

# Script with data справка

## Описание программы

Программа "Script with data" представляет собой инструмент для [выполнения скриптов](#) с [подстановкой данных](#). Она разработана для [записи](#) и [воспроизведения](#) действий пользователя на компьютере, таких как перемещение мыши, клики мышью и нажатия клавиш клавиатуры.

Все действия пользователя записываются в виде [команд](#) для программы и могут быть [отредактированы](#) пользователем. Затем эти действия могут быть воспроизведены, позволяя компьютеру повторить все действия пользователя. Таким образом, программа позволяет автоматизировать выполнение рутинных задач, освобождая человека от их выполнения. Некоторые примеры использования включают отправку электронных писем, тестирование программ и другие подобные задачи.

## Состав программы

Программа состоит из 3 компонентов:

1. [Редактор скриптов](#) - отдельное окно, с помощью которого можно записать действия пользователя и преобразовать их в [список команд](#) (скрипт). Скрипт может быть набран в [ручном режиме](#), дополнен [вспомогательными командами](#) и настроен в соответствии с требованиями пользователя.
2. [Менеджер проектов](#) - окно, в котором отображаются [проекты](#) из [рабочей папки](#). Тут проектам присваиваются и переназначаются пользователем уникальные (в рамках [рабочей папки](#)) [коды для запуска](#). Создаются [штрих-коды](#) и [ярлыки](#) на рабочем столе для быстрого запуска.
3. [Окно быстрого запуска](#). Служит для [запуска скрипта](#) из [рабочей папки](#) по [коду](#) или посредством сканера [штрих-кодов](#).

## Отличительные возможности программы

Script with data является мощным инструментом автоматизации рутинных задач на компьютере. В этом разделе мы рассмотрим отличительные функциональные возможности данной программы. Среди них:

- [Запись и сверка пиксельных изображений](#): возможность записи изображений участков интерфейса приложений или веб-страниц позволяет программе решать проблему изменения внешнего вида интерфейса. [Сверка изображений](#) во время воспроизведения позволяет реагировать на изменения интерфейса для принятия решений о [выполнении действий](#).
- [Реакция на ошибки](#): программа обладает возможностью автоматического принятия решения на основе возникших ошибок.
- Вставка текстов в поля приложений или веб-страниц: данные для вставки могут быть [заранее записаны](#) в скрипт или [взяты из табличных файлов](#).

- [Организация циклов](#): возможность организации циклов команд на основе [нужного количества повторений](#) или [перебора записей файла](#).
- Переход по [меткам](#): программа поддерживает возможность перехода по меткам, что улучшает управление скриптом.
- [Организация подпрограмм](#): участок скрипта, который можно выполнить из разных частей программы.
- Гибкое [редактирование команд кликов мыши](#): программа предоставляет гибкий механизм редактирования команд кликов мышки, что позволяет избегать ошибок при выполнении скрипта.
- Сокращение некоторых комбинаций клавиш в отдельные команды: программа позволяет сократить [некоторые комбинации клавиш](#) в [отдельные команды](#) для более эффективного управления.
- Работа с буфером обмена: программа поддерживает возможность работы с буфером обмена через [комбинации клавиш](#).
- Набор скрипта в редакторе: возможность полностью [набрать скрипт в редакторе](#) или [записать действия пользователя](#) с последующим редактированием.

## Основные понятия

### Проект и скрипт

Скрипт - это основное понятие, которым оперирует программа. Он представляет собой не просто набор команд, а включает в себя также [настройки](#), [изображения элементов интерфейса](#) и [файлы с данными](#). В процессе [выполнения скрипта](#) может использоваться один [источник данных](#) (файл) или не использоваться вовсе. Однако в данной программе понятие "скрипт" почти не используется. Вместо этого используется понятие "проект".

[Проект](#) представляет собой папку с определенным именем, имя папки - это и есть имя проекта. Внутри папки находится файл со скриптом и настройками, имеющий то же имя, что и папка, и расширение json. Также в ней есть две дополнительных папки: "[data](#)", где хранятся [источники данных](#) для скрипта, файлы в формате xls, xlsx, csv, и "[elements\\_img](#)", где хранятся [изображения элементов интерфейса](#).

Таким образом, скрипт является основным содержимым проекта, который включает в себя дополнительные [настройки](#), [изображения элементов интерфейса](#) и [файлы с данными](#). Проект представляет собой единую структуру, облегчающую использование и управление скриптами в программе.

### Источник данных

Источник данных - это табличный файл в формате xls, xlsx или csv, который может быть [подключен](#) к программе для предоставления текстовых данных. Эти данные используются для заполнения полей форм или других элементов интерфейса. Механизм источника данных позволяет программе использовать один и тот же [скрипт](#) с разными данными, не изменяя сам скрипт. Для этого в табличном документе первая строка должна содержать названия полей. Скрипт будет использовать эти названия

для определения, из какого столбца брать данные. В процессе выполнения скрипта указатель будет перемещаться по строкам таблицы, чтобы показывать, из какой строки брать данные.

Переход по записям таблицы (перемещение указателя на следующую строку) может осуществляться автоматически в [цикле](#) или с помощью [специальных команд](#).

Положение указателей (для каждого поля свой) нигде явно не отображается.

Проект может содержать несколько источников данных, которые хранятся в папке "[data](#)". Однако, к одному [скрипту](#) может быть [подключен](#) только один источник данных одновременно.

## Рабочая папка

Концепция рабочей папки в программе - это папка на диске, в которой хранятся [проекты](#) программы. Программа хранит путь к [рабочей папке](#) в настройках. По умолчанию рабочей назначается папка программы, но пользователь может изменить место нахождения рабочей папки через [меню редактора](#) или [кнопкой](#) в [менеджере проектов](#). Этот путь используется как путь по умолчанию при [открытии](#) и [сохранении проектов](#). А также как место, в котором программа ищет проект, когда для его запуска используется [цифровой код](#) или [штрих-код](#). Этим способом скрипт может быть запущен из [Окна быстрого запуска](#) или из [Менеджера проектов](#).

В рабочей папке [Менеджер проектов](#) автоматически создает файл "[projects\\_list.json](#)", в котором хранит информацию обо всех [проектах](#) в [папке](#) и назначает каждому проекту и файлу в проекте [уникальный код](#), комбинация которых и используется для [запуска скриптов](#). Поэтому скрипт по коду может быть запущен из рабочей папки, только когда папка назначена таковой и [просканирована Менеджером проектов](#).

