

Встроенная команда echo

Синописис

```
эхо [-урожденная] [арг...]
```

Описание

echo выводит аргументы в стандартный вывод, разделенные пробелами, за которыми следует перевод строки. Статус возврата всегда 0. Если установлен параметр short xpg_echo, Bash динамически определяет, следует ли echo разворачивать ескаре-символы (перечисленные ниже) по умолчанию в зависимости от текущей платформы. echo не интерпретируется -- как конец параметров, а просто выводит эту строку, если она задана.

Опции

Вариант	Описание
-n	Завершающий символ новой строки подавляется.
-e	Включена интерпретация следующих символов, экранированных обратной косой чертой (ниже).
-E	Отключает интерпретацию этих управляющих символов даже в системах, где они интерпретируются по умолчанию.

Экранирующие последовательности

Побег	Описание
\a	тревога (звонок)
\b	обратное пространство
\c	подавление дальнейшего вывода
\e	

Побег	Описание
<code>\E</code>	экранирующий символ
<code>\f</code>	подача формы
<code>\n</code>	новая строка
<code>\r</code>	возврат каретки
<code>\t</code>	горизонтальная вкладка
<code>\v</code>	вертикальная вкладка
<code>\\</code>	обратная косая черта
<code>\0nnn</code>	восьмибитный символ, значением которого является восьмеричное значение nnn (от нуля до трех восьмеричных цифр).
<code>\xHH</code>	восьмибитный символ, значением которого является шестнадцатеричное значение HH (одна или две шестнадцатеричные цифры).
<code>\unnnn</code>	символ Unicode (ISO / IEC 10646), значением которого является шестнадцатеричное значение NNNN (от одной до четырех шестнадцатеричных цифр)
<code>\UNNNNNNN</code>	символ Unicode (ISO / IEC 10646), значением которого является шестнадцатеричное значение NNNNNNN (от одной до восьми шестнадцатеричных цифр)

Примеры

Соображения по переносимости

- `echo` это крушение поезда мобильности. Ни одна крупная оболочка не следует POSIX полностью, и любая оболочка, которая пытается это сделать, должна считаться ужасно сломанной. SUSv4 (http://pubs.opengroup.org/onlinepubs/9699919799/utilities/echo.html#tag_20_37) указывает, что `echo` **не должен** включать никаких опций. Кроме того, он указывает, что поведение `-n` в качестве первого аргумента должно определяться реализацией, если только не используется XSI, и в этом случае `-n` он всегда обрабатывается как строка, а экранирование обратной косой черты интерпретируется по умолчанию. `dash` имеет недостаток следовать этому и интерпретировать экранирование по умолчанию, но, тем не менее, включает `-n` функцию для подавления перевода строк.

На практике, если вы можете использовать оболочку, подобную `korn`, включая `bash`, `mksh` или `zsh`, `echo` при использовании в простых случаях, как правило, надежна. Например, в очень распространенной ситуации, когда `echo`

поставляется с одним аргументом и выводом которого является добавление новой строки, использование `echo` считается обычной практикой.

- **Никогда не используйте опции для `echo` ! *Никогда!*** В любое время, когда у вас возникнет соблазн использовать `echo -e -n` или любую другую специальную функцию `echo`, **используйте вместо этого `printf`!** Если переносимость является обязательным требованием, вам следует рассмотреть возможность использования `printf` *исключительно* и просто игнорировать то, что `echo` вообще существует. Если вы должны использовать `echo -e` и отказываетесь от использования `printf`, обычно приемлемо использовать `echo $'...' if`, ориентируясь только на оболочки, поддерживающие этот специальный стиль цитирования.
- `ksh93` имеет `print` команду, которая, если кодирование специально для `ksh93` нее должно быть предпочтительнее `echo`. `printf` по-прежнему включает в себя большую часть функциональности обоих и обычно должна быть наиболее предпочтительным вариантом.

См . также

- Команда `printf`
- <http://cfajohnson.com/shell/cus-faq.html#Q0b> (<http://cfajohnson.com/shell/cus-faq.html#Q0b>)
- <http://www.in-ulm.de/~проверка/разное/echo+printf/> (<http://www.in-ulm.de/~mascheck/various/echo+printf/>)

Обсуждение

М. Антлер (<http://www.example.com>), [2016/05/01 04:12 \(\)](#)

1. Использование-Е.
2. Правильно укажите ваши переменные.
3. Используйте только 1 аргумент, не являющийся опцией.

```
$ a='-e'; b='1 \ t2'
$ echo -E $a $b
1 2
$ echo -E "$a $b"
-e 1\t2
$
```

This site is supported by Performing Databases - your experts for database administration

Bash Hackers Wiki



Except where otherwise noted, content on this wiki is licensed under the following license:
GNU Free Documentation License 1.3