Sprachverarbeitung: Übung

SoSe 25

Janis Pagel
Department for Digital Humanities, University of Cologne

8 April 2025

Submit your solution via Ilias, either as a Jupyter Notebook (`.ipynb`, you can export your Notebook in Jupyter by going to File > Download) or as a Python script (`.py`) if you are not working in Jupyter.

Exercise 1.

Given are three lists of planets and some of their moons:

```
l1 = ["Titan", "Prometheus", "Saturn", "Janus", "Enceladus"]
l2 = ["Io", "Europa", "Ganymede", "Jupyter"]
l3 = ["Neptune", "Nereid", "Triton", "Proteus"]
```

Write a Python scripts that creates two lists out of the given three lists, one for the planets and one for the moons, i.e. the output should be:

```
# The ordering of the items is not important
~> planets = ["Saturn", "Jupyter", "Neptune"]
# The ordering of the items is not important
~> moons = ["Titan", "Prometheus", "Janus", "Enceladus", "Io", "Europa", "Ganymede", "
                                            Nereid", "Triton", "Proteus"]
```

In your solution, you have to use **indexing**, **slicing** and **list concatenation**.

Exercise 2.

Given is a list

```
fruits = ["cranberry", "strawberry", "coconut", "pear", "peach"]
```

Implement a **for loop** that iterates over the items of that list and prints 'Berry: {item}' if the item ends in 'berry' (For example "Berry: cranberry"). If not the case, print 'Not a berry {item}' (For example "Not a berry: coconut"). Also use an **if statement** in your implementation.

Exercise 3.

Write a **function** that takes a dictionary (keys: string, values: int) as an input and creates the inverse dictionary, i.e. the values are now the keys and the keys are now the values.
Example:

```
# Input for the function
dictionary = {"red": 1, "blue": 2, "green": 3, "yellow": 4, "violet": 5}
# Desired output
~> {1: "red", 2: "blue", 3: "green", 4: "yellow", 5: "violet"}
```

Exercise 4.

Write a **function** that reverses a dictionary (keys: string, values: int) when the values are not unique by writing all matching key values into a list.
Example:

```
# Input for function
{"red": 1, "blue": 1, "green": 2, "yellow": 3, "violet": 3}
# Desired output
~> {1: ["red", "blue"], 2: ["green"], 3: ["yellow", "violet"]}
```