

Управляющие последовательности ANSI

Управляющие символы ANSI ([англ. ANSI escape code](#)) — символы, встраиваемые [\[править | править код \]](#) в текст, для управления форматом, цветом и другими опциями вывода в [текстовом терминале](#). Почти все [эмуляторы терминалов](#), предназначенные для отображения текстового вывода с удалённого компьютера (за исключением [Microsoft Windows](#)) и для отображения текстового вывода локального программного обеспечения, способны интерпретировать по крайней мере некоторые управляющие последовательности ANSI.

Содержание [\[скрыть\]](#)

- [История](#)
- [Поддержка](#)
 - [2.1 Windows и DOS](#)
- [Элементы последовательности](#)
- [CSI-коды](#)
- [Цвета](#)
- [См. также](#)
- [Примечания](#)
- [Ссылки](#)

История [\[править | править код \]](#)

Почти все разработчики текстовых терминалов создавали свои, несовместимые с другими версиями терминалов, управляющие последовательности, выполняющие какие-либо преобразования над содержимым терминала; например, перемещение курсора на произвольную позицию на экране. Сложные библиотеки, такие как [termcap](#), были созданы для того чтобы в программах можно было использовать единый [API](#), независимый от реализации текстового терминала. Кроме того, большинство терминалов требовали отправки чисел (таких как строка и столбец) в виде бинарных значений символов; для некоторых языков программирования и операционных систем, не имеющих внутренней поддержки [ASCII](#), часто было сложно или невозможно поставить в соответствие числу корректный символ.

Первым стандартом **управляющих символов ANSI** стал [ECMA-48](#), принятый в 1976 году. Это было продолжение серии стандартов кодировок, первым из которых был ECMA-6 1961 года — 7-битный стандарт от которого происходит ASCII. ECMA-48 обновлялся несколько раз, текущая версия — пятая, 1991 года. Он также принят [ISO](#) и [IEC](#) как стандарт [ISO/IEC 6429](#). Название «Управляющие символы ANSI» ([англ. «ANSI escape sequence»](#)) появилось в 1981, когда ANSI принял ECMA-48 как стандарт **ANSI X3.64** (позднее, в 1997 году, отозван^{[\[1\]](#)}).

Первым популярным терминалом, поддерживающим эти последовательности, стал [VT100](#), представленный в 1978 году^{[\[2\]](#)}. Среди многочисленных «клонов» VT100, самым ранним и наиболее популярным стал наиболее доступный [Zenith Z-19](#), вышедший в 1979 году^{[\[3\]](#)}. Их популярность постепенно приводила к появлению большого количества программного обеспечения (особенно [BBS](#)), поддерживающего управляющие последовательности, а это, в свою очередь, привело к тому, что практически все новые терминалы и эмуляторы терминалов поддерживали их.

Поддержка [\[править | править код \]](#)

Большинство [эмуляторов терминалов UNIX-подобных](#) операционных систем (таких как [xterm](#) или [OS X Terminal](#)) интерпретируют управляющие последовательности ANSI. [Linux-консоль](#) (видна когда [X Window System](#) не запущена) также интерпретирует их. Эмуляторы терминалов [Microsoft Windows](#), предназначенные для текстового вывода из внешних источников (последовательный порт, модем или сокет), также интерпретируют их. Некоторая поддержка для текстового вывода локальных программ в Windows часто осуществляется при помощи альтернативных [процессоров команд](#), например [Take Command Console](#) (прежде 4NT) от *JP Software*, ANSI.COM от *Michael J. Mefford*, ansicon от *Jason Hood*.

Многие [консольные приложения](#)^[англ.] [Unix](#) (например, [ls](#), [grep](#), [Vim](#), [Emacs](#)) могут генерировать их. Некоторые утилиты ([tput](#)^[англ.]), низкоуровневые ([termcap](#), [terminfo](#)) и высокоуровневые ([curses](#)) библиотеки, выводят их.

В операционной системе OS/2 от IBM, начиная с версии 3.0 и старше, включая ECs, поддержка ANSI в полноэкранном и оконном режиме консоли включена по умолчанию.

Windows и DOS [\[править | править код \]](#)

[MS-DOS 1.0](#) не поддерживала ни ANSI, ни какие-либо другие управляющие последовательности. Интерпретировались только несколько управляющих символов ([CR](#), [LF](#), [BS](#)), что не давало возможности создавать какие-либо полноэкранные приложения. Каких-либо эффектов дисплея можно было достичь только вызовами [BIOS](#) (или, гораздо чаще, прямыми манипуляциями над оборудованием [IBM PC](#)).

