

User selections

Synopsis

```
select <NAME>; do
    <LIST>
done
```

```
выберите <ИМЯ> в <СЛОВАХ>; выполните
<СПИСОК>
Выполнено
```

```
# альтернативный, исторический и недокументированный синтаксис

выберите <ИМЯ>
{
    <СПИСОК>
}

выберите <ИМЯ> в <СЛОВАХ>
{
    <СПИСОК>
}
```

Описание

Эта составная команда предоставляет своего рода меню. Пользователю предлагается *пронумерованный список* заданных слов, и его просят ввести порядковый номер слова. Если было выбрано слово, переменная `<NAME>` устанавливается на это слово, и список `<LIST>` выполняется.

Если `in <WORDS>` задано значение `no`, то позиционные параметры принимаются как слова (как если `in "$@"` бы было написано).

Независимо от функциональности, введенный пользователем *номер* сохраняется в переменной `REPLY`.

Bash знает альтернативный синтаксис для `select` команды, заключая тело цикла в `{...}` вместо `do ... done`:

```
выберите x в 1 2 3
{
  echo $ x
}
```

Этот синтаксис **не документирован** и не должен использоваться. Я нашел определения синтаксического анализатора для него в коде 1.x и в современном коде 4.x. Я предполагаю, что это сделано по соображениям совместимости. Этот синтаксис не задан POSIX(R) .

Примеры

```
echo
```

```
echo
```

```
очистить
```

```
# значение, например:
```

```
# готово # <СПИСОК> # выберите <ИМЯ> в <СЛОВАХ>;  нажимаем цифровую клавишу 1, 2 или 3, затем ВВОД -клавиша
```

```
echo один ввод является пустым выбором и будет бесконечно повторяться до тех пор, пока
```

```
Ctrl-C или Ctrl-D
```

```
echo не выберут опцию X в beer whiskeyвинный ликер;
```

```
    повторите, что вы заказали $ OPTIONX
```

```
    break # break позволяет избежать бесконечного цикла - вторая строка, которая должна выполняться, всегда
```

```
выполняется
```

```
# разместите здесь некоторые, если еще какие-то дела
```

```
# и объясните, как имеет смысл, что $ OPTIONX красный, а OPTIONX черный
```

```
# хотя оба являются переменными
```

Соображения о переносимости

See also



Discussion

