

Regular expressions & Debug & other tools

Sugarkhuu Radnaa

Py4Econ in Ulaanbaatar

py4econ@gmail.com

Week 7: Learning objectives

Get to know:

- 1 Regular expressions
- 2 Debugging
- 3 Exception handling

Regular expressions

Search "pattern" from text and do an action Package "re":

- compile
- findall – find all matches
- match – match at the beginning of string
- search – first match anywhere in string
- split – split string into pieces at pattern points
- sub – replace match by user input
- group
- special characters

Regular expressions special characters

Regular Expressions Cheat Sheet by [DaveChild](#)

A quick reference guide for regular expressions (regex), including symbols, ranges, grouping, assertions and some sample patterns to get you started.

Anchors

<code>^</code>	Start of string, or start of line in multi-line pattern
<code>\A</code>	Start of string
<code>\$</code>	End of string, or end of line in multi-line pattern
<code>\Z</code>	End of string
<code>\b</code>	Word boundary
<code>\B</code>	Not word boundary
<code>\<</code>	Start of word
<code>\></code>	End of word

Character Classes

<code>\c</code>	Control character
<code>\s</code>	White space
<code>\S</code>	Not white space
<code>\d</code>	Digit
<code>\D</code>	Not digit
<code>\w</code>	Word
<code>\W</code>	Not word
<code>\x</code>	Hexadecimal digit
<code>\O</code>	Octal digit

POSIX

<code>[upper:]</code>	Upper case letters
-----------------------	--------------------

Quantifiers

<code>*</code>	0 or more	<code>{3}</code>	Exactly 3
<code>+</code>	1 or more	<code>{3,}</code>	3 or more
<code>?</code>	0 or 1	<code>{3,5}</code>	3, 4 or 5

Add a `?` to a quantifier to make it ungreedy.

Escape Sequences

<code>\</code>	Escape following character
<code>\Q</code>	Begin literal sequence
<code>\E</code>	End literal sequence

"Escaping" is a way of treating characters which have a special meaning in regular expressions literally, rather than as special characters.

Common Metacharacters

<code>^</code>	<code>[</code>	<code>.</code>	<code>\$</code>
<code>{</code>	<code>*</code>	<code>(</code>	<code>\</code>
<code>+</code>	<code>)</code>	<code> </code>	<code>?</code>
<code><</code>	<code>></code>		

The escape character is usually `\`.

Special Characters

<code>\n</code>	New line
-----------------	----------

Groups and Ranges

<code>.</code>	Any character except new line (<code>\n</code>)
<code>(a b)</code>	a or b
<code>(...)</code>	Group
<code>(?...)</code>	Passive (non-capturing) group
<code>[abc]</code>	Range (a or b or c)
<code>[^abc]</code>	Not (a or b or c)
<code>[a-q]</code>	Lower case letter from a to q
<code>[A-Q]</code>	Upper case letter from A to Q
<code>[0-7]</code>	Digit from 0 to 7
<code>\x</code>	Group/subpattern number "x"

Ranges are inclusive.

Pattern Modifiers

<code>g</code>	Global match
<code>i *</code>	Case-insensitive
<code>m *</code>	Multiple lines
<code>s *</code>	Treat string as single line
<code>x *</code>	Allow comments and whitespace in pattern
<code>e *</code>	Evaluate replacement
<code>U *</code>	Ungreedy pattern
<code>*</code>	PCRE modifier

String Replacement

In IDEs (VSCode):

- Actions at breakpoints: continue, step over, step into, step out
- Additional actions at breakpoints - expression, hit count, log message
- Watch
- Exceptions

Hand inputted breakpoint:

- `import pdb, pdb.set_trace()`

Exception handling

Exceptions occur from time to time in any sort of job. If you know what kind of errors potentially could occur and implement “remedy(ies)” on it beforehand

```
373  
374     try:  
375         # do main job  
376     except:  
377         # do another - Hereby, you are handling an exception!  
378
```

- ① Task 1
 - ② Task 2
 - ③ Task 3
- Submit your result as a Github repository
 - Deadline: 1 week

Task 1: Regex

Өөрөө зохиох эсвэл бэлэн текст интернетээс олж regex-ийн дараах үйлдлүүдийг ашиглан текстүүд гаргаж авах жишээнүүд үзүүл (нэг бүр дээр нь бус хамтад нь ашиглаж болно. Гэхдээ доорх бүх тэмдгээс ядаж 1 удаа ашиглаарай)

- `[a-zA-Z0-9]`, `[a-z]`, `[A-Z]`, `[0-9]`
- `\d`, `\D`, `\w`, `\W`, `\s`
- `^`, `$`, `?`, `*`, `+`, `.`
- `{m,n}`, `{,n}`, `{m,}`, `{n}`
- Look behind, Look ahead, Negative look behind, Negative look ahead

Task 2: Exception

- 1 Generate array of 1000 random integers in numpy
- 2 Create a sub-array of the above with only the negative even numbers.
- 3 Loop one by one through this array ...
- 4 ...when negative odd, then raise “odd error” and continue to next loop,
- 5 ...when even but positive, raise “sign error” and continue to next loop ...
- 6 ...when “negative even”, then append your list.

Task 3: Debugging

- Debugging гэж юу вэ?
- Breakpoint – ийн ямар 3 төрөл (log message г.м) Vscode дээр байдаг вэ?
- Debug-ийн step into, step over, step out – ийн ялгааг тайлбарла

Thank you!