

FireBug Console API

`console.log(object[, object, ...])`

Логгирует сообщение в консоль. Может принимать любое количество параметров. Если их передано больше одного, то на консоли они будут отображены в одной строке и разделены пробелами.

```
console.log("строка_1", переменная_1, ...);
```

Может использоваться для вывода, форматированного в printf-стиле.


```
console.log("Это строка: %s, а это целое: %d", "текст", 12);
```

| Спецификатор | Значение |
|--------------|--------------------------|
| %d, %i | Целое число |
| %f | Число с плавающей точкой |
| %s | Строка |
| %o | Ссылка на объект |

`console.debug(object[, object, ...])`

Логгирует в консоль сообщение и ссылку на строку, где был вызван.

`console.info(object[, object, ...])`

Логгирует в консоль сообщение, отмеченное иконкой , и ссылку на строку, где был вызван.


console.warn(object[, object, ...])

Логгирует в консоль предупреждение, отмеченное иконкой , и ссылку на строку, где был вызван.

console.error(object[, object, ...])

Логгирует в консоль сообщение об ошибке, отмеченное иконкой , и ссылку на строку, где был вызван.

console.assert(expression[, object, ...])

Проверяет истинность выражения, переданного в качестве аргумента. При отрицательном результате логгирует в консоль пользовательское сообщение, если оно было передано вторым аргументом, или «Assertion Failure» по-умолчанию, ссылку на строку из которой был вызван. Сообщение отмечается .

console.dir(object)

Логгирует в консоль все свойства переданного объекта в таком же формате, как представлены свойства объектов документа на вкладке «DOM».

console.dirxml(node)

Логгирует в консоль содержимое узла XML или DOM элемента, в формате, аналогичном вкладке «HTML». Каждый узел является ссылкой на самого себя во вкладке «HTML».

console.trace()

Логгирует в консоль стек вызовов функций на момент исполнения, где был вызван.

Если какая-либо из вызванных функций принимает аргументы, они тоже отображаются на консоли.

По клику на имя функции Вы будете перемещены на вкладку «Scripts», где можно увидеть ее код. По клику на аргумент функции Вы будете перемещены на вкладку «DOM», где можно увидеть его значение, если это объект, то все его свойства.

console.group(object[, object, ...])

Открывает в консоли блок, куда будут выводиться последующие сообщения, что позволяет их группировать. Блок получает заголовок, составленный из значений полученных аргументов.

Для закрытия блока используется метод `.groupEnd()`.

console.groupEnd()

Закрывает блок, открытый вызовом `.group()`, или самый последний из открытых блоков, если их было открыто несколько (вложенность).

console.time(name)

Создает таймер, с именем `name`. Используется для определения времени выполнения блока кода.

console.timeEnd(name)

Останавливает таймер, запущенный вызовом `.time()` и логирует значение прошедшего времени.

console.count([title])

Логгирует количество раз, которое был вызван, и сообщение, переданное в качестве аргумента.

console.profile([title])

Устанавливает начала профайла с заголовком `title` для блока кода.

Профайлинг позволяет узнать:

- какие функции/методы были вызваны в данном блоке кода
- сколько раз
- среднее/максимальное/минимальное время выполнения каждой, если она была вызвана более одного раза
- процент времени, затраченный на ее выполнение относительно времени, затраченного на выполнение всего заданного блока кода.

console.profileEnd()

Завершает процесс профайлинга и логирует отчет.