

Стек-Консоль

Материал из Stack

Стек-Консоль - инструмент для разработчиков наподобие консоли в веб-браузерах. При отладке избавляет от необходимости вставлять в код вызовы методов **Сообщить** и **Отладить**. В **Стек-Консоли** удобнее, чем в **Калькуляторе**, выполнять функции и вычислять выражения. **Стек-Консоль** есть в Стекe версии **201704** и выше. Она работает как в **Muzzle** так и в **JavaScript**.

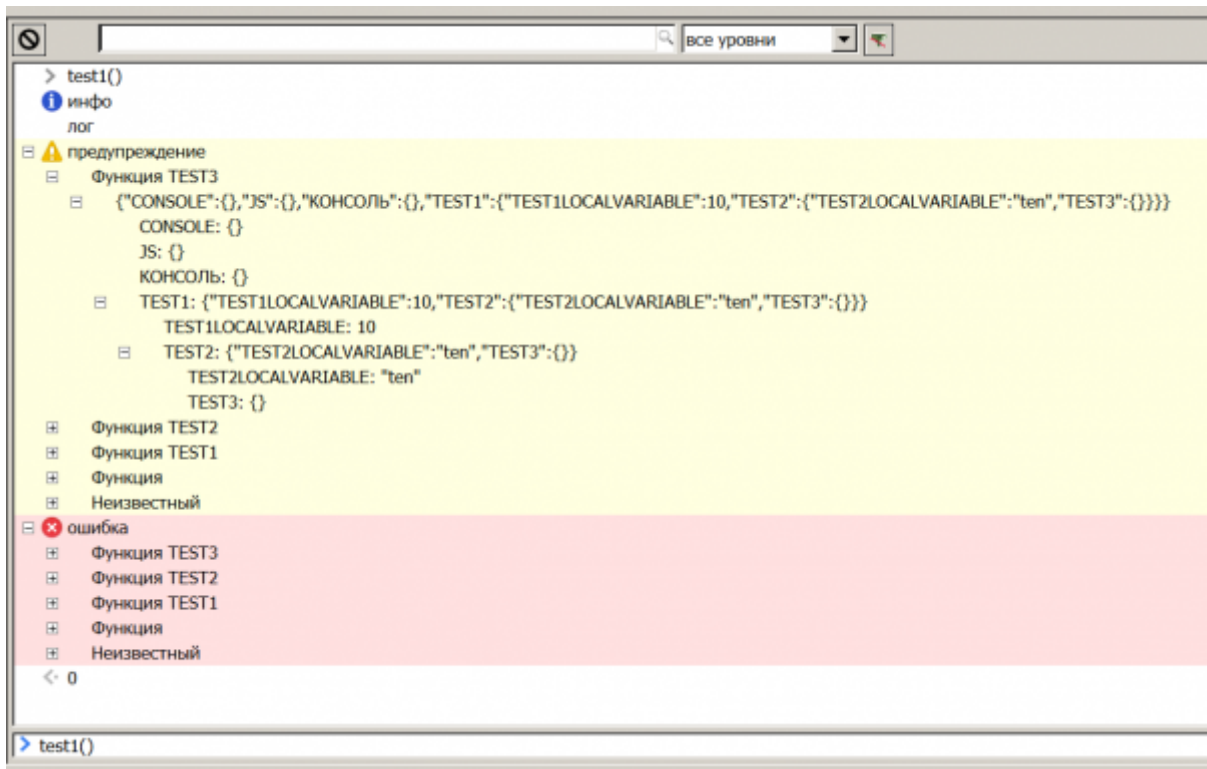
Содержание

- 1 Описание
- 2 Вывод сообщений
- 3 Интерактивное выполнение выражений
- 4 Методы
 - 4.1 console.open / консоль.открыть
 - 4.2 console.close / консоль.закрыть
 - 4.3 console.clear / консоль.очистить
 - 4.4 console.log / консоль.лог
 - 4.5 console.info / консоль.инфо
 - 4.6 console.warn / консоль.предупреждение
 - 4.7 console.error / консоль.ошибка
 - 4.8 console.group / консоль.группа
 - 4.9 console.groupEnd / консоль.группаСтоп
 - 4.10 console.trace / консоль.трейс
 - 4.11 console.count / консоль.подсчитать
 - 4.12 console.time / консоль.таймер
 - 4.13 console.timeEnd / консоль.таймерСтоп
 - 4.14 console.assert / консоль.проверить
 - 4.15 console.file / консоль.файл
 - 4.16 console.fileEnd / консоль.файлСтоп

