**PHILOSOPHY AND HISTORY OF SCIENCE WITH COMPUTATIONAL MEANS**

**PROF. DR. GERD GRAßHOFF**

# Regular Expressions

Imagine you have to search a string like 'world' in the first proposition of Wittgenstein's *Tractatus Logico-Philosophicus*. You can use the keyword 'in' as follows:

```
'world' in 'The world is all that is the case.'
```

If we run the cell, we will get 'true' or 'false' depending on whether the string is there.

Now imagine that you do not want a specific string, but to search for a pattern, like all the numbers that have format #.##, then you need a regular expression. Each character type has a pattern code. For example, digits have the pattern code \d (for any number). Thus the regular expression would be r'\d.\d{2}'.

First, we have to import re:

```
import re
```

Next, we define a text and a pattern. To get the first match, we use the re.search() function:

```
re.search(pattern, text)
```

To get all the matches, we use the re.findall() function. The 'r' before the pattern stands for 'regular expression.'

```
re.findall(r'\d.\d{2}', text)
```

To see how many patterns did Python find, use the len() function. Please check the Jupyter notebook for all the details.

## Regex Cheat Sheet

| Identifier | Legend | Example | Sample Match |
|---|---|---|---|
| \d | One digit from 0 to 9 | file_\d\d | file_25 |
| \w | Words; letter, digit or underscore | \w-\w\w\w | A-b_1 |
| \s | Whitespace character | a\sb\sc | a b c |
| \D | One character that is not a digit | \D\D\D | ABC |
| \W | One character that is not a word | \W\W\W\W | =+-) |

| Identifier | Legend | Example | Sample Match |
|---|---|---|---|
| \S | One character that is not a whitespace | \S\S\S\S | Xam1 |

| Quantifier | Legend | Example | Sample Match |
|---|---|---|---|
| + | One or more | Version \w-\w+ | Version A-b1_1 |
| {3} | Exactly three times | \D{3} | ABC |
| {2,4} | Two to four times | \d{2,4} | 156 |
| {3,} | Three or more times | \w{3,} | regex_tutorial |
| * | Zero or more times | A*BC** | AAACC |
| ? | Once or none | plurals? | plural |

| Character | Legend | Example | Sample Match |
|---|---|---|---|
| . | Any character except line break | a.c | abc |
| . | Any character except line break | .* | whatever, man. |
| **.** | A period (special character: needs to be escaped by a ) | a.c | a.c |
| \ | Escapes a special character | .*+\? $\^\/\ | .*+? $^/\ |
| \ | Escapes a special character | [{()}] | [{()}] |

| Logic | Legend | Example | Sample Match |
|---|---|---|---|
| \| | Alternation / OR operand | 22\|33 | 33 |
| ( … ) | Capturing group | A(nt\|pple) | Apple (captures "pple") |
| \1 | Contents of Group 1 | r(\w)g\1x | regex |
| \2 | Contents of Group 2 | (\d\d)+(\d\d)=\2+\1 | 12+65=65+12 |
| (?: … ) | Non-capturing group | A(?:nt\|pple) | Apple |