КОММЕРЦИЯЛЫҚ ЕМЕС АКЦИОНЕРЛІК ҚОҒАМЫ
«ӘБІЛҚАС САҒЫНОВ АТЫНДАҒЫ ҚАРАҒАНДЫ ТЕХНИКАЛЫҚ УНИВЕРСИТЕТІ»

Киберқауіпсіздік және жасанды интеллект кафедрасы

Аубакирова А.Е.

«PYTHON ОБЬЕКТІГЕ БАҒЫТТАЛҒАН БАҒДАРЛАМАЛАУ»

ПӘНІ БОЙЫНША
ДӘРІСТЕРГЕ
ӘДІСТЕМЕЛІК НҰСҚАУЛАР

Қарағанды 2026

КОММЕРЦИЯЛЫҚ ЕМЕС АКЦИОНЕРЛІК ҚОҒАМЫ
«ӘБІЛҚАС САҒЫНОВ АТЫНДАҒЫ ҚАРАҒАНДЫ ТЕХНИКАЛЫҚ УНИВЕРСИТЕТІ»

Киберқауіпсіздік және жасанды интеллект кафедрасы

Аубакирова А.Е.

«PYTHON ОБЬЕКТІГЕ БАҒЫТТАЛҒАН БАҒДАРЛАМАЛАУ»

ПӘНІ БОЙЫНША
ДӘРІСТЕРГЕ
ӘДІСТЕМЕЛІК НҰСҚАУЛАР

6B06105 «Data Science» білім беру бағдарламасы

Қарағанды 2026

ДӘРІС КОНСПЕКТІЛЕРІ

Тақырып 1. Обьектіге бағытталған Python бағдарламалау тілі

Python - бұл барлық қарапайым жұмыс үстелі операциялық жүйелерінде жұмыс істейтін заманауи объектіге бағытталған бағдарламалау тілі.
1980 жылдардың аяғында голландиялық Гвидо ван Россум жасаған Python тілі ондаған жылдар бойы модернизацияланып, жетілдіріліп программисттерге бүгінде бағдарламалар мен қосымшаларды құруға, жобаларды әртүрлі бағытта жүзеге асыру үшін қолдануға мүмкіндік берді.
Қазіргі уақытта тілдің екі нұсқасы белсенді қолданылады — ескі 2x нұсқасы және қазіргі 3x нұсқасы. 2x нұсқасы енді дамымайды, бірақ әлі де қолданылады, өйткені көптеген бағдарламалық жасақтама мен кітапханалар 2x нұсқасына арналған.
Нұсқалар арасында айтарлықтай сәйкессіздік бар, соның ішінде енгізу-шығару пәрмендерінің синтаксисінде (2-ші нұсқадағы Python бағдарламасы 3-ші нұсқада жұмыс істемеуі мүмкін және керісінше), бірақ олар өте ұқсас. Біз 3x нұсқасын неғұрлым заманауи және жетілдірілген ретінде қарастырамыз.

