Prof. dr. sc. Tomislav Pribanić
Izv. prof. dr. sc. Marija Seder
Doc. dr. sc. Jurica Babić
Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva
Vojni studijski program: Vojno inženjerstvo
Vojno vođenje i upravljanje

Informatika

Uvod u Python III

Sadržaj

- Stringovi
- Liste

Primjer: string

- Napišite program koji će izdvojiti riječi u unesenom tekstu.

- Tekst → niz riječi odvojenih razmacima
- Riječ → niz znakova
- Niz znakova → STRING

Operatori sa stringovima

Operator	Opis
`+`	Nadovezivanje
`*`	Uvišestručenje
`in`	Prvi string sadržan u drugom stringu
`not in`	Prvi string nije sadržan u drugom stringu

```
>>> 'abcdefghij'
'abcdefghij'
>>> 'abc'+'de'+'fghi'
'abcdefghi'
>>> 'JKL' * 4
'JKLJKLJKLJKL'
>>> 'na' in 'Kuća na plaži'
True
>>> 'na' not in 'Kuća na plaži'
False
```

Ugrađene funkcije za stringove (1)

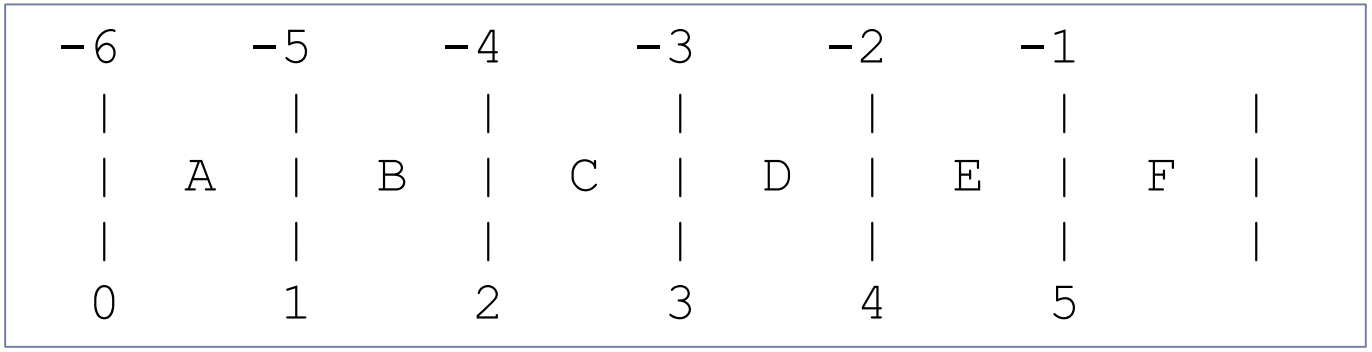
Funkcija	Opis
`len(s)`	vraća duljinu stringa
`min(s)`	vraća znak s najmanjom kodnom vrijednošću (ascii kod)
`max(s)`	vraća znak s najvećom kodnom vrijednošću
`ord(s)`	vraća dekadski kod pojedinog znaka
`chr(n)`	vraća znak pojedinog dekadskog koda
`str(n)`	vraća znakovni prikaz broja n

Ugrađene funkcije za stringove (2)

```
>>> s = 'Moj prvi string'
>>> len(s)
15
>>> min(s)
' '
>>> max(s)
'v'
>>> ord(' ')
32
>>> ord('i')
105
>>> chr(105)
'i'
>>> str(105)
'105'
>>> len('')
0
```