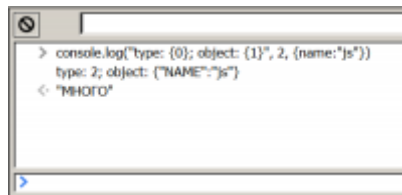
Описание

Чтобы открыть **Стек-Консоль**, нажмите сочетание клавиш **Ctrl+Shift+I** или выполните в калькуляторе **console.open()**.

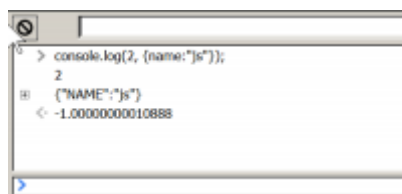
Стек-Консоль состоит из области вывода сообщений, области ввода выражений, фильтра сообщений и двух кнопок - очистки и закрытия:



У всех функций вывода сообщений (`console.info`, `console.log`, `console.warn` и `console.error`) есть две формы вызова. Первая форма, когда первым параметром идет строка с форматом, а за ней объекты подставляемые в этот формат. Результатом будет одна строка:



Вторая форма, когда выводятся произвольные объекты через запятую. В этом случае каждый объект будет выведен с новой строки:

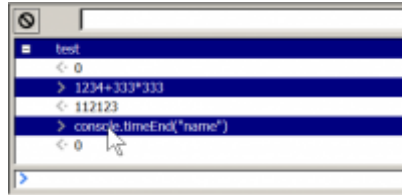


Также сообщение в консоль можно вывести из кода самого стека:

```
GetStackConsole()->AddMessage(scERROR, error);
```

Чтобы очистить область вывода сообщений, нажмите на кнопку в верхнем левом углу консоли или выполните **console.clear()**.

Чтобы скопировать сообщения, выберите необходимые сообщения и нажмите **Ctrl+C**:



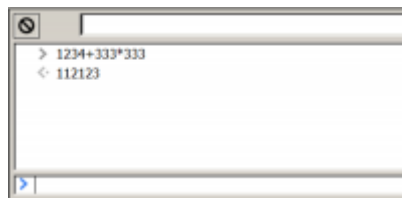
```
test
< 0
> 1234+333*333
< 112123
> console.timeEnd("name")
< 0
```

С помощью **console.file** можно направить вывод сообщений в файл.

Интерактивное выполнение выражений

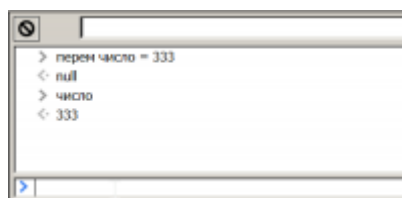
Стек-Консоль предоставляет простую интерактивную среду для выполнения выражений также известную как **Read-Eval-Print Loop (REPL)** - цикл **чтение — вычисление — вывод**.

Чтобы выполнить выражение в **Стек-Консоли**, встаньте в область ввода выражений, введите выражение и нажмите **Enter**. В области вывода сообщений появится введенное выражение, а через некоторое время результат его выполнения:



```
> 1234+333*333
< 112123
```

Стек-Консоль имеет свой собственный исполнитель выражений. Это позволяет создавать в консоли локальные переменные и потом использовать их. После обновления скриптов по **Ctrl+F12** исполнитель консоли будет пересоздан, и поэтому локальные переменные в консоли сотрутся:



```
> переменное = 333
< var
> число
< 333
```

Стек-Консоль запоминает некоторое количество выполненных выражений. Эта история сохраняется даже после закрытия стека. Чтобы получить доступ к истории выполненных выражений, встаньте в область ввода выражений и нажмите стрелку вверх. Для навигации по истории используйте стрелки вверх и вниз.