## Код запуска. Штрих-код

Код запуска - это комбинация номера [проекта](#) и номера [файла данных](#), которая служит для [запуска](#) конкретного [скрипта](#) с определенным [источником данных](#). Номера проектов и файлов состоят из двух цифр (от 01 до 99). Код запуска образуется из 2 пар, где первая пара обозначает номер [проекта](#), а вторая - номер [файла данных](#). Если скрипту не требуется [источник данных](#), вместо номера файла добавляются нули.

Коды генерируются и назначаются проектам автоматически [Менеджером проектов](#), там же они могут быть [переназначены](#) пользователем.

С помощью кода скрипт может быть [запущен](#) только из [рабочей папки](#) в [Окне быстрого запуска](#) или в [Менеджере проектов](#), где он вводится в [соответствующее поле](#).

Для ввода кода можно воспользоваться сканером штрих-кодов для сканирования [предварительно созданного](#) штрих-кода. [Штрих-код](#) может быть сгенерирован и сохранен в указанном месте на диске с помощью кнопки [Получить штрих-код](#) в [Менеджере проектов](#).

## Выполнение скрипта

Процесс, при котором программа выполняет заданный [скрипт](#). Воспроизводятся [записанные действия пользователя](#), такие как нажатия клавиш, клики мыши. Кроме того выполняются другие вспомогательные [команды](#).

Скрипт выполняется до получения команды остановки от пользователя, записанной в [скрипте](#) или пришедшей в результате завершения списка команд.

Скрипт может выполняться в разных режимах, в зависимости от компонента программы из которого [запускается](#):

- в скрытом режиме - окно программы скрывается на время выполнения ([Окно быстрого запуска](#), [Менеджер проектов](#)). После выполнения появляется.
- в фоновом режиме - окно не появляется вообще (запуск скрипта [ярлыком](#))
- в режиме редактирования - окно редактора не скрывается ([Редактор скриптов](#)).

Независимо от места запуска скрипта, пользователь может остановить его выполнение в любой момент дважды нажав левый Ctrl.

Выполнение скрипта может быть прервано диалоговым окном [Остановка скрипта для диалога](#).

## Режим разработчика

Режим разработчика включается и выключается через основное меню в [Редакторе скриптов](#): [Опции](#) -> [Режим разработчика](#).

В текущей версии программы опция влияет только на отображение [дополнительных кнопок](#) в [Окне быстрого запуска](#).

## Изображения элементов интерфейса

Для выполнения [кликов мыши](#) и в команде [Проверка изображения](#) программа делает скриншоты участков экрана (кнопки, иконки, ярлыки) и сохраняет их в папку проекта "elements\_img" (такая папка есть у всех [проектов](#)). При [воспроизведении скрипта](#) эти изображения используются для ориентации на экране, с этой целью выполняется [Проверка изображений](#).

## Проверка изображений

Это один из ключевых механизмов программы, позволяющий ей ориентироваться в интерфейсах приложений с которыми она работает. Проверка изображений используется во всех командах [кликов мыши](#) и в одноименной команде [Проверка изображения](#).

Принцип работа механизма основан на [изображениях элементов интерфейса](#), которые программа [сохраняет](#) при [записи скрипта](#). Например, [клик мыши](#) по кнопке создает [команду](#) с координатами курсора в момент клика и фрагментом изображения этой кнопки, который записывается в [папку изображений проекта](#). При [воспроизведении](#), программа сравнит сохраненное изображение с местом на экране, куда должен быть совершен клик мышью и на основании [результата сравнения](#) примет решение о совершении этого или другого [действия](#).

Проверка изображений осуществляется по следующей [базовой схеме](#):

1. Поиск имеющегося участка изображения на месте проверки с заданными координатами и размером.
2. Если элемент присутствует в области с заданными координатами, [совершается клик](#) или [иное действие](#) предусмотренное командой.
3. Если элемент не найден, проверка начинается сначала через 1 секунду. После 3 попыток выполняется следующий пункт.
4. Производится поиск [элемента](#) на всем экране, если он найден, [совершается клик](#) в координатах его местонахождения или выполняется [другое действие](#), основанное на информации о наличии искомого [изображения на экране](#).
5. Если [элемент](#) не найден, вырабатывается [ошибка изображения](#), на основании которой принимается решение о каком-либо действии.

Настройки такой проверки заданы по умолчанию для любого скрипта и описаны в разделе [Настройки скрипта](#). Они могут быть изменены пользователем для любого скрипта через пункт меню [Настройки скрипта](#). Также доступны [локальные настройки](#) схемы проверки для каждой команды использующей механизм проверки изображений.

## Реакция на ошибку

На обработке ошибок основан механизм принятия решений в программе.

Применяемые решения позволяют контролировать ход [выполнения скрипта](#).

Есть 2 типа ошибок которые контролируются программой:

- [Ошибки изображений](#) - к ним относятся все ошибки возникающие при работе с [изображениями элементов интерфейса](#). Это отсутствие изображения в папке [проекта](#) (такого быть не должно, но изображение можно удалить из [папки](#) вручную), отсутствие изображения элемента на экране при [проверке](#). На контроле этой ошибки основан механизм [Проверка изображений](#).
- Ошибки данных - это ошибки связанные с отсутствием данных в [источнике данных](#). С их помощью контролируется выполнение [циклов](#) основанных на получении данных из источника. И может быть построено ветвление скрипта, основанное на количестве элементов в полях таблицы.

В большинстве случаев ошибки обрабатывают команды, которые их генерируют. Для этого в их параметрах есть специальный виджет [Реакция на ошибку](#). В нем указывается что делать при возникновении ошибки:

- Остановить [выполнение скрипта](#)
- Игнорировать ошибку
- Вывести [диалоговое окно](#) для пользователя
- Перейти к [метке](#) в скрипте. При этом выборе нужно указать [метку](#).

Реакция на ошибки по умолчанию определена в [Настройках скрипта](#). И может быть изменена в [локальных настройках](#) отдельных команд и в скрипте, при выполнении, командами [Реакция на ошибку "Нет элемента"](#), [Реакция на ошибку "Нет данных"](#).

К командам обрабатывающим ошибки относятся все команды [проверяющие изображение](#). И команда [конца цикла](#).

Команда [Следующий элемент поля](#) может генерировать ошибку данных, при этом она не обрабатывает ошибку, она обрабатывается автоматически, в соответствии с [Настройками скрипта](#).

## Остановка скрипта для диалога

Это диалоговое окно появляется при [выполнении скрипта](#). Оно может быть вызвано командой скрипта [Диалоговое окно](#) или как [Реакция на ошибку](#).

В окне выводится причина остановки выполнения скрипта и предлагаются варианты продолжения:

- Перейти к [метке](#) или [блоку](#) - можно выбрать метку и скрипт продолжит выполняться с нее.
- Перезапустить - скрипт начнет выполняться с начала.
- Остановить - выполнение скрипта будет остановлено.
- Продолжить - скрипт продолжит выполняться дальше.