Dohvaćanje pojedinačnih znakova

- Svaki član se može dohvatiti indeksiranjem
- Indeks - položaj znaka u nizu (stringu)
  - Prvi znak u nizu → indeks `0`
  - Zadnji znak u nizu → indeks `n = len(s)-1`



```
>>> s = 'Slika je na zidu.'
>>> s[0]
'S'
>>> s[5]
' '
>>> s[17]

Traceback (most recent call last):
  File "<pyshell#31>", line 1, in <module>
    s[17]
IndexError: string index out of range
>>> len(s)
17
>>> s[16]
'.'
>>> s[len(s)]

Traceback (most recent call last):
  File "<pyshell#34>", line 1, in <module>
    s[len(s)]
IndexError: string index out of range
>>> s[len(s)-1]
'.'
```

```
>>> s[-1]
'.'
>>> s[-5]
'z'
>>> s[-17]
'S'
>>> s[-18]

Traceback (most recent call last):
  File "<pyshell#39>", line 1, in <module>
    s[-18]
```

Primjer: riječ naopako

- Napišite program koji će zadanu riječ napisati naopako.
- Npr. `slika` → `akils`
- Pomoć:
- `rijec = input('upisi rijec: ')`

```
s_naopako=''
for i in range(len(s)):
    s_naopako = s_naopako + s[len(s)-i-1]
```

```
s_naopako=''
for i in range(len(s)):
    s_naopako = s[i] + s_naopako
```

Primjer: prebroji samoglasnike

- Napiši program koji će prebrojati koliko ima samoglasnika u unesenoj riječi.

```
Unesi rijec: računalo
U unesenoj rijeci ima 4 samoglasnika.
```

```
samoglasnik = 0
for i in range(len(rijec)):
    if rijec[i]=='a' or rijec[i]=='e' or rijec[i]=='i' \
       or rijec[i]=='o' or rijec[i]=='u':
        samoglasnik += 1
```

```
samoglasnik = 0
for i in range(len(rijec)):
    if rijec[i] in 'aeiou':
        samoglasnik += 1
```

Primjer: izdvoji samoglasnike

- Napiši program koji će izdvojiti sve samoglasnike iz riječi.

```
>>>
Unesi rijec: računalo
U unesenoj rijeci ima 4 samoglasnika.
Ti samoglasnici su: auao
>>>
```

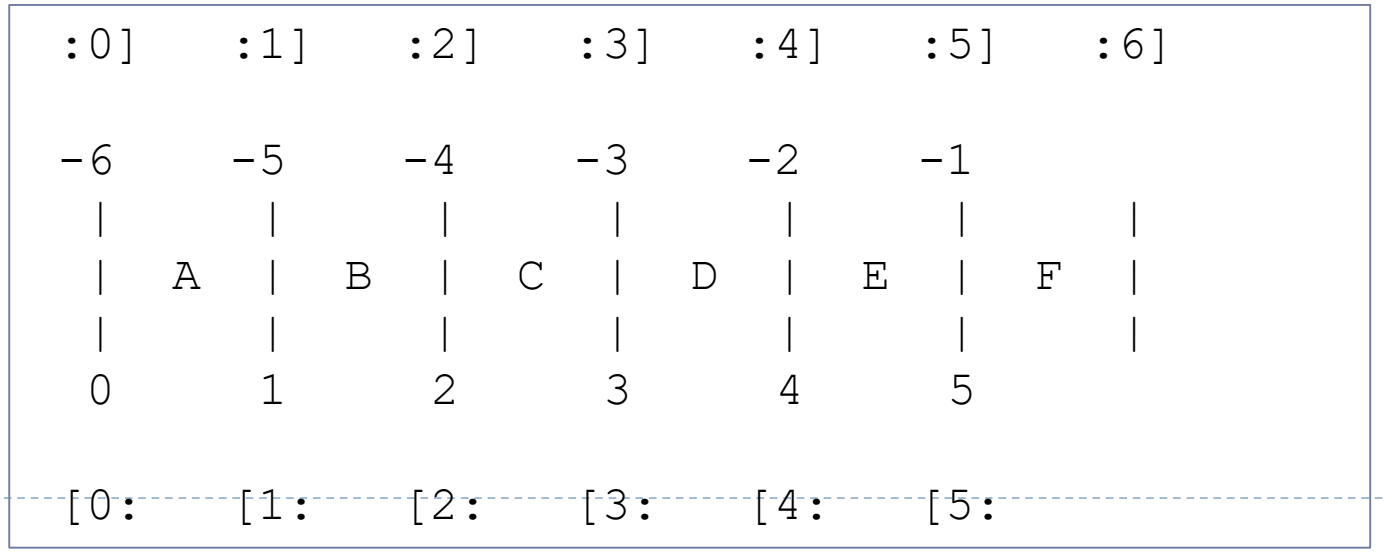
```
samoglasnici = ''
for i in range(len(rijec)):
    if rijec[i]=='a' or rijec[i]=='e' or rijec[i]=='i' or rijec[i]=='o' or rijec[i]=='u':
        samoglasnici = samoglasnici + rijec[i]
```

