

[Каталог документации](#) / [Раздел "Программирование, языки"](#) / [Оглавление документа](#)**Advanced Bash-Scripting Guide: Искусство программирования на языке сценариев командной оболочки**[Назад](#)

Глава 4. Переменные и параметры. Введение.

[Вперед](#)

## 4.3. Переменные Bash не имеют типа

В отличие от большинства других языков программирования, Bash не производит разделения переменных по "типам". По сути, переменные Bash являются строковыми переменными, но, в зависимости от контекста, Bash допускает целочисленную арифметику с переменными. Определяющим фактором здесь служит содержимое переменных.

### Пример 4-4. Целое число или строка?

```
#!/bin/bash
# int-or-string.sh: Целое число или строка?

a=2334                                # Целое число.
let "a += 1"
echo "a = $a "                        # a = 2335
echo                                  # Все еще целое число.

b=${a/23/BV}                          # замена "23" на "BV".
echo "b = $b"                         # Происходит трансформация числа в строку.
declare -i b                          # b = BV35
echo "b = $b"                         # Явное указание типа здесь не поможет.
                                      # b = BV35

let "b += 1"                          # BV35 + 1 =
echo "b = $b"                         # b = 1
echo

c=BV34
echo "c = $c"                         # c = BV34
d=${c/BV/23}                          # замена "BV" на "23".
echo "d = $d"                         # Переменная $d становится целочисленной.
let "d += 1"                          # d = 2334
echo "d = $d"                         # 2334 + 1 =
echo                                  # d = 2335

# А что происходит с "пустыми" переменными?
```

```
e=""
echo "e = $e"          # e =
let "e += 1"           # Арифметические операции допускают использование
"пустых" переменных?
echo "e = $e"          # e = 1
echo                   # "Пустая" переменная становится целочисленной.

# А что происходит с необъявленными переменными?
echo "f = $f"          # f =
let "f += 1"           # Арифметические операции допустимы?
echo "f = $f"          # f = 1
echo                   # Необъявленная переменная трансформируется в
целочисленную.
```

# Переменные Bash не имеют типов.

```
exit 0
```

Отсутствие типов -- это и благословение и проклятие. С одной стороны -- отсутствие типов делает сценарии более гибкими (чтобы повеситься -- достаточно иметь веревку!) и облегчает чтение кода. С другой -- является источником потенциальных ошибок и поощряет привычку к "неряшливому" программированию.

Бремя отслеживания типа той или иной переменной полностью лежит на плечах программиста. Bash не будет делать это за вас!

---

[Назад](#)[Присваивание значений  
переменным](#)[К началу](#)[Наверх](#)[Вперед](#)[Специальные типы  
переменных](#)

Спонсоры:



При поддержке  
**inferno solutions\***

Хостинг:



**Hoster.ru**  
хостинг провайдер

---

[Закладки на сайте](#)  
[Проследить за страницей](#)

---

Created 1996-2022 by **Maxim Chirkov**  
[Добавить](#), [Поддержать](#), [Вебмастеру](#)