С помощью диалогового окна разработчики скрипта может напомнить о необходимости выполнения каких-либо действий, без которых продолжение скрипта не возможно.

## Запуск программы

Для запуска программы нужно открыть файл main.exe. При открытии файла без параметров откроется [Окно быстрого запуска](#).

Для открытия других компонентов программы нужно использовать ключи:

**main.exe -c** или **main.exe --code** - откроется [Окно быстрого запуска](#)

**main.exe -m** или **main.exe --manager** - откроется окно [Менеджера проектов](#)

**main.exe -e** или **main.exe --editor** - откроется окно [Редактора скриптов](#)

При установке программы стандартным инсталлятором на рабочем столе автоматически создаются 3 ярлыка для запуска программы в разных режимах.

## Запуск скрипта

Готовый скрипт может быть [запущен](#) несколькими способами:

1. в Редакторе скриптов кнопкой [Выполнить скрипт](#)
2. в [Окне быстрого запуска](#) нужно ввести код проекта в [поле ввода](#) и с помощью клавиши Enter или кнопки [Выполнить скрипт](#) запустить скрипт. При вводе [кода](#) посредством сканера штрих-кодов скрипт запустится автоматически.
3. в [Менеджере проектов](#) нужно набрать [код](#) проекта в [поле Поиск](#) и запуск по коду и запустить скрипт клавишей Enter или кнопкой “Выполнить скрипт”. При вводе [кода](#) посредством сканера штрих-кодов скрипт запустится автоматически.
4. [ярлыком](#) на рабочем столе.

## Параметры ярлыка для запуска скрипта

Можно создать 2 типа ярлыков. В обоих случаях используется ключ “-r” или “--run”, за которым следует 1 или 2 параметра.

Первый тип запускает скрипт из [рабочей папки](#), установленной в программе.

**main.exe -r, main.exe --run <код проекта>**

Пример: **main.exe -r --run 0101**

Если проекта с нужным [кодом](#) в [рабочей папке](#) не окажется, скрипт не [запустится](#).



Второй тип ярлыков запускает скрипт из любого места, в нем в качестве параметров нужно указать полный путь к [проекту](#) и если нужно, вторым параметром имя [файла данных](#).

**main.exe -r** или **main.exe --run** <путь к проекту> <имя файла данных>

Пример: **main.exe --run C:\Scripts\Script\_1 data.xlsx**

(путь и имя файла можно брать в кавычки, если в них есть пробелы)

В программе предусмотрена возможность создания ярлыков на Рабочем столе из [Менеджера проектов](#) кнопкой [Сохранить ярлык на рабочем столе](#). Она создает ярлыки второго типа.

## Редактор скриптов

Редактор [скриптов](#) открывается с помощью соответствующего [ярлыка](#) на рабочем столе. Также он может быть вызван кнопками в [Менеджере проектов](#) и в [Окне быстрого запуска](#) (кнопка видна, когда включен [Режим разработчика](#)) или комбинацией клавиш Ctrl+E.

При открытии редактора загружается последний [проект](#), который был [загружен](#) в редакторе до его закрытия. Если этот проект не доступен по каким-либо причинам или отсутствует файл, в котором хранится конфигурация редактора, то создается новый проект с именем "script\_<n>", где вместо <n> первый свободный номер проекта с таким именем в [рабочей папке](#). Проект сохраняется сразу при создании в текущей рабочей папке. В дальнейшем, для сохранения изменений необходимо нажать [Сохранить проект](#) в меню [Проект](#) или подтвердить сохранение при [закрытии окна редактора](#).

## Окно Редактора скриптов

Окно разделено на 2 части. В левой части находится [Список команд скрипта](#). Под списком [кнопки](#) записи, воспроизведения и остановки этих действий. В правой части 4 раздела: [Источник данных](#), [Редактор команд](#), [Редактор скрипта](#), [Информация](#). Еще ниже 2 кнопки для управления [Историей редактирования](#).

## Основное меню

### Проект

#### Новый проект

Откроет окно Создание нового [проекта](#), в котором нужно указать имя для нового проекта и папку, в которой он будет создан.

#### Открыть проект

Откроется окно выбора папки, нужно выбрать папку [проекта](#), который необходимо открыть.

#### Сохранить проект

Производит запись изменений в текущий [проект](#).

### **Сохранить проект как**

Делает копию текущего [проекта](#). Откроет окно Сохранить проект как, в котором нужно указать имя для нового проекта и папку, в которой он будет сохранен.

### **Переименовать проект**

Откроет окно Новое имя [проекта](#), в котором нужно указать новое имя для текущего проекта.

### **Удалить лишние изображения**

Эта опция служит для очистки папки с [изображениями](#) текущего проекта от тех, которые в данный момент не используются [командами](#) скрипта.

### **Просмотр изображений**

Откроет папку "elements\_img", в которой хранятся [изображения](#) для текущего [проекта](#).

### **Выход**

[Закроет окно редактора](#) скриптов.

## **Данные**

### **Подключить источник данных**

Откроет диалоговое окно выбора файла (xls,xlsx, csv). Если файл будет выбран вне папки файлов данных текущего [проекта](#), то он будет скопирован в проект и выбран как [источник данных](#), в разделе [Источник данных](#) появится список полей (столбцов) таблицы файла.

### **Сбросить источник данных**

Устанавливает указатели всех полей [источника данных](#) на начало. Эта операция выполняется автоматически при каждом [старте скрипта](#), если выбрана соответствующая настройка в [Настройках скрипта](#).

### **Отключить источник данных**

Раздел [Источник данных](#) будет очищен, скрипт не сможет получать данные из источника. При попытке получения данных будет возникать ошибка.

### **Открыть файл данных**

Откроет файл текущего [источника данных](#) в редакторе для редактирования табличных документов (например, Excel). Если файл будет изменен, то, чтобы обновления появились в программе, его нужно будет [подключить](#) повторно.

### **Открыть папку с данными**

Откроет папку "data" текущего [проекта](#), в которой хранятся его [источники данных](#).



## Опции

### Быстрый запуск

Открывает [Окно быстрого запуска](#)

### Менеджер проектов

Открывает [Менеджер проектов](#)

### Выбрать рабочую папку

Открывает диалоговое окно для выбора [папки](#). Настройка применяется для всей программы.

### Открыть рабочую папку

Открывает текущую [рабочую папку](#)

### Режим разработчика

Включен/Выключен. Включает или выключает [режим разработчика](#)

### Свернуть окно при записи

Да/Нет. Устанавливает, сворачивать ли окно редактора при включении [Режима записи](#)

### Настройки скрипта

Открывает окно, которое позволяет выбрать [настройки](#) по умолчанию для текущего [скрипта](#). Они могут меняться [командами](#) скрипта в процессе [выполнения](#), но это не меняет [настроек](#) по умолчанию.

