.
  3. ағымдағы терезенің тақырыбын қайтару.
  4. ағымдағы уақытта орнатылған дисплей бетінен сілтеме алу.
  5. терезе атауын тағайындау.

# pygame.display.get\_surface –

* 1. дисплей модулін баптау.
  2. дисплей бетін толығымен жаңарту.
  3. ағымдағы терезенің тақырыбын қайтару.
  4. ағымдағы уақытта орнатылған дисплей бетінен сілтеме алу.
  5. терезе атауын тағайындау.

# pygame.display.flit –

* 1. дисплей модулін баптау.
  2. дисплей бетін толығымен жаңарту.
  3. ағымдағы терезенің тақырыбын қайтару.
  4. ағымдағы уақытта орнатылған дисплей бетінен сілтеме алу.
  5. терезе атауын тағайындау.

# Спрайт –

* 1. дисплей модулін баптау.
  2. дисплей бетін толығымен жаңарту.
  3. ағымдағы терезенің тақырыбын қайтару.
  4. ағымдағы уақытта орнатылған дисплей бетінен сілтеме алу.
  5. терезе атауын тағайындау.

# Глоссарий

**Абакус** – қарапайым есептеулерді шығаруға мүмкіндік беретін және нұсқалары бір қатар елде табылған санау құралы.

**Айнымалылар** – программа орындалу барысында әртүрлі мәндерді қабылдай алатын шамалар. Айнымалылар программадағы ақпараттың орналасу орнын анықтау үшін қолданылады.

**Есептің кіріс мәліметтері** – есепті шешу үшін пайдаланылатын мәліметтер.

**Есептің шығыс мәліметтері** – есепті шешу нәтижесінде алынған мәліметтер.

**Логикалық қателер** – алгоритмдегі немесе программа логикасындағы жаңылыстардан туындайтын қателер.

**Логикалық функциялар** – бір немесе бір неше шарттарды тексеруге арналған функциялар.

**Массив** – бір атаумен біріктірілген, бір типтегі деректер ден тұратын айнымалылардың жиынтығы.

**Математикалық функциялар** – қарапайым және күрделі математикалық есептеулер орындайтын функциялар. Мысалы, 200 ұяшықтар ауқымының қосындысын есептеу, санның абсолютті мәнін табу, сандарды дөңгелектеу және т.б.

**Мәселені қалыптастыру** – кіріс және шығыс мәліметтерін сипаттайтын есеп шартының нақты құрылымы.

**Мәтіндік функциялар** – пайдаланушыға мәтінді өңдеуге мүмкіндік беретін функциялар.

**Модель** – нысанның, құбылыстың немесе процестің маңызды ерекшеліктерін көрсететін белгілі бір жаңа нұсқа.

**Модельдеу** – практикалық мәселелерді шешу үшін модельдерді құру және пайдалану процесі.

**Модульдер** – Visual Basic кіріктірілген программалау тілінде жазылған программалары бар нысандар.

**Программа құрудың кіріктірілген ортасы** – программалаушыға программалық жасақтаманы дамытуға жағдай жасайтын қосымша.

**Программалау тілі** – компьютерде орындауға арналған және оған түсінікті тілде жазылған алгоритм.

**Программаны орындаушылар** – компьютер немесе басқа автоматты құрылғылар.

**Санның факториалы** – 1-ден n-ге дейінгі барлық натурал сандардың көбейтіндісі, яғни n! = 1 · 2· 3 ·... · n.

**Транслятор** – (ағылш. translator – аудармашы) берілген программалау тіліндегі программалау кодын машиналық кодқа түрлендіретін арнайы программа.

**Тіл әліпбиі** – программаның элементтерін құруда қолдануға болатын символдар жиынтығы.

**Break операторы** – циклді аяқтамай циклден шығуға мүмкіндік беретін оператор.

**Continue операторы** – цикл денесінің кейбір мәліметтерін өткізіп жіберіп, жаңа итерацияны бастауға мүмкіндік беретін оператор.

**Python** – қазіргі заманға лайықты программалау тілі.

**Функция** – y = f(x) өрнегін қанағаттандыратын (x, y) нүктелер жиынтығы.

**Цикл** – бірнеше рет қайталанатын операциялардың жиынтығы.

**Цикл алгоритмі** – қайталау шарты дұрыс болғанда белгілі бір әрекеттердің қайталануын анықтайды.

**Цикл денесі** – бір неше рет орындалатын код тізбегі.

**Цикл параметрі** – циклден шығу жағдайына байланысты және циклдегі өз мәнін өзгертетін айнымалы шама.

**Шартты алдын ала тексеру циклі** – циклдің қайталану саны алдын ала белгісіз болған жағдайда қолданылады.

**Шексіз цикл** – программаны шексіз орындау үшін қолданылатын әдіс.

**Пайдаланылған әдебиеттер**

1. Златопольский Д.М. Занимательная информатика. М.: Бином. Лаборатория знаний, 2016. – 424 с.
2. Шапошникова С. Основы программирования на Python. Вводный курс. 2016.
3. Николай Прохорёнок – Python 3 и PyQt. Разработка приложений, 2018 г. 832 с.
4. Гаврилов М.В., Климов В.А. Информатика и информационные технологии: Учебник. – Люберцы: Юрайт, 2016. – 383 с.
5. Гэддис Т. Начинаем программировать на Python. – 4-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2019. – 768 с.
6. Лучано Рамальо Python. К вершинам мастерства. – М.: ДМК Пресс, 2016. – 768 с.
7. Свейгарт, Эл. Автоматизация рутиных задач с помощью Python: практическое руководство для начинающих. Пер. с англ. — М.: Вильямc, 2016. – 592 с.
8. Рейтц К., Шлюссер Т. Автостопом по Python. – СПб.: Питер, 2017. – 336 с.: ил. – (Серия «Бестселлеры O’Reilly»).
9. Любанович Билл Простой Python. Современный стиль программирования. – СПб.: Питер, 2016. – 480 с.: – (Серия «Бестсепперы O’Reilly»).
10. Федоров Д.Ю. Программирование на языке высокого уровня Python: учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2019. – 161 с. – (Бакалавр. Прикладной курс). – ISBN 978-5-534-10971-9. – Текст: электронный // ЭБС Юрайт [сайт]. – URL: <https://urait.ru/bcode/437489> (дата обращения: 13.02.2020).
11. Шелудько, В. М. Основы программирования на языке высокого уровня Python: учебное пособие / В. М. Шелудько. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. – 146 c. – ISBN 978-5-9275-2649-9. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: <http://www.iprbookshop.ru/87461.html> (дата обращения: 13.02.2020). – Режим доступа: для авторизир. пользователей
12. Шелудько, В. М. Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули: учебное пособие / В. М. Шелудько. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. – 107 c. – ISBN 978-5-9275-2648-2. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: <http://www.iprbookshop.ru/87530.html> (дата обращения: 13.02.2020). – Режим доступа: для авторизир. пользователей
13. Коноплева И.А., Хохлова О.А., Денисов А.В. Информационные технологии. – М.: Проспект, 2015. – 328 б.
14. Златопольский Д.М. Основы программирования на языке Python. М.: ДМК Пресс, 2017. – 284 с.: ил. – ISBN 978-5-97060-552-3.
15. Пилгрим Марк. Погружение в Python 3 (Dive into Python 3 на русском)
16. Босова Л.Л., Босова А.Ю., Информатика: учебник для 9 класса. М.: БИНОМ. Лаборатория знаний, 2017. – 208 с.