9. Programování - řetězce

Řetězce - zápis, indexování

Řetězec je seřazená neměnná kolekce znaků.

SYNTAX

- uzavřeny v apostrofech **' '**, v uvozovkách **" “**
- trojité uvozovky/apostrofy uzavírají víceřádkové řetězce - častěji používané jako víceřádkové komentáře
- jsou neměnné - pro změnu se musí vytvořit nový řetězec

VYUŽITÍ

Využívají se pro uložení textu, hesel, konstant.

INDEXOVÁNÍ

K jednotlivým částem řetezce lze přistoupit pomocí [ ]. Kladné hodnoty číslují znaky zleva - začíná 0, záporné od konce.

print(seznam[0]) výpis znaku na indexu 0 (první)

print(seznam[-1]) výpis posledního znaku

operace s řetězci

SČÍTÁNÍ / SPOJOVÁNÍ - *ret=ret + “ : ” + ret2*

NÁSOBENÍ CELÝM ČÍSLEM - *ret*=10*

PROCHÁZENÍ CYKLEM FOR

```
for znak in ret:
    print(znak)
```

ROZBALENÍ

ret=”abc”

*print(*ret) # stejné jako print(“a”, “b”, “c”)* *>a b c*

ZJIŠTĚNÍ PŘÍTOMNOSTI (in) - “a” in ret

ZJIŠTĚNÍ DÉLKY - funkce len(s)

PŘEVODNÍ FUNKCE - str()

převod libovolného typu na řetězec

METODY

- .**split**(s, n) - vrátí seznam podřetězců oddělených řetězcem s (není-li zadán, rozděluje mezerou), nepovinné n určuje maximální počet oddělení (od počátku)
- .**replace**(s1, s2, n) - vrátí řetězec s nahrazenými výskyty řetězce s1 řetězcem s2, nepovinné n určuje maximální počet náhrad (od počátku)
- .**count**(s, od, do) - vrátí počet výskytů podřetězce; je možné nepovinně zadat start a end jako pozice začátku a konce hledání
- .**index**(s) - vrátí index prvního znaku, na kterém podřetězec s začíná; pokud ho nenajde, nastane výjimka
- .**find**(s) - vrátí index prvního znaku, na kterém podřetězec s začíná; pokud ho nenajde, vrací hodnotu -1
- .**upper**(), .**lower**() - vrátí řetězec převedený na velké/malé znaky
- .**capitalize**(), .swapcase() - první vélké, přehození

- .**join**(seznam) - vrátí řetězec prvků seznamu spojených původním řetězcem

Knihovna **string** obsahuje mnoho užitečných proměnných a funkcí, např.: string.ascii_letters, string.digits…

escape znaky

Existují tzv. escape znaky, které slouží hlavně ke zkracování kódu a zpřehledňování tisku. Jsou to speciální sekvence začínající lomítkem:

\' - apostrof

\" - uvozovka

\\ - lomítko

\n - zalomení na nový řádek

\t - tabulátor

RAW řetězce

Někdy se stává, že je potřebujeme zapsat řetězec, který obsahuje hodně zpětných lomítek. Nebo chceme zapsat speciální znaky, které ve skutečnosti nebudou speciální (např. regulární výraz). Pro tyto případy existuje předpona r (jako RAW), která zruší všechny speciální znaky.

s = r"ahoj\nnazdar\n\tcau"

funkce format() nebo fstring

- Vrátí řetězec v němž jsou speciální konstrukce ve složených závorkách {...} nahrazeny hodnotami z n-tice
- Do složených závorek lze zapsat volitelně pořadí hodnoty v n-tici
- Obecný tvar je ret="{index:specifikátor}".format(n-tice)
- Pro vložení znaku { nebo } je třeba jej zdvojit

*“jedna: {2}, dva: {0}, tri: {1}; znak {{“.**format**(“ahoj”, 77, 3.1415)*

Dále lze přidat za dvojtečku formátovací řetězec ve tvaru:

<výplň><zarovnání><znaménko><šířka><,_>.<přesnost><typ>

<výplň> je libovolný znak.

<zarovnání> může být **>**, **<** , **^** a **=** (pro zarovnávání čísel).

<znaménko> může být + (plus i mínus), - (jen mínus) nebo mezera (mezera nebo mínus)

<šířka> a **<přesnost>** jsou celá čísla.

<,_> určuje oddělovač řádů (, nebo _).

<typ>	význam
s	řetězec
d	číslo v desítkové soustavě
b	číslo ve dvojkové soustavě
o	číslo v osmičkové soustavě
x, X	číslo v šestnáctkové soustavě
f, F	reálné číslo

fstring

- další zjednodušení formátovaní
- vytvoříme tak, že před apostrof nebo uvozovku napíšeme f

- podobný fci format, akorát místo indexů můžeme do složených závorek psát rovnou názvy proměnných nebo hodnoty

***f**"Dnes je {d:02}. {m:02}. {r}"*

operátor ":" (slice)

Se všemi seznamovými datovými typy můžeme provádět **vyřezy** (slice). Výřez je nový objekt.

[:n] - vrátí prvních n znaků

[n:] - vrátí podřetězec od pozice n do konce

[m:n] - vrátí podřetězec od pozice m do n (n se nepočítá)

datum a čas

Pro práci s datem a časem můžeme použít modul **datetime** (nebo time).

datetime.**now**() - vrací aktuální datum a čas jako DateTime objekt ve formátu rrrr-mm-dd hh:mm:ss.ms

Jednotlivé složky datumu získáme z jeho polí hour, minute atd. Formátování umožňuje metoda **strftime**.

```
from datetime import *
dnes = datetime.now()
print(f"Aktualni rok: {dnes.year}")
print(f"Aktualni mesic: {dnes.month}")
print(f"Aktualni den: {dnes.day}")
print(f"Aktualni hodina: {dnes.hour}")
print(f"Aktualni minuta: {dnes.minute}")
print(f"Aktualni vterina: {dnes.second}")
print(f"Aktualni mikrosekunda: {dnes.microsecond}")
```

Datové objekty lze **odečítat**:

```
cas1 = datetime.now()
x=1
for i in range(1000000):
  x+=1
cas2 = datetime.now()
cas = cas2 - cas1
```