

Дәріс №3 Python тілінің негіздері. Python бағдарламалау тілінің қарапайым конструкциялары.

Қолданыстағы барлық бағдарламалар мен қолданбалар пайдаланушымен бір немесе басқа жолмен өзара әрекеттеседі. Мұны істеу үшін олар пайдаланушыдан деректерді алады, оны өңдейді және жауап ретінде бір нәрсені шығарады. Пайдаланушымен әрекеттеспейтін қолданбаларда ешқандай пайда жоқ.

Python-да толыққанды қолданбада пайдаланушының өзара әрекеттесуін жүзеге асыру үшін жеткілікті пішімделген енгізу және шығару үшін кірістірілген қуатты құралдар бар.

Түбіртек деректер енгізу

Деректерді жинау кез келген бағдарламаның ең маңызды құрамдастарының бірі болып табылады. Мысалы, қарапайым калькулятор пайдаланушыдан деректерді пернетақтадан сандарды енгізгенде, калькулятор интерфейсінің элементтерін басқанда, тіпті көшірілген өрнекті қойғанда алады.

Енгізу деректер В Python әдетте өңдеуге болатындай айнымалыларда немесе тізімдерде сақталады .

Синтаксис операциялар енгізу көрінеді Сонымен :

```
a=input()
```

айнымалысына жазылатын енгізуді күтеді .

Сонымен қатар, оператор `енгізу()` енгізуді сұрамас бұрын кейбір мәтінді көрсетуге мүмкіндік береді, мысалы:

```
a=input("Введитесвоеимя")
```

Осылайша сіз пайдаланушыға не енгізу керектігі туралы кеңес бере аласыз. Қатты терілген бағдарламалау тілдерінен айырмашылығы, Python кез келген деректер түрін нақты көрсетпей енгізуге мүмкіндік береді. Яғни, пайдаланушы жолды да енгізе алады саны . Команда синтаксисі бұл кірісті өзгертпейді.

Дегенмен, жиі белгілі бір типтегі деректерді алу қажет, мысалы, калькулятор санды күтеді және пайдаланушы жолды енгізеді.

Ол үшін `cast` түрі пайдаланылады, бұл жағдайда бүтін санға :

Пайдаланушы жолды немесе бөлшекті енгізсе ,

```
a=int(input("Введитецелоечисло"))
```

Python аудармашысы ерекше жағдайды шығарады . Сондықтан **`try-except` операторы енгізу үшін пайдаланылады :**

```
try:
    a=int(input("Введитецелоечисло"))except:
    print("Ошибка,введенонецелоечисло")
```

Саннның енгізілгенін тексеру туралы қосымша ақпаратты бөлек мақаладан оқи аласыз .

Сондай-ақ пайдаланушы деректердің қажетті түрін енгізген кезде ғана шығатын шексіз циклде енгізуді сұрауға болады :

```
whileTrue:try:
    a=int(input("Введитецелоечисло:"))except:
    print("Вывелинецелоечисло,ошибка")continue
break
```

Шығару тек деректерді экранда көрсету үшін ғана емес, сонымен қатар бір программаның шығысын басқасының кірісіне қайта бағыттау немесе файлға шығаруды жазу және т.б.

Python деректерді басып шығару үшін print() функциясын пайдаланады. Оны жолдарды, сандарды және тізбектерді көрсету үшін пайдалануға болады. **Оның синтаксисі келесідей көрінеді :**

```
print("Строка")

print(5)
```

Бағдарламашы функционалды аргументтерді пайдаланып деректер шығысын теңшей алады :

***объектілер**

Бұл экранда көрсетілетін деректер. Бағдарламашы функцияға кез келген объектілер санын береді, олар автоматты түрде жолдық форматқа түрленеді және шығыс ағынына жіберіледі.