Методы

Чтобы вызвать метод консоли, нужно обратиться к объекту **console** или **консоль**. Ко всем методам консоли можно обращаться, используя русские или английские имена.

console.open / **консоль.открыть**

console.open() открывает консоль в пользовательском интерфейсе.

console.close / консоль.заккрыть

console.close() закрывает консоль в пользовательском интерфейсе.

console.clear / консоль.очистить

console.clear() удаляет накопившиеся сообщения.

console.log / консоль.лог

console.log(obj1 [, obj2, ..., objN]) - выводит один или более объектов в консоль. Каждый объект выводится отдельным сообщением.


console.log(msg [, subst1, ..., substN]) - выводит сообщение с подставленными параметрами. Формат у сообщения такой же как в кодовой функции **ФорматСтроки**. Например,

```
console.log("второй: \${1}\"; первый: \${0}\\"", "строка", 123) // второй: "123"; первый: "строка"
```


console.info / консоль.инфо

Тоже самое что **console.log** плюс перед каждым сообщением идет иконка .

console.warn / консоль.предупреждение

Тоже самое что **console.log** плюс перед каждым сообщением идет иконка , у сообщений бледно желтый фон и есть стек вызовов.

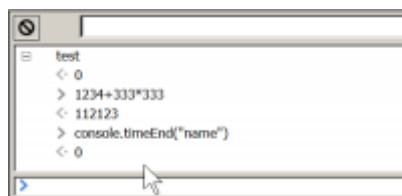
console.error / консоль.ошибка

Тоже самое что **console.log** плюс перед каждым сообщением идет иконка , у сообщений бледно красный фон и есть стек вызовов.

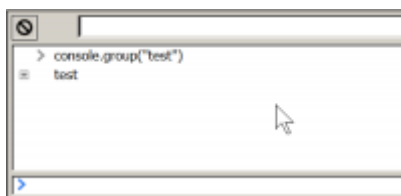
console.group / консоль.группа

console.group(name) - создает новую группу с именем **name** для сообщений консоли. Все входящие сообщения будут попадать в эту группу до тех пор пока не будет вызван метод **console.groupEnd()**.

Так выглядит группа в развернутом виде:



Так в свёрнутом:



console.groupEnd / консоль.группаСтоп

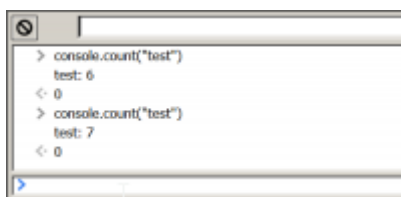
console.groupEnd() - закрывает группу, если таковая существовала.

console.trace / консоль.трейс

console.trace() - выводит в консоль стек вызовов.

console.count / консоль.подсчитать

console.count(name) - выводит количество раз, которое был вызван **count** с этим конкретным **name**. Этот метод можно использовать для подсчета количества вызовов определенной функции:

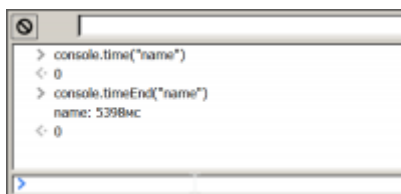


console.time / консоль.таймер

console.time(name) - запускает таймер с именем **name**, который можно остановить вызвав **console.timeEnd(name)**.

console.timeEnd / консоль.таймерСтоп

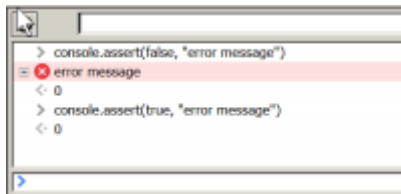
console.timeEnd(name) - останавливает таймер с именем **name** и выводит в консоль сообщение с количеством миллисекунд прошедших с запуска таймера:



console.assert / консоль.проверить

console.assert(condition, obj1 [, obj2, ..., objN]) или **console.assert(condition, msg [,**

subst1, ..., substN] - тоже самое что и `console.error`, только сообщение об ошибке пишется, если **condition** является ложью. Если **condition** истинно, ничего не происходит:



console.file / консоль.файл

console.file() - начинает выводить сообщения в лог файл по умолчанию. **В консоли сообщения не отображаются.**

console.file(path) - тоже самое, только сообщения выводятся в файл **path**.

console.fileEnd / консоль.файлСтоп

console.fileEnd() - прекращает запись сообщений консоли в файл.

Источник — [«http://wiki.stack-it.ru/index.php?title=Стек-Консоль&oldid=6955»](http://wiki.stack-it.ru/index.php?title=Стек-Консоль&oldid=6955)

-
- Последнее изменение этой страницы: 11:40, 2 апреля 2018.

