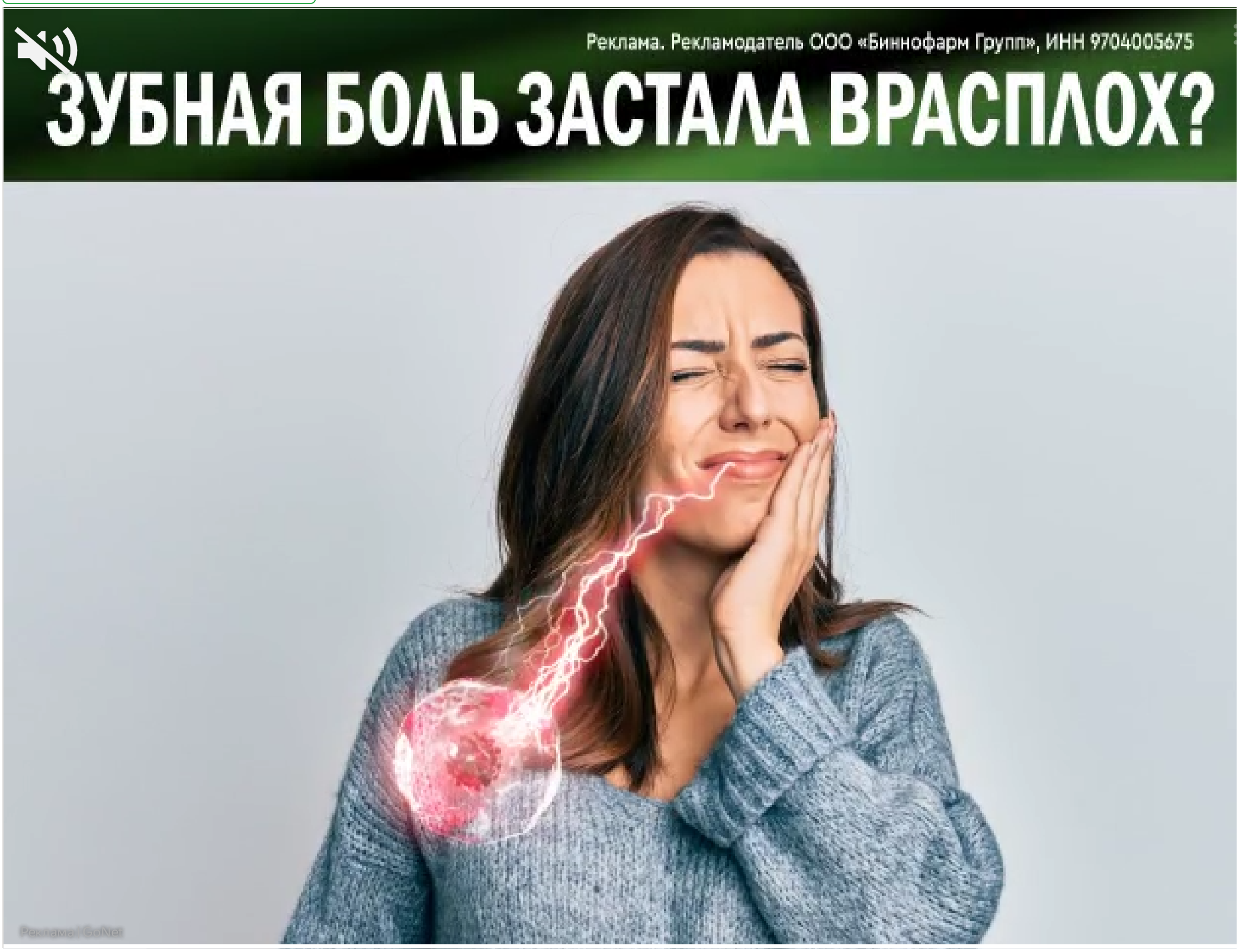
Использование регулярных выражений



Справочник по языку Python3. / Использование регулярных выражений

Регулярные выражения - это шаблоны соответствия текста, описанные в формальном синтаксисе. Шаблоны интерпретируются как набор инструкций, которые затем выполняются со строкой в качестве входных данных для создания соответствующего подмножества или модифицированной версии оригинала. Регулярные выражения могут включать в себя буквальное сопоставление текста, повторение, ветвление и другие сложные правила. Регулярные выражения обычно используются в приложениях, которые требуют тонкую обработку текста.

Модуль re предоставляет операции сопоставления шаблонов регулярных выражений, аналогичные тем, которые встречаются в языке Perl.

Важно отметить, что в языке Python большинство операций с регулярными выражениями доступны как функции и методы уровня модуля для скомпилированных регулярных выражений. Функции модуля re не требуют, чтобы вы сначала компилировали объект регулярного выражения, но не имеют некоторых параметров тонкой настройки шаблона для поиска регулярного выражения.