Нысандарды функцияға бергенде, бағдарламашы оларды үтірмен ажырата алады немесе біріктіруді қолдана алады. Бірнеше аргументтерді беру *objects және басқа қосымша дәлелдер арасында қайшылық тудырмайды, себебі аудармашы автоматты түрде таниды кілт сөз бойынша басқа параметрлер (sep, end, file, flush)!

Міне , мысал :

```
print("apple","orange"+"banana","cherry")

appleorangebananacherry
```

қыркүйек

Бұл аргумент өткізілген нысандар арасында не пайда болатынын анықтайды. Әдепкі бойынша, sep параметрінде « » мәні болады, яғни шығару кезінде бірнеше объектілер, арасында олар ерік бос орын орнатылса, **оны басқа таңбаға өзгертіңіз, мысалы “-“ :**

```
print("Один","Два","Три",sep="-")
```

Один-Два-Три

Соңы

Аргумент print() функциясы аяқталатын таңбаны көрсетеді. Әдепкі бойынша, бұл «\n» жол үзілімінің таңбасы, яғни барлық деректер шығарылғаннан кейін каретка келесі жолға өтеді және басып шығару функциясының келесі шығысы жаңа жолда басталады. Бірнеше басып шығарудың бір жолда болуы қажет болса, соңғы таңбаны ауыстыру керек.

Міне , мысал :

```
print("Один")print("Два",end="")  
print("Три")
```

ОдинДваТ
ри

файл

Бұл аргумент шығыс ағынын қайта анықтауға мүмкіндік береді. Әдепкі мән «sys.stdout», яғни деректер консольге шығарылады. Оның орнына файлды көрсетсеңіз , деректер оған жазылады.

Міне temp.txt файлын ашатын және басып шығару функциясын пайдаланып оған мәтін жазатын Python сценарийінің мысалы :

```
withopen("temp.txt","w")asf: print("Write to file",  
file=f)
```

Бұл қосымша аргумент шығыс ағынын тазалауға мүмкіндік береді. Әдепкі мән False болып табылады, шығыс буферленген кезде, бағдарламашы шығысты дереу шығару үшін «True» мәнін көрсете алады. Мысалы, егер сіз файлға деректерді жазсаңыз, ол буферге түседі және буфер толған немесе файл жабылған кезде файлға жазылады.

Консольге шығару сияқты деректер буферге жазылады және одан консольге шығады. Егер тазалау True мәніне орнатылса, шығыс бірден шығарылады.

```
print("Sometext",flush=True)
```

Пішімделген қорытынды

Консольге шығатын ақпарат оңай оқылып, түсінікті болуы үшін оны пішімдеу керек. Python бағдарламашыға консоль шығысын әртүрлі жолдармен пішімдеуге арналған құралдарды ұсынады. Олардың көмегімен сіз тіпті тегіс және оқылатын кесте жасай аласыз.

Оператор %

% операторы пішімдеу үшін жиі пайдаланылады және оны тек Python бағдарламашыларына ғана емес, сонымен қатар C/C++ бағдарламашыларына да жақсы біледі. Негізінде, % операторы орнына мән ауыстырылатын белгі болып табылады айнымалы немесе өрнек. % операторы тікелей шығысқа орналастырылады түзу, А ауыстырылды құндылықтар — В кейін кортеж екінші

%. Мысал :

```
print("Программаномер%d,получиларезультат:%5.2f"%(4,7.3333))
```

```
Программаномер4,получиларезультат:7.33
```

Сонымен қатар символы % көрсетілген түрі мағыналары, А Сондай-ақ, Авторы қалаған, ұзындық және дәлдік. **Синтаксис келесідей көрінеді :**

```
%(флаги)(ширина)(точность)тип
```

Егер қарастыру мысал, мүмкін мынаны қараңыз :

- " % 2d " кортеждің бірінші элементі, бүтін "4" үшін пайдаланылады. Ұзындығы «2» % кейін көрсетілгендіктен және кортеждегі сан тек бір цифрдан тұратындықтан, шығыста қосымша бос орын қосылады.
- " % 5.2f " - өзгермелі нүкте сандарына арналған пішім. «5» болып табылады жолда болу керек цифрлардың жалпы саны, ал ".2" - ондық бөлшектен кейінгі цифрлар саны.

