

Переменные

Как и во многих других языках программирования, в Bash можно создавать переменные. Переменные нужны для сохранения данных в оперативной памяти компьютера и последующего доступа к ним по имени переменной. Переменная начинает существовать после того, как ей было присвоено значение. Значения присваиваются с помощью оператора `=`. Пример:

```
#!/bin/bash

n="123"
greeting="Hello"
```

В данном скрипте создаются две переменные с именами `n` и `greeting` с соответствующими значениями. Обратите внимание! Пробелы вокруг знака `=` ставить нельзя. Если их поставить, то bash будет интерпретировать имя переменной и её значение как имена команд и аргументов:

```
#!/bin/bash

# Будет попытка запустить команду my_var с аргументами = и hello
my_var = "hello"

# Будет попытка запустить команду my_var с аргументом =hello
my_var ="hello"

# Будет попытка запустить команду hello, установив перед этим
# переменную окружения с именем my_var
my_var= "hello"

# В любом случае будет ошибка!
```

Типы переменных в Bash фактически отсутствуют. Значениями переменных могут быть только строки, которые записываются в одинарных кавычках `'string'`, в двойных кавычках `"string"` или вообще без кавычек. Разница между записью строк в двойных и одинