

REŤAZCE A FORMÁTOVANIE REŤAZCOV

a) Reťazce

Reťazec je nemenná postupnosť znakov Unicode údajového typu str (skratka slova string). Hodnoty typu str sú uzavreté v úvodzovkách alebo apostrofoch (je jedno, ktoré ohraničenie použijeme; na oboch stranách reťazca musí byť rovnaké). Prázdny reťazec je reťazec, ktorý nemá medzi ohraničením ani jeden znak. K jednotlivým znakom v reťazci sa dá dostať pomocou výrazu *reťazec[index]* , pričom prvý znak v reťazci má index 0.

Zmiešaná ukážka:

```
--
cele_cislo=123
print(cele_cislo)

real_cislo=12.345
print(real_cislo)

--
znak="a"
print(znak)

print(str(real_cislo))

retazec_znakov="retazec"

spajanie=znak + " toto je " + retazec_znakov + " znakov."
print(spajanie)
```

(„+“ v ukážke znamená spojiť dva reťazce)

Dĺžka reťazca je počet jeho znakov. Dĺžku reťazca, čo je prirodzené číslo, vracia funkcia *len(reťazec)* (názov funkcie zo slova length). Dĺžka prázdneho reťazca je nula.

```
dlzka=len(retazec_znakov)
print(dlzka)
```

Ukážka indexovania v reťazci s hodnotou Python

s="Python"

s[0]	s[1]	s[2]	s[3]	s[4]	s[5]
P	y	t	h	o	n
s[-6]	s[-5]	s[-4]	s[-3]	s[-2]	s[-1]

Možné a užitočné sú aj záporné indexy, index s hodnotou -1 vždy ukazuje na posledný znak v reťazci. Uvedomte si aj, že funkcia *len()* vracia celé číslo o 1 väčšie ako je index posledného znaku v reťazci, resp. *s[len(s)-1] == s[-1]*.

V rámci konverzie (prevodu) možno iný údajový typ previesť na reťazec funkciou *str()*. Konverziu musíme použiť aj pri vytváraní výstupného reťazca

```
a=987654
b=str(a)
print(a)

x=retazec_znakov
print(x)
```

x[0] vráti jednoznakový reťazec *r*; *x[1]* vráti *e*; " *x[4]* vráti *z* .

```
print(x[0],x[1],x[4])
```

Ukážky s cyklom while

```
y=0
while y<dlzka:
    print(x[y])
    y=y+1
```

```
y=0
while y<dlzka:
    print(x[y],end="+")
    y=y+1
```

b) Formátovanie

```
print("{}{}{}".format(x[0],x[1],x[4]))
print("{}{}{}{}".format(x[0],x[1],x[4],"anec"))
```

Funkcia **str.format()**

- vracia nový reťazec v zadanom tvare
- má tvar: *reťazec.format(parametre)*

kde *reťazec* je text v úvodzovkách (alebo apostrofoch) obsahujúci aj formátovacie polia uzavreté v zložených zátvorkách {}. Na miesta formátovacích polí sa do nového reťazca vložia v predpísanom formáte hodnoty parametrov. Názvy formátovacích polí, v najjednoduchších zápisoch, sú 0, 1, 2,...

Funkcia format sa najviac využíva v príkaze print, ale môže sa použiť kdekoľvek, kde potrebujeme upraviť formát reťazca.

Napríklad:

zápis "Súčet = {0}".format(sucet) pre hodnotu 123 premennej sucet vráti reťazec Súčet = 123. Ukážka:

```
>>> sucet = 123
>>> print("Súčet = {0}".format(sucet))
Súčet = 123
```

Zápis "Súčet prvých {0} prirodzených čísel je {1}".format(pocet, sucet) pre pocet = 100 a sucet = 5050 vráti reťazec Súčet prvých 100 prirodzených čísel je 5050.

```
>>> pocet = 100
>>> sucet = 5050
>>> print("Súčet prvých {0} prirodzených čísel je {1}".format(pocet, sucet))
Súčet prvých 100 prirodzených čísel je 5050
```

Ako vidieť z ukážok, v reťazci sa uvádza v zložených zátvorkách¹ poradové číslo parametra, ktorého hodnota sa má na danom mieste v reťazci zobrazíť (číslovanie začína nulou!). V mieste parametrov sa v požadovanom poradí uvedú príslušné parametre (najčastejšie premenné, hodnoty ktorých chceme zobrazíť).

Ak sa formátujú štruktúrované dáta, používa sa zápis, napr. pre n-ticu t = (12.5, 18.6, 16.3),

"Teplota ráno: {0}, na obed: {1}, večer: {2}".format(t[0], t[1], t[2])

```
>>> t = (12.5, 18.6, 16.3)
>>> print("Teplota ráno: {0}, na obed: {1}, večer: {2}".format(t[0], t[1], t[2]))
Teplota ráno: 12.5, na obed: 18.6, večer: 16.3
```

alebo zápis "Teplota ráno: {0}, na obed: {1}, večer: {2}".format(*t)

```
>>> print("Teplota ráno: {0}, na obed: {1}, večer: {2}".format(*t))
Teplota ráno: 12.5, na obed: 18.6, večer: 16.3
```

¹ Zloženú zátvorku { alebo } môžeme vložiť v slovenskej klávesnici - stlačené pravé Alt+B alebo pravé Alt+N.

Niečo navyše – pre rozšírenie rozhl'adu

Ak chceme formátovať parameter (predpísať počet znakov jeho hodnoty, zarovnať vľavo, na stred alebo vpravo v poli, určiť počet desatinných miest u reálneho čísla a pod.), za poradovým číslom parametra (názvom formátovacieho poľa) napíšeme **dvojbodku** a špecifikáciu formátu. V špecifikácii (určení) formátu môžeme použiť napríklad špecifikátory:

- pre zarovnanie: < vľavo, > vpravo, ^ na stred (^ - ľavé Alt+94)
- šírka: minimálna šírka poľa (prirodzené číslo)
- .presnosť: pre reálne čísla udáva počet desatinných miest (musí sa uviesť aj typ, t.j. f); pre reťazce udáva maximálnu šírku reťazca (zvyšok reťazca odreže)
- typ: napr. f pre reálne číslo

Ukážka pre reálne číslo:

```
>>> priemer = (t[0] + t[1] + 2*t[2])/4
>>> print("Priemerná teplota dňa bola ", priemer)
Priemerná teplota dňa bola 15.925
>>> print("Priemerná teplota dňa bola {0}".format(priemer))
Priemerná teplota dňa bola 15.925
>>> print("Priemerná teplota dňa bola {0:.2f}".format(priemer))
Priemerná teplota dňa bola 15.93
>>> print("Priemerná teplota dňa bola {0:.1f}".format(priemer))
Priemerná teplota dňa bola 15.9
```

Ukážka použitia rôznych špecifikátorov:

```
>>> meno1 = "Peter"
>>> meno2 = "Magdaléna"
>>> vek1 = 17
>>> vek2 = 9
>>> print("{0:10} {1:2}".format(meno1, vek1))
Peter      17
>>> print("{0:10} {1:2}".format(meno2, vek2))
Magdaléna  9
>>> print("{0:>10} {1:2}".format(meno1, vek1))
      Peter 17
```

Po zadaní presnosti reťazca meno2 na 5 znakov:

```
>>> print("{0:10.5} {1:2}".format(meno2, vek2))
Magda      9
```

Použitie parametra end príkazu print:

```
print("{0:10}{1:2}".format(meno1, vek1), end="; ")
print("{0:10}{1:2}".format(meno2, vek2))
Peter      17; Magdaléna  9
```