Regular expressions

The Topic: What?

•An introduction to regular expressions using Python

- Pattern matching
- Python standard library module re.py

Applications: Why?

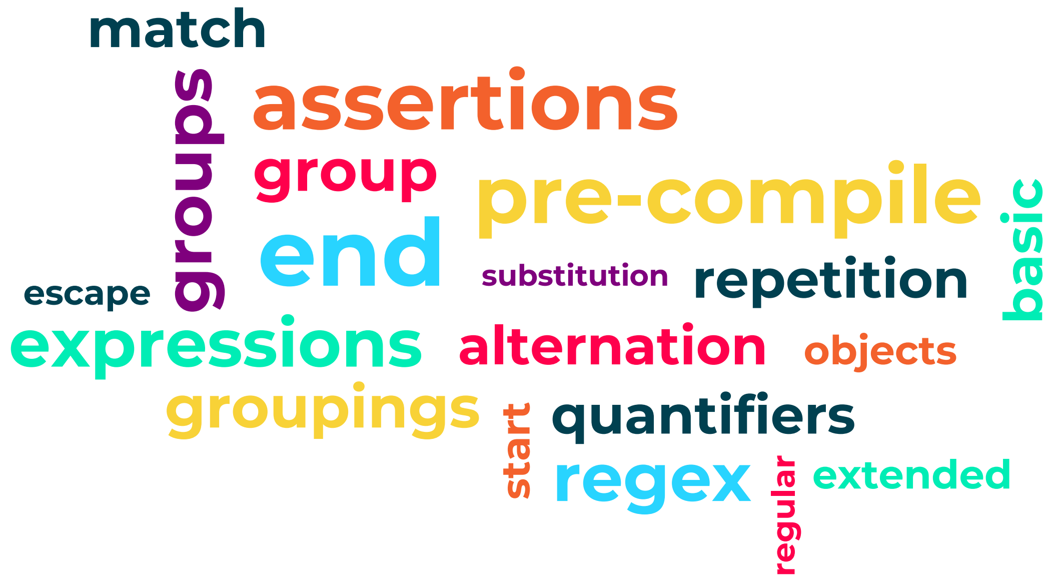
•To be able to match a pattern in a searched for string

Expectations: Who?

•Learners are expected to have covered fundamental programming in Python previously.

Regular expressions

Have you used Regular Expressions before?



