Әл-Фараби атындағы Қазақ ұлттық университеті

Механика-математика факультеті

Python құрылымы

Темирбеков Нурлан Муханович ф-м.ғ.д., профессор

Жоспар

1. Python бойынша ақпарат және қол жеткізу.
2. Python негізі.
3. Шартты және циклдік операторлар.
4. Деректерді оқу және басып шығару.

Мақсаты

Python бағдарламалық тілі бойынша негізгі ақпараттармен, тілдің негізін құрайтын операторлармен танысу, сандық әдістердің есептерін шешуге қажетті ақпарат беру.

Python бойынша ақпарат

Python–объектіге бағытталған тіл, ол қарапайым, икемді және өте танымал тіл.

Python бағдарламалары машиналық кодта компиляцияланбайды, бірақ интерпретатор арқылы орындалады. Интерпретацияланатын тілдің үлкен артықшылығы – бағдарламаларды тез тексеруге және жөндеуге болады.

Теріс жағы, интерпретацияланған бағдарламалар автономды қолданбаларды шығармайды.

Python-ның басқа артықшылықтары:

- Python – ашық кодты бағдарламалық жасақтама, яғни ол тегін;
- Python барлық негізгі операциялық жүйелер үшін қолжетімді.
- Python үйрену оңай және көптеген тілдерге қарағанда жеңіл оқылатын кодты құрайды.
- Python және оның қосымша кеңейтімдерін орнату оңай.

Python-ға қол жеткізу

Python интерпретаторын жүктеп алу: http://www.python.org/getit

Windows жүйесінде Pythonмен есеп шығару үшін Anaconda, Jupyter notebook орталарын жүктеп алу ұсынылады: https://www.anaconda.com/products/distribution#windows

Python-ң кеңінен қолданылатын екі модулі:

1) Numpy үшін сілтеме: https://numpy.org/

2) Matplotlib үшін сілтеме: http://matplotlib.sourceforge.net/contents.html

Python бойынша қосымша кітап:

Hans P. Langtangen A Primer on Scientific Programming with Python (Springer- Verlag, 2016).

Python негізі

Айнымалылар

Python интерпретаторын қосу

(Пуск→Python 3.10→IDLE):

```
>>> b = 2          # b бүтін типті
>>> print(b)
2
>>> b = b*2.0      # Енді b өзгермелі нүкте типті
>>> print(b)
4.0
```

Жолдар(Strings)

Жол – бір немесе қос тырнақшаға алынған таңбалар тізбегі. Мысал:

```
>>> string1 = Сандық әдісті Python'
>>> string2 = 'ортасында жасау. '
>>> print(string1 + '  ' + string2)      # Біріктіру
Сандық әдісті Python ортасында жасау.
>>> print(string1[0:11])                 # Бөліктеу
Сандық әдіс
```

Жолды **split** командасы арқылы оның құрамдас бөліктеріне бөлуге болады. Мысалға,

```
>>> s = '3 9 81'
>>> print(s.split())          # Бөлгіш – бос орын
['3', '9', '81']
```

Кортеждер (Tuples)

Кортеж – үтірмен бөлінген және жақшаға алынған ерікті нысандар тізбегі.

```
>>> rec = ('Kasenov', 'Syrym',(16,3,83)) # Mynau kortej
>>> Familia, Aty, tugankuni = rec # jaqshadan shygaru
>>> print(Aty)
Syrym
>>> tuganjyly = tugankuni [2]
>>> print(tuganjyly)
83
>>> name = rec[1] + ' ' + rec[0]
>>> print(name)
Syrym Kasenov
>>> print(rec[0:2])
('Kasenov', 'Syrym')
```

Тізімдер (Lists)

Тізім кортежге ұқсас, бірақ ол өзгермелі. Тізім оны квадрат жақшаға алу арқылы анықталады:

```
>>> a = [1.0, 2.0, 3.0]        # Тізім құру
>>> a.append(4.0) # Тізімге 4.0 элементін енгізу
>>> print(a)
[1.0, 2.0, 3.0, 4.0]
>>> a.insert(0,0.0)  # 0 позициясына 0.0 енгізіңіз
>>> print(a)
[0.0, 1.0, 2.0, 3.0, 4.0]
>>> print(len(a))   # Тізім ұзындығын анықтаңыз
5
>>> a[2:4] = [1.0, 1.0, 1.0]  #Таңдалған элем-ді өзгерту
>>> print(a)
[0.0, 1.0, 1.0, 1.0, 1.0, 4.0]
```

Арифметикалық операторлар

+	Қосу
−	Алу
*	Көбейту
/	Бөлу
**	Дәрежеге шығару
%	Модульдік бөлу

$a \mathrel{+}= b$	$a = a + b$
$a \mathrel{-}= b$	$a = a - b$
$a \mathrel{*}= b$	$a = a * b$
$a \mathrel{/}= b$	$a = a/b$
$a \mathrel{**}= b$	$a = a ** b$
$a\% = b$	$a = a\%b$

Салыстыру операторлар True немесе False мәнін қайтарады.

