Логика функций backend.py:

по необходимым параметрам подключаются таблицы, все условия соединяются через AND

\* функция для создания JOIN части sql запроса - `join\_search\_tables(params)`

\* функций для создания WHERE части sql запроса - `join\_search\_conditions(conds)`

\* функция для поиска заглавных слов по созданному запросу - `search\_query(tables, conditions)`

\* функция для поиска текста для конкретного слова - `load\_page(word)`

\* функция для поиска по жанрам источника - `resource\_genre(param)`

на вход мы получаем список списков, где каждый элемент `[имя\_заполненного\_поля, [значение]]` или `[имя\_заполненного\_поля, [значение\_1, значение\_2, и т.д.]]` для параметров со множественным выбором

имя\_заполненного\_поля состоит из имени общего параметра и конкретного признака.

`source\_name` = в источниках смотрим название

`example\_date\_start\_y` = в примерах смотрим раннюю дату в годах

`join\_search\_tables(params)`

1. имеет список соответствия общего параметра в названии условия поиска и конкретных таблиц, необходимых для поиска

2. в соответствии со списком добавляет необходимые таблицы без повторений

3. возвращает строку `"JOIN.... JOIN..."`

`join\_search\_conditions(conds)`

1. имеет список соответствия условия поиска и конкретного поискового запроса WHERE, учитывает множественность выбора

2. склеивает запросы в один

3. возвращает строку `"WHERE ... AND ..."`

`search\_query(search\_tables, conditions)`

1. склеивает все в один запрос и ищет

2. выдает список `[head\_word, head\_word, head\_word]` без повторений, если встретились

`load\_page(word)`

1. принимает конкретное слово head\_word из списка, составленного ранее

2. возвращает список из `full\_text` для слова

т.к. некоторые поиски состоят из нескольких, как поиск по жанру источника, такие поиски будут отдельной функцией

`resource\_genre(param)` - поиск по жанрам источников

https://github.com/aRahna/rus\_dict\_hse\_project/tree/with-site