Zadaci za vježbu

- Napiši funkciju koja će izdvojiti sve suglasnike.
- Napiši funkciju koja će izdvojiti sve samoglasnike, i koja će ispravno raditi i za velika i za mala slova.
- Napiši funkciju koja će iza svakog samoglasnika staviti znak *

Isječak znakovnog niza

- `s[pocetak : kraj]`
  - `pocetak` ☐ indeks prvog znaka
  - `kraj` ☐ zadnji znak će biti s indeksom `kraj-1`
- **Isječak počinje od prvog znaka:** `s[0:kraj]` **ili** `s[:kraj]`
- **Isječak završi zadnjim znakom:** `s[pocetak:len(s)]` **ili** `s[pocetak:]`
- **Isječak je cijeli string:** `s[:]`



Primjer: isječak znakovnog niza

```
>>> s = 'abcdefghij'
>>> len(s)
10
>>> s[:]
'abcdefghij'
>>> s[0:]
'abcdefghij'
>>> s[4:len(s)]
'efghij'
>>> s[4:]
'efghij'
>>> s[0:5]
'abcde'
>>> s[:5]
'abcde'
>>> s[3:5]
'de'
>>> s[6:6]
''
>>> s[6:4]
''
```

Korak isijecanja

- `s[pocetak : kraj : korak]`
  - `poetak` → indeks prvog znaka
  - `kraj` → zadnji znak će biti s indeksom `kraj-1`
  - `korak` → preskoči dio znakova
- ako je `korak<0` → ispis je obratnim redom

```
>>> s[0:10:1]
'abcdefghij'
>>> s[0:10:2]
'acegi'
>>> s[2:6:2]
'ce'
>>> s[::2]
'acegi'
>>> s[9:0:-1]
'jihgfedcb'
>>> s[9::-1]
'jihgfedcba'
>>> s[::-1]
'jihgfedcba'
>>> s[6:4:-1]
'gf'
>>> s[9:1:-2]
'jhfd'
```

Promjena znaka u stringu

- Napišite program koji će znak na i-tom indeksu zamijeniti novim znakom.

```
>>> rijec='racunalo'
>>> rijec[2]='T'
Traceback (most recent call last):
  File "<pyshell#49>", line 1, in <module>
    rijec[2]='T'
TypeError: 'str' object does not support item assignment
```

- rijec=rijec[:2]+'T'+rijec[3:]

```
>>>
Unesite riječ: racunalo
Unesite i-ti indeks: 2
Unesite novi znak: T
raTunalo
```

Metode za stringove (1)

Metoda	Upotreba	Opis
`center(w)`	`s.center(w)`	string `s` ispisan u centru polja širine `w`
`ljust(w)`	`s.ljust(w)`	string `s` ispisan lijevo u polju širine `w`
`rjust(w)`	`s.rjust(w)`	string `s` ispisan desno u polju širine `w`

```
>>> s = 'olovka je na stolu.'
>>> print(s.center(50))
                  olovka je na stolu.
>>> print(s.ljust(50))
olovka je na stolu.
>>> print(s.rjust(50))
                                   olovka je na stolu.
>>> s.center(50)
'                  olovka je na stolu.                  '
```

Metode za stringove (2)

