ADAM MICKIEWICZ UNIVERSITY IN POZNAŃ
Faculty of English

Robert Dyzman, M.Sc.Eng.

PYTHON PROGRAMMING
CLASS 09

Run „Teams”

Start your IDE

AGENDA:

- Quiz
- Create a file *„class_pp_09.py”*
- Pt_02.py solved
- Regular expressions
- https://www.w3schools.com/python/python_regex.asp
- https://pythonexamples.org/python-regular-expression-regex-tutorial/
- Push to GitHub, alternatively Copy/Paste to Teams

REGULAR EXPRESSIONS

- Allows us to search for and match the specific patterns of text;
- **It is a way of assesing whether a text matches a specific pattern e.g., contains a certain sequence of characters or a specific number of characters.**

RE METHODS

- **search(pattern, string, flags=0)** - returns a Match object if there is a match anywhere in the string
- **findall(pattern, string, flags=0)** – returns a list containing all matches
- **split(pattern, string, maxsplit=0, flags=0)** - split string by the occurrences of pattern and returns the list of them
- **sub(pattern, repl, string, count=0, flags=0)** - replaces one or many matches with a string

RE METHODS

- *pattern* – [mandatory] the pattern which has to be found in the string
- *string* – [mandatory] the string in which the pattern has to be found
- *repl* - [mandatory] the value which has to be replaced in the string in place of matched pattern.
- *flag* – [optional]
- *maxsplit* - [optional] the maximum limit on number of splits

search() method

```
import re
if re.search('ab', sys.argv[1]):
    print('a match')
else:
    print('no match')
```

Returns
„Match object”
or
„None”

search() method - summary

PATTERN MATCHING

- `import re`
- `re.search(pattern, string)` → this expression returns ***Match object*** or ***None***
- If we use it inside if statement, a match object will evaluate to True and a None object will evaluate to False

Causes the resulting RE to match 0 or more repetitions of the preceding RE, as many repetitions as are possible. ab* will match 'a', 'ab', o

RE PATTERNS

- A single symbol, for example: 'a', '3', 'k', etc. These will match a string that consists of just the symbol indicated
- A concatenation or sequence of characters. For example: 'ab', '3g', 'kk', etc. Such an expression matches if the given text contains them.
- '.' → any single character (except newline character), e.g.,'a.b', 'a..b'
- '.*' → zero or more occurrences (of any characters), except newline character

REGULAR EXPRESSIONS

```
# 1. test with -> abc, acb, bac
import re
import sys
if re.search('ab', sys.argv[1]):
    print('a match')
else:
    print('no match')
```

REGULAR EXPRESSIONS

```
# 2. test with acdeb, zacdebr, ab, ba
import re
import sys

if re.search('a*b', sys.argv[1]):
    print('a match')
else:
    print('no match')
```