

PROG1: Prednáška 8

Retazce (strings)

Domaca uloha

1. Precitajte si kapitolu 8 v knihe.
2. Vyrieste ulohy z Cvicenia 8.

Retazce

Retazec je postupnost znakov.

Příklad:

```
>>> ret='semester'
```

Pristupovanie k znakom v retazci

Priklad:

```
>>> ret='semester'  
>>> ret[1]  
'e'
```

Pozor! Znaky v retazci su indexovane od **nuly!**
Ak chceme ziskat prvý znak retazca *ret*, musime zadat *ret[0]*. (*ret[1]* vrati druhy znak retazca)

Dlžka retazca

Priklad:

```
>>> ret='semester'  
>>> len(ret)  
8
```

Prikaz `len(ret)` vrati dlžku retazca `ret`.

Prechadzanie cez retazec

Prvy sposob: prechadzanie cez znaky retazca

Priklad:

```
ret='semester'  
for znak in ret:  
    print(znak)
```

Prechadzanie cez retazec

Druhy sposob: prechadzanie cez indexy retazca

Priklad:

```
ret='semester'  
for i in range(len(ret)):  
    print(ret[i])
```

V knihe je prechadzanie cez indexy retazce spravene pomocou while cyklu. Pozrite si to!

Prechadzanie cez retazec

Prvy sposob je jednoduchsi, niekedy je ale prirodzene pouzit druhy sposob.

Uloha:

Definujte funkcia, ktorej argumentom bude retazec a ktora vrati poziciu prveho vyskytu pismena a v retazci. Ak sa pismeno a v retazci nenachadza, funkcia vrati hodnotu -1.

Prechadzanie cez retazec

Prvy sposob je jednoduchsi, niekedy je ale prirodzene pouzit druhy sposob.

Uloha:

Definujte funkcia, ktorej argumentom bude retazec a ktora vrati poziciu prveho vyskytu pismena a v retazci. Ak sa pismeno a v retazci nenachadza, funkcia vrati hodnotu -1.

```
def find_a(ret):  
    for i in range(len(ret)):  
        if ret[i]=='a':  
            return i  
    return -1
```

Vyseky z retazca

```
>>> ret='Monty Python'
```

```
>>> ret[0:5]
```

```
'Monty'
```

```
>>> ret[6:12]
```

```
'Python'
```

Retazce su nemenne

```
>>> ret='Monty Pyhton'
```

```
>>> ret[1]='u'
```

```
Traceback (most recent call last):
```

```
  File "<pyshell#5>", line 1, in <module>
```

```
    ret[1]='u'
```

```
TypeError: 'str' object does not support item assignment
```

Metody pre retazce

Metody pre retazce = zabudovane “funkcie” v Pythone, ktore funguju iba pre retazce

Pozor! Syntax pri pouzivani metod je ina ako pri obycajnych funkciach!

Priklad: metoda upper

```
>>> ret='semester'  
>>> ret2=ret.upper()  
>>> ret2  
'SEMESTER'
```

Operator in

Pre dva retazce vrati operator in hodnotu True, ak sa prvý retazec nachádza v druhom.

```
>>> ret='Monty Python'  
>>> 'y' in ret  
True  
>>> 'a' in ret  
False  
>>> 'Monty' in ret  
True  
>>> 'Mnt' in ret  
False
```

ASCII tabulka

- ASCII = American Standard Code for Information Interchange
- ASCII tabulka prideluje kazdemu znaku ciselny kod

97	a
98	b
99	c
100	d

Funkcie ord a chr

- Funkcia ord berie ako argument znak a vrati ciselny kod znaku v ASCII tabulke

```
>>> ord('a')  
97
```

- Funkcia chr berie ako argument cislo a vrati znak, ktoremu v ASCII tabulke koresponduje dane cislo

```
>>> chr(97)  
'a'
```

Porovnavanie retazcov

Operatory

<

>

<=

>=

porovnaju retazce vzhľadom na kody znakov v ASCII tabulke.

```
>>> 'abeceda' < 'bonboniera'
```

```
True
```

```
>>> 'sanitka' < 'semester'
```

```
True
```