



Использование регулярных выражений в Python



0+

hh.ru РЕКЛАМА · 16+

hh.ru для малого бизнеса: разместите вакансию

Более 50 млн резюме в нашей базе. Ищите помощников для своего бизнеса на hh.ru

Зарегистрироваться

Реклама. Рекламодатель ООО "Хэдхантер", ИНН 7718620740, www.hh.ru.

[Стандартная библиотека Python3.](#) / Использование регулярных выражений в Python

[Модуль re](#) предоставляет операции сопоставления шаблонов регулярных выражений, аналогичные тем, которые встречаются в языке Perl.

Новое в Python 3.11: Механизм сопоставления регулярных выражений [модуля re](#) был частично переработан и теперь использует вычисляемые переходы (или "поточный код") на поддерживаемых платформах. В результате Python 3.11 выполняет тесты регулярных выражений *superformance* на 10% быстрее, чем Python 3.10.

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.

Регулярные выражения используют символ обратной косой черты '\', чтобы указать специальные формы или разрешить использование специальных символов, используемых в шаблонах поиска без вызова их специального значения. Это противоречит использованию в Python одного и того же символа для той же цели в [строковых литералах](#). Например, чтобы сопоставить обратную косую черту литерала '\', может потребоваться записать '\\\\' как строку шаблона поиска, потому что регулярное выражение должно быть \\, и каждая обратная косая черта должна быть выражена как \\ внутри обычного строкового литерала Python.

Решение заключается в использовании необработанной строковой нотации Python для шаблонов регулярных выражений. Обратная косая черта не обрабатывается особым образом в строковом литерале с [префиксом 'r'](#). Так что r"\n" это двухсимвольная строка, содержащая '\' и 'n', в то время как "\n" односимвольная строка, содержащая новую строку. Обычно шаблоны поиска будут выражаться в коде Python с использованием этой необработанной строковой записи.

Важно отметить, что 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 допускают некоторые параметры тонкой настройки шаблона для поиска регулярного выражения.

[Синтаксис регулярных выражений в Python](#)

Регулярные выражения могут быть объединены для формирования новых регулярных выражений; если A и B оба являются регулярными выражениями, то AB также является регулярным выражением. Большинство обычных символов, таких как 'A', 'a'или '0', являются простейшими регулярными выражениями; они просто соот

[Функция compile\(\) модуля re в Python](#)

Функция compile() модуля re компилирует шаблон регулярного выражения pattern в объект регулярного выражения, который может быть использован для поиска совпадений

[Флаги, используемые в функции re.compile\(\)](#)

Флаги, используемые для компиляции регулярного выражения.

[Функция search\(\) модуля re в Python](#)

Функция search() модуля re сканирует строку string в поисках первого совпадения с шаблоном pattern регулярного выражения и во

Вверх

 соответствующий объект соответствия

[Функция match\(\) модуля re в Python](#)

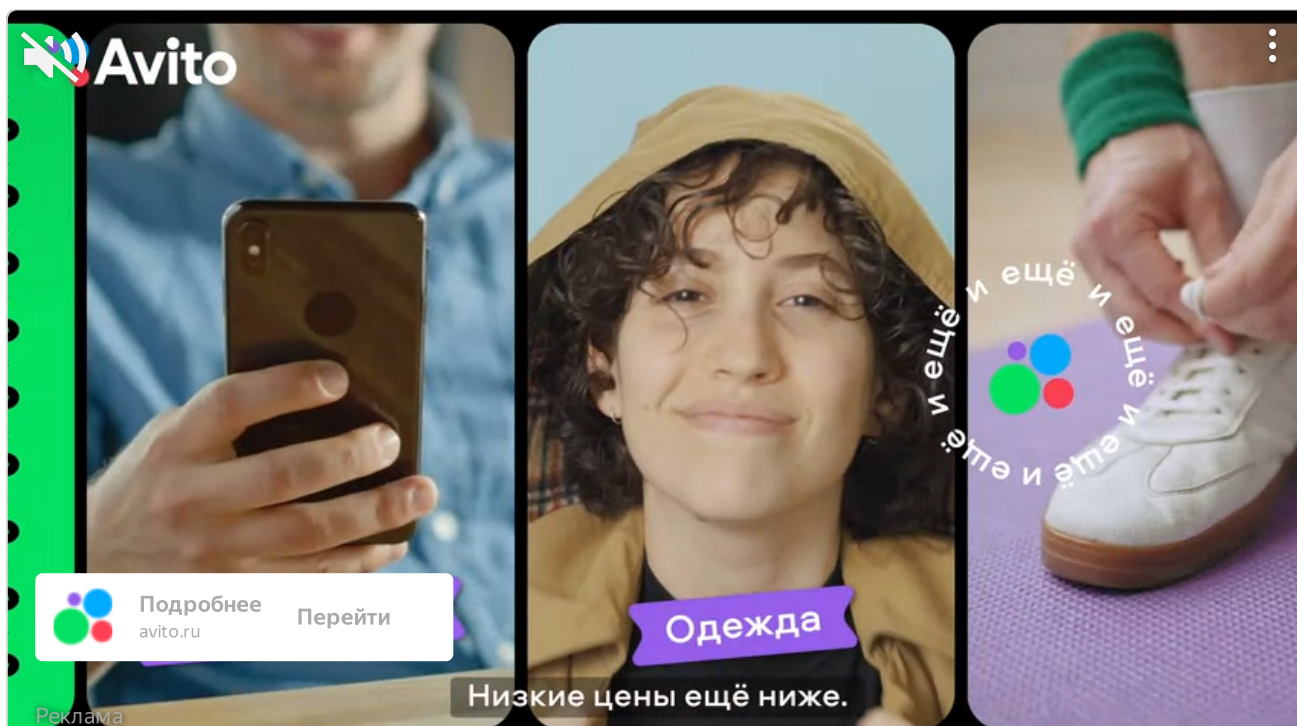
Функция `match()` модуля `re` вернуть соответствующий объект сопоставления, если ноль или более символов в начале строки `string` соответствуют шаблону регулярного выражения `pattern`.