Әдіс пішім

Бұл құрал Python 2.6 нұсқасында қосылды. Алдыңғы әдіспен салыстырғанда пішім сәл қиынырақ болып шықты. Бағдарламашы қай орынды белгілеу үшін «{ }» таңбасын пайдаланады айнымалы немесе өрнек ауыстырылады. Бұйра жақшалардың ішінде атын, мән түрін, енін және дәлдігін көрсетуге болады.

Ауыстырылған мәндер пішім кілт сөзінен кейін кортежде көрсетіледі. Олар позициялық түрде көрсетілуі немесе аты бойынша тағайындалуы мүмкін. Мысалы:

```
print('{0}{1}январяпразднует{other}деньрождения'\n      .format('Никита','25',other='15-ый'))
```

```
Никита25январяпразднует15-ыйденьрождения
```

Пішімдеу әдісі сонымен қатар сандарды пішімдеуге, яғни көрсетуге мүмкіндік береді олардың ені мен дәлдігі:

```
print("Робот№{r:4d}весит{w:6.2f}"
      .format(r=1134,w=112.5534))
```

Робот№1134весит112.55

Бұл әдіс деректерді шығаруды өте икемді түрде пішімдеуге мүмкіндік береді. Мысалы, бағдарламашы дайын сөздіктен жолға деректерді бере алады:

```
data = dict(name ="Бендер", race ="Робот")
print("Это{name},он-{race}".format(**data))
```

ЭтоБендер,он-Робот

Түрлендіру түрі деректер

Кейде кірістірілген түрлер арасында түрлендіру қажет болуы мүмкін. Түрлерді түрлендіру үшін сіз жай ғана түр атауларын функция ретінде пайдаланасыз.

Бір деректер түрінен екіншісіне түрлендіруге арналған бірнеше кірістірілген функциялар бар. Бұл функциялар түрлендірілген мәнді көрсететін жаңа нысанды қайтарады.

Функция	Сипаттама
int(x [,база])	Түрлендіреді x B бүтін сан .
қалқымалы(x)	Түрлендіреді x B саны бірге қалқымалы үтір.
күрделі (нақты [,сурет])	жасайды жан-жақты саны.
str(x)	Түрлендіреді объект x B жол өнімділік.
қайталау(x)	Түрлендіреді объект x B жолды өрнегі.
бағалау(str)	Жолды өңдейді және нысанды қайтарады. Жолмен көрсетілген бағдарлама кодын орындау үшін қолданылады.
кортеж(лер)	Түрлендіреді c B кортеж.
тізім(дер)	Түрлендіреді c B тізім.
жинақ(лар)	Түрлендіреді c B жинақ.
дикт(д)	Сөздік жасайды . d керек (кілт, мән) кортеждердің тізбегі болуы.
мұздатылған жинақ(тар)	Түрлендіреді c B өзгермейтін жинақ.
chr(x)	Түрлендіреді тұтас саны B символы.
Unichr(x)	Түрлендіреді тұтас саны B символы Юникод.

Жүйелер өлі есеп

Python әртүрлі санау жүйелеріндегі сандармен жұмыс істеуге және сандарды бір санау жүйесінен екіншісіне түрлендіруге мүмкіндік береді. Ол үшін бірнеше кірістірілген функциялар қолданылады:

- `int(сан, радикус)` – қажетті санау жүйесіндегі санды бүтін санға түрлендіреді. Әдепкі санау жүйесі ондық.
- `себет(сан)` — түрлендіреді саны В екілік.
- `он алтылық(сан)` — түрлендіреді саны В он алтылық.
- `қазан(х)` – түрлендіреді саны В сегіздік.

```
print(oct(10)) # Выведет '0o12' print(hex(11)) #  
Выведет '0xb' print(bin(11))#Выведет'0b1011'
```