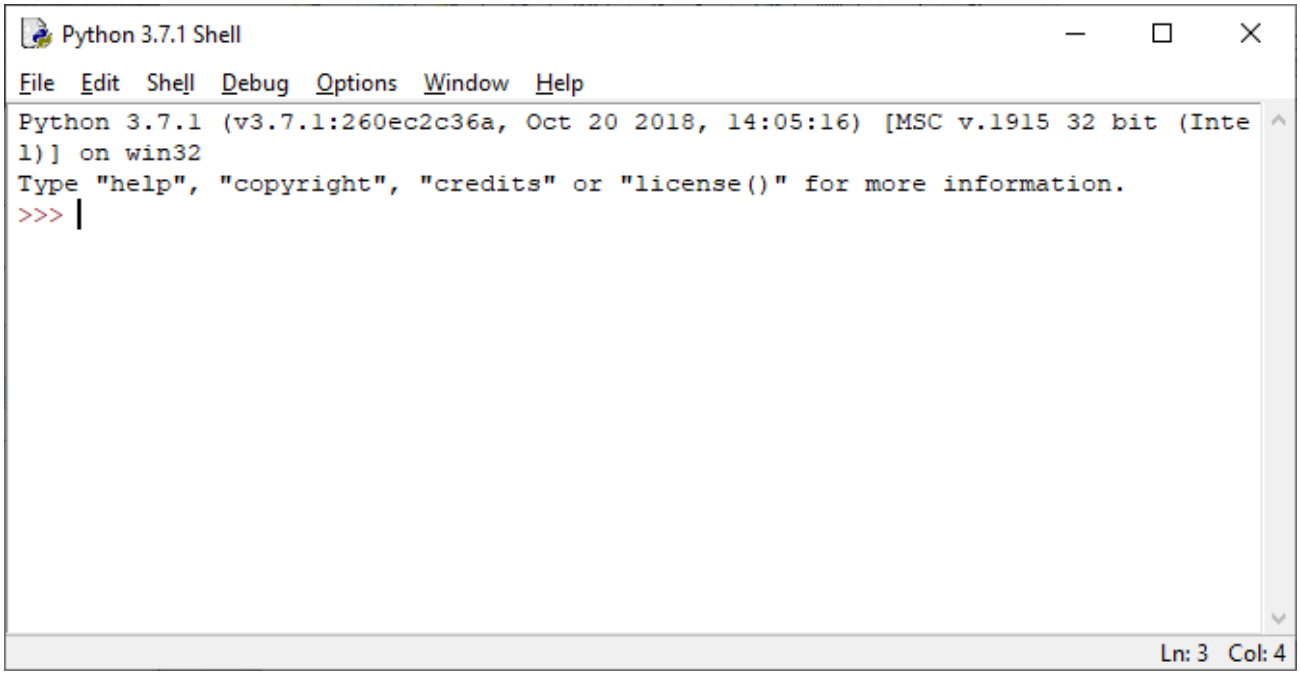
Python бағдарламалау тілінің артықшылықтары:

1. Кросс-платформа және тегін.
2. Қарапайым синтаксис пен мүмкіндіктері зор бағдарламаларды өте қысқа, бірақ сонымен бірге түсінікті етіп жазуға мүмкіндік береді.
3. Игерудің қарапайымдылығы бойынша тілді бейсикпен салыстыруға болады, бірақ әлдеқайда зор және әлдеқайда заманауи.
4. Стандартты кітапханасы күштірек.
5. Ойындарға арналған қосымшаларды әзірлеу, интернет, суреттер, есептеу жүйелері, мәліметтер базасы және т. б.

Python стандартты пакетіне қарапайым мәтіндік редакторға қарағанда бағдарламаларды өңдеу әлдеқайда ыңғайлы IDLE интеграцияланған даму ортасы кіреді.

IDLE (Integrated Development and Learning Environment) - Python тілінде интеграцияланған даму және оқыту ортасы.
IDLE - қолдана отырып, сіз интеграцияланған орта үшін әдеттегі тапсырмаларды орындай аласыз: Python тілінде бағдарламаларды қарау, өңдеу, іске қосу, күйін келтіру. Код редакторы синтаксисті бөлектеуді қолданады.
IDLE Python іске қосылғаннан кейін бағдарлама терезесі пайда болады (сурет. 1). Бұл терезеде қарапайым командаларды орындауға және тіпті оны калькулятор ретінде пайдалануға болады. Файл мәзірі арқылы сіз Python тілінде

жаңа бағдарлама жасай аласыз немесе бар бағдарламаны аша аласыз. Меню Option - мәзірі арқылы шрифттің өлшемі мен түрін өзгертуге болады.



Сурет 1-интеграцияланған терезе ортасы.

IDLE Python терезесінде > > > таңбалары пәрменді енгізуге шақыруды білдіреді. Пәрменді енгізгеннен кейін <Enter>пернесін басыңыз. Келесі жолда нәтиже бірден көрсетіледі, содан кейін жаңа пәрменді енгізуге шақыру пайда болады.