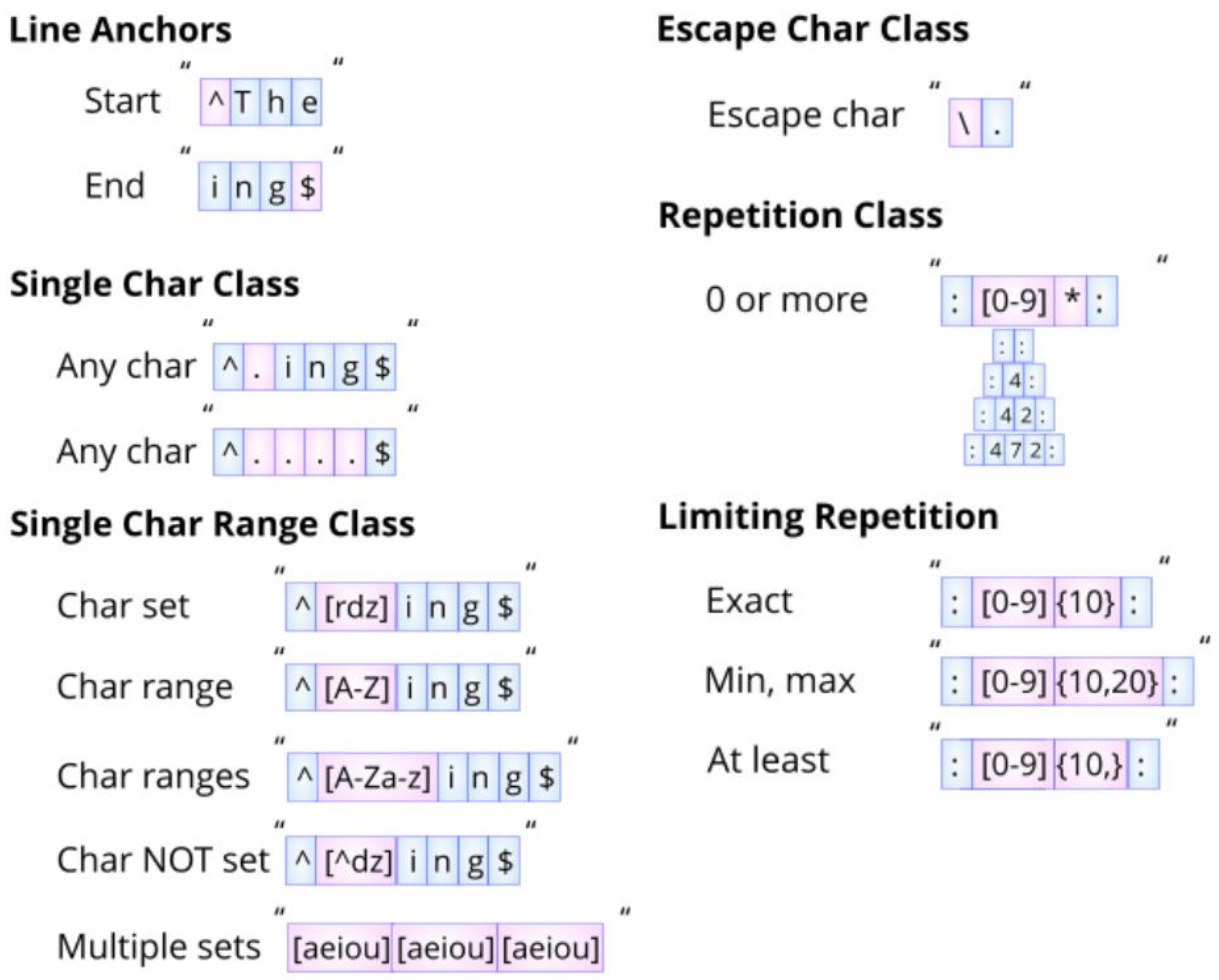
Regex

Regular Expressions (Regex, or just RE) are used in many tools, sql, and programming languages, and have their roots in Unix command line tools such as ed, ex, grep, sed, and awk. They provide a more extensive and powerful set of meta-characters for pattern matching and come in several dialects. The Python standard library module re.py provides matching functions and support for Basic, Extended and Python pattern characters.

To learn how to use Regex, we need to do two things - learn the theory behind the meta-characters, and then how to apply them using Python code.

The website https://regex101.com/ is useful for learning and testing your Python Regular Expressions.

Basic regular expressions (B.R.E.)



Line Anchors

Start	"^The"
End	"ing$"

Single Char Class

Any char	"^.ing$"
Any char	"^....$"

Single Char Range Class

Char set	"^[rdz]ing$"
Char range	"^[A-Z]ing$"
Char ranges	"^[A-Za-z]ing$"
Char NOT set	"^[^dz]ing$"
Multiple sets	"[aeiou][aeiou][aeiou]"

Escape Char Class

Escape char	"\."

