

lecture 3 - Exercises



Programming in geoinformatics
Autumn 2015

1.TUPLES

- máme definované 3 souřadnice a tři odpovídající názvy měst v rámci dvou proměnných typu tuple.
- `coords = ([52.23, 21.04], [54.41, 18.65], [53.14, 23.21])`
- `cities = ("Warszawa", "Gdansk", "Bialystok")`

- I) přidejte do `ntic` hodnoty `[53.4, 14.5]` a `"Szczecin"`
- II) zaokrouhlete koordináty na jedno desetinné místo
- III) doplňte do názvů měst text `"_railway_station"`

2. SETS 1

- máme pole `values1`, `values2` a `values3`:
 - `values1 = [1, 5, 17, 25, 35, 69, 12, 13, 17]`
 - `values2 = [2, 7, 17, 25, 36, 13, 15, 18]`
 - `values3 = [3, 5, 7, 36, 16]`
- definujte list `values`, který se skládá z hodnot `values1`, `values2` a `values3`, ale neobsahuje duplikáty
- použijte datový typ `set`

3. SETS 2

- vyřešte domácí úlohu 1 z předcházejícího týdne pomocí setů
 - hranice krajiny A a krajiny B jsou definovány neuspořádanými listy s hraničními body
 - `countryA_bounds = ["A", "W", "X", "C", "M", "K"]`
 - `countryB_bounds = ["W", "O", "L", "A", "K", "U", "S"]`
 - vytvořte nový list `countryAB_bounds`, do kterého uložíte hraniční body pro hypotetickou krajinu, která by vznikla spojením krajiny A a B
- použijte metody `difference/union/intersection`

4. DICTIONARIES 1

máme definovanou dictionary

```
countries_stats = {  
    'Nigeria' : {'GDP': 1109000, 'rank': 20, 'languages': ['English']},  
    'South Africa' : {'GDP': 725004, 'rank': 30, 'languages': ['Zulu', 'Xhosa', 'Afrikaans', 'English'] },  
    'Ethiopia' : { 'GDP': 132000, 'rank': 65, 'languages': ['Amharic']}  
}
```

I) najděte na wikipedii svůj oblíbený africký stát a přidejte ho do seznamu

II) spočtěte průměr GDP pro státy

III) vypište jazyky, kterými se v těchto státech mluví (bez duplikátů)

5. FUNCTIONS 1

máme koordináty vertexů řek v podobě

```
river_a = [[3,7], [3,9], [4,11], [6,12]]
```

```
river_b = [[12,4], [10,6], [6,7], [3,9], [2,4]]
```

- I) pomocí pythagorovi věty udělejte funkci “distance”, která vezme argumenty 2 bodů a spočte délku
- II) pomocí volání funkce “distance” udělejte funkci “river_length”, která veme argument hodnoty řeky a vrátí délku řeky
- III) funkce intersect() vezme argumenty 2 řek a vrátí koordináty, které jsou totožné pro obě řeky

HOMEWORK 1 - Population counters

- Uděláme počítač obyvatel pro obce A, B, C a D
- Obce a počet obyvatel máme uložené ve slovníku
 - `towns = { 'A': 150, 'B': 279, 'C': 869, 'D': 974 }`
- Vytvořte funkci `born`, která veme jako argument seznam obcí, kde se narodilo dítě
- Funkce `died` má obdobně zemřelé obyvatele odečítat
- Funkce berou argumenty listu nebo budou fungovat pomocí `*args` (výběr je na vás)
 - příklad: `born('A', 'B', 'A', 'A', 'C', 'D', 'C', 'D', 'C', 'D')` nebo `born(['A', 'B', 'A', 'A', 'C', 'D', 'C', 'D', 'C', 'D', 'C', 'D'])`

HOMEWORK 2 - Calendar application

- Máme týdenní kalendář uložený ve slovníku, kde klíče jsou jednotlivé dny a hodnoty jsou listy o délce 24, pro každou hodnotu jeden element
- Definice takového kalendáře vypadá (False hodnota značí prázdné místo):

```
- week_calendar = {  
    'mon': [False] * 24,  
    'tue': [False] * 24,  
    'wen': [False] * 24,  
    'thu': [False] * 24,  
    'fri': [False] * 24,  
}
```

- Na obsluhu napište následující funkce:
 - `check_free(day, hour)` -> vrátí True pokud je v dané době volno (volno = False)
 - `add_event(event, day, hour_from, hour_to)` -> přidá do daného rozmezí novou událost, pokud tam žádná není
 - `clear_hour(day, hour)` -> vymaže událost na daném místě