

IB113 Úvod do programování a algoritmizace

Cvičení 4

Jaromír Plhák



Řetězec (string)

- Sekvence znaků (čísla, písmena, symboly) uzavřených:
- a) V apostrofech: `my_string = 'Brian'`
- b) V úvozovkách: `my_string = "Brian"`
- Pokud má řetězec obsahovat apostrof?

```
>>> print('I'm Brian !')
```

```
SyntaxError : invalid syntax
```

```
>>> print('I\'m Brian !') # Escaping
```

```
I'm Brian !
```

```
>>> print("I'm Brian !")
```

```
I'm Brian !
```

Délka řetězce

- Funkce len():

```
>>> example = "Python"
```

```
>>> len(example)
```

```
6
```

- Prázdný řetězec = prázdné apostrofy / uvozovky:

```
>>> empty = ""
```

```
>>> type(empty)
```

```
<class 'str '> # It really is a string
```

```
>>> len(empty)
```

```
0
```

Jednotlivé znaky řetězce

- Jednotlivé znaky řetězce jsou indexované celými čísli
 - Indexuje se od 0 po `len(string) - 1`!

```
>>> example = "Python"
```

Znak	P	y	t	h	o	n
Index	0	1	2	3	4	5

- Syntaxe zpřístupnění znaků: `string[index]`

```
>>> character = example[1]
```

```
>>> character
```

```
'y'
```

- Řetězce jsou neměnitelné (*immutable*): nefunguje např. `example[1] = 'i'!`

Iterace nad řetězcem

- Cyklus for

```
string = "spam!"
```

```
for character in string:
```

```
    print (character)
```

- Proměnná character postupně nabývá hodnoty všech znaků řetězce
- Druhá možnost:

```
string = "spam!"
```

```
for i in range(len(string)):
```

```
    print(string[i])
```

Úkoly – část 1

- 1. Vytvořte funkci `n_factorials_print(n)`, která **vypíše** všechny faktoriály od $0!$ do $(n - 1)!$ oddělené mezerou.
 - Například `n_factorials_print(8)` vypíše "1 1 2 6 24 120 720 5040"
- 2. Vytvořte funkci `n_factorials_return(n)`, která **vrátí** všechny faktoriály od $0!$ do $(n - 1)!$ oddělené mezerou jako **řetězec**.
 - Nezapomeňte na přetypování pomocí funkce `str()`
- 3. Vytvořte funkci `string_with_spaces(string)`, která **vypíše** zadaný řetězec tak, že za každým písmenem bude mezera.
 - Například `string_with_spaces("Test")` vypíše "T E S T "

Zadání domácí úlohy

- V ISu v odevzdávárně
 - https://is.muni.cz/auth/el/1433/podzim2017/IB113/ode/72006396/72006443/hw01_zadani.py (skupina 1)
 - https://is.muni.cz/auth/el/1433/podzim2017/IB113/ode/72006400/72006457/hw01_zadani.py (skupina 2)
- Veškeré informace v souboru
- Odevzdejte pouze tento soubor do odevzdávárny
 - <https://is.muni.cz/auth/el/1433/podzim2017/IB113/ode/72006396/72006429/> (skupina 1)
 - <https://is.muni.cz/auth/el/1433/podzim2017/IB113/ode/72006400/72006450/> (skupina 2)
- Do 13. 10. 2017, 23:59
 - Výjimečně do pátku, kvůli využití řetězců

Funkce pro generování náhodných čísel v Pythonu

- Modul random
- `from random import randrange, randint, random`
- `randrange(a, b)` – int n takový, že $a \leq n < b$
- `randint(a, b)` – int n takový, že $a \leq n \leq b$
- `random()` – float v rozsahu $[0, 1)$
- <https://docs.python.org/3.5/library/random.html>

Úkoly – část 2

- 4.1.2. Dokud padá sudé číslo
 - Napište funkci, která bude provádět házení obyčejnou šestistěnnou kostkou tak dlouho dokud nepadne liché číslo. Poté funkce vrátí celkový součet všech hozených hodů.
- 4.1.3. Statistiky
 - Napište funkci, která vygeneruje a vypíše count náhodných čísel v intervalu [lower, upper] a následně vypíše nejmenší, největší a průměrné číslo.
- 4.2.1. Opilec na cestě domů
 - Opilec je na půli cesty mezi domovem a hospodou, každý krok udělá náhodně jedním směrem. Napište funkci, která bude simulovat opilcův pohyb. Parametry budou vzdálenost mezi domovem a hospodou a počet kroků do opilcova usnutí (tj. maximální délka simulace). Simulace skončí buď tehdy, když opilec dojede domů nebo do hospody, případně po vyčerpání počtu kroků.