Repetition Class

0 or more	":[0-9]*:"

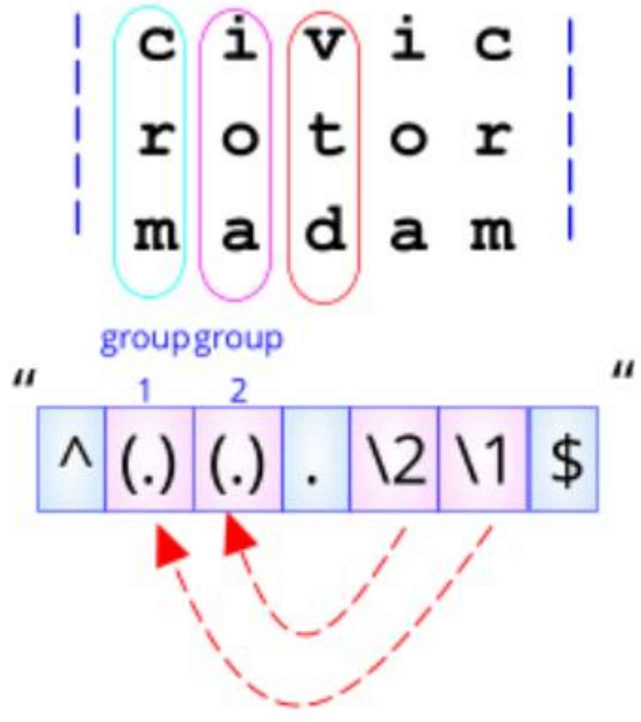
```
::
:4:
:42:
:472:
```

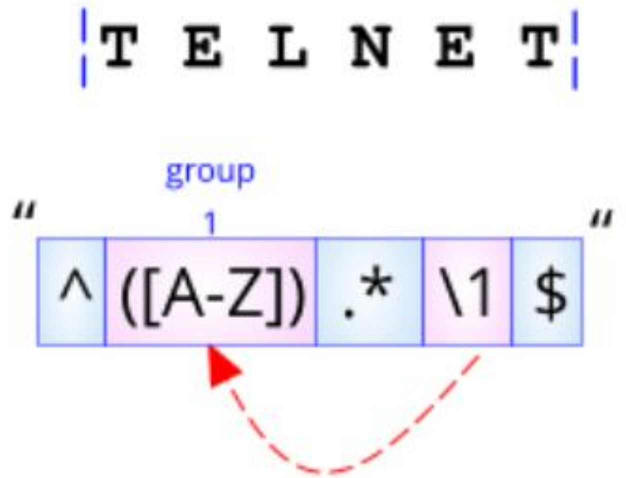
Limiting Repetition

Exact	":[0-9]{10}:"
Min, max	":[0-9]{10,20}:"
At least	":[0-9]{10,}:"

BRE (cont.)

