|  |  |  |
| --- | --- | --- |
| **Продолжение re, requests, urllib, webcrawler** | | |
| Этап занятия | Листинг кода | комментарии |
| Виды квантификаторов | <https://regex101.com/r/hLeEZX/3> | Как мы уже посмотрели в прошлом уроке, метасимвол + обозначает совпадение с одним или несколькими символами, т.е. хотя бы с одним символом, а иначе совпадения не происходит. Но что если требуется обнаружить совпадение с совершенно произвольным количеством символов: от нуля и больше?  Для этой цели служит метасимвол \*. Он указывает на совпадение с нулевым количеством символов или большим. |
| Применение подвыражений | <https://regex101.com/r/LDP4xv/2/> | Группирование подвыражений применяется в том случае, если вам нужно описать повторяющиеся символы. К примеру, нам нужно написать регулярное выражение, которое может отыскать нам валидные ip – адреса. Его можно представить следующим образом:  \d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}  Однако благодаря группировке подвыражений это можно написать более коротко:  (\d{1,3}\.){3}\d{1,3} |
| Просмотр вперед | <https://regex101.com/r/nrojHB/1> | Все регулярные выражения, которые рассматривались до сих порт, служили для обнаружения совпадений с текстом, но иногда требуются выражения для отметки позиции в совпавшем тексте, в отличии от самого текста. Для этой задачи служит позиционный просмотр – возможность выполнять просмотр вперед. |
| Просмотр вперед, вариант валидизации пароля | <https://regex101.com/r/qH6JRw/3> | Приведу примеру регулярного выражения, с помощью которого можно проводить валидизацию пароля, в котором должно быть от 6 до 25 символов, как минимум одна заглавная буква, одна строчная буква, и одна цифр, а также группа спецсимволов: |
|  |  |  |
| Библиотека requests | | |
| Bs4 | нету | Библиотека Requests – библиотека, которая представляет собой набор различных http функций, с помощью которых вы сможете как получать какую-либо информацию из интернета, так и отправлять ее ( причем с помощью это библиотеки вы сможете даже авторизироваться на различных сайтах). В прошлый раз мы воспользовались ей в связке с BeautifulSoup, чтобы получить информацию, а bs4 смог ее распарсить. Теперь мы узнаем больше ее возможностей. |
| API requests | request\_first.py | Давайте посмотрим на основные функции этой библиотеки:  r = requests.get(’<https://ru.wikipedia.org/wiki/Python>’)  print(r) - выведет http код 200(ок), если запрос был выполнен успешно.  print(r.text) – получить текст (содержимое) страницы. |
| http | <https://developer.mozilla.org/ru/docs/Web/HTTP> | Существует довольно много методов http, которые используются в современных web – приложениях. Если вы собираетесь связать свою жизнь с разработкой бекенд веб-приложений, вам стоит изучить этот вопрос максимально подробно, но так как мы прохожим на этом курсе Python, мы не будем останавливаться на этом подробно. Изучить http, на котором работает весь современный интернет, можно по данной ссылке:  <https://developer.mozilla.org/ru/docs/Web/HTTP> |
| Get запросы тренировка | requests\_get.py | Проверяем работу наших get запросов. Для этого есть замечательный сервис httpbin, вполне можно на немного потренироваться и остановиться поподробнее. |
| Get запросы практика | requests\_get\_yandex.py | Делаем запросы к Yandex и скачиваем оттуда все сслыки подряд. Конечно же, ищет Ozon на Украине( ну или что может предложить группа) |
| Запросы с помощью других библиотек | нет | В Python существует встроенная библиотек urllib, которая тоже может использоваться для получения информации о странице. Единственное, она сложная и включает в себя достаточно большое количество классов, в которых можно потеряться. Дополнительные материалы на эту тему:  <https://www.youtube.com/watch?v=LosIGgon_KM>  <https://khashtamov.com/ru/python-requests/> |
| Web-crawler | | |
| Теория web-crawler |  | Давайте напишем несложный crawler для википедии, который сможет выдавать нам случайные ссылки с каждой страницы. Для генерации мы воспользуемся интересной функцией random.seed.  Если не говорить о том, как работает генератор псевдослучайных чисел, то random.seed дает вам возможность поменять направления выдачи псевдослучайных чисел. Попробуйте написать рандом с random.seed и позапускать его в разные дни. |
| Работа с datetime | нет | Вообще тут нет лстинга, потому что время занятия будет заканчиваться. Можно попробовать немного поработать с различными методами datetime, но вообще мы будем проходить это позже. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |