Язык программирования

python

Лекция № 9. Регулярные выражения в Python

Евгений Сергеевич Чухланцев

Регулярное выражение [A-Za- z] \w+ означает, что первый символ должен быть алфавитным, т.е. должен относиться к диапазону A-Z или a-z, а за ним следует по крайней мере один (+) алфавитно-цифровой символ (\w).

Обозначение	Описание	Пример регулярного выражения
Знаки		
`literal`	Строка содержит символьный литерал `literal`	`foo`
`re1|re2`	Строка содержит регулярные выражения `re1` или `re2`	`foo|bar`
`.`	Соответствует *любому символу* (кроме `\n`)	`b.b`
`^`	Соответствует *началу строки*	`^Dear`
`$`	Соответствует *концу строки*	`/bin/*sh$`
`*`	Предшествующее регулярное выражение встречается в строке *любое количество* (или не встречается вообще)	`[A-Za-z0-9]*`
`+`	Предшествующее регулярное выражение встречается в строке *не менее одного раза*	`[a-z]+\.com`
`?`	Предшествующее регулярное выражение встречается в строке *только один раз или вообще в ней не встречается*	`goo?`

`{N}`	Предшествующее регулярное выражение встречается в строке *N* раз	`[0-9]{3}`
`{M,N}`	Предшествующее регулярное выражение встречается в строке от *M до N* раз	`[0-9]{5,9}`
`[...]`	Соответствует любому отдельному символу из *класса символов*	`[aeiou]`
`[..x-y..]`	Соответствует любому отдельному символу *в диапазоне от x до y*	`[0-9],[A-Za-z]`
`[^...]`	*Не соответствует* ни одному символу из класса символов, включая любые диапазоны, если они заданы	`[^aeiou],` `[^A-Za-z0-9_]`
`(*|+|?|{})?`	Предусматривает применение "нежадных" версий приведенных выше знаков вхождения/повторения (*, +?, {})	`.*?[a-z]`
`(...)`	Соответствует регулярному выражению, заключенному в круглые скобки, и сохраняет его в памяти как *подгруппу*	`([0-9]{3})?,` `f(oo|u)bar`

Специальные символы

\d	Соответствует любой *десятичной цифре*, так же, как [0-9] (\D является обратным по отношению к \d: не соответствует ни одной цифре)	data\d+.txt
\w	Соответствует любому *алфавитно-цифровому символу*, эквивалентно [A-Za-z0-9_] (\W является обратным по отношению к \w)	[A-Za-z_]\w+
\s	Соответствует любому пробельному символу, эквивалентно [\n\t\r\v\f] (\S является обратным по отношению к \s)	of\sthe
\b	Позиция, соответствующая границе слова (\B является обратным по отношению к \b),	\bThe\b
\N	Соответствует *сохраненной* подгруппе *N* (см. (...) выше)	price: \16
\c	Буквально соответствует *любому специальному символу c* (т.е. игнорирует особое значение этого литерала)	\., \\, *
\A (\Z)	Позиция, соответствующая *началу (концу)* строки (см. также ^ и $ выше)	\ADear

Расширенная система обозначений

`(?iLmsux)`	Внедряет один или несколько специальных "флагов" непосредственно в регулярное выражение (применяется вместо способа, основанного на использовании функции/метода)	`(?x), (?im)`
`(?:...)`	Обозначает группу, содержимое которой *не сохраняется в памяти*	`(?:\w+\.)*`
`(?P<name>...)`	Обозначает группу, заданную именем, а не числовым идентификатором	`(?P<data>)`
`(?P=name)`	Сопоставляется с текстом, который был перед этим сгруппирован оператором `(?P<name>)` в той же строке	`(?P=data)`
`(?#...)`	Задает примечание; все содержимое примечания игнорируется	`(?#comment)`
`(?=...)`	Обеспечивает сопоставление, если далее следует шаблон `...`, без сохранения в памяти сопоставленной части входной строки; эта операция именуется *положительной опережающей проверкой* (positive lookahead assertion)	`(?=.com)`

