



Получение значения

Чтобы получить значение, которое хранится в переменной, нужно поставить перед её именем знак доллара `$`. Пример:

```
#!/usr/bin/bash

number=123
some_string="Hello"

echo $number $some_string # 123 Hello
```

Здесь число записано без кавычек, а строковое значение в кавычках. Так можно писать просто, чтобы подчеркнуть смысл переменных. Для Bash все значения всё равно строки.

Значение переменной можно встраивать в строку в двойных кавычках или без кавычек, также поставив перед именем переменной `$`. Это иногда называют интерполяцией строк. Пример:

```
#!/usr/bin/bash

name="Ariannita"

msg="Hello, $name!"

echo $msg # Hello, Ariannita!

# Со строками в одинарных кавычках интерполяция не работает
echo 'Hello, $name!' # Hello, $name!
```

Но бывают неоднозначные случаи. Пусть мы пишем скрипт, который создаёт файл, а потом переименовывает его, добавив `1` к названию файла:

```
#!/usr/bin/bash

filename="myfile"

# Создадим файл
touch $filename

# Переименуем файл
mv $filename $filename1 # Работать не будет
```

Почему скрипт выше не работает? Потому что `filename1` трактуется как имя переменной, а такой переменной у нас нет. Исправить это можно с помощью другого варианта обращения к переменной:

```
#!/usr/bin/bash

filename="myfile"

# Создадим файл
touch $filename

# Переименуем файл
mv $filename ${filename}1 # Работает как ожидалось
```

Таким образом, обратиться к значению переменной можно, поставив `$` перед её именем `$var_name` или так `${var_name}`. Второй способ предпочтительнее, так как позволяет избежать неоднозначностей.