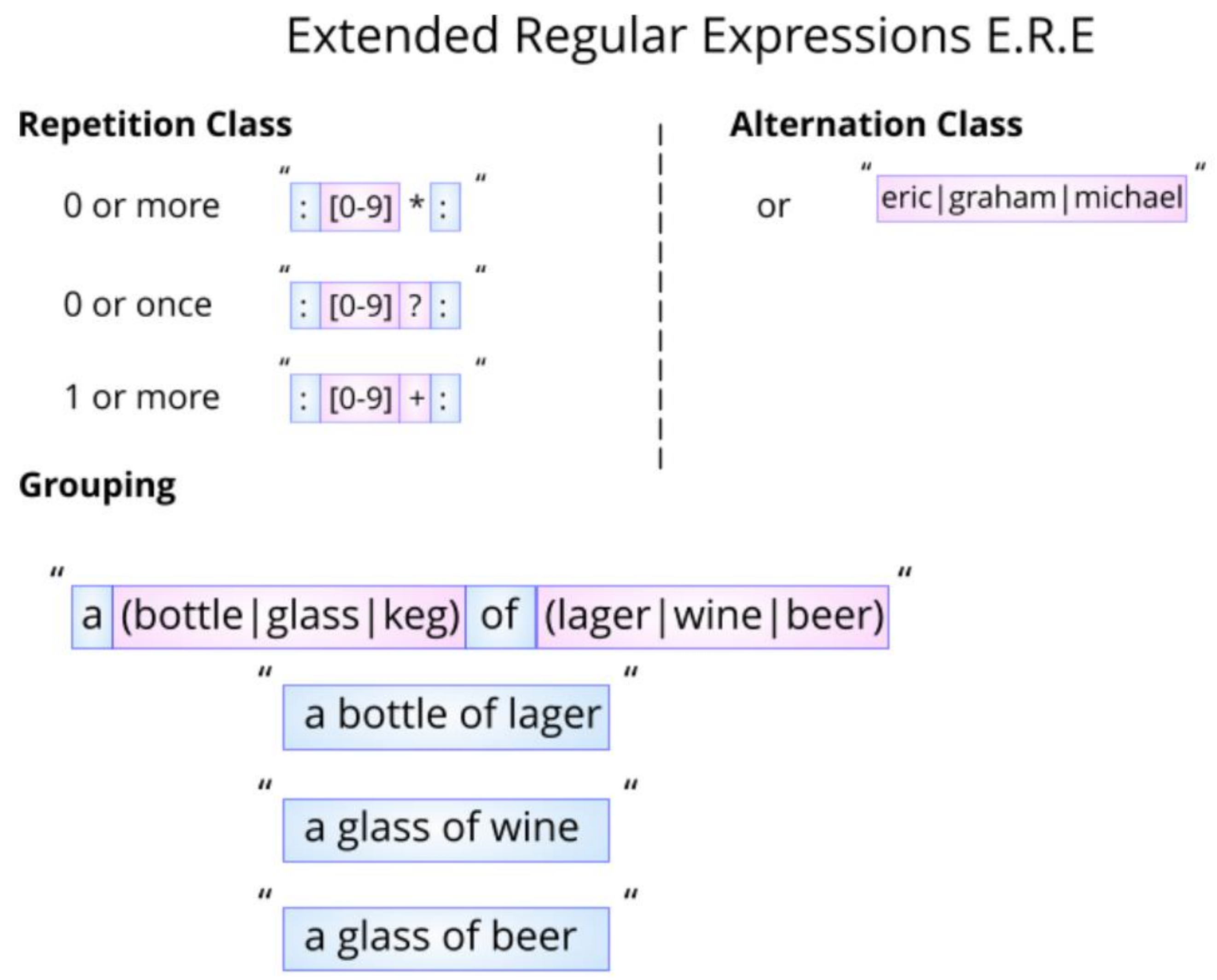
Grouping/Back Referrals





Extended RE

Extended Regular Expressions or E.R.E added some extra repetition chars (?+), groupings (rhubarb)+ and alternation (or).



Extended Regular Expressions E.R.E

Repetition Class

0 or more	":[0-9]*:"
0 or once	":[0-9]?:"
1 or more	":[0-9]+:"

Alternation Class

or "eric|graham|michael"

Grouping

"a (bottle|glass|keg) of (lager|wine|beer)"