### Менеджер проектов

Открывает [Менеджер проектов](#).

## Список команд скрипта

Первая строка списка всегда остается пустой.

Список отображает все [команды](#) скрипта. У левой границы списка расположен номер команды, далее идет название команды или ее [псевдоним](#), заданный пользователем. С помощью мыши в списке можно выбрать 1 команду или группу команд. Для выбора группы команд используются клавиши Ctrl и Shift.

Для списка команд работают сочетания клавиш для [редактирования скрипта](#), они описаны в разделе [Редактор скрипта](#). Так же работает сочетание Ctrl+A для выбора всех строк в списке.

## Кнопки управления скриптом

Под [списком команд скрипта](#) расположены 3 кнопки для [записи](#), [воспроизведения](#) скрипта и остановки этих операций.

- кнопка Запись скрипта - программа переходит в [Режим записи](#)
- кнопка Стоп - выходит из режимов записи и воспроизведения
- кнопка Выполнение скрипта - переходит в [Режим воспроизведения](#) скрипта.

## Источник данных

В разделе [Источник данных](#) перечисляются названия полей (столбцов) выбранного файла с данными. Если файл не выбран, то поле будет пустым.

## Редактор команд

В [Редакторе команд](#) выводятся [параметры](#) для выбранной в [Списке команд скрипта команды](#). При этом, краткая справка по команде выводится в раздел [Информация](#). Изменения [полей](#) в редакторе не приводят к немедленному изменению команды в [списке](#). В редакторе производится работа с копией выбранной команды, поэтому все изменения будут применены после сохранения с использованием [кнопок](#) Добавить команду или Изменить команду. Если изменения не сохранены, то они будут потеряны при выборе другой команды.

В Редакторе команд есть [постоянные элементы](#) и [изменяемые](#). Изменяемые элементы служат для отображения и изменения параметров команды, которая выбрана в [Списке команд скрипта](#) (параметры команд и выводимые для них элементы описаны в разделе [Команды](#)).

### Постоянные элементы:

- выпадающий список команд, в котором указано название редактируемой [команды](#). В нем можно выбрать или заменить текущую команду.
- поле Псевдоним под выпадающим списком команд. В нем можно указать описание (псевдоним) для редактируемой команды, оно будет отображаться в [Списке команд скрипта](#) для этой команды, вместо ее названия.
- кнопка Добавить команду - добавляет редактируемую [команду](#) в [Список команд скрипта](#) после выделенной команды, при этом в списке выделяется новая команда.
- кнопка “Изменить команду” - применяет изменения для выделенной в [Списке команд скрипта](#) команде.

### Изменяемые элементы. Параметры команд

Передача параметров [командам](#) осуществляется несколькими типами элементов. В этом разделе описаны элементы интерфейса и способы ввода данных в них. Информация о том, какие данные требуется команде выводится в разделе [Информация](#) при выборе команды в редакторе и описана в разделе [Команды](#) этого руководства.

#### Целое число

Обычное поле для ввода. Позволяет вводить только цифры с клавиатуры. Может иметь ограничение по количеству цифр и максимальному числу.

#### Строка

Обычное поле для ввода. Позволяет вводить любые символы с клавиатуры. Может иметь ограничение по количеству символов.

## Дробное число

Или число с плавающей точкой. Поле принимает только цифры и символ разделителя (запятая или точка). Может иметь ограничение по количеству символов.

## Выпадающий список

Нужно выбрать один вариант из предложенного списка.

## Реакция на ошибку

Это комбинированный виджет, позволяющий указать что делать в случае возникновения ошибки. Он состоит из 2 выпадающих списков. В первом нужно указать [реакцию на ошибку](#) из списка: "stop", "ignore", "dialog", "run". Второй активируется только когда в первом поле установлено значение "run" и позволяет указать название [метки](#) к которой нужно [перейти](#) скрипту.

## Выбор изображения

Элемент для выбора [изображения](#) состоит из поля с изображением и 2 кнопок под ним. В поле выводится выбранное изображение или символ, сообщающий о его отсутствии.

Клик по полю открывает диалоговое окно позволяющее выбрать изображение в формате "png". Диалоговое окно открывается на папке с изображениями текущего проекта "[elements\\_img](#)", но позволяет выбрать [изображение](#) и из другой папки, тогда оно будет скопировано в папку [проекта](#). Выбранное изображение отобразится на поле. Для удаления изображения используется кнопка Удалить изображение (справа под полем). Она открепляет изображение от команды, но не удаляет его физически. Для полного удаления из папки, после открепления изображения, нужно сохранить команду (если она ранее была сохранена с изображением), после чего [Удалить лишние изображения](#).

Левая кнопка под полем называется Скриншот, она позволяет получить изображение участка экрана и сохраняет его в папку изображений проекта.

## Инструкция для получения изображения участка экрана

После нажатия кнопки Скриншот активируется режим, при котором работают только клавиши Ctrl.

Нажатие правого Ctrl запоминает [координаты](#) центра участка экрана с которого нужно получить [изображение](#). Поэтому после первого нажатия Ctrl курсор можно увести от элемента, если он меняет свой стиль при наведении мыши и это мешает получить изображение в изначальном виде.

Повторное нажатие этой клавиши делает скриншот и записывает его в нужную папку. При этом оно будет проверено на уникальность и если такое изображение уже есть в папке, сохранено не будет.

*Обратите внимание. Перед получением скриншота желательно удалить из [папки](#) с изображениями неиспользуемые, чтобы новое изображение не совпало с похожим, уже имеющимся, что не позволит заменить старое. Для этого нужно использовать команду [Удалить лишние изображения](#) в меню [Проект](#). Если неудачное изображение используется [командами](#), его следует удалить из них чтобы оно удалилось из папки этой командой.*

Выход из режима получения скриншота осуществляется двойным нажатием левого Ctrl.

Если нужно получить только [координаты](#), а [изображение](#) не нужно (координаты будут записаны в соответствующие [поля](#) редактора команды), то после нажатия правого Ctrl первый раз следует нажать 3 раза подряд левый Ctrl.