Metoda	Upotreba	Opis
`capitalize()`	`s.capitalize()`	vraća kopiju stringa `s` s prvim slovom velikim
`lower()`	`s.lower()`	vraća kopiju stringa `s` sa svim slovima malim
`upper()`	`s.upper()`	vraća kopiju stringa `s` sa svim slovima velikima

```
>>> s
'olovka je na stolu.'
>>> s.capitalize()
'Olovka je na stolu.'
>>> s
'olovka je na stolu.'
>>> s1 = s.upper()
>>> s1
'OLOVKA JE NA STOLU.'
>>> s2 = s1.lower()
>>> s2
'olovka je na stolu.'
>>> s.upper().center(60)
'                    OLOVKA JE NA STOLU.                     '
```

Metode za stringove (3)

Metoda	Upotreba	Opis
`replace(stari, novi)`	`s.replace(stari, novi)`	vraća kopiju stringa `s` u kojem su sve pojave podstringa `stari` zamijenjene podstringom `novi`
`strip()`	`s.strip('izbaci')`	vraća kopiju stringa `s` iz kojeg su izbačeni znakovi `izbaci` s početka i kraja stringa (ako nema argumenata izbacuju se razmaci)
`index(s1)`	`s.index(s1)`	vraća poziciju prvog pojavljivanja stringa `s1` u stringu `s`

```
>>> s
'olovka je na stolu.'
>>> s1 = s.strip('.')
>>> s1
'olovka je na stolu'
>>> s2 = ' ' + s1 + '  '
>>> s2
' olovka je na stolu  '
>>> s2.strip()
'olovka je na stolu'
>>> s1.replace('je', 'NIJE')
'olovka NIJE na stolu'
>>> s1.index(' ')
6
```

Primjer: riječi u rečenici

- Napišite program koji će unositi rečenicu (niz znakova odvojenih jednim razmakom). Program treba ispisivati svaku riječ u novi redak, malim slovima.

```
>>>
Unesi recenicu: Olovka je na stolu.
olovka
je
na
stolu
```

```
recenica = input('Unesite recenicu: ')
recenica=recenica.lower().strip('.?!')
# print(recenica)
while ' ' in recenica:
    polozaj=recenica.index(' ')
    print(recenica[:polozaj])
    recenica=recenica[polozaj:].strip()
print(recenica) #ispisi posljednju rijec
```

Liste (polje, niz, array)

Liste (polje, niz, array)

- U liste se smještaju podaci koju su srodni na neki način

- Elementi liste mogu biti različitog tipa podataka

Stvaranje liste (1)

```
>>> lista = [4, 6, 1, 9, 3]
>>> lista
[4, 6, 1, 9, 3]
>>> bla = [4, 3.5, 'r', 5]
>>> bla
[4, 3.5, 'r', 5]
>>> lista = [i for i in range(20)]
>>> lista
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
>>> lista = [i for i in range(15, 20)]
>>> lista
[15, 16, 17, 18, 19]
>>> lista = [i for i in range(15, 20, 2)]
>>> lista
[15, 17, 19]
>>> lista = [1 for i in range(15)]
>>> lista
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
>>> lista = [i for i in range(20) if i%2==0]
>>> lista
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
>>> lista2 = [3*i for i in lista]
>>> lista2
[0, 6, 12, 18, 24, 30, 36, 42, 48, 54]
>>> lista = list(range(14))
>>> lista
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
>>> lista = list('neki string')
>>> lista
['n', 'e', 'k', 'i', ' ', 's', 't', 'r', 'i', 'n', 'g']
```

Stvaranje liste (2)

- Navođenje elemenata, odvojiti zarezom

```
lista = [prvi_element, drugi_element, treci_element]
```

- Korištenje for petlje

```
lista = [element for i in range(pocetak, kraj+1,
  korak)]
```

- Korištenje for petlje i uvjeta

```
lista = [element for i in range(pocetak, kraj+1,
  korak) if uvjet]
```

- Kopiranje dijelova druge liste

```
lista = [element for i in druga_lista]
lista = [element for i in druga_lista if uvjet]
```