[MS-DOS 2.0](#) предоставляла возможность установки [драйвера](#) для поддержки управляющих последовательностей ANSI — [де-факто](#) стандартом был [ANSI.SYS](#), но использовались и другие ([ANSI.COM](#)^[4], [NANSI.SYS](#)^[5]). Медлительность и тот факт, что драйвер не был установлен по умолчанию, привели к тому, что они почти не использовались; программное обеспечение продолжало напрямую манипулировать оборудованием для получения необходимых эффектов в текстовом терминале. ANSI.SYS и подобные драйверы продолжали использоваться в Windows 98, и даже в Windows ME в режиме native DOS (полноэкранный текстовый режим). ANSI.SYS также работает на системах семейства Windows NT в 16-битном режиме для устаревших программ при помощи [NTVDM](#).

[Win32-консоль](#)^[англ.] вообще не поддерживала управляющие последовательности ANSI до версии Windows 10 "Threshold 2". Программное обеспечение может управлять консолью посредством [ioctl](#)^[англ.]-подобных системных вызовов [консольных API](#) ↗, чередующихся с текстовым выводом. Некоторые программы внутренне интерпретируют управляющие последовательности ANSI из текстового ввода и переводят их в соответствующие системные вызовы.

Элементы последовательности [\[править | править код \]](#)

Управляющие последовательности начинаются с символа [ESC](#) ([ASCII](#): 27 / [0x1B](#) / 033). Для двусимвольных последовательностей вторым символом может быть символ [ASCII](#) в диапазоне от [0x40](#) до [0x5F](#) (от @ до _). Однако большинство последовательностей включают более двух символов и начинаются с символов [ESC](#) и [. Совокупность символов [ESC](#) и [называют [CSI](#) или **Control Sequence Introducer** (также Control Sequence Initiator). Последним символом такой последовательности может быть [ASCII](#)-символ в диапазоне от [0x40](#) до [0x7E](#) (от @ до ~).

Существует также односимвольный [CSI](#) ([ASCII](#): 155 / [0x9B](#) / 0233). [CSI](#) вида [ESC](#) + [используется чаще, чем односимвольная альтернатива (см. также [C0 and C1 control codes](#)). Только [ESC](#) + [распознаются устройствами, поддерживающими лишь [ASCII](#) (7-битные байты) или устройствами, поддерживающими 8-битные байты, но использующие управляющие символы в диапазоне [0x80](#) - [0x9F](#) для других целей. В

терминалах, использующих кодировку [UTF-8](#), обе формы занимают 2 байта (**CSI** в UTF-8: `0xC2`, `0x9B`), но **ESC** + [выглядит разборчивей.

Хотя некоторые кодировки и используют несколько байтов для кодировки одного символа, далее в статье мы будем рассматривать только ASCII-символы, которые представляются одним байтом.

CSI-коды [\[править | править код \]](#)

Структура большинства управляющих последовательностей ANSI выглядит следующим образом:

CSI [символ(-ы) режима] *n1* ; *n2*... [закрывающий(-ие) символ(-ы)] буква

Последний байт (**буква**), изменённый **символами режима** и **закрывающими символами**, определяет команду. Числа ***n1***, ***n2***, ... — необязательные параметры. Если их значения не указаны, то по умолчанию обычно используются 0 или 1, в зависимости от операции. Если не указаны следующие числа, то и точки с запятыми тоже могут быть пропущены.

Последним байтом, формально, может быть любой символ от 64 до 126 (от `0x40` до `0x7e`, ASCII: от @ до ~), он может быть изменён впереди стоящими промежуточными байтами в диапазоне от 32 до 47 (от `0x20` до `0x2f`).

Символ двоеточия (`0x3a`) не может содержаться в последовательности. Он был оставлен для будущей стандартизации, поэтому любая последовательность, содержащая двоеточие, игнорируется.

Хотя наличие нескольких символов режима и нескольких закрывающих символов разрешено, это не используется.

Некоторые управляющие последовательности ANSI (неполный список)

Код	Название	Эффект
CSI <i>n</i> A	CUU — Cursor Up	Перемещает курсор на <i>n</i> (по умолчанию 1) позиций в указанном направлении. Если курсор уже находится с краю окна терминала, ничего не происходит.
CSI <i>n</i> B	CUD — Cursor Down	
CSI <i>n</i> C	CUF — Cursor Forward	
CSI <i>n</i> D	CUB — Cursor Back	
CSI <i>n</i> E	CNL — Cursor Next Line	Перемещает курсор в начало <i>n</i> -ой (по умолчанию 1-й) строки снизу относительно текущей.
CSI <i>n</i> F	CPL — Cursor Previous Line	Перемещает курсор в начало <i>n</i> -ой (по умолчанию 1-й) строки сверху относительно текущей.
CSI <i>n</i> G	CHA — Cursor Horizontal Absolute	Перемещает курсор в столбец <i>n</i> .

