Основы программирования в Python
Проект: документация
Стихеева Дарья

В данном разделе будет подробно описано, как работает программа. Для удобства код разделен на пронумерованные чанки.

В первом чанке мы загружаем необходимые для работы пакеты. Пакет requests нужен, чтобы мы могли собрать код с нужной html-страницы и сохранить его в отдельную переменную для дальнейшей работы. Функция BeautifulSoup из пакета bs4 позволяет превращает код со необходимой страницы в текст. Функция sleep из пакета time позволит делать остановки при парсинге данных, чтобы программа могла сделать вид, будто это человек переходит по всем ссылкам и сайт нас не заблокировал. Пакет pandas, который в нашей прграмме импортируется под именем pd, нужен для создания датафрейма и дальнейшей работы с ним. Наконец, пакет matplotlib позволит нам построить гистограммы по нужным переменным.

В чанке 2 в переменную url мы сохраняем ссылку на первую страницу с приключенческими настолками в виде строки. Далее в этом чанке создается список с анализируемыми страницами с подборками настолок. Для начала в этот список включается только первая страница. Затем при помощи цикла for, который проходится по номерам оставшихся страниц, в список добавляются и они. Делается это через f-строку, в которой i - номер необходимой страницы. По итогу мы получаем список из 7 страниц (нужно заметить, что остальные страницы не включены в анализ, поскольку товаров, представленных на них, нет в наличии).

В чанке 3 мы непосредственно собираем интересубщие нас данные со страниц из предыдущего списка. Для начала создаем пять пустых список, в которые будем сохранять ссылки на игры, их названия, цены, время игры и максимальное число игроков соответственно. Затем прогоняем кицл for для каждой страниц из списка page_urls. Внутри этого цикла создаем несколько переменных: с помощью метода get из пакета requests в переменную page сохраняется код с нужной страницы; данный код преобразуется в текст с помощью функции BeautifulSoup и сохраняется в переменную soup; в переменную info сохрянется весь код (преобразованный в текст), касающийся определенного товара на странице: это делается с помощью метода find_all, который в данном случе ищет все строчки, начинающиеся с тега div, в которых также class = product-item, и сохраняет их в список, на выходе мы получаем список из кусков кода, в котором каждый элемент посвящен отдельному товару на странице. Затем внутри нашего цикла программа запускает еще один цикл for, который ходит уже по элементам списка info и собирает информацию по кажому из товаров, представленных на странице. Внутри этого цикла в список urls добавляем ссылку на каждый товар: в куске кода с информацией о товаре программа находит первую строку, начинающуюся с тега a и class=name и с помощью функции get забирает ссылку, которая стоит после href=. В список names добавляем название игры: программа находит первую строку, начинающуюся с тега a и class=name и с помощью функции get забирает название, которое стоит после title=. В список prices добавляем цену настолки: программа находит строку, начинающуюся с тега span и class=price и забирает оттуда весь текст без пробелов, заменяя при этом ненужные символы на пустоту и переводя цену в целое число. Некоторые игры на сайте имеют цену в 100 бонусов, поскольку мы ими не располагаем, заменим эту информацию на 0 и потом удалим из датафрейма. В список times добавляем строку, в которой содержится информация о примерном времени игры: если ифнормации о нем нет, в список добавляется значение «Неизвестно», в обратном случае программа берет первую строку, начинающуюся с тега div и class=params__item time, и с помощью функции get забирает содержимое, которое стоит после title=. В список players добавляем строку, в которой содержится информация о максимальном числе игроков: если ифнормации о нем нет, в список добавляется значение «Неизвестно», в обратном случае программа берет первую строку, начинающуюся с тега div и class=params__item players, и с помощью функции get забирает содержимое, которое стоит после title=. В конце цикла мы также применяем функцию sleep, которая обязывает программу ждать (в нашем случае) 2 секунды после каждого прогона цикла, чтобы сымитировать человека.

В чанке 4 преобразуем список times, в котором находятся строки с информацией о времени: программа разбирвает каждый элемент на список по пробелам, берет оттуда предпоследний элемент (максимальное время игры) и преобразует его в целое число.

В чанке 5 преобразуем список players, в котором находятся строки с информацией о числе игроков: программа азбирвает каждый элемент на список по пробелам, берет оттуда последний элемент (максимальное число

игроков) и преобразует его в целое число.

В чанке 6 создаем словарь games, в котором будет собрана вся информация о каждой настолке. Для этого с помощью цикла for проходимя по каждому индексу элементов списка urls (это мог бы быть любой другой список, поскольку у них одинаковая длина) и добавляем в словарь ключ с названием игры и значение в виде списка с информацией о цене, времени, игроках и ссылкой.

В чанке 7 создаем из словаря датафрейм. Программа делает это с помощью функции Dataframe из пакета pandas и метода transpose (нам нужно, чтобы по столюцам были переменные, а по строкам - игры). Далее с помощью метода columns присваиваем столбцам названия. Теперь нам необходимо удалить из датафрейма пропущенные значения. Для переменной price мы кодировали их как 0, поэтому сохраняем в наш датафрейм только строки, где в колонке price нет значения 0. Ту же процедуру проделываем с пколонками time и players, только в них пропущенные значения были закодированы как «Неизвестно».

В чанке 8 программа считает среднее значение для колонки с ценой настолок. Для этого мы пользуемся методом для датафрейма из пакета pandas.

В чанке 9 программа считает среднее значение для колонки с временем.

В чанке 10 программа считает среднее значение для колонки с числом игроков.

В чанке 11 программа строит гистограмму по колонке с ценой настолок. Для этого нам необходимо было подключить пакет matplotlib. Внутри метода hist мы также задаем цвет гистограммы и ее название.

В чанке 12 программа строит гистограмму по колонке с временем.

В чанке 13 программа строит гистограмму по колонке с числом игроков.

В чанке 14 программа выбирает строки только с теми играми, которые удовлетворяют условиям, на которые мы согласны. Поскольку большая часть настолок имеет цену ниже 5-6 тысяч, рассмотрим все варианты с цной меньше 4000, чтобы настолки не были чересчур дорогими и при этом позволяли играть в них большой компании. Если взять цену меньше, игры могут оказаться либо слишком короткими, либо рассчитанными на маленькое число игроков. Средняя продолжительность игры составлять около 85 минут, при этом на гисограмме видно, что достаточно много значений превышают его, поэтому мы может выбрать игры дольше, например, 100 минут, не боясь, что по итогу получим всего одну-две опции на выбор. Такая же ситуация обстоит и с максимумом игроков: в среднем это число составляет 5, однако его превышает большое число игр. Мы же, согласно нашей содержательной задаче, ищем игры на компанию из 5 человек или больше, поэтому ограничим значение игроков так, чтобы оно превышало 4. Таким образом, в этом чанке программа выводит строки из датафрейма, которые удовлетворяют всем трем условиям одновременно. На выходе мы получаем датафрейм, в котором содержатся только подходящие нам игры. Всего их оказалось 12.