



IB111 – cvičení 5

ŘETĚZCE, SEZNAMY

Miroslav Kadlec



Obsah

- . Seznam, pole
 - . Indexování
 - . Slicing
 - . Funkce
- . Řetězec znaků
 - . Znak
 - . Práce s řetězcí

Příklady

Seznam, pole

- Obecně je pole základní struktura pro uchování více hodnot stejného typu
- V Pythonu je pro tyto účely **seznam** (list) – velmi obecný
 - Lze ukládat i **různé datové typy**
 - ExampleList = [1, "únor", 3.14]
 - Má předprogramované určité funkce
 - append(),
 - Definujeme do **hranatých** závorek, oddělujeme čárkami
 - temperatures = [20, 22, 28, 19] # s hodnotami
 - results = [] # prázdný
-

Indexování v seznamu

- Přístup k jednotlivým prvkům
 - Čtení, změna hodnoty
 - **Každý prvek** určen celým číslem (= **indexem**) vyjadřujícím jeho pořadí
 - Zadáváme za jméno seznamu do hranatých závorek
 - První prvek má **index 0**
 - př.: `print(seznam[0])`
 - Lze indexovat i **od konce** zápornými čísly
 - [-1] vrátí poslední prvek, [-2] předposlední
 - Co když zadáme větší číslo, než (počet prvků - 1)?

Funkce pro práci se seznamy

- Kopírování
 - Přiřazením "newList = oldList" se nevytvoří nový seznam, jen máme dvě proměnné, **skrz které** můžeme k seznamu **přístupovat**
 - "Mělká" kopie a = list(x) nebo a = x[:]
 - V případě seznamu seznamů se **nevytvoří** nové "vnitřní,, seznamy
 - Ekvivalentní konstrukce a = copy(list)
 - Potřeba importovat z modulu "copy"
 - "Hluboká" kopie a = deepcopy(x)
 - Vytvoří se **kompletně nová struktura**
 - Třeba **importovat** z modulu "copy"

Funkce pro práci se seznamy

- Funkce pro agregaci:
 - len(x)
 - sum(x)
 - min(x)
 - max(x)
 - sorted(x)
 - y in x
- Jak se chovají s nečíselnými typy?
 - sum()
 - Seznam řetězců
 - Seznam seznamů
 - max/min()
 - Seznam řetězců
 - Seznam seznamů

Funkce pro práci se seznamy

- Pokročilé indexování – **slicing**
 - Umožňuje vybírat podčásti seznamu
 - S pomocí dvojtečky specifikujeme, které indexy se vyberou
 - seznam[x:y:z] vybere indexy
 - Od **x** včetně
 - Do **y** nevčetně
 - S krokem **z**
 - Lze hodnotu vynechat – defaultní hodnoty:
 - Od začátku (0)
 - Do konce
 - S krokem 1 (při vynechání kroku není třeba psát 2. dvojtečku)

Řetězec (string)

- Posloupnost znaků, se kterými pracujeme jako s celkem – různé funkce, operace
 - Setkali jsme se s
 - len()
 - konkaténace (skládání) řetězců operátorem +
 - opakování řetězců operátorem *
 - escape sekvence

Znak (char)

- Celé číslo reprezentováno jako **znak**
 - Písmeno, číslice, interpunkce, ...
 - Zapisujeme jako jednoznakový string např.: "a"
- V Pythonu – kódování UNICODE (<http://www.tamasoft.co.jp/en/general-info/unicode-decimal.html>)
 - Každý používaný znak má jednoznačně přiřazenou hodnotu
 - **Některé** logické návaznosti jsou zachovány

- **chr()**

- Číslo -> znak

- **ord()**

- Znak -> číslo

UNICODE dec. 0-499

| | | | | | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|
| 000 | 001 | fi | 002 | fl | 003 | □ | 004 | □ | 005 | □ | 006 | □ | 007 | □ | 008 | □ | 009 | □ | |
| 010 | □ | 011 | □ | 012 | □ | 013 | □ | 014 | □ | 015 | □ | 016 | □ | 017 | □ | 018 | □ | 019 | □ |
| 020 | □ | 021 | □ | 022 | □ | 023 | □ | 024 | □ | 025 | □ | 026 | □ | 027 | □ | 028 | □ | 029 | □ |
| 030 | □ | 031 | □ | 032 | | 033 | ! | 034 | " | 035 | # | 036 | \$ | 037 | % | 038 | & | 039 | ' |
| 040 | (| 041 |) | 042 | * | 043 | + | 044 | , | 045 | - | 046 | . | 047 | / | 048 | 0 | 049 | 1 |
| 050 | 2 | 051 | 3 | 052 | 4 | 053 | 5 | 054 | 6 | 055 | 7 | 056 | 8 | 057 | 9 | 058 | : | 059 | ; |
| 060 | < | 061 | = | 062 | > | 063 | ? | 064 | @ | 065 | A | 066 | B | 067 | C | 068 | D | 069 | E |
| 070 | F | 071 | G | 072 | H | 073 | I | 074 | J | 075 | K | 076 | L | 077 | M | 078 | N | 079 | O |
| 080 | P | 081 | Q | 082 | R | 083 | S | 084 | T | 085 | U | 086 | V | 087 | W | 088 | X | 089 | Y |
| 090 | Z | 091 | [| 092 | \ | 093 |] | 094 | ^ | 095 | _ | 096 | ` | 097 | a | 098 | b | 099 | c |
| 100 | d | 101 | e | 102 | f | 103 | g | 104 | h | 105 | i | 106 | j | 107 | k | 108 | l | 109 | m |
| 110 | n | 111 | o | 112 | p | 113 | q | 114 | r | 115 | s | 116 | t | 117 | u | 118 | v | 119 | w |

Další práce s řetězcí

- Indexování
 - Funguje jako u seznamů, např.: "ahoj"[2]
- Slicing
 - Opět funguje jako u seznamů (podřetězce)
- Iterace - procházení jednotlivých znaků cyklem
 - for znak in text:
- Změna velikosti písmen
 - "AhOj SvEtE".upper()
 - "AhOj SvEtE".lower()

Příklady - https://www.fi.muni.cz/IB111/sbirka/05-retezce_a_seznamy.html

- **Rozcvička**: prokládání, cenzura
- **Výškový profil**
 - height_profile(), elevation() - ve slidech
 - **generování**:
 - generateProfile(size, startingHeight, maxDiff = 2)
 - generateProfileAdv(size, maxDiff = 2)
- **Vigenerova šifra**
 - Uvažujme pouze malá písmena, bez mezer
 - Klíč se opakuje podél plaintextu
 - Ke každému znaku plaintextu se “přičte” znak klíče, který na něj vyšel
 - plaintext lze procházet for-cyklem, klíč indexovat proměnnou -> inkrementovat a hlídat pomocí %