Python-да "hello world" бағдарламасын жазу үшін тек бір жол жеткілікті:
>>>print("Hello world!")
Бұл кодты IDLE терезесінде енгізіп, Enter пернесін басыңыз.
Көп жолды пәрменді енгізген кезде (мысалы, for) <Enter> пернесін басқаннан кейін редактор автоматты түрде шегіністі енгізеді және одан әрі енгізуді күтеді. Редакторға көп жолды командасын енгізу туралы хабарлау үшін <Enter > пернесін екі рет басу керек.

Мысалы:

```
for n in range(1, 4):
        print(n)
print(“Hello”)
```

Нәтижесінде экранда:
1
2
3
Hello

Python интерактивті жұмысының тағы бірнеше мысалын қарастырайық. IDLE терезесінде келесі командаларды теріңіз:
print (3 + 4) немесе жай 3+4 теріңіз.
print (3 * 5) немесе жай 3*4 теріңіз.
print (3 * * 2) немесе жай 34 теріңіз.**

Дегенмен, Интерактивті режим негізгі болмайды. Негізінен біз бағдарламалар жазып, оларды файлға жазамыз. Болашақта бағдарламаны оқылған файлдан бастауға болады.
Ол үшін IDLE терезесінде File / New File мәзірін таңдаңыз (немесе Ctrl + N басыңыз). Ашылған терезеде келесі бағдарламалық кодты енгізіңіз:
name = input("Сіздің аты-жөніңіз? ")
print("Сәлем ", name)

Бірінші жол сұрақты басып шығарады ("Сіздің аты жөніңіз? "), сіз бірдеңе теріп, Enter пернесін басуды күтіңіз. Содан кейін сіз енгізген мәнді name айнымалысында сақтайды.
Екінші жолда мәтінді экранға шығару үшін print нұсқауын қолданамыз, бұл жағдайда "сәлем" сөзін және Name айнымалысында не сақталатынын көрсетеміз.
Бағдарламаны іске қосу үшін F5 пернесін басыңыз. Іске қоспас бұрын IDLE файлды сақтауды ұсынады. Бағдарламаны сізге ыңғайлы жерде сақтаңыз, содан кейін бағдарлама іске қосылады.

Python-дағы айнымалылар

Айнымалылар деректерді сақтауға арналған. Python-дағы айнымалы атау алфавиттік таңбадан немесе астын сызу белгісінен басталуы керек және алфавиттік-сандық таңбалар мен астын сызу белгісін қамтуы мүмкін. Сонымен қатар, айнымалы атау Python тілінің кілт сөздерінің атауымен сәйкес келмеуі керек. Кілт сөздердің мысалы: True, else, for, and, return, import, not, or, def, pass, while және т. б.
Айнымалы-бұл бағдарламаның аралық немесе соңғы нәтижесін сақтауға болатын қарапайым деп аталатын мәліметтер құрылымы.
Python — да айнымалы құру өте қарапайым - "="тағайындау операторын қолдана отырып, белгілі бір идентификаторға мән беру керек.
Мысалы:

a = 10
b = 3.1415
c = “Hello”
d = [1, 2, 3]

Бұл мысалда төрт айнымалы қолданылады:

- **a** айнымалысы int типінің мәнін сақтайды (бүтін сан),
- **b** айнымалысы - float типті өзгермелі (нақты сан),
- **c** айнымалысы - str типті (жол),
- **d** айнымалысы - list типті (тізім, бұл жағдайда үш бүтін сан).