CSI $n ; m$ H	CUP — Cursor Position	Перемещает курсор в строку n , столбец m . Значения n и m по умолчанию 1 (левый верхний угол), если указан только один аргумент без точки с запятой, считается что это n . Например, последовательность CSI ;5H синонимична CSI 1;5H, CSI 17;H синонимична CSI 17H и CSI 17;1H.
CSI n J	ED — Erase Data	Очищает часть окна терминала. Если n равно нулю (или отсутствует), очищает всё от курсора до конца окна терминала. Если n равно единице, очищает всё от курсора до начала окна терминала. Если n равно двум, очищает всё окно терминала (и перемещает курсор в левый верхний угол в MS-DOS ANSI.SYS). В случае n равного 2, вместо очистки может произойти просто создание новых строчек так чтобы предшествующее содержимое терминала вытеснилось вверх.
CSI n K	EL — Erase in Line	Удаляет часть строки. Если n равно нулю (или отсутствует), очищает всё от курсора до конца строки. Если n равно единице, очищает всё от курсора до начала строки. Если n равно двум, очищает всю строку. Положение курсора не меняется.
CSI n S	SU — Scroll Up	Прокручивает текущую страницу терминала вверх на n (по умолчанию 1) строк. Новые строки добавляются снизу, строки в верхней части текущей страницы затираются. (не ANSI.SYS)
CSI n T	SD — Scroll Down	Прокручивает текущую страницу терминала вниз на n (по умолчанию 1) строк. Новые строки добавляются сверху, строки в нижней части текущей страницы удаляются. (не ANSI.SYS)
CSI $n ; m$ f	HVP — Horizontal and Vertical Position	Перемещает курсор в строку n , столбец m . Значение по умолчанию обоих параметров 1. Тот же эффект, что и CUP.
CSI $n [;k]$ m	SGR — Select Graphic Rendition	Установка SGR параметров , определяющих цвета текста и фона. После CSI могут находиться ноль или более параметров, разделённых точками с запятыми. Если параметры не указаны, CSI m ведёт себя как CSI 0 m (reset / normal).
CSI 6 n	DSR — Device Status Report	Сообщает текущее положение курсора в формате ESC[n ; m R, где n — строка, m — столбец. (Может не работать в MS-DOS)
CSI s	SCP — Save Cursor Position	Сохраняет положение курсора.
CSI u	RCP — Restore Cursor Position	Восстанавливает положение курсора.
CSI ?25l	DECTCEM	Скрывает курсор.
CSI ?25h	DECTCEM	Делает курсор видимым.

SGR (Select Graphic Rendition) параметры

Код	Эффект	Примечание
0	Reset / Normal	выключение всех атрибутов
1	Жирный или увеличить яркость	
2	Блёклый (уменьшить яркость)	Не везде поддерживается
3	Курсив: вкл.	Не везде поддерживается. Иногда обрабатывается как инверсия.
4	Подчёркнутый: один раз	
5	Мигание: Медленно	менее 150 раз в минуту
6	Мигание: Часто	MS-DOS ANSI.SYS; 150+ в минуту; не везде поддерживается
7	Отображение: Негатив	инвертирует или обращает; меняет цвета фона и текста
8	Скрытый	Не везде поддерживается.
9	Зачёркнутый	Символы разборчивы, но помечены как удалённые. Не везде поддерживается.
10	Основной (по умолчанию) шрифт	
11–19	<i>n</i> -й альтернативный шрифт	Выбирает <i>n</i> -й альтернативный шрифт (от 14 — четвёртый альтернативный шрифт, до 19 — 9й альтернативный шрифт).
20	Фрактура	вряд ли поддерживается
21	Жирный: выкл. или Подчёркивание: Двойное	жирный не всегда поддерживается; двойное подчёркивание вряд ли поддерживается.
22	Обычный цвет или яркость	Ни жирный ни блеклый
23	Не курсивный, не фрактура	
24	Подчёркивание: Нет	Подчёркивание ни одиночное ни двойное
25	Мигание: выкл.	
26	Зарезервирован	
27	Отображение: обычное	не негатив
28	Отображающийся	выключить скрывание
29	Не зачёркнутый	
30–37	Установить цвет текста (на переднем плане)	<code>30</code> + <i>n</i> , где <i>n</i> из таблицы цветов ниже
38	Зарезервировано для дополнительных цветов	обычно поддерживаются следующие аргументы: <code>5;n</code> где <i>n</i> индекс цвета (<code>0..255</code>) или <code>2;r;g;b</code> где <i>r</i> , <i>g</i> , <i>b</i> — красный, зелёный и синий каналы цвета (в пределах <code>255</code>)
39	Цвет текста по умолчанию (на переднем плане)	зависит от реализации (в соответствии со стандартом)

