

Formátovanie reťazcov

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
```

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

Podrobnejšie pozri nižšie odsek **Všetky možnosti špecifikácie formátu**.

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
```

¹ Zloženú zátvorku `{}` alebo `}` môžeme vložiť v slovenskej klávesnici - stlačené pravé `Alt+B` alebo pravé `Alt+N`. Pozri <http://www.gpnr.sk/obsah/predmety/subory/informatika/klavesnica.pdf>

```
>>> print("Priemerná teplota dňa bola {}".format(priemer))
Priemerná teplota dňa bola 15.925
>>> print("Priemerná teplota dňa bola {:.2f}".format(priemer))
Priemerná teplota dňa bola 15.93
>>> print("Priemerná teplota dňa bola {:.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
```

Poznámky:

Názvy polí ({názov_poľa}) môžeme kombinovať, ako to vidieť v ukážke:

```
>>> pocet = 100
>>> sucet = pocet/2*(1+pocet)
>>> print("Súčet {1} dostaneme po sčítaní prvých {0} prirodzených čísel, t.j. od 1
po {0}".format(pocet, sucet))
Súčet 5050.0 dostaneme po sčítaní prvých 100 prirodzených čísel, t.j. od 1 po 100
```

Pri použití celočíselného delenia, t.j. `sucet = pocet//2*(1+pocet)`, je `sucet` typu `int`, čo sa prejaví aj v tvare výpisu reťazca:

```
Súčet 5050 dostaneme po sčítaní prvých 100 prirodzených čísel, t.j. od 1 po 100
```

Názvy polí nemusíme použiť, ak vyhovuje poradie parametrov:

```
>>> slovo1 = "Python"
>>> slovo2 = "veľmi dobrý"
>>> slovo3 = "jazyk"
>>> text = "{} je {} programovací {}".format(slovo1, slovo2, slovo3)
>>> print(text)
Python je veľmi dobrý programovací jazyk
```

Ako názov poľa môžeme použiť aj názov lokálnej premennej, keďže Python si vytvára slovník *názov lokálnej premennej*: *hodnota*; a zápisom parametra `**locals()` slovník „rozbalíme“ vo funkcii `format`:

```
>>> print("{slovo1} je dobrý programovací {slovo3}".format(**locals()))
Python je dobrý programovací jazyk
```

Vhodnejšie označenie premenných:

```
>>> jazyk = "Python"
>>> hodnotenie = "dobrý"
>>> print("{jazyk} je {hodnotenie} programovací jazyk".format(**locals()))
Python je dobrý programovací jazyk
```

Všetky možnosti špecifikácie formátu: [[vyplniť]zarovnanie] [znamienko][#][0][šírka][,][.presnosť] [typ]

kde [] znamená nepovinná časť a možnosti jednotlivých častí sú:

vyplniť: ľubovoľný znak

zarovnanie: "<" | ">" | "=" | "^" („|“ sa číta „alebo“)

znamienko: "+" | "-" | ""

šírka: celé číslo

presnosť: celé číslo

typ: "b" | "c" | "d" | "e" | "E" | "f" | "F" | "g" | "G" | "n" | "o" | "s" | "x" | "X" | "%"

Príklad použitia vypĺňacieho znaku x:

```
>>> pi = 3.14159
>>> print("PI = {0:x^10.2f}".format(pi))
PI = xxx3.14xxx
```