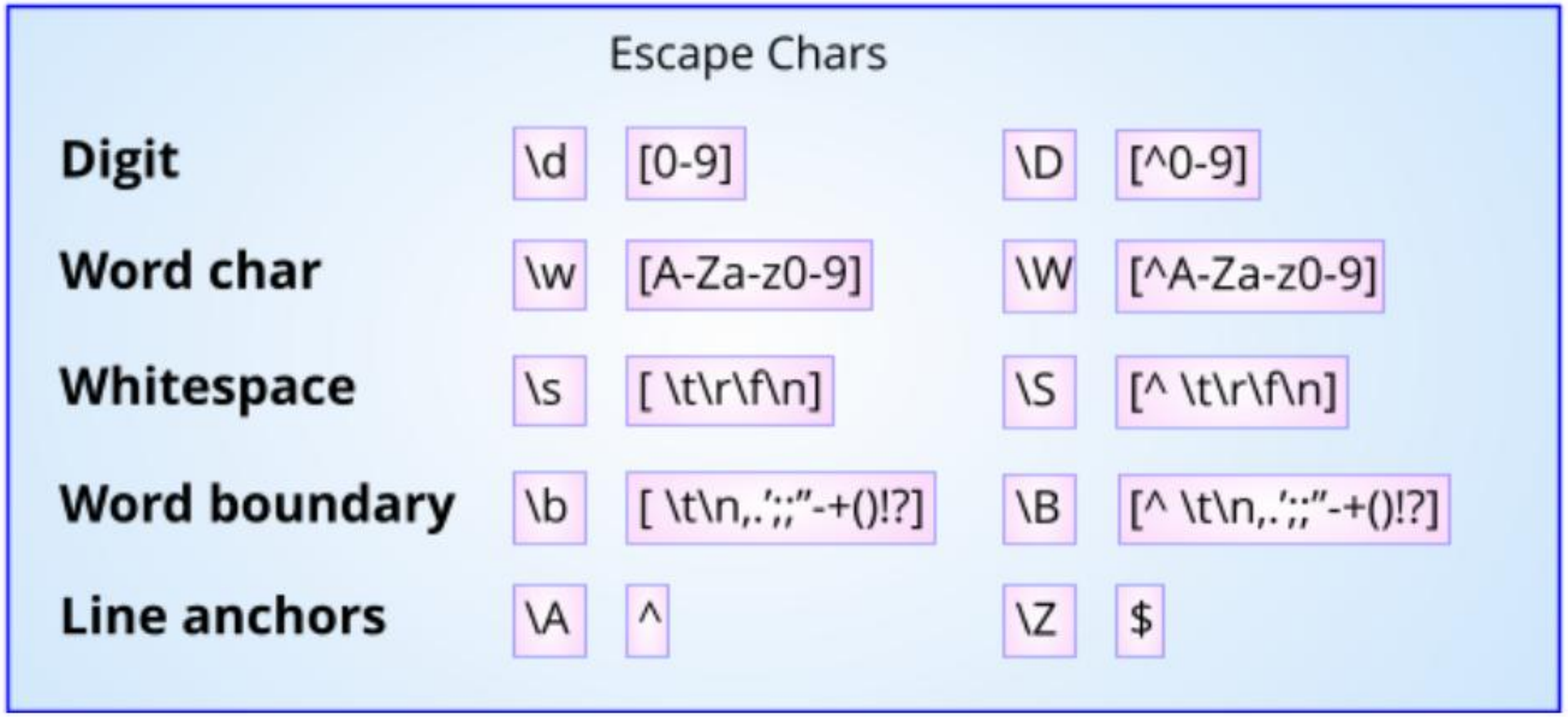
"a bottle of lager"

"a glass of wine"

"a glass of beer"

Python assertions/escape chars

Python Assertions / Escape chars



Escape Chars

Digit	\d	[0-9]	\D	[^0-9]
Word char	\w	[A-Za-z0-9]	\W	[^A-Za-z0-9]
Whitespace	\s	[\t\r\f\n]	\S	[^ \t\r\f\n]
Word boundary	\b	[\t\n,.’;:”-+()!?]	\B	[^ \t\n,.’;:”-+()!?]
Line anchors	\A	^	\Z	$

Regex

Trainer demonstration

demo_regex_1.py

demo_regex_2.py

demo_regex_4.py

Learning check

5-10 mins

Quiz!

1. **What does B.R.E and E.R.E stand for?**
2. **What prefix should you use with your Regex patterns?**
3. **Can you think of a better way to write the following pattern, which is 15 chars long?**

r"^...............$"

4. **Given this code, what will the pattern match?**

```
>>> text = "Brave Sir Robin ran away. Bravely ran away away. When danger
reared its ugly head, he bravely turned his tail and fled."
>>> m = re.search(r"(brave).*\1", text, flags=re.IGNORECASE)
>>> print(m.group())
```

5. **Given this code, what would be printed?**

```
>>> text = "racecar"
>>> m = re.search(r"^(.)(.)(.).\3\2\1$", text)
>>> print(m.groups(), m.groups()[0], m.group(1))
```

