LU04.L12 - Sortieren grösserer Daten

Funktion: sort_countries_by_name()

```
def sort_countries_by_name(data):
    """Return a copy of the data, sorted by name."""
    # Sort countries by their names
    return sorted(data, key=lambda x: x['name'])
```

Diese Funktion nimmt eine Liste von Dictionaries (`data`) entgegen, in der jedes Dictionary Informationen zu einem Land enthält. Die Funktion gibt eine sortierte Kopie dieser Liste zurück, in der die Länder nach ihrem Namen sortiert sind.

1. `sorted()` ist eine eingebaute Python-Funktion, die eine sortierte Liste zurückgibt.
2. `key=lambda x: x['name']` definiert das Sortierkriterium (in diesem Fall den Namen des Landes).

Funktion: sort_countries_by_capital()

```
def sort_countries_by_capital(data):
    """Return a copy of the data, sorted by capital."""
    # Sort countries by their capitals
    return sorted(data, key=lambda x: x['capital'])
```

Ähnlich zur vorherigen Funktion, aber diesmal werden die Länder nach der Hauptstadt sortiert.

Funktion: sort_countries_by_population()

```
def sort_countries_by_population(data):
    """Return a copy of the data, sorted by population."""
    # Sort countries by their population
    return sorted(data, key=lambda x: x['population'], reverse=True)
```

Diese Funktion sortiert die Länder nach ihrer Bevölkerungszahl, wobei das am dichtesten besiedelte Land zuerst kommt (`reverse=True`).

Funktion: get_ten_most_spoken_languages()

```
def get_ten_most_spoken_languages(data):
    """
    Return the most spoken language by country.
    As Array with tuples like (name, population)
```

```
    Example: [('English', 91), ('French', 45), ...]

    Have a look at collections.Counter if you're stuck:
    https://docs.python.org/3/library/collections.html#collections.Counter
    """
    from collections import Counter

    # Collect all languages spoken in all countries
    all_languages = [language for country in data for language in
country['languages']]

    # Count the occurrence of each language
    language_counter = Counter(all_languages)

    # Sort and get the ten most spoken languages by location
    return language_counter.most_common(10)
```

Diese Funktion ermittelt die zehn am häufigsten gesprochenen Sprachen. Dazu wird zuerst die `Counter`-Klasse aus dem `collections`-Modul importiert. Anschließend werden alle gesprochenen Sprachen in einer einzigen Liste gesammelt und gezählt. Die zehn am häufigsten vorkommenden Sprachen werden dann als Liste von Tupeln zurückgegeben.

Erklärung der List Comprehension

Der Code:

```
all_languages = [language for country in data for language in
country['languages']]
```

ist eine List Comprehension, die eine Liste von Listen in eine einzelne Liste umwandelt. Es ist eine verkürzte Art und Weise, alle gesprochenen Sprachen aus allen Ländern in einer einzigen Liste zu sammeln. So funktioniert es:

1. `for country in data:` - Diese Schleife durchläuft jedes Dictionary (das ein Land repräsentiert) in der Liste `data`.
2. `for language in country['languages']:` - Diese innere Schleife durchläuft die Liste der Sprachen, die im aktuellen Land gesprochen werden.
3. `language` - Hier wird jede Sprache gesammelt, die während der inneren Schleife gefunden wird.

Das Ergebnis ist eine einzelne Liste, `all_languages`, die jeden Fall einer gesprochenen Sprache in allen Ländern enthält, einschließlich Wiederholungen, wenn eine Sprache in mehreren Ländern gesprochen wird.

Der gleiche Effekt könnte mit verschachtelten Schleifen erzielt werden:

```
all_languages = []
for country in data:
    for language in country['languages']:
```

```
            all_languages.append(language)
```

Erklärung zu "from collections import Counter"

`Counter` ist eine Unterklasse des `dict`-Datentyps in Python und wird aus dem Modul `collections` importiert. Es wird verwendet, um die Häufigkeit von Elementen in einer Sammlung (wie einer Liste) zu zählen.

Ein `Counter`-Objekt kann wie folgt verwendet werden:

```
from collections import Counter

beispiel_liste = ['a', 'b', 'c', 'a', 'b', 'b']
zaehler = Counter(beispiel_liste)

# zaehler wäre nun Counter({'b': 3, 'a': 2, 'c': 1})
```

In diesem Beispiel zählt `Counter` die Häufigkeit jedes Elements in `beispiel_liste`. Das Ergebnis ist ein Dictionary-ähnliches Objekt, das die Elemente als Schlüssel und ihre Häufigkeiten als Werte speichert.

In unserem Fall haben wir `Counter` benutzt, um die Häufigkeit jeder Sprache in der Liste `all_languages` zu zählen, und dann die zehn am häufigsten vorkommenden Sprachen ermittelt.

Funktion: get_ten_most_populated_countries()

```
def get_ten_most_populated_countries(data):
    """
    Return the ten most populated countries.
    As Array with tuples like (name, population)
    Example: [('China', 1377422166), ('India', 1295210000), ...]
    """
    # Get the ten most populated countries
    top_ten_populated_countries = sort_countries_by_population(data)[:10]

    # Extract just the names and populations for display
    return [(country['name'], country['population']) for country in
                                        top_ten_populated_countries]
```

Diese Funktion gibt die zehn am dichtesten besiedelten Länder zurück. Sie verwendet die bereits definierte Funktion `sort_countries_by_population()` und extrahiert die zehn Länder mit der höchsten Bevölkerungszahl. Das Ergebnis ist eine Liste von Tupeln, die den Namen des Landes und seine Bevölkerung enthalten.