`(?!...)`	Обеспечивает сопоставление, если далее не следует шаблон `...`, без сохранения в памяти сопоставленной части входной строки; эта операция именуется *отрицательной опережающей проверкой* (negative lookahead assertion)	`(?!.net)`
`(?<=...)`	Обеспечивает сопоставление, если далее находится шаблон `...`, без сохранения в памяти сопоставленной части входной строки; эта операция именуется *положительной ретроспективной проверкой* (positive lookbehind assertion)	`(?<=800-)`
`(?<!...)`	Обеспечивает сопоставление, если далее не находится шаблон `...`, без сохранения в памяти сопоставленной части входной строки; эта операция именуется *отрицательной ретроспективной проверкой* (positive lookbehind assertion)	`(?<!192\.168\.)`
`(?(id/name)Y|N)`	Обеспечивает сопоставление регулярного выражения *Y* по условию, если группа с указанным идентификатором или именем существует; в противном случае возвращает *N*; часть *|N* является необязательной	`(?(1)y|x`

В регулярных выражениях операция чередования обозначается с помощью символа канала (|), который представлен на клавиатуре вертикальной чертой (pipeline symbol). Символ канала используется для отделения друг от друга разных регулярных выражений.

Шаблон регулярного выражения	Сопоставленные строки
`at|home`	`at, home`
`r2d2|c3po`	`r2d2, c3po`
`bat|bet|bit`	`bat, bet, bit`

Знак точки (.) обеспечивает сопоставление с любым отдельным символом, кроме \n. Знак точки сопоставляется с любой буквой, цифрой, пробельным символом (не включая " \n"), печатаемым или непечатаемым знаком.

Шаблон регулярного выражения	Сопоставленные строки
`f.o`	В этом примере точка между "`f`" и "`o`" представляет любой символ, например, как в строках `fao`, `f9o`, `f#o` и т.д.
`..`	Любая пара символов
`.end`	Любой символ перед строкой `end`

Для сопоставления с шаблоном, начиная с начала строки, необходимо использовать знак вставки (^) или специальный символ \A (прописная буква "A", которая следует за обратной косой чертой). Последний вариант применяется в основном на компьютерах с клавиатурой, на которой отсутствует знак вставки (такой как международная клавиатура). Аналогичным образом знак доллара ($) или специальный символ \Z применяется для сопоставления с шаблоном, начиная с конца сроки.

Шаблон регулярного выражения	Сопоставленные строки
`^From`	Любая строка, которая начинается с подстроки `From`
`/bin/tcsh$`	Любая строка, которая заканчивается подстрокой `/bin/tcsh`
`^Subject: hi$`	Любая строка, полностью совпадающая со строкой `Subject: hi`

Шаблон регулярного выражения	Сопоставленные строки
`the`	Любая строка, содержащая подстроку `the`
`\bthe`	Любое слово, которое начинается с подстроки `the`
`\bthe\b`	Сопоставляется только со словом `the`
`\Bthe`	Любая строка, которая содержит подстроку `the`, но с этой подстроки не начинается слово

Безусловно, знак точки хорошо подходит для тех случаев, когда необходимо обеспечить сопоставление с любым знаком, но иногда требуется провести сопоставление лишь с конкретным набором символов. По этой причине была предусмотрена возможность применения в шаблонах знаков квадратных скобок ([ ]).

Шаблон регулярного выражения	Сопоставленные строки
`b[aeiu]t`	`bat, bet, bit, but`
`[cr][23][dp][o2]`	Строка из четырех символов: в начале следует `"c"` или `"r"`, затем — `"2"` или `"3"`, после этого — `"d"` или `"p"` и, наконец, `"o"` или `"2"`. Примерами могут служить подстроки `c2do`, `r3p2`, `r2d2`, `c3po` и т.д.

Квадратные скобки позволяют задавать не только наборы из отдельных симво лов, но и диапазоны символов. Для обозначения диапазона символов применяется пара символов, заключенных в квадратные скобки, между которыми проставлен знак дефиса. В качестве примера можно указать диапазоны A-Z, a-z и 0-9, применяемые для обозначения прописных букв, строчных букв и цифровых знаков соответственно.

Шаблон регулярного выражения	Сопоставленные строки
`z.[0-9]`	Буква "z", за которой следует любой символ, за которым следует одна цифра
`[r-u][env-y][us]`	Буква "r", "s", "t" или "u", за которой следует буква "e", "n", "v", "w", "x" или "y", за которой следует буква "u" или "s"
`[^aeiou]`	Знак, отличный от гласной буквы (**Упражнение.** Можно ли в этой формулировке вместо "знак, отличный от гласной буквы" применить "знак согласной буквы"?)
`[^\t\n]`	Знак, отличный от знака табуляции или \n
`["-a]`	В системе ASCII — все символы, которые расположены между '"' и "a", т.е. между знаками с порядковыми номерами 34 и 97

