

Stringid ehk Sõned

Sõne loomine. Sõnede loomiseks kasutatakse topeltjutumärke (ühekordseid) " või (kahekordseid) "":

In []:

```
s1 = 'Python'  
s2 = "Python"
```

Sõne lugemine - funktsioon input():

In []:

```
s = input() # tekst  
num = int(input()) # teksti sisestamine ja selle täisarvuks teisendamine
```

Sõne pikkus. Sõne pikkuse (sümbolite arvu) määramiseks - funktsioon len():

In []:

```
s = 'Hello, Python'  
n = len(s)  
print(n)
```

13

Operaatoreid + ja * saab kasutada sõnede jaoks. Operaator + seob kaks või enam sõnet. Seda nimetatakse sõnede konkatenatsiooniks. Operaator * kordab sõnet määratud arvu kordi.

In []:

```
print('Hello'+ ' '+ 'Python'*5)
```

Hello PythonPythonPythonPythonPython

Kuuluvuse operator in. In operaatori abil saab kontrollida, kas üks sõne on teise alamsõne

In []:

```
s = 'Python is a popular programming language.'  
if 'Python' in s:  
    print('YES')  
else:  
    print('NO')
```

YES

Sõnede indekseerimine. Kui on vaja pöörduda konkreetse sümboli poole sõnes, kasutatakse nurksulgusid [], kus näidatakse soovitud sümboli indeksit (numbrit) sõnes. Numeratsioon algab 0-st.

In []:

```
s = 'Python'  
print(s[0], s[5])
```

P n

Erinevalt paljudest programmeerimiskeeltest on Pythonil võimalik töötada negatiivsete indeksitega. Kui sõne esimesel sümbolil on indeks 0, siis viimasele elemendile omistatakse indeks -1.

In []:

```
s = 'Python'  
print(s[-1], s[-6])
```

n P

Iga sümboli töötlemine sõnes

In []:

```
s = 'Python'  
for i in range(len(s)):  
    print(s[i])
```

P
y
t
h
o
n

In []:

```
# Kui pole indeksit vaja  
s = 'Python'  
for c in s:  
    print(c)
```

P
y
t
h
o
n

Mida näitab alljärgnev koodiosa?

```
s = 'abcdefg'
```

```
print(s[0]3 + s[-1]3 + s[3]2 + s[3]2)
```

```
s = '01234567891011121314151617'
```

```
for i in range(0, len(s), 5):
```

```
    print(s[i], end='')
```

```
</code>
```

Ülesanne 1

Kasutaja sisestab sõnet. Kirjutage programm, mis kuvab

- sõne elemendid veerus vastupidises järjekorras
- sõne elemendid ühel real vastupidises järjekorras

Ülesanne 2

Programmi sisendiks on numbritest koosnev sõne. Kirjutage programm, mis arvutab selle sõne numbrite summa.

Ülesanne 3

Programmi sisendiks on sõne. Kirjutage programm, mis määrab, mitu korda sümbolid a ja b sõnes esinevad.

Sõne lõik

Lõigu saab kasutada sõnest mitme sümboli saamiseks, luues kooloniga eraldatud indeksivahemiku.

`s[n:m]` esimene number - kust lõik algab (kaasa arvatud) ja teine number - kus lõik lõpeb (välja arvatud). Luuakse uus sõne.

In []:

```
s = 'Python'  
print(s[2:4])
```

th

- `s[n:]` lõik sõne lõpuni, alustades indeksist `n` (Pythonis algab indekseid nummerdamine 0-st).
- `s[:n]` lõik sõne algusest `n-1` indeksini.
- Lõik `s[:]` langeb sõnega `s` kokku.

In []:

```
s = 'Python'  
print(s[2:])  
print(s[:2])  
print(s[:])
```

thon
Py
Python

Lõikude loomiseks võib kasutada negatiivseid indekseid. Negatiivsed reaindeksid algavad `-1` ja loetakse kuni sõne alguseni.

Negatiivsete indekseid kasutamisel peab esimene lõiguparameeter olema väiksem kui teine või vahele jäetud.

In []:

```
s = 'Python'  
print(s[-2:])  
print(s[:-2])  
print(s[-4:-2])
```

on
Pyth
th

Lõigu samm on kolmas valikuline parameeter.

Kui lõigu sammuks on määratud negatiivne arv, lähevad sümbolid vastupidises järjekorras.

In []:

```
s = 'Python Programming Language'  
print(s[::2])  
print(s[:10:3])  
print(s[::-1])  
print(s[::-3])
```

Pto rgamn agae
Ph o
egaugnaL gnimmargorP nohtyP
euagmrrnt

Ülesanne 4

On antud sõne `s = 'Python is a programming language that lets you work quickly and integrate systems more effectively'`

Printige:

- sõne `s` esimesed 10 sümbolit
- sõne `s` viimased 10 sümbolit
- sõne `s` iga 3. sümbol alates sõne algusest
- sõne `s` vastupidises järjekorras
- sõne, millest on eemaldatud esimene ja viimane sümbol

Ülesanne 5

Kirjutage program, mis tükeldab sõne kaheks võrdseks osaks, paigutab need ümber ja kuvab need ekraanil. Kui sõne pikkus on paaritu, peab esimene osa olema ühe sümboli võrra pikem.

