**Введение**

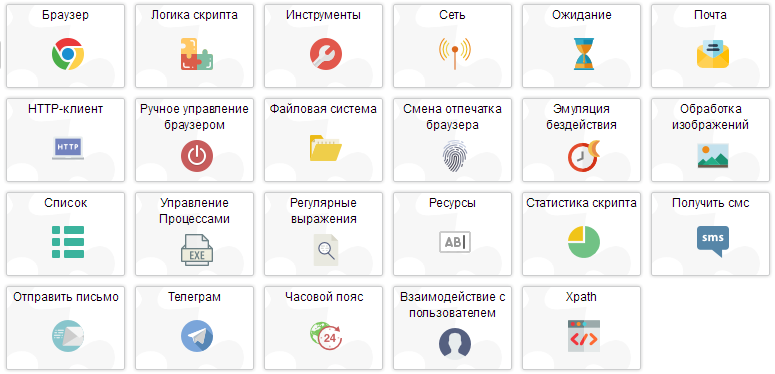
Browser Automation Studio предназначена для автоматизации рутинных действий в интернете и на локальном сервере. В основе программы лежит интеграция с прогрессивным и быстрым браузером Google Chrome.

Все возможности софта заключены в готовых модулях, позволяющих решать конкретные задачи и заточенные на метагпрограммирование (создание утилит при помощи таких же утилит). Конструкторная парадигма позволяет решать проблемы логически последовательно и не требует глубоких познаний в софтостроении.

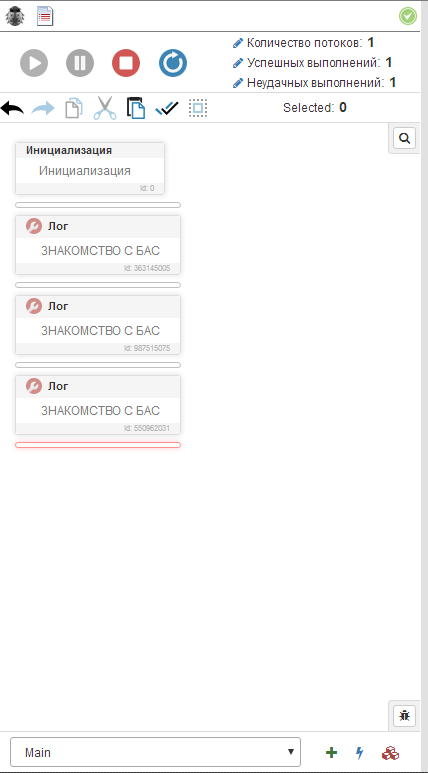
Это делает BAS монстром среди родственных программ, от которых он отличается упрощенным подходом к решению многоплановых задач. Уже сейчас программе пророчат славу небезызвестной системы автоматизации ZennoPoster, в отличии от которой BAS распространяется под лицензией shareware, где функционал программы по своей сути доведен до кондиции личного пользования, но не ограничен гибким функционалом. Тем не менее, если лавры создателей и коммерсантов успешных утилит не дают вам покоя, придется купить лицензионную версию, которая позволит оперативно обновлять исходный код и защищаться от несанкционированного доступа «любознателей».

В сообществе единомышленников BAS блоки кода принято называть скриптом, хотя, однажды взглянув на проект, заключенный в xml-документе, у вас не повернется язык называть Си-образный код скриптом.

Все экшены программы распределены по большим разделам, критерием деления которых является функциональное назначение (см. рисунок 1).



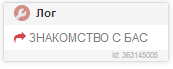
Так, например, чтобы выставить задержку между действиями скрипта, потребуется исследовать раздел «ожидание».

Блочная структура кода легко отслеживается при помощи редактора кода, который изображен на рисунке. Таким образом вы можете удалять, копировать, вырезать, перемещать – проще говоря, редактировать блоки, словно текст.

Запустить всё это дело можно, конечно же, в несколько потоков сразу, установив, соответственно, количество возможных ошибок и выполнений.

Кроме того, в самом низу редактора располагается менеджер функций и переменных, что удобно в отслеживании их маршрута и деятельности по ходу работы скрипта. Опытный программист скажет вам, как порой бывает сложно управлять всем этим богатством, когда имеется внушительный набор.