Айнымалылардың арнайы декларациясы қажет емес, айнымалы мәнді бірінші тағайындау оның жарнамасы болып табылады. Python идентификаторы жадта сақталған деректерге "сілтеме" болып табылады. Python-динамикалық тілі бар тип: уақыттың әр сәтінде әр айнымалы белгілі бір типке ие, бірақ бұл түрі бағдарламаны орындау барысында өзгеруі мүмкін, оған басқа типтегі жаңа мән беру жеткілікті.
Python-дағы айнымалы тек жадтағы объектіге сілтеме болып табылады. Кез-келген айнымалы (Сан, жол немесе массив) жасалған кезде оған объектіге сілтеме жазылады, ал объектінің өзі сілтемесі бар айнымалыдан алыс жерде жедел жадта болады. Осылайша, бірнеше айнымалылар бір нысанды көрсете алады және объект өзгерген кезде (мысалы, тізім) әр айнымалы мәнді қолдана отырып, оған жүгіну кезінде нәтижесі өзгереді.

Python-дағы қарапайым бағдарламаның құрылымы

Қарапайым Python бағдарламасы келесі бөліктерден тұруы керек:
- деректерді енгізу;
- мәселені шешу;
- нәтижені шығару.

Мысалы, екі бүтін санды қосу бағдарламасы келесідей болуы мүмкін:

```
a = int(input())
b = int(input())
sum = a + b
print(sum)
```

Екі жолды қосу бағдарламасы келесі түрде болуы мүмкін:

```
s1 = input()
s2 = input()
s3 = s1 + s2
print(s3)
```

Python тілінің синтаксисі

Python тілінің синтаксисі, тілдің өзі сияқты, өте қарапайым. Онда күрделі құрылымдар жоқ, сондықтан оны үйрену өте оңай.
Python тілінің синтаксисінің негізгі принциптері:
1. Жолдың соңы - Нұсқаулықтың соңы болып табылады (үтір қажет емес).
Мысалы:

```
a = 5
b = 3
print(a + b)
```
2. Еңгізілген нұсқаулар шегініс өлшемі бойынша блоктарға біріктіріледі. Шегініс кез-келген болуы мүмкін, бастысы, бір кірістірілген блоктың ішінде шегініс бірдей болады.

Мысалы:
```
if a == 5:
  print('yes')
    a += 1
y=a*12
```
Негізгі нұсқаулық қос нүктемен аяқталған кезде, одан кейін негізгі нұсқаулық жолының астында шегінісі бар кірістірілген код блогы орналасқан.
Командаларды бір жолға жазып, оларды "; " арқылы бөлу мүмкіндігі бар. Алайда, бұл әдісті жиі қолдануға болмайды, бұл көрінуді азайтады.
```
a = 1; b = 2; print(a)
print (a + b)
```

Python тіліндегі деректер түрлері
Python тілінде көптеген мәліметтер бар: бүтін сандар (int), өзгермелі нүктелер (float), жолдар (str), логикалық тип (Boolean), тізім (List) және т. б.
Әр айнымалының түрі бағдарламаның орындалу барысында динамикалық түрде өзгеруі мүмкін. Type () командасын қолдана отырып, айнымалының типін анықтауға болады.
Мысалы:
```
a = 72
print(a, "is of type", type(a))
```

Шығады
```
72 is of type <class 'int'>
a = 2.0
print(a, “is of type”, type(a) )
2.0 is of type <class 'float'>
```

Бүтін санды сақтау үшін компьютердің жадында 4 байт қолданылады. Нақты Сан-бұл өзгермелі нүкте саны, оны сақтау үшін компьютердің жадында 8 байт қолданылады. Мысалы, 2.0 немесе 58.241. Жолдар Юникодты кодтаудағы таңбалар жиынтығын білдіреді.
Сандарды нақты заттардан бүтін санға және керісінше Python-да түрлендіру үшін int() және float () функциялары анықталған. Мысалы, int(12.6) нәтижесінде 12, ал float (12) нәтижесінде 12.0 (ондық бөлгіш - нүкте) береді.