### Кнопка ЕЩЕ

Открывает окно Дополнительных настроек команды. В нем можно активировать локальные настройки, которые заменят при [воспроизведение скрипта](#) действия основных [Настроек скрипта](#). Эти настройки в данной версии программы применяются для изображений и используются всеми командами [клика мыши](#) и командой [Проверка изображения](#). Они включают в себя индивидуальные настройки для [опций](#):

- Включить локальную проверку
- Зона локальной проверки
- Сколько секунд ждать, после 1 попытки
- Искать на всем экране
- Выполнить [действие](#) если изображение ...
- Какое действие выполнить
- [Сообщение](#) при выполнении действия

Описание опций можно посмотреть в разделе [Настройки скрипта](#). В этом списке присутствует только один пункт, не применяемый в основных настройках, это Выполнить действие если изображение .... Он позволяет инвертировать значение нахождения изображения. Другими словами, [действие](#) выполняется, если [изображение](#) не найдено, как это и предусмотрено в [основных настройках](#). Но можно выбрать и обратное, действие будет выполнено, если изображение будет найдено. Т.е. что считать ошибкой, присутствие или отсутствие изображения.

### Редактор скрипта

В этом разделе несколько кнопок позволяющие управлять [командами скрипта](#). Для выполнения какой-либо операций нужно выделить в [Списке команд скрипта](#) одну или несколько команд, а затем нажать на соответствующую кнопку. Нажатие кнопки выведет сообщение в раздел [Информация](#) о выполненном действии. А результаты операции будут отображены в [Списке команд скрипта](#).

Следует учитывать, что при выборе нескольких команд идущих не подряд при операциях перемещения или вставки они будут вставлены подряд, одна за другой, даже если в месте откуда они копируются между ними были другие команды.

Операции выполняемые кнопками также могут быть активированы сочетаниями клавиш, они указаны в списке рядом с названиями кнопок:

- кнопка Сдвинуть вверх (Ctrl+Up) - переместит выделенную команду или группу команд на 1 строку вверх
- кнопка Копировать команды (Ctrl+C) - запомнит выделенные команды, сообщит об этом в разделе [Информация](#) и будет ждать нажатия кнопки Вставить команды для завершения операции копирования
- кнопка Переместить команды (Ctrl+X) - запомнит выделенные команды, сообщит об этом в разделе [Информация](#) и будет ждать нажатия кнопки Вставить команды для завершения операции перемещения

- кнопка Сдвинуть вниз (Ctrl+Down) - переместит выделенную команду или группу команд на 1 строку вниз
- кнопка Вставить команды (Ctrl+V) - завершает операции копирования и перемещения, вставляя запомненные команды после выделенной строки
- кнопка Удалить команды (Del) - удаляет выделенные команды

## Информация

В разделе информация выводятся различные справочные сообщения, сообщения о начале или завершении каких-то действий, информация о ходе [выполнения скрипта](#), [ошибки](#) и другие сведения позволяющие понимать состояние [редактора](#). Сообщения выводятся разными цветами в зависимости от их важности.

## Кнопки управления историей

Кнопки Отменить и Вернуть позволяют отменить выполненные действия и вернуть их, если отмена сделана ошибочно или не привела к ожидаемому результату. История может хранить до 100 действий. Она хранится только при работе программы и не сохраняется на диск.

## Закрытие окна редактора

При закрытии окна редактора или при переходе к другому [компоненту программы](#) может появиться сообщение о необходимости [сохранить](#) изменения. Пользователь может подтвердить сохранение изменений, отказаться от них или остаться в [редакторе](#), нажав кнопку Отмена в диалоговом окне.

## Режим записи

Режим записи активируется нажатием [кнопки Запись скрипта](#) расположенной под [Списком команд скрипта](#). В разделе [Информация](#) появляется надпись Идет запись. Если в меню [Опции](#) в пункте [Свернуть окно при записи](#) установлено “Да”, то окно при начале записи будет свернуто автоматически. В любом варианте элементы окна [редактора](#) в этом режиме не работают.

Для завершения записи можно воспользоваться [клавишей Стоп](#), расположенной рядом с [клавишей Запись скрипта](#) в окне редактора или дважды нажать левый Ctrl.

В режиме записи происходит фиксирование [действий](#) пользователя с мышью и клавиатурой и запись их в виде команд в [Список команд скрипта](#). Кроме того, для команды [Проверка изображения](#) и всех [кликов](#) мыши в папку проекта "[elements\\_img](#)" записываются [изображения элементов интерфейса](#), на которых произведен клик, для дальнейшей проверки при [воспроизведении скрипта](#).

[Команды](#) записываются в строку, следующую за выделенной в [Списке команд скрипта](#). После записи каждой команды она автоматически выделяется.

## Как записываются действия?

Несмотря на то, что полный [список команд скрипта](#) достаточно обширный, основные команды фиксирующие [действия](#) клавиатуры и мыши не многочисленны.

Нажатие клавиши на клавиатуре записывается двумя командами: [Нажать клавишу](#), [Отпустить клавишу](#). В [параметрах](#) этой команды указывается название клавиши. [Клик](#) мышью записывается одной из 3 команд: [Клик левой кнопкой мыши](#), [Двойной щелчок мышью](#), [Клик правой кнопкой мыши](#). В [параметрах](#) команд указываются координаты щелчка и [изображение участка экрана](#) на котором был произведен щелчок. Полный перечень команд приведен в разделе [Команды](#).

## Специальные клавиши при записи

Для ввода отдельных команд используются стандартные комбинации клавиш или отдельные последовательности. Важно помнить, что для ввода комбинации клавиш первой должна быть нажата специальная клавиша (Ctrl, Shift, Alt, Win), затем буквенная клавиша. Порядок отпускания - не важен.

### Стандартные сочетания клавиш:

- Ctrl+C - копировать в буфер обмена. Добавляет команду [Копировать](#)
- Ctrl+X - вырезать в буфер обмена. Добавляет команду [Вырезать](#)
- Ctrl+V - вставить из буфера обмена. Добавляет команду [Вставить](#)
- Ctrl+A - выделить все. Добавляет команду [Выделить все](#)
- Alt+Shift - переключить раскладку клавиатуры. Добавляет команду [Сменить язык](#)
- Ctrl+T - новая вкладка. Добавляет команду [Новая вкладка](#)
- Ctrl+Tab - новая вкладка. Добавляет команду [Следующая вкладка](#)
- Alt+Tab - следующее окно. Добавляет команду [Следующее окно](#)
- Win+D (Ctrl+Alt+D для Linux) - свернуть все окна. Добавляет команду [Свернуть все окна](#)

### Клавиши для ввода команд:

- Ctrl (левый), Ctrl (левый) нажатие левой клавиши Ctrl подряд 2 раза - команду не вводит, останавливает [запись скрипта](#).
- Ctrl (правый), Ctrl (правый) нажатие правой клавиши Ctrl подряд 2 раза - добавляет команду [Проверка изображения](#).

## Ручной ввод и редактирование команд

Для более тонкой настройки [скрипта](#) можно воспользоваться ручным вводом [команд](#) или отредактировать и дополнить команды после [записи](#).