Административная панель имеет возможность управлять процессом запуска, паузы, остановки и перезапуска скрипта (чем-то напоминает плеер).

Здесь так же имеется программный секундант, который сопроводит скрипт от начала до конца и покажет, на каком этапе выполнения кода находится программа. Такой секундант обозначен красной стрелочкой на блоке:

**Типы данных**

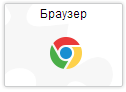
Если Вы программист, то представленная в этой главе информация не будет полезной, если дилетант, ознакомьтесь как можно более вкрадчиво.

Фундаментом любого программирования является обработка данных на основе его типа и, поскольку BAS, претендует на звание отдельной среды программирования, то его концепция устроена по аналогии с известными языками.

Конструктор BAS целиком и полностью устроен так, что, прежде чем использовать определенную функцию программы, туда нужно ввести какое-либо значение, а затем явно указать к какому типу данных это значение принадлежит. Запомним эту цепочку:

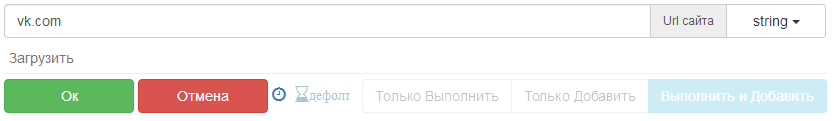
**ВЫБОР ФУНКЦИИ = ВВОД ЗНАЧЕНИЯ = УКАЗАНИЕ ТИПА ДАННЫХ**

Давайте, к примеру, пошагово попробуем зайти на сайт vk.com благодаря этой цепочке. Итак, для начала зайдем в раздел *«браузер»,* а затем выберем интересующую нас функцию *«загрузить»:*

**1**

2

Вторым шагом введем значение vk.com. В завершение выставим тип данных *«string»* (строка) и нажмем зеленую кнопку *«ОК»*.



**Поздравляю, только что мы исполнили с вами функцию, ввели для нее значение и явно указали тип данных!**

|  |
| --- |
| **Основные типы данных для ввода значения:**   1. *Текст;* 2. *Числа;* 3. *Код.* |

**ОСНОВНЫЕ ТИПЫ ДАННЫХ**

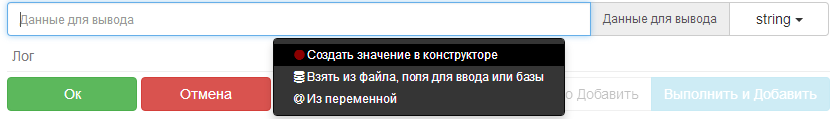
**Текст.** Система естественных и специфических символов, выраженных в буквенных и знаковых значениях. Проще говоря, все, что вы можете набрать на клавиатуре, относится к этому типу данных. Текст впоследствии можно обрезать, склеивать, удалять, заменять и т.д. Этот тип данных называется string. Сюда позволено вводить любой **ТЕКСТ**, переменные вида **[[VAR]]**, а так же ресурсы вида **{{resourse}}**. Широко известна практика кодеров помещать текст в кавычки, в типе данных string этого не требуется. Но, если вы используете тип данных «код» (expression), текст всё же придется брать в кавычки.



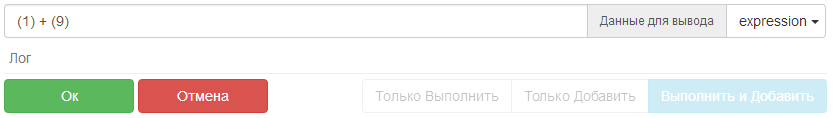
**Числа.** Цифирная система знаков, обрабатываемая BAS в парадигме математической модели исчисления. С помощью этого типа данных впоследствии можно проводить любые арифметические действия: сложение, вычитание, умножение. Этот тип данных называется int. Сюда позволено вводить только целое или десятичное число с точкой.



**Код.** Интегрированная в ядро программы совокупность языков, призванная дополнить недостающий функционал BAS. В число таких языков входит Java Script и node.js. Стоит сказать, что типы данных string и int задают значение явно и не могут их обрабатывать самостоятельно. Иными словами, если вам нужно обрезать текст или сложить числа, то придется прибегнуть к типу данных «код» (expression). Автор предусмотрел тот случай, что вы не знаете Java Script и создал легкий конструктор кода, который можно вызвать левой кнопкой мыши по полю для ввода данных.