<	Кем
>	Үлкен
<=	Кем немесе тең
>=	Үлкен немесе тең
==	Тең
! =	Тең емес

Шартты (Conditionals) және циклді (Loops) операторлар

1 – sign(a) функциясын анықтайтын бағдарлама:

```
def sign_of_a(a):
    if a < 0.0:
        sign = 'теріс сан'
    elif a > 0.0:
        sign = 'оң сан'
    else:
        sign = 'нөл'
    return sign
a = 1.5
print('a - ' + sign_of_a(a))
```

Бағдарлама нәтижесі:

a - оң сан

2 – $[1, 1/2, 1/3, \ldots]$ тізімді жасайтын мысал:

```
nMax = 5
n = 1
a = [] # Бос тізім жасаңыз
while n < nMax:
    a.append(1.0/n) # Элем-ті тізімге қосу
    n = n + 1
print(a)
```

Бағдарлама нәтижесі:

[1.0, 0.5, 0.33333333333, 0.25]

Алдыңғы мысалды for операторымен :

```
nMax = 5
a = []
for n in range(1,nMax):
    a.append(1.0/n)
print(a)
```

Мұндағы n – мақсат, ал диапазоны $[1, 2, \ldots, nMax - 1]$ (range функциясын шақыру арқылы жасалған) – тізбек.

Цикл break операторы арқылы аяқталуы мүмкін.

Continue операторы кейінгі командаларды орындамай-ақ бірден циклдің басына апарады.

Деректерді оқу және басып шығару

```
a = input('a енгізіңіз: ')
print(a, type(a)) # a және оның типін басып шығару
b = eval(a)       # eval() жолды санға түрлендіреді
print(b,type(b)) #  b және оның типін басып шығару
```

```
 a енгізіңіз: 10.0
10.0 <class 'str'>
10.0 <class 'float'>
 a енгізіңіз: 11**2
11**2 <class 'str'>
121 <class 'int'>
```

print() функциясымен нәтижені басып шығару

```
>>> a = 1234.56789
>>> b = [2, 4, 6, 8]
>>> print(a, b)
1234.56789 [2, 4, 6, 8]
>>> print('a =',a, '\n b =',b)
a = 1234.56789
b = [2, 4, 6, 8]
```

print функциясы әрқашан жолдың соңына жаңа жол таңбасын қосады. end кілт сөзінің аргументін пайдалану арқылы '\n' таңбаны алмастыра аламыз.

Мысалы, print(нысан1,нысан2,...,end=’ ’)

Файлды ашу және жабу

Деректер файлына қол жеткізу үшін file_object = open(файл аты, әрекет) команда арқылы жасалады.

Мұндағы файл аты – ашылатын файлды көрсететін жол және әрекет келесілердің бірі болып табылады:

'r'	Бар файлдан оқу.
'w'	Файлға жазу. Егер файл атауы жоқ болса, ол жасалады.
'a'	Файлдың соңына қосыңыз.
'r+'	Бар файлды оқу және одан жазу.
'w+'	'r+' сияқты, бірақ файл аты жоқ болса жасалады.
'a+'	'w+' сияқты, бірақ деректер файлдың соңына қосылады.

Файлға кіру қажет болмаған кезде оны жабу:*file_object.close()*

Файлдан деректерді оқу

Мысал:

sunspots.txt атты файлды ашып оқу.

Әр жолдың келесі форматта (жыл/ай/күн/қарқындылық):

1896 05 26 40.94

1896 05 27 40.58

1896 05 28 40.20

т.б.

Біздің міндетіміз – файлды оқу және тек қарқындылықты қамтитын x тізімін жасау.

Файлдағы әрбір жолды split командасы арқылы бөліктерге бөлеміз: ['1896','05','26','40.94'].

Қарқындылық тізімнің [3] элементін x-ке енгіземіз.

Мынау алгоритмі:

```
x = []
data = open('sunspots.txt','r')
for line in data:
    x.append(eval(line.split()[3]))
data.close()
```

Деректерді файлға жазу

Мысал ретінде, $k = 101$-ден 110-ға дейін k және k^2 форматталған кестесін **testfile** файлына жазайық. Файлға жазуды жүзеге асыратын бағдарлама:

```
f = open('testfile.txt','w')
for k in range(101,111):
    f.write('{:4d} {:6d}'.format(k,k**2))
    f.write('\n')
f.close()
```

'testfile.txt мазмұны мыналар

```
 101  10201
 102  10404
 103  10609
 104  10816
 105  11025
 106  11236
 107  11449
 108  11664
 109  11881
 110  12100
```

Қорытынды

1. Python тілі жайлы ақпарат және орнату.
2. Python негізі: айнымалы, жол, кортеж, тізім.
3. If шартты және while, for циклдік операторлар.
4. Деректерді файлдан оқу және жазу.

Пайдаланылған әдебиеттер тізімі

1. Jaan Kiusalaas. Numericalmethods in engineering with Python. Cambridge University Press. ISBN 978-1-107-03385
2. Вабищевич П.Н. Численные методы: Вычислительный практикум. — М.: Книжный дом «ЛИБРОКОМ», 2010. — 320 с.
3. Киреев В. И., Пантелеев А. В. Численные методы в примерах и задачах: Учебное пособие. —СПб.: Издательство «Лань», 2015. — 448 с.