Такая настройка позволит определить [паузы](#) в различных местах скрипта, [циклы](#), [реакцию на ошибки](#) и др. Что позволит сократить объем скрипта, сделать его более универсальным, устойчивым к непредвиденным задержкам и ошибкам, предусмотреть варианты альтернативных действий в неожиданных ситуациях.

Редактирование команды происходит в [Редакторе команд](#). В нем можно задать [дополнительные настройки](#) для существующей команды или создать новую команду и вставить ее в любое место скрипта.

Список команд с описанием параметров для их настройки можно найти в разделе [Команды](#).

## Что можно настроить и изменить в командах и скрипте?

Чтобы понимать какие возможности предоставляет программа в базовом варианте прямой записи действий можно ознакомиться с разделом [Режим записи](#). Он работает с настройками скрипта по умолчанию, описанному в разделе [Настройки скрипта](#). И использует базовую схему [проверки изображений](#).

Для настройки команд и дополнения скрипта доступны следующие возможности:

- изменить названия команд в списке на [псевдонимы](#)
- установить [задержки](#) при исполнении команд, нажатий клавиш, кликов мыши.
- изменить базовую схему [проверки изображений](#), применив [локальные настройки](#).
- выбирать действия которые нужно выполнить при возникновении [ошибок](#) с изображением или данными.
- организовать выполнение действия в зависимости от наличия или отсутствия [изображения](#) в заданном месте экрана или на всем экране.
- вывести [диалоговое окно](#) с сообщением пользователю и возможностью выбрать дальнейший вариант действий.
- вывести текст в место где находится фокус (куда установлен курсор). [Текст](#) может быть задан заранее или взят из [Источника данных](#).
- подключить [источник данных](#).
- [переходить](#) по записям источника данных.
- организовать [повторное выполнение](#) участка скрипта заданное количество раз. Или принимать решение о повторе на основе доступных данных в [Источнике данных](#).
- выделить участок скрипта в [отдельный блок](#) и вызывать его исполнение из разных мест скрипта.
- поставить [метку](#) в любом месте скрипта с целью [перехода](#) к ней в результате возникновения [ошибок](#), по желанию пользователя из [диалогового окна](#) или без условий.
- сделать [паузу](#) или [остановить выполнение скрипта](#) в любом месте.

## Настройки скрипта

Настройки для всего [скрипта](#) задаются в отдельном окне программы вызываемом пунктом меню [Настройки скрипта](#). Частично они могут быть [переопределены](#) отдельными [командами](#) при [ручном редактировании](#) или во время [выполнения](#) скрипта. Применение локальных настроек не меняет состояние этих, они могут быть изменены только в окне настроек.

В списке описаны настройки и возможности их переопределения:

- Пауза между нажатием клавиш клавиатуры - задается [числом с плавающей точкой](#) только в окне настроек.
- Пауза после [клика](#) мыши - задается [числом с плавающей точкой](#) только в окне настроек.
- Пауза между [командами](#) (всеми) - задается [числом с плавающей точкой](#) только в окне настроек.
- Сбрасывать источник данных при [запуске скрипта](#) - при установленном флажке выполняется команда как через меню [Данные](#) - [Сбросить источник данных](#)



каждый раз когда запускается скрипт, это позволяет при каждом [выполнении скрипта](#) использовать данные из [источника](#) сначала.

- Включить локальную проверку - относится к схеме [проверки изображений](#), может быть изменена в [локальных настройках](#) команды. Определяет, нужно ли выполнить поиск изображения в области, размер которой задан настройкой [Зона локальной проверки](#) (сторона квадрата), при [выполнении скрипта](#).
- Зона локальной проверки (сторона квадрата) - относится к схеме [проверки изображений](#), может быть изменена в [локальных настройках](#) команды. Определяет размер квадрата в пикселях, в котором будет производиться поиск [изображения элемента](#) если включена проверка в настройке [Включить локальную проверку](#).
- Сколько секунд ждать, после 1 попытки - относится к схеме [проверки изображений](#), может быть изменена в [локальных настройках](#) команды. Если первая попытка проверки изображения в [зоне локальной проверки](#) показала, что оно отсутствует, будут предприняты еще попытки поиска в этой зоне, каждая, через 1 секунду. Изображение могло не прогрузиться, по причине, например, медленного интернета. Эта опция показывает, сколько попыток всего нужно совершить, а следовательно, сколько секунд будет ожидание появления изображения.
- Искать на всем экране - относится к схеме [проверки изображений](#), может быть изменена в [локальных настройках](#) команды. Включение этой опции заставит программу искать [изображение](#) на всем экране, если ожидание изображения в [зоне локальной проверки](#) не принесло результата. Если изображение будет найдено в другом месте, [ошибки](#) отсутствия изображения не будет, а программа получит координаты найденного изображения и совершит [клик мышью](#) в этом месте (если результатом проверки был клик).
- Какое действие выполнить если нет изображения - относится к [обработке ошибок](#), может быть изменено в [локальных настройках](#) команды и в скрипте. Для указания параметров использует специальный виджет [Реакция на ошибку](#).
- Какое действие выполнить если нет данных - относится к [обработке ошибок](#), может быть изменено в скрипте. Для указания параметров использует специальный виджет [Реакция на ошибку](#).
- Описание скрипта - справочная информация предоставляемая разработчиком [скрипта](#).

## Режим воспроизведения

Для начала [выполнения скрипта](#) из [редактора](#), нужно запустить его кнопкой [Выполнение скрипта](#), расположенной под [Списком команд скрипта](#). В разделе [Информация](#) при этом выведется сообщение Выполнение скрипта. Там же будут появляться другие сообщения информирующие пользователя о ходе выполнения. Скрипт начнет выполняться со строки которая выбрана в списке. Если выбрана первая, пустая строка, выполнение начнется с первой [команды](#).

Во время [выполнения скрипта](#) текущая команда подсвечивается в списке. Действия пользователя с мышью и клавиатурой, при работе скрипта, могут помешать его успешному выполнению.

Выполнение скрипта может быть прервано диалоговым окном [Остановка скрипта для диалога](#).

Для остановки скрипта можно воспользоваться [кнопкой Стоп](#) в окне [Редактора скриптов](#) или дважды нажать левый Ctrl.

## Команды

Команды скрипта можно разбить на 3 группы в зависимости от их назначения:

- [Команды действий](#)
- [Сервисные команды](#)
- [Команды настроек](#)

Для своей работы команды используют параметры, которые отображаются и могут быть изменены в [Редакторе команд](#). Параметр [Псевдоним](#) есть у всех команд, наличие [остальных параметров](#) зависит от самих команд и описано для каждой отдельно.