[Функция `fullmatch\(\)` модуля `re` в Python](#)

Функция `fullmatch()` модуля `re` вернет объект сопоставления, если вся строка `string` соответствует шаблону регулярного выражения `pattern`.

[Функция `finditer\(\)` модуля `re` в Python](#)

Функция `finditer()` модуля `re` возвращает итератор объектов сопоставления по всем неперекрывающимся совпадениям для шаблона регулярного выражения в строке.



[Функция `split\(\)` модуля `re` в Python](#)

Функция `'split()'` модуля `'re'` делит строку по появлению шаблона регулярного выражения `'pattern'` и возвращает список получившихся подстрок.

[Функция `findall\(\)` модуля `re` в Python](#)

Функция `findall()` модуля `re` возвращает все неперекрывающиеся совпадения шаблона `pattern` в строке `string` в виде списка строк. Строка сканируется слева направо, и совпадения возвращаются в найденном порядке.

[Функция `sub\(\)` модуля `re` в Python](#)

Функция `sub()` модуля `re` возвращает строку, полученную путем замены крайнего левого неперекрывающегося вхождения шаблона регулярного выражения `pattern` в строке `string` на строку замены `rep1`. Если шаблон регулярного выражения не найден, строка возвращается без изменений.

[Функция `subn\(\)` модуля `re` в Python](#)

Функция `subn()` модуля `re` выполняет ту же операцию, что и функция `sub()`, но возвращает кортеж (`new_string`, `number_of_subs_made`)

[Функция `escape\(\)` модуля `re` в Python](#)

Функция `'escape()'` модуля `'re'` выполняет экранирование специальных символов в шаблоне. Это полезно, если требуется сопоставить произвольную строку литерала, которая может содержать метасимволы регулярных выражений

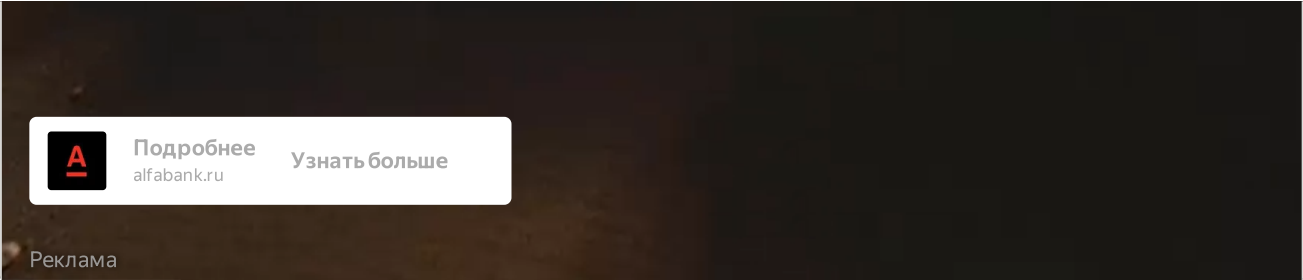
[Функция `purge\(\)` модуля `re` в Python](#)

Функция `'purge()'` модуля `'re'` очищает кэш от регулярных выражений.

[Исключение `error\(\)` модуля `re` в Python](#)

Исключение `'error()'` модуля `'re'` возникает, когда строка, переданная одной из функций модуля, не является допустимым регулярным выражением, например шаблон может содержать несоответствующие скобки или когда возникает какая-либо другая ошибка во время компиляции шаблона или сопоставления со строкой.





Объект регулярного выражения Pattern модуля re в Python

Объект регулярного выражения Pattern получается в результате компиляции шаблона регулярного выражения. Скомпилированные объекты регулярных выражений поддерживают рассмотренные ниже методы и атрибуты.

Объект совпадения с шаблоном Match модуля re в Python

Объект сопоставления регулярного выражения со строкой всегда имеет логическое значение True. Можно проверить, было ли совпадение, с помощью простого утверждения if...else. Объекты сопоставления поддерживают методы и атрибуты.

Содержание раздела:

- [ОБЗОРНАЯ СТРАНИЦА РАЗДЕЛА](#)
- [Синтаксис регулярных выражений](#)
- [Функция compile\(\) модуля re](#)
- [Флаги, используемые в функции re.compile\(\)](#)
- [Функция search\(\) модуля re](#)
- [Функция match\(\) модуля re](#)
- [Функция fullmatch\(\) модуля re](#)
- [Функция finditer\(\) модуля re](#)
- [Функция split\(\) модуля re](#)
- [Функция findall\(\) модуля re](#)
- [Функция sub\(\) модуля re](#)
- [Функция subn\(\) модуля re](#)
- [Функция escape\(\) модуля re](#)
- [Функция purge\(\) модуля re](#)
- [Исключение error\(\) модуля re](#)
- [Объект регулярного выражения Pattern модуля re](#)
- [Объект совпадения с шаблоном Match модуля re](#)

ХОЧУ ПОМОЧЬ
ПРОЕКТУ