Бүтін сандардың негізгі операциялары

A + B-сомма;
A-B-айырмашылық;
A * B-шығарма;
A / B — бөлу, (бұл әрекеттің нәтижесі нақты сан болып табылады, тіпті егер A B-ге бөлінсе де);
A % B-A-ның B-ға бөлінуінен қалған қалдық;
A / / B - A-ның B-ға бөлінуінің тұтас бөлігі
A * * B — дәрежеге аудару.
Сонымен қатар, Python-да сандық операциялар үшін функциялар қолданылады:
abs() - абсолюттік мәнді есептеу;
pow () - дәрежеге ауыстыру, pow (2,3) = 8;
div mod () - бүтін санды бөлу мен қалдықтың нәтижесін есептеу, div mod(17,5) аламыз (3,2);
round () - дөңгелектеу (шамамен), round(5.7) = 6.
Бұл мүмкіндіктер" кіріктірілген", яғни оларды пайдалану үшін қосымша кітапханаларды қосудың қажеті жоқ.
Мысалы:

```
x = 17
y=divmod(x,4)
print("бүтін бөлік",y[0])
print("қалдық", y[1])
```

Шығады

```
бүтін бөлік 4
қалдық 1
```

Квадраттық түбір, синус, тангенс және т.б. есептеу сияқты сандармен жұмыс істеуге арналған барлық басқа математикалық функциялар **math** кітапханасын қосуды қажет етеді. Бұл модульмен жұмыс істеу үшін оны алдымен **import math** командасымен импорттау керек.
Мысалы:

```
import math
x=16
y=math.log2(x)
print(y)
```

Әдепкі бойынша, стандартты сандар ондық жүйеде сандар ретінде қарастырылады. Бірақ Python екілік, сегіздік және он алтылық жүйелердегі сандарды да қолдайды.
Сан екілік жүйені білдіретінін көрсету үшін санның алдына 0b префиксі қойылады:

```
a = 0b11
b = 0b1011
c = 0b100001
print(a)    # 3 ондық жүйеде
print(b)    # 11 ондық жүйеде
```

```
print(c)    # 33 ондық жүйеде
```

Санның сегіздік жүйені білдіретінін көрсету үшін санның алдына 0o префиксі қойылады:

```
a = 0o7
b = 0o11
c = 0o17
print(a)    # 7 ондық жүйеде
print(b)    # 9 ондық жүйеде
print(c)    # 15 ондық жүйеде
```

Санның он алтылық жүйені білдіретінін көрсету үшін санның алдына 0x префиксі қойылады:

```
a = 0x0A
b = 0xFF
c = 0xA1
print(a)    # 10 ондық жүйеде
print(b)    # 255 ондық жүйеде
print(c)    # 161 ондық жүйеде
```

Операциялардың басымдықтары

Python-дағы операциялардың басымдықтары математикадағы операциялардың басымдылығымен сәйкес келеді:

- Дәрежеге айналдыру;
- Көбейту және бөлу операциялары бірдей басымдыққа ие.
- Қосу және азайту операциялары бірдей басымдыққа ие.

Әрекет тәртібін өзгерту үшін жақшаларды пайдалану керек.

Өзгермелі нүкте сандары (нақты сандар)

Python-дағы нақты сан float түріне ие. Ол сандар тізбегі ретінде жазылады, оның алдында минус белгісі де болуы мүмкін. Нүкте бүтін және бөлшек бөлгіш ретінде қолданылады.

```
height = 1.68
pi = 3.14
weight = 68.
print(height)   # 1.68
print(pi)       # 3.14
print(weight)   # 68.0
```

Өзгермелі нүкте санын экспоненциалды жазбада анықтауға болады:

```
x = 3.9e3
print(x)  # 3900.0

x = 3.9e-3
print(x)  # 0.0039
```

Нақты сандар мен негізгі операциялар

A + B-қосынды;

A-B-айырма;
A * B-көбейту;
A * * B-дәрежеге ауыстыру.
A / B — бөлу, (бұл әрекеттің нәтижесі нақты сан болып табылады, тіпті егер A мақсатты түрде B - ге бөлінсе де);
A % B-A-ны B-ға бөлудің қалдығы, (толық емес бөлік бүтін сан дегенді білдіреді);
A / / B - A-ны B-ға бөлудің бүтін бөлігі, (толық емес бөлік бүтін сан дегенді білдіреді).