Meetodid ja funktsioonid

`min()`, `max()`, `len()`, `int()`, `float()` on funktsioonid

Meetod on funktsioon, mida rakendatakse objektile.

Objekti_nimi.meetodi_nimi(parameetrid)

Näiteks meetod `lower()` tagastab s sõne koopia, kus kõigil sümbolitel on alaregister.

```
s = 'Python'
```

```
s_lower = s.lower()
```

```
</code>
```

Sõnede meetodid

- `S.find(str, [start],[end])` Alamsõne otsimine sõnest. Tagastab esimese siseneise numbri või `-1`.
- `S.rfind(str, [start],[end])` Alamsõne otsimine sõnest. Tagastab viimase siseneise numbri või `-1`.
- `S.index(str, [start],[end])` Alamsõne otsimine sõnest. Tagastab esimese siseneise numbri või kutsub `ValueError`
- `S.rindex(str, [start],[end])` Alamsõne otsimine sõnest. Tagastab viimase siseneise numbri või kutsub `ValueError`
- `S.replace(mall, asendus[, maxcount])` Malli asendamine asenduse vastu. `maxcount` piirab asenduste arvu
- `S.split(символ)` Sõne tükeldamine eraldajaga
- `S.isdigit()` Kas sõne koosneb numbritest
- `S.isalpha()` Kas sõne koosneb tähtedest
- `S.isalnum()` Kas sõne koosneb numbritest või tähtedest
- `S.islower()` Kas sõne koosneb alaregistris sümbolitest
- `S.isupper()` Kas sõne koosneb ülaregistris sümbolitest
- `S.isspace()` Kas sõne koosneb kuvamata sümbolitest
- `S.istitle()` Kas sõnad sõnes algavad suure algustähega
- `S.upper()` Sõne teisendamine ülaregistrile
- `S.lower()` Sõne teisendamine alaregistrile
- `S.startswith(str)` Kas sõne `S` algab `str` malliga
- `S.endswith(str)` Kas sõne `S` lõpeb `str` malliga
- `ord(символ)` Sümbol -> ASCII
- `chr(арв)` ASCII -> Sümbol
- `S.capitalize()` Viib sõne esimese sümboli ülaregistrisse ja kõik teised alaregistrisse
- `S.lstrip([chars])` Tühisümbolite eemaldamine sõne alguses
- `S.rstrip([chars])` Tühisümbolite eemaldamine sõne lõpus
- `S.strip([chars])` Tühisümbolite eemaldamine sõne alguses ja lõpus

- `S.swapcase()` Muudab sümbolite alaregistri ülaregistriks ja ülaregistri alaregistriks
- `S.title()` Muudab iga sõna esimese tähe suurtäheks ja kõik teised väiketähtedeks
- `S.zfill(width)` Muudab sõne pikkuse vähemalt laiuseks (`width`), täites vajaduse korral esimesed sümbolid nullidega
- `S.ljust(width, fillchar=" ")` Muudab sõne pikkuse vähemalt laiuseks (`width`), täites vajaduse korral viimased sümbolid sümboliga `fillchar`
- `S.rjust(width, fillchar=" ")` Muudab sõne pikkuse vähemalt laiuseks (`width`), täites vajaduse korral esimesed sümbolid sümboliga `fillchar`
- `S.count(str, [start],[end])` Tagastab alamsõne sisestuste arvu vahemikus [`algus`, `lõpp`] (`0` ja vaikimisi sõne pikkus)

[Link \(https://pythonworld.ru/typy-dannyx-v-python/stroki-funkcii-i-metody-strok.html\)](https://pythonworld.ru/typy-dannyx-v-python/stroki-funkcii-i-metody-strok.html)

In []:

```
print('S.find(str, [start],[end])Alamsõne otsimine sõnest. Tagastab esimese sisenemise numbri või -1')
s = 'Hello, world'
n = s.find('o')
print(n)
n = s.find('o', 6)
print(n)
```

```
S.find(str, [start],[end])Alamsõne otsimine sõnest. Tagastab esimese sisenemise numbri või -1
4
8
```

Ülesanne 6

Kirjutage programm, mis kuvab ekraanile sümboli, mis ilmub kõige sagedamini.

Ülesanne 7

Kirjutage programm teksti kodeerimiseks Caesari šifriga([Wikipedia \(https://et.wikipedia.org/wiki/Caesari_nihe\)](https://et.wikipedia.org/wiki/Caesari_nihe)).

Kasutaja sisestab sõne ja sümbolite arvu paremale nihutamiseks.