### Команды действий

К ним относятся команды, которые каким-либо образом взаимодействуют с интерфейсом операционной системы и программ.

В большинстве случаев они повторяют [записанные действия](#) пользователя. Но есть и такие, которые добавляются в скрипт вручную, при [редактировании](#).

#### Клик левой кнопкой мыши

Ввод: [Режим записи](#), [Редактор команд](#).

Параметры:

- два [целых числа](#) “x=”, “y=” которыми заданы координаты клика
- [изображение](#)
- [кнопка ЕЩЕ](#) для настройки работы с изображением

Действие: выполнит клик левой клавишей мыши в соответствии с логикой [Проверки изображений](#) или выполнит другие действия в соответствии с [Реакцией на ошибку](#).

#### Двойной щелчок мышью

Работает как команда “Клик левой кнопкой мыши”, только выполняет двойной клик левой клавишей мыши.

#### Клик правой кнопкой мыши

Работает как команда “Клик левой кнопкой мыши”, только выполняет двойной клик правой клавишей мыши.

#### Нажать клавишу

Ввод: [Режим записи](#), [Редактор команд](#).

Параметры:

- [клавиша](#), которая должна быть нажата

Действие: имитирует нажатие (только нажимает, не [отпускает](#)) выбранной клавиши.

#### Отпустить клавишу

Ввод: [Режим записи](#), [Редактор команд](#).

Параметры:

- [клавиша](#), которая должна быть отпущена

Действие: отпускает (клавиша должна быть [нажата](#) ранее) выбранную клавишу.

### Вывести текст

Ввод: [Редактор команд](#).

Параметры:

- [текст](#) до 500 символов

Действие: напечатает указанный текст в месте, где установлен курсор.

### Вывод из поля

Ввод: [Редактор команд](#).

Параметры:

- [имя поля](#) в таблице ([источнике данных](#))
- к скрипту должен быть [подключен Источник данных](#).

Действие: напечатает текст из указанного поля в месте, где установлен курсор.

### Сочетания клавиш

Группа команд без специальных параметров.

Ввод: [Режим записи](#), [Редактор команд](#).

Параметры: без параметров.

Действие: имитирует набор на клавиатуре [Стандартных сочетаний клавиш](#).

## Сервисные команды

Эти придают гибкость [скрипту](#) и добавляют новые возможности.

### Проверка изображения

Ввод: [Режим записи](#), [Редактор команд](#).

Параметры:

- два [целых числа](#) “x=”, “y=” которыми заданы координаты центра проверяемого изображения
- [изображение](#)
- [кнопка ЕЩЕ](#) для настройки работы с изображением

Действие: определит, есть ли в месте с указанными координатами требуемое [изображение](#). Не выполнит никаких действий, в соответствии с логикой [Проверки изображений](#) или выполнит действия в соответствии с [Реакцией на ошибку](#).

### Следующий элемент поля

Ввод: [Редактор команд](#).

Параметры:

- [имя поля](#) в таблице (источнике данных)
- к скрипту должен быть подключен [Источник данных](#).

Действие: перемещает указатель на следующую строку (запись) для указанного поля.

### Цикл по полю

Ввод: [Редактор команд](#).

Параметры:

- [имя поля](#) в таблице (источнике данных)
- к скрипту должен быть подключен [Источник данных](#).

Действие: служит началом блока команд, который повторяется заданное количество раз. Конец блока определяется командой [Конец цикла](#). Количество итераций (повторений) зависит от количества записей (строк) в указанном поле. Блок команд выполнится для каждой записи.

## Цикл

Ввод: [Редактор команд](#).

Параметры:

- [целое число](#)

Действие: служит началом блока команд, который повторяется указанное количество раз. Конец блока определяется командой [Конец цикла](#).

## Конец цикла

Ввод: [Редактор команд](#).

Параметры: без параметров.

Действие: служит окончанием блока команд, который повторяется заданное количество раз. Начало блока определяется командами [Цикл по полю](#) или [Цикл](#). Для цикла по полю переводит указатель поля в [Источнике данных](#) к следующей записи. Для простого цикла меняет счетчик. После отработки последней итерации выполняются команды следующие за этой. Если команда встретится без объявленного ранее начала цикла - она будет проигнорирована.

## Блок команд

Ввод: [Редактор команд](#).

Параметры:

- [имя блока](#) (метка)

Действие: служит началом блока команд, который выполнится при переходе к его метке. Конец блока определяется командой [Конец блока](#). Переход к метке осуществляется командой [Выполнить](#), как [Реакция на ошибку](#) или из [диалогового окна](#). Если команда встречается по ходу скрипта, блок будет обойден, осуществится переход к команде следующей за командой [Конец блока](#). Если [Конец блока](#) не будет обнаружен, то эта команда будет просто проигнорирована.

## Конец блока

Ввод: [Редактор команд](#).

- Параметры: без параметров.

Действие: служит окончанием блока команд, который выполнится при переходе к его метке. Начало блока определяется командой [Блок команд](#). После этой команды скрипт продолжит [выполняться](#) от команды вызвавшей блок. Если команда встретится без объявленного ранее начала блока - она будет проигнорирована.

## Метка

Ввод: [Редактор команд](#).

Параметры:

- [имя](#) (метка)

Действие: не выполняет действий. Служит меткой, к которой может быть совершен переход командой [Выполнить](#), как [Реакция на ошибку](#) или из [диалогового окна](#).

## Выполнить

Ввод: [Редактор команд](#).

Параметры:

- [имя](#) (метка)

Действие: совершает переход к метке заданной командами [Блок команд](#) или [Метка](#).

## Диалоговое окно

Ввод: [Редактор команд](#).

Параметры:

- [текст](#) до 500 символов

Действие: выведет [диалоговое окно](#) с указанным текстом и будет ожидать реакции пользователя.

## Пауза

Ввод: [Редактор команд](#).

Параметры:

- [число с плавающей точкой](#) (секунды)

Действие: остановит [выполнение скрипта](#) на указанное время.

## Конец скрипта

Ввод: [Редактор команд](#).

- Параметры: без параметров.

Действие: завершит [выполнение скрипта](#). В раздел [Информация](#) выведет сообщение "Пользователь остановил выполнение скрипта".

## Команды настроек

Временно меняют [настройки скрипта](#) заданные по умолчанию.

## Реакция на ошибку "Нет элемента"

Ввод: [Редактор команд](#).

Параметры:

- [реакция на ошибку](#)

Действие: изменит одноименную [настройку](#) на указанную до конца [выполнения скрипта](#) или до нового переназначения.

## Реакция на ошибку "Нет данных"

Ввод: [Редактор команд](#).

Параметры:

- [реакция на ошибку](#)

Действие: изменит одноименную [настройку](#) на указанную до конца [выполнения скрипта](#) или до нового переназначения.

