

Области видимости

Область видимости определяет участок кода, в котором видна переменная. В Bash существует две основные области видимости: глобальная и локальная.

Если переменная объявляется в глобальной области видимости, то она доступна во всём скрипте, ниже её объявления. Рассмотрим пример:

```
#!/bin/bash

# Не выведет значение переменной, так как
# в этом месте она ещё не существует
echo "g is $g" # g is

g=123

echo "g is $g" # g is 123

func() {
    echo "in function g is $g" # in function g is 123
}

func
```

Важно отметить то, что функция будет иметь доступ к переменной, только если она вызывается ниже объявления этой переменной. Думаю, не сложно догадаться почему.

В Bash по умолчанию все переменные глобальные. Это может несколько удивить, но даже переменная, объявленная внутри функции, имеет глобальную область видимости. Пример:

```
#!/bin/bash

func1() {
    # Переменная объявлена внутри функции
    v=500
}

func2() {
    echo "in func2 v is $v"
}

echo "v is $v" # v is
# Здесь переменная ещё недоступна

func1

echo "v is $v" # v is 500
# А теперь уже доступна

func2 # in func2 v is 500
```

Таким образом имеется ещё один способ "вернуть" значение из функции: просто создать глобальную переменную для него. **Никогда** так не делайте! Даже в совсем простых скриптах.