Solutions

Regular Expressions quiz

1. What does B.R.E and E.R.E stand for?

   Answer: Basic and Extended Regular Expressions

2. What prefix should you use with your Regex patterns?

   Answer: The r-prefix for RAW pattern string!

   r"pattern"

3. Can you think of a better way to write the following pattern, which is 15 chars long?

   r"^...............$"

   Answer: r"^.{15}$"

4. Given this code, what will the pattern match?

   text = "Brave Sir Robin ran away. Bravely ran away away. When danger reared its ugly head, he bravely turned his tail and fled."

   m = re.search(r"(brave).*\1", text, flags=re.IGNORECASE)

   print(m.group())

   [Note: Regular expressions are greedy!]

   Answer: Brave Sir Robin ran away. Bravely ran away away. When danger reared its ugly head, he brave

5. Given this code, what would be printed?

   text = "racecar"

   m = re.search(r"(.)(.).\3\2\1$", text")

   print(m.groups(), m.groups()[0], m.group(1))

   Answer: ('r', 'a', 'c') r r

Labs

1. For the following exercises, we will be installing the IMDbPY Python package (https://pypi.org/project/IMDbPY/). This package is used to retrieve and manage data of the IMDb online movie database about movies, people, characters and companies.

Install the IMDb Package, on command line or within Pycharm.

C:> pip install IMDbPY.

2. Load the script **C:\labs\top250.py** into Pycharm. This script imports the IMDb module and creates an IMDB HTTP access class for searching for movie information. Compile and run the script and confirm the output displays the top 250 movies.

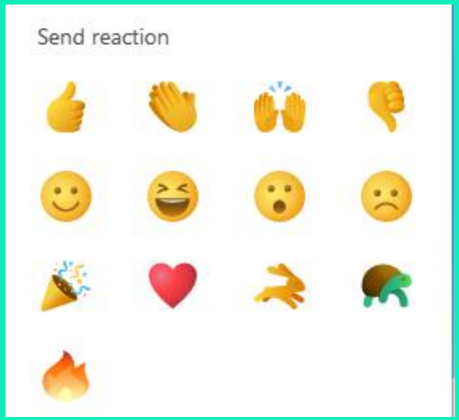
3. Modify the script so that the movie names are displayed alongside their ranking from 1 to 250. Ensure the ranking is right justified to 4 characters so that the ':' is aligned. For example:

```
   1: The Shawshank Redemption
   2: The GodFather
   ..
  99: North by Northwest
```

4. Using regular expressions, modify the script to prompt the user to enter their favourite movie and display its ranking, if found. Ensure that the case is ignored.