40–47	Установить цвет фона	<code>40</code> + <i>n</i> , где <i>n</i> из таблицы цветов снизу
48	Зарезервировано для установки расширенного цвета фона	обычно поддерживает следующие аргументы <code>5;n</code> где <i>n</i> индекс цвета (<code>0..255</code>) или <code>2;r;g;b</code> где <i>r</i> , <i>g</i> , <i>b</i> — красный, зелёный и синий каналы цвета (в пределах <code>255</code>)
49	Цвет фона по умолчанию	зависит от реализации (в соответствии со стандартом)
50	Зарезервировано	
51	Обрамлённый	
52	Окружённый	
53	Надчёркнутый	
54	Не обрамлённый и не окружённый	
55	Не надчёркнутый	
56–59	Зарезервировано	
60	ideogram underline or right side line	вряд ли поддерживается
61	ideogram double underline or double line on the right side	вряд ли поддерживается
62	ideogram overline or left side line	вряд ли поддерживается
63	ideogram double overline or double line on the left side	вряд ли поддерживается
64	ideogram stress marking	вряд ли поддерживается
65	ideogram attributes off	вряд ли поддерживается, отменяет эффекты кодов <code>60</code> – <code>64</code>
90–97	Set foreground text color, high intensity	aixterm (не стандарт)
100–107	Set background color, high intensity	aixterm (не стандарт)

Цвета [\[править | править код \]](#)

Цвета текста (и SGR параметры в целом) управляются при помощи последовательностей `CSI n1 [;n2 [; ...]] m`, где каждый `n1`, `n2`, ... является SGR-параметром, как показано выше. Таким образом, например, вы используете коды `30+i` для задания цвета текста, `40+i` для задания цвета фона, где *i* — номер цвета из таблички ниже. Следующие примеры могут использоваться с утилитой `printf`, где `\x1b[` или `\e[` реализуют CSI: Для переключения цвета текста в чёрный используйте `\x1b[30m`; в красный `\x1b[31m`; установка жирного серого `\x1b[30;1m`; установить жирный красный `\x1b[31;1m`. Для установки цветов по умолчанию используйте `\x1b[39;49m` (не поддерживается некоторыми терминалами) (или сбросьте все атрибуты `\x1b[0m`).

Таблица цветов^[6]

Intensity	0	1	2	3	4	5	6	7
Normal	Black	Red	Green	Yellow	Blue	Magenta	Cyan	White
Bright	Black	Red	Green	Yellow	Blue	Magenta	Cyan	White

См. также [править | править код]

- [terminfo](#)
- [ANSI-графика](#)
- [Управляющие символы](#)

Примечания [править | править код]

- ↑ [NIST list of withdrawn standards](#)
- ↑ *Paul Williams*. *Digital's Video Terminals* [↗](#). VT100.net (2006). Дата обращения: 17 августа 2011. Архивировано из оригинала [↗](#) 23 июля 2012 года.
- ↑ *Heathkit Company*. *Heathkit Catalog 1979* [↗](#). Heathkit Company (1979). Дата обращения: 4 ноября 2011. Архивировано из оригинала [↗](#) 13 января 2012 года.
- ↑ *Michael Mefford*. *ANSI.com: Download It Here* [↗](#). PC Magazine (7 февраля 1989). Дата обращения: 10 августа 2011. Архивировано [↗](#) 18 декабря 2012 года.
- ↑ *Dan Kegel, Eric Auer*. *Nansi and NNansi - ANSI Drivers for MS-DOS* [↗](#). Dan Kegel's Web Hostel (28 февраля 1999). Дата обращения: 10 августа 2011. Архивировано [↗](#) 18 декабря 2012 года.
- ↑ имена стандартны, но точные значения цветов не стандартизированы и зависит от устройства, которое их отображает.

Ссылки [править | править код]

- [Standard ECMA-48, Control Functions For Coded Character Sets](#) [↗](#). *(5th edition, June 1991)*, European Computer Manufacturers Association, Geneva 1991 (also published by ISO and IEC as standard ISO/IEC 6429)
- [vt100.net DEC Documents](#) [↗](#)
- [Xterm / Escape Sequences](#) [↗](#)
- [AlXterm / Escape Sequences](#) [↗](#) (недоступная ссылка)
- [A collection of escape sequences for terminals that are vaguely compliant with ECMA-48 and friends.](#) [↗](#)
- [ANSI Escape Sequences](#) [↗](#) Архивная копия от 14 декабря 2020 [↗](#)

Категории: [Компьютерные и телекоммуникационные стандарты](#) | [Стандарты ECMA](#) | [ANSI](#)