additional materials for SET 4

kacper.topolnicki@uj.edu.pl

Simple script to demonstrate the process of extracting floating point numbers from a string of characters. More information on regular expressions can be found:

- https://docs.python.org/3/library/re.html
- https://www.w3schools.com/python/python_regex.asp

Regular expressions

We begin by importing the "re" module [set_4.py line: 38] :

```
import re
```

This module contains methods that will allow us to work with regular expressions (RE). More information on RE can be found **here** or **here**.

In order to extract floating point numbers we will use the simple RE `\d+\.\d+`. You read this from left to right as: `\d+\.\d+` will match any substring of text that starts with a digit (`\d`) repeated one or more times (`+`), followed by a period (`\.`), and again followed by a digit (`\d`) repeated one or more times (`+`).

We will be testing this RE on [set_4.py line: 55] :

```
text = """
1 1.2
this is some text without numbers
this is line number 3
1.2 3.5716
2.1 3.1
Point (1a) from line 233.
"""
```

First we have to compile the expression [set_4.py line: 66] :

```
floatingPoint = re.compile(r"\d+\.\d+")
```

The result is compiled RE object. Notice the `r` before `r"\d+\.\d+"`. This notation means "raw string" and effects the way python handles backslashes.

Next we will use the compiled RE under `floatingPoint` to look through `text` and find all substrings that can be interpreted as floating point numbers [set_4.py line: 77] :

```
matches = floatingPoint.findall(text)

print(matches)
```

Finally we turn all the character strings into numbers [set_4.py line: 95] :

```
floats = map(lambda s : float(s) , matches)

print(floats)
```

Where we map the anonymous lambda function `lambda s : float(s)` over each element of the list. The result is a `map object` that needs to be turned into a list [set_4.py line: 101] :

```
floats = list(floats)

print(floats)
```