Stretch Exercises 5-7

END OF SECTION



An introduction to regular expressions using Python

- Pattern matching
- Python standard library module re.py

•To be able to match a pattern in a searched for string

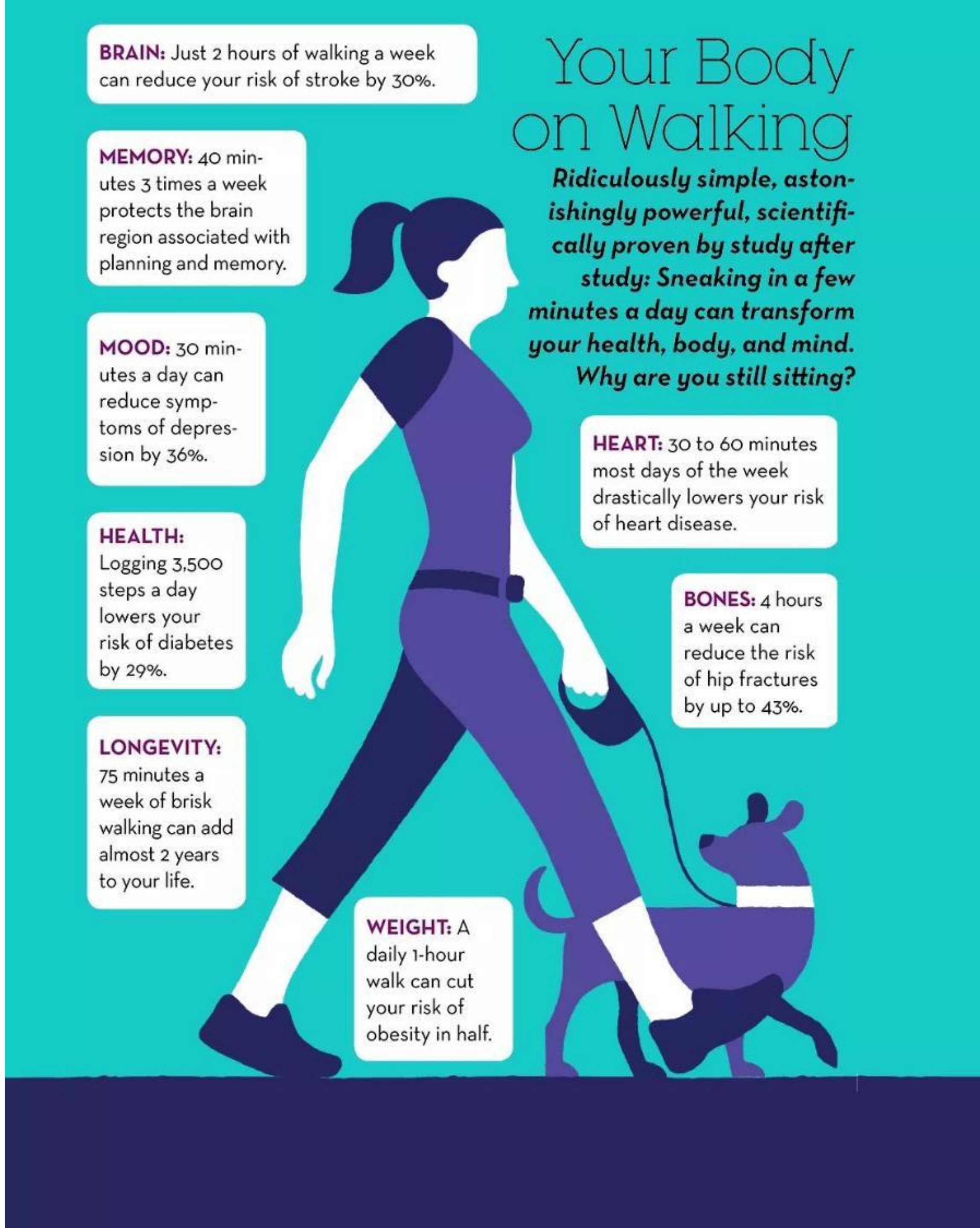
Extension materials

Pre-Compile

- demo_regex_3.py

REMINDER: TAKE A BREAK!

10.30 - 10.40
11.40 – 11.50
12.50 – 13.30
14.30 – 14.40
15.40 – 15.50



Your Body on Walking

Ridiculously simple, astonishingly powerful, scientifically proven by study after study: Sneaking in a few minutes a day can transform your health, body, and mind. Why are you still sitting?

BRAIN: Just 2 hours of walking a week can reduce your risk of stroke by 30%.

MEMORY: 40 minutes 3 times a week protects the brain region associated with planning and memory.

MOOD: 30 minutes a day can reduce symptoms of depression by 36%.

HEALTH: Logging 3,500 steps a day lowers your risk of diabetes by 29%.

LONGEVITY: 75 minutes a week of brisk walking can add almost 2 years to your life.

HEART: 30 to 60 minutes most days of the week drastically lowers your risk of heart disease.

BONES: 4 hours a week can reduce the risk of hip fractures by up to 43%.

WEIGHT: A daily 1-hour walk can cut your risk of obesity in half.