Путем нехитрых манипуляций можно получить заветный код, который позволит сложить числа 1 и 9, причем тип данных автоматически сменится на «expression».



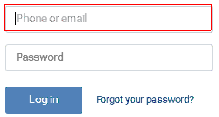
Тип данных *«код»* принимает свой синтаксис, основанный на JS, любой текст вида **«ТЕКСТ»**, переменные вида **[[VAR]]**, а так же ресурсы вида **{{resourse}}**.

**Селекторы**

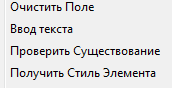
В браузерной автоматизации программисту всегда приходится работать с внешними данными: формы, текст, картинки на сайте и т.д. Каждый элемент веб-сайта имеет свое уникальное название, как имена у людей. Эти имена называются *селекторы.*

|  |
| --- |
| **Основные типы селекторов:**   1. *>CSS>* 2. *>MATCH>* 3. *>XPATH>*   **Второстепенные типы селекторов:**   1. *AT* 2. *NONE* |

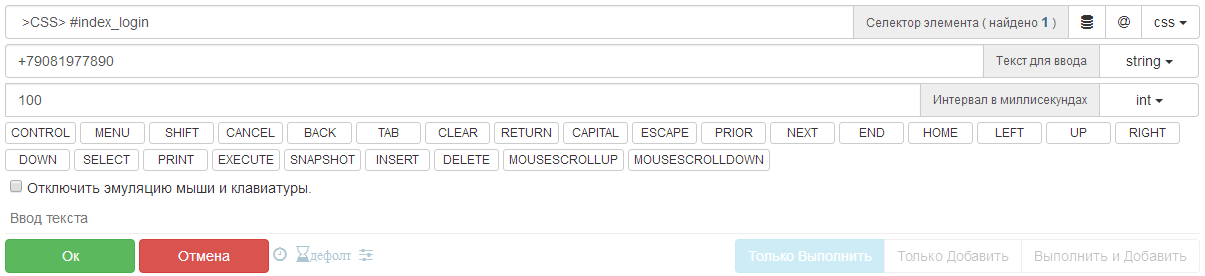
Зайдем на сайт vk.com, как это делать вы уже знаете из предыдущей главы. Нас интересует поля для ввода логина и пароля и кнопка авторизации.



Поводя мышкой, вы легко обнаружите, что текущий селектор, на который указывает курсор, подсвечивается красной рамочкой. Выделите поле для ввода логина и нажмите правую кнопку мыши. Выпадает список возможных действий для данного селектора. Выберем «ввод текста».

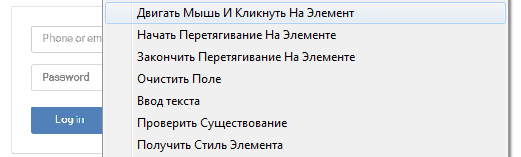


Перед нами возникают три поля. Второе – для ввода значения и указания типа данных, нам уже знакомо. Введем туда номер телефона и укажем тип данных «string». Обратим внимание на первое поле, в котором нам автоматически сгенерировалось имя селектора **>CSS>**. Правее мы можем изменить тип селектора. В конце манипуляций нажмем «выполнить и добавить» и «ОК».

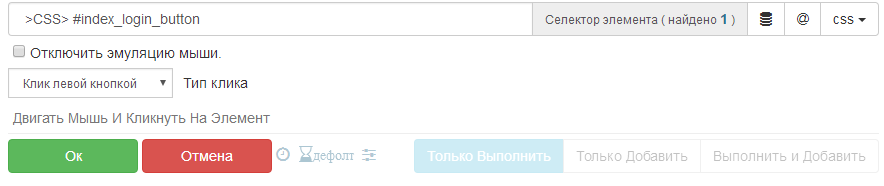


Все те же самые операции проделаем и с полем ввода для пароля.

Дальше, когда у нас заполнены поля, наводим мышку на кнопку авторизации и выбираем «двигать мышь и кликнуть на элемент».



Возникает поле с именем селектора. Тип выберите тот, какой вам больше нравится. В конечном счете нажмите «выполнить и добавить» и «ОК».



Поздравляю, только что мы авторизовались в ВК, использую для этого знания о типах данных, селекторах и типах селекторов!