## Менеджер проектов

Менеджер проектов служит для создания различных способов [запуска скриптов проектов](#). Он работает только с проектами находящимися в [Рабочей папке](#). При каждом запуске сканирует [проекты](#), проверяет и обновляет информацию о них в [Файле с описанием проектов](#).

Автоматически генерирует [коды запуска](#) скриптов, позволяет редактировать их, получать [штрих-коды](#), создавать [ярлыки](#) на рабочем столе для запуска. В Менеджере проектов предусмотрена возможность изменения [Рабочей папки](#), поиск и запуск скрипта с подключенным [Источником данных](#) по коду или с использованием сканера штрих-кодов.

Менеджер проектов открывается с помощью соответствующего [ярлыка](#) на рабочем столе. Также он может быть запущен через [меню](#) Редактора скриптов и в [Окне быстрого запуска](#), [кнопка](#) видна, когда включен [Режим разработчика](#).

## Окно менеджера проектов

Окно [Менеджера проектов](#) состоит из Списка проектов и ряда управляющих элементов под ним.

### Список проектов

В списке отображаются все [проекты](#) находящиеся в текущей [Рабочей папке](#). Каждая строка - это отдельный проект, в ней указаны: [код запуска](#), дата создания, дата последнего изменения, имя проекта.

Если проект имеет [Источники данных](#), то перед кодом запуска будет отображаться указатель, клик на котором (или двойной клик по строке проекта) откроет их список.

Список источников отобразится под строкой проекта. В нем указаны: [код запуска](#) и имя файла Источника данных.

Проект в списке выбирается кликом мыши. Выбор строки файла подразумевает выбор проекта, которому он принадлежит, с подключением указанного файла в качестве [Источника данных](#).

### Элементы управления Менеджера проектов

Для выполнения действий с выбранным в [Списке проектов](#) проектом и некоторых других операций, служат поля ввода и кнопки расположенные под списком.

### Номер проекта или файла данных

Поле ввода для изменения номера [проекта](#) или источника данных, из которых формируется [Код запуска](#). В нем отображается двузначный номер выбранного в списке объекта, который может быть изменен пользователем. Если пользователь вводит номер который уже принадлежит другому проекту или файлу в проекте, то его номер

меняется на указанный, а объект которому принадлежал этот номер, получает первый свободный номер.

Подтверждение изменения номера совершается клавишей Enter или расположенной справа кнопкой [Изменить код](#).

### **Изменить код**

Кнопка для подтверждения изменения [номера](#) проекта или [Источника данных](#).

### **Получить штрих-код**

Кнопка открывает диалоговое окно сохранения [штрих-кода](#) для запуска выбранного в [Списке проектов](#) скрипта. Его изображение будет сохранено в формате “png” в указанном месте на диске.

### **Сохранить ярлык на рабочем столе**

Кнопка создает на рабочем столе ярлык для запуска выбранного в [Списке проектов](#) скрипта. Имя ярлыка составляется из слова “Старт “ и имени проекта с большой буквы.

### **Открыть в редакторе**

Кнопка используется для перехода в [Редактора скриптов](#). Редактор открывается с загруженным [проектом](#) и подключенным [Источником данных](#), если они были выбраны в [Списке проектов](#). Если проект не был выбран, то редактор загрузится с последним открытым проектом. Для открытия Редактора скриптов также может быть использовано сочетание клавиш Ctrl+E.

### **Открыть окно быстрого запуска**

Кнопка совершает переход к [Окну быстрого запуска](#).

### **Открыть рабочую папку**

Открывает текущую [Рабочую папку](#).

### **Выбрать рабочую папку**

Открывает диалоговое окно для выбора [папки](#). Настройка применяется для всей программы.

### **Выполнить скрипт**

Начнет [Выполнение скрипта](#) с подключенным или нет [Источником данных](#) из [проекта](#) который в данный момент выбран в [Списке проектов](#).

### **Поиск и запуск по коду**

Поле ввода для поиска проекта по [коду запуска](#). Допускает ввод 4 цифр. [Проект](#) с введенным кодом запуска выбирается в [Списке проектов](#) и может быть запущен клавишей Enter в окончании ввода, кнопкой [Выполнить скрипт](#) или автоматически, если для ввода использовать сканер [штрих-кодов](#).



## Файл с описанием проектов

Файл "projects\_list.json" создается [Рабочей папке](#) при первом запуске [Менеджера проектов](#) с использованием этой папки в качестве [рабочей](#). В дальнейшем, при каждом запуске [Менеджер проектов](#) обновляет информацию в файле при обнаружении изменений в [проектах](#) Рабочей папки.

Файл содержит информацию обо всех проектах в [Рабочей папке](#) и их [Источниках данных](#), хранит уникальные коды для них.

Если в [выбранной](#) Рабочей папке нет файла с описанием проектов, [запуск скрипта](#) из [Окна быстрого запуска](#) будет не возможен.

## Окно быстрого запуска

Окно быстрого запуска предназначено для [запуска скрипта](#) посредством сканером [штрих-кодов](#). Но скрипт можно запустить и вводом [кода запуска](#) с клавиатуры. Окно всегда отображается поверх всех окон. Оно представляет минимум функций для запуска скриптов и перехода к другим [компонентам программы](#).

## Элементы управления в Окне быстрого запуска

Слева всегда отображается поле ввода кода, когда окно активно это поле имеет фокус ввода. Справа расположены 2 или 4 кнопки. Кнопки перехода в [Редактор скриптов](#) и [Менеджер проектов](#) отображаются только если включен [Режим разработчика](#).

### Поле ввода кода запуска

В поле вводится [код запуска](#) проекта, [скрипт](#) которого должен быть [выполнен](#). Допускает ввод 4 цифр. Скрипт может быть запущен клавишей Enter в окончании ввода, кнопкой [Выполнить скрипт](#) или автоматически, если для ввода использовать сканер [штрих-кодов](#).

### Выполнить скрипт

Начнет [Выполнение скрипта](#) с подключенным или нет [Источником данных](#) из [проекта](#), [Код запуска](#) которого в данный момент введен в [Поле ввода кода запуска](#).

### Перейти в редактор скриптов

Кнопка не отображается если выключен [Режим разработчика](#). Однако, в любом случае, для перехода в [Редактор скриптов](#) можно воспользоваться сочетанием клавиш Ctrl+E. Кнопка используется для перехода в [Редактора скриптов](#).

### Перейти в Менеджер проектов

Кнопка не отображается если выключен [Режим разработчика](#). Используется для перехода в [Менеджер проектов](#).

## Закрыць

Кнопка закрывае [окно](#), завершае праграму.