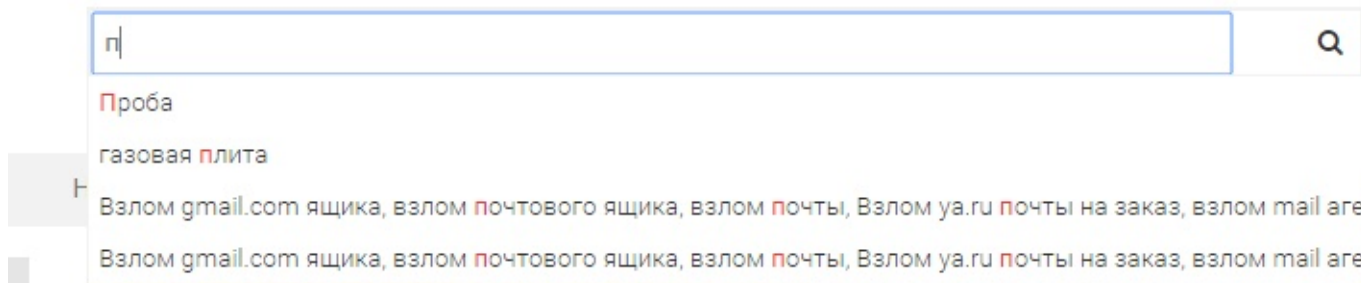


# Автодополнение в строке поиска

В этой статье описано, как реализовать такое автодополнение вариантов поиска в редакции Халява



такую функцию выполняет плагин Autocomplete | jQuery UI.

Свежую версию плагина можно скачать на Github'e:

<https://github.com/devbridge/jQuery-Autocomplete>

1. Необходимо добавить загруженные файлы скрипта и стилей на сервер `styles.css` и `jquery.autocomplete.min.js` в папки `css` и `js` соответственно.
2. Подключить эти файлы в основном макете сайта
3. В основной макет сайта в блок `head` добавить следующий код:

```
<script type="text/javascript">
  $(function() {
    //для строки поиска в шапке
    $('#search_text').autocomplete({
      serviceUrl: '/shop/?autocomplete=1', // Страница для обработки
запросов автозаполнения
      minChars: 1, // Минимальная длина запроса для срабатывания
автозаполнения
      maxHeight: 300, // Максимальная высота списка подсказок, в пикселях
      width: 200, // Ширина списка
      zIndex: 9999, // z-index списка
      deferRequestBy: 300, // Задержка запроса (мсек)
      // Дополнительные параметры
      onSelect: function (suggestion) {
        $(this).closest("form").submit();
      }
    });
  });
</script>
```

4. Для `<input>`, у которого должен выводиться выпадающий список добавляете `id="search_text"`
5. В ТДС, в Настройки страницы интернет магазина добавить код: (данный пример кода ищет варианты автозаполнения по имени и артикулу)

```

/**
 * Поиск товаров для автозаполнения
 *
 * @author Malinovskii Aleksandr, KAD Systems (©) 2015
 * @date 24-11-2015
 * Начало >>
 */

if (Core_Array::getGet('autocomplete'))
{
    $arResultJson = array();
    $sQuery=Core_Array::getGet('query');

    $arResultJson['query'] = $sQuery;
    $arResultJson['suggestions'] = array();

    //ищем совпадающие запросу имена
    $arResultName = Core_QueryBuilder::select(array('shop_items.name',
'value'))
        ->from('shop_items')
        ->where('shop_items.name', 'LIKE', '%'.$sQuery.'%')
        ->setAnd()
        ->where('shop_items.shop_id', '=', '5')->setAnd()
        ->where('shop_items.active', '=', '1')->setAnd()
        ->where('shop_items.deleted', '=', '0');

    //аналогично ищем артикулы
    $arResultMarking =
Core_QueryBuilder::select(array('shop_items.marking', 'value'))
        ->from('shop_items')
        ->where('shop_items.marking', 'LIKE', '%'.$sQuery.'%')
        ->setAnd()
        ->where('shop_items.shop_id', '=', '5')->setAnd()
        ->where('shop_items.active', '=', '1')->setAnd()
        ->where('shop_items.deleted', '=', '0');

    $arResultJson['suggestions'] =
array_merge($arResultName->execute()->asAssoc()->result(),
$arResultMarking->execute()->asAssoc()->result()) ;

    echo json_encode($arResultJson);
    exit();
}

/**
 * << Конец
 * @author Malinovskii Aleksandr, KAD Systems (©) 2015
 * @date 24-11-2015
 */

```