

RegEx Exercises for the Class

Hello happy classmates! At the end of the class, Eric mentioned the dot operator.

The dot operator, which uses the period key, matches any character. (Note: it does not match the newline by default but that isn't relevant to these exercises.)

Since I didn't discuss it in class, you can read up on the dot operator [here](#).

The 'Use the Dot Sparingly' section discusses potential problems with overusing the dot operator.

I have updated regex_demo.py to give you examples of code that use the dot operator in the regular expression.

The Python Software Foundation's article, [Regular Expressions HOW TO](#), provides enough information to program any of these exercises. (I hope! I haven't gotten them all working yet.)

Note: These exercises are also described in regex_exercises.py

Exercise: Parse the title and author from 'The Stone Sky, by N.K. Jemisin'

Note: This is similar to the 'Jack and Jill' parsing example in regex_demo.py

Expected Result:

- Title: 'The Stone Sky'
- Author: 'N.K. Jemisin'

Exercise: Parse the title and author from ' The Stone Sky, by N.K. Jemisin'

Note: adding whitespace at the start of the line

Expected Result:

- Title: 'The Stone Sky'
- Author: 'N.K. Jemisin'

Exercise: Parse the title, author, and publisher from the line ' The Stone Sky, by N.K. Jemisin (Orbit)'

Note: adding the publisher

Expected Result:

- Title: 'The Stone Sky'
- Author: 'N.K. Jemisin'
- Publisher: 'Orbit'

Exercise: Parse publishers from a variety of lines

Note: adds metacharacters to the pattern string

Input String	Expected Result
' (Orbit)'	'Orbit'
' (Tor.com Publishing)'	'Tor.com Publishing'
'(Uncanny, March/April 2017)'	'Uncanny, March/April 2017'
'(Harper Voyager / Spectrum Literary Agency)'	'Harper Voyager / Spectrum Literary Agency'
'(Twelfth Planet Press)'	'Twelfth Planet Press'

Exercise: Parse '"And Then There Were (N-One)," by Sarah Pinsker (Uncanny, March/April 2017)'

Note: adding metacharacters to the title

Expected Results:

- Title: 'And Then There Were (N-One)'
- Author: 'Sarah Pinsker'
- Publisher: '(Uncanny, March/April 2017)'

Exercise: Parse 'Binti: Home, by Nnedi Okorafor (Tor.com Publishing)'

Expected Results:

- Title: 'Binti: Home'
- Author: 'Nnedi Okorafor'
- Publisher: 'Tor.com Publishing'

Exercise: Parse "'Children of Thorns, Children of Water," by Aliette de Bodard (Uncanny, July-August 2017)'

Note: Adding a comma to the tile adds a complication because a comma is part of our separator pattern. You will need to use a negative lookahead assertion which you can read about here:

<https://docs.python.org/2/howto/regex.html>

Expected Results:

- Title: 'Children of Thorns, Children of Water'
- Author: 'Aliette de Bodard'
- Publisher: 'Uncanny, July-August 2017'

Exercise: Parse ' Luminescent Threads: Connections to Octavia E. Butler, edited by Alexandra Pierce, and Mimi Mondal (Twelfth Planet Press)'

Note: Adding a new, and optional, element to the separator pattern. Also adding a second author

Expected Results:

- Title: 'Luminescent Threads: Connections to Octavia E. Butler'
- Author: 'Alexandra Pierce, and Mimi Mondal'
- Publisher: 'Twelfth Planet Press'

Exercise: Write a pattern that can parse all these strings. That is, the same pattern should work on 'The Stone Sky, by N.K. Jemisin' and ' Luminescent Threads: Connections to Octavia E. Butler, edited by Alexandra Pierce, and Mimi Mondal (Twelfth Planet Press)' and each string in betweenFor

Exercise: parse title, author, and publisher for each title in hugo_2018.txt