Использование операторов

Шаблон регулярного выражения	Сопоставленные строки
`[dn]ot?`	Буквы "d" или "n", за которыми следует буква "o", а затем, самое большее, — одна буква "t". Таким образом, это регулярное выражение сопоставляется со словами do, no, dot, not
`0?[1-9]`	Любой цифровой символ, которому может предшествовать цифра "0". В качестве примера можно указать множество числовых обозначений месяцев от января до сентября, представленных с помощью одной или двух цифр
`[0-9]{15,16}`	Пятнадцать или шестнадцать цифр (например, номера кредитных карточек)
`</?[^>]+>`	Строки, которые сопоставляются со всеми допустимыми (и недопустимыми) тегами HTML
`[KQRBNP][a-h][1-8]-[a-h][1-8]`	Допустимый шахматный ход в алгебраической нотации (только ходы; взятия фигур, шахи и др. не представлены); т.е. строки, которые начинаются с любой из букв "K", "Q", "R", "B", "N" или "P", обозначающих шахматные фигуры, за которой следует разделенная дефисами пара обозначений клеток шахматной доски от "a1" до "h8" (и всего, что находится между ними), в которой первое обозначение клетки указывает прежнее местонахождение фигуры, а второе — новое местонахождение

Специальные символы, обозначающие наборы символов

Шаблон регулярного выражения	Сопоставленные строки
`\w+-\d+`	Алфавитно-цифровая строка и число, разделенные дефисом
`[A-Za-z]\w*`	Первый символ — алфавитный; следующие символы (если они присутствуют) могут быть алфавитно-цифровыми. Это регулярное выражение почти эквивалентно выражению, которое описывает множество допустимых идентификаторов Python (см. примеры)
`\d{3}-\d{3}-\d{4}`	Номера телефонов в формате, принятом в США, с префиксом кода города, как в примере 800-555-1212
`\w+@\w+\.com`	Простые адреса электронной почты в форме `XXX@YYY.com`

Пара круглых скобок (( )) в регулярном выражении позволяет решить любую из следующих задач (или обе эти задачи).

- Выполнить группирование регулярных выражений.
- Провести сопоставление с подгруппами.

Шаблон регулярного выражения	Сопоставленные строки
`\d+(\.\d*)?`	Строки, представляющие числа с плавающей точкой в простом формате, т.е. любое количество цифр, за которым следует необязательная отдельная десятичная точка, затем нуль или большее количество цифровых символов, например, `"0.004"`, `"2"`, `"75."` и т.д.
`(Mr?s?\. )?[A-Z] [a-z]* [ A-Za-z-]+`	Имя и фамилия, в которых имя может быть представлено сокращенно (должно начинаться с прописной буквы, за которой следуют строчные буквы остальной части имени, если она задана), фамилия представлена полностью, а впереди находится необязательное обращение, `"Mr."`, `"Mrs."`, `"Ms."` или `"M."`. При указании фамилии предусмотрена возможность задавать несколько ее компонентов с использованием дефисов, прописных и строчных букв

Шаблон регулярного выражения	Определение расширенных обозначений
`(?:\w+\.)*`	Строки, которые оканчиваются точкой, такие как `"google."`, `"twitter."`, `"facebook."`. Обнаруженные при этом сопоставления не сохраняются для дальнейшего использования и не могут быть извлечены
`(?#comment)`	Эта конструкция не определяет сопоставление и применяется исключительно для ввода комментария
`(?=.com)`	Сопоставление происходит, только если далее следует подстрока `".com"`; обработчик не изымает ни одной части целевой строки.
`(?!.net)`	Сопоставление происходит, только если далее не следует подстрока `".net"`
`(?<=800-)`	Сопоставление происходит, только если данной строке предшествует подстрока `"800-"`; эта конструкция может применяться, допустим, для поиска номеров телефонов. В этом случае обработчик не изымает ни одной части входной строки
`(?<!192\.168\.)`	Сопоставление происходит, только если данной строке не предшествует подстрока `"192.168."`. Эта конструкция может применяться, например, для фильтрации IP-адресов класса C
`(?(1)y|x)`	Если сопоставленная группа 1 (\1) существует, провести сопоставление с *y*; в противном случае — с *x*

Спасибо за внимание !

Домашнее задание

Продолжаем читать книгу: Лутц М. “Изучаем Python" (4-е издание, в 2-х томах) (2011, PDF) !