- Naredba `list`

```
lista = list(range(pocetak, kraj+1, korak))
lista = list('neki string')
```

Dohvaćanje elemenata liste

- Kao kod stringova
- Prvi element ima indeks 0

```
>>> lista = list(range(1,15))
>>> lista
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
>>> lista[5]
6
>>> lista[0]
1
>>> lista[3:6]
[4, 5, 6]
```

Mijenjanje vrijednosti elementa liste

- Direktno pridruživanje nove vrijednosti na željeno mjesto (može li tako i kod stringova?)

```
>>> lista = list('ABCDEFGHIJK')
>>> lista
['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K']
>>> lista[0]
'A'
>>> lista[0]='Z'
>>> lista
['Z', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K']
>>> lista[-1]
'K'
>>> lista[-1]='Y'
>>> lista
['Z', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'Y']
>>> lista[4:6]
['E', 'F']
>>> lista[4:6]=['e', 'f']
>>> lista
['Z', 'B', 'C', 'D', 'e', 'f', 'G', 'H', 'I', 'J', 'Y']
```

Brisanje elemenata liste

- Elementi liste se brišu naredbom:

```
del lista[pocetak:kraj+1:korak]
```

```
>>> lista
['Z', 'B', 't', 'D', 'e', 'f', 'G', 'H', 'I', 'J', 'Y']
>>> del lista[2]
>>> lista
['Z', 'B', 'D', 'e', 'f', 'G', 'H', 'I', 'J', 'Y']
>>> del lista[-3:-1]
>>> lista
['Z', 'B', 'D', 'e', 'f', 'G', 'H', 'Y']
```

```
>>> del lista[2:6:2]
>>> lista
['Z', 'B', 'e', 'G', 'H', 'Y']
```

Kopiranje liste

- Ne može direktno znakom =, (plitka kopija)
- Promjena elemenata nove liste → mijenja i elemente originalne liste

```
>>> lista
['Z', 'B', 'C', 'D', 'e', 'f', 'G', 'H', 'I', 'J', 'Y']
>>> lista2 = lista
>>> lista2
['Z', 'B', 'C', 'D', 'e', 'f', 'G', 'H', 'I', 'J', 'Y']
>>> lista2[2]='t'
>>> lista2
['Z', 'B', 't', 'D', 'e', 'f', 'G', 'H', 'I', 'J', 'Y']
>>> lista
['Z', 'B', 't', 'D', 'e', 'f', 'G', 'H', 'I', 'J', 'Y']
```

- Načiniti isječak duljine cijele liste: `nova_lista = lista[:]`

```
>>> lista
['Z', 'B', 't', 'D', 'e', 'f', 'G', 'H', 'I', 'J', 'Y']
>>> lista3 = lista[:]
>>> lista3
['Z', 'B', 't', 'D', 'e', 'f', 'G', 'H', 'I', 'J', 'Y']
>>> lista3[1]='u'
>>> lista3
['Z', 'u', 't', 'D', 'e', 'f', 'G', 'H', 'I', 'J', 'Y']
>>> lista
['Z', 'B', 't', 'D', 'e', 'f', 'G', 'H', 'I', 'J', 'Y']
```

Operatori za liste

Operator	Opis
`+`	Nadovezivanje
`*`	Uvišestručenje
`in`	Element je sadržan u listi
`not in`	Element nije sadržan u listi

Ugrađene funkcije za liste (povezati s znanjem o stringovima ☺)

Funkcija	Opis
`len(lista)`	vraća duljinu liste
`min(lista)`	vraća najmanju vrijednost elementa liste
`max(lista)`	vraća najveću vrijednost elementa liste

Primjer: najviši student (1)