Шаблоны регулярных выражений и строки для поиска могут быть как Unicode strings, так и 8-битными строками . Однако строки Unicode и 8-битные строки не могут быть смешаны. То есть вы не можете сопоставить строку Unicode с байтовым шаблоном регулярного выражения или наоборот. Аналогично, при замене на основе регулярного выражения строка замены должна быть того же типа, что и регулярное выражение и строка поиска.

Символьные классы и сокращенные обозначения классов

Квадратные скобки `[]` используются для указания класса символов, который представляет собой набор символов, которым нужно соответствовать. Специальные последовательности могут быть включены в класс символов для обозначения групп по[Вверх]тельности букв, слова, цифр и т. д.

Повторение набора символов в регулярном выражении

Регулярных выражениях можно указывать повторение определенное количество раз для какого-то набора символов. Символы операций повторения называются квантификаторы, это плюс, звездочка и знак вопроса.

К выражений в Python

Ре уются в объекты шаблонов, которые имеют методы для различных операций, таких как поиск со олнение подстановок строк.

Об рных выражениях Python

Чт сую черту литерала, нужно написать ``\\\\`` как строку шаблона RegExp в Python, потому что ре ыть `\\` и каждая обратная косая черта должна быть выражена как `\\` внутри обычного ст

Ме ого регулярного выражения в Python

Ме о совпадает ли выражение в начале строки. Метод search() сканирует всю строку и заканчивает ра Метод findall() возвращает список всех совпадающих подстрок.

Фу re в Python

В ужно создавать/компилировать объект шаблона и вызывать его методы. Модуль re определяет функции верхнего уровня, такие как match(), search(), findall(), sub() и так далее. Шаблон регулярного выражения добавляется в качестве первого аргумента функции.

Флаги объекта регулярного выражения

Флаги объекта re.compile() позволяют изменить некоторые аспекты работы регулярных выражений. Флаги доступны в модуле re под двумя именами: длинное имя, например, re.IGNORECASE и короткая однобуквенная форма, например re.I. Несколько флагов в функции компиляции re.compile() могут быть зада

Метасимволы нулевой ширины в RegExp Python

Некоторые из оставшихся метасимволов, являются утверждениями нулевой ширины. Они не заставляют движок регулярных выражений продвигаться по строке. Они вообще не сравниваются с символами, они просто сообщают движку регулярных выражений о присутствии данного условия в строке или терпят неудачу.

Группы с захватом в регулярных выражениях Python

Группировка результатов в шаблоне регулярного выражения происходит при помощи круглых скобок '()'. Метасимволы '(' и ')' имеют то же значение, что и в математических выражениях. Они группируют содержащиеся в них выражения для последующего извлечения.

Обратные ссылки регулярного выражения

Обратные ссылки в шаблоне позволяют указать, что содержимое более ранней группы также должно быть найдено в текущем месте строки. Например обозначение `\1` в шаблоне регулярного выражения будет соответствовать содержимому группы с номером `1`.

Именованные группы регулярных выражениях Python

Синтаксис имени группы является одним из расширений Python: (?P<name>...). Переменная name это название группы. Именованные группы ведут себя точно так же, как группы захвата и дополнительно связывают имя с группой.

Вверх

Группы без захвата в регулярных выражениях

Иногда требуется использовать группу для обозначения части регулярного выражения при этом содержимого группы в последствии получать не нужно. Для этого необходимо [использовать группу без захвата: `(?:...)`где `...` заменяется любым регулярным выражением.

Опережающая и ретроспективная проверка позиции в RegExp

Оп ует текст, расположенный справа, и проверяет возможность совпадения подвыражения. Если со считается успешной. Также существует ретроспективная проверка, при которой текст ан авлении, к левому краю.

Де улярному выражению

Фу ку string по соответствиям регулярного выражения. Если в регулярном выражении используются за одержимое также будет возвращено как часть результирующего списка. Можно ограничить ко ний, передав значение maxsplit.

По улярному выражению

Ра ти все совпадения для шаблона и заменить их другой строкой. Рассмотрены методы sub() и su

ma

Фу совпадение регулярного выражения только в начале строки, в то время как функция re.search() будет сканировать строку вперед на предмет первого совпадения с шаблоном. Важно держать это различие в виду.

Жадный квантификатор против не жадного

При повторении регулярного выражения при помощи `a*`, результирующее действие состоит в том, чтобы захватить как можно больше строки для анализа. Такой шаблон как `''<.*>''` для сопоставления одного HTML тега не работает из-за жадной природы выражения `.*`.

Вверх