- Napišite program koji će tražiti unos broja studenata u grupi. Zatim će za svakog studenta tražiti unos njegove visine. Unesene visine pohranite u listu. Program treba odrediti najveću visinu studenta u grupi.

```
>>>
Unesi broj studenata u grupi: 5
Unesite visinu 1. studenta: 11
Unesite visinu 2. studenta: 13
Unesite visinu 3. studenta: 15
Unesite visinu 4. studenta: 14
Unesite visinu 5. studenta: 16
Lista svih visina studenata: [11, 13, 15, 14, 16]
Najvisi student ima visinu 16 (ugradena funkcija).
Najvisi student ima visinu 16 (vlastita funkcija).
```

Primjer: najviši student (2)

```
sve_visine=[]
N=5
najvisi=[-1]
for i in range(5):
    element = int(input('Unesite visinu {0} elementa liste: '.format(i+1)))
    #bez int-a shvaća unos kao string a ne int broj
    sve_visine = sve_visine + [element]
    if najvisi<[element]:
        najvisi=[element]
print('Najveća visina je ', max(sve_visine))
print('Najveća visina je ', najvisi)
```

Metode za liste (1)

Metoda	Upotreba	Opis
`append()`	`lista.append(element)`	dodaje `element` na kraj liste `lista`
`extend()`	`lista.extend(lista2)`	dodaje `lista2` na kraj liste `lista`
`insert()`	`lista.insert(indeks, element)`	dodaje `element` prije `indeks` elementa liste `lista`

```
>>> lista = list('ABCDEFGHIJK')
>>> lista
['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K']
>>> lista.append('L')
>>> lista
['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L']
>>> lista.extend(['M', 'N'])
>>> lista
['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N']
>>> lista2 = ['O', 'P']
>>> lista.extend(lista2)
>>> lista
['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P']
```

Metode za liste (2)

Metoda	Upotreba	Opis
`remove()`	`lista.remove(element)`	izbacuje `element` iz liste `lista` (ako ih ima više – prvog po redu)
`pop()`	`lista.pop(indeks)`	izbacuje element sa indeksom `indeks` iz liste `lista` i vraća ga ako `indeks` nije specificiran – izbacuje i vraća zadnji element liste

```
>>> lista.insert(2, 'b')
>>> lista
['A', 'B', 'b', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P']
>>> lista.remove('b')
>>> lista
['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P']
>>> visak = lista.pop(3)
>>> visak
'D'
>>> lista
['A', 'B', 'C', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P']
>>> visak = lista.pop()
>>> visak
'P'
>>> lista
['A', 'B', 'C', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O']
```

Metode za liste (3)

Metoda	Upotreba	Opis
`reverse()`	`lista.reverse()`	okreće redoslijed elemenata liste `lista`
`sort()`	`lista.sort()`	sortira listu `lista` prema rastućim vrijednostima elemenata

```
>>> lista = [4, 2, 6, 8, 3, 5, 7]
>>> lista
[4, 2, 6, 8, 3, 5, 7]
>>> lista.reverse()
>>> lista
[7, 5, 3, 8, 6, 2, 4]
>>> lista.sort()
>>> lista
[2, 3, 4, 5, 6, 7, 8]
```

Primjer: liste

- Super pametni mobitel prepoznaje ljudski govor, te može nazvati telefonski broj koji mu se izgovori znamenku po znamenku. No, mobitel je upao u lokvu vode i sada prepoznaje samo svaku drugu znamenku. Za unesenih 6 znamenaka koje ste rekli mobitelu, ispišite ono što je mobitel "čuo".

```
>>>
1: 4
2: 7
3: 2
4: 4
5: 7
6: 6
[7, 4, 6]
```

```
ucitani_broj = []
for i in range(6):
    znamenka = input('{0}: '.format(i+1))
    ucitani_broj.append(znamenka)

ucitani_broj.pop(4)
ucitani_broj.pop(2)
ucitani_broj.pop(0)

print(ucitani_broj)
```

```
ucitani_broj = []
for i in range(6):
    znamenka = input('{0}: '.format(i+1))
    ucitani_broj.append(znamenka)

for i in range(5,0,-2):
    ucitani_broj.pop(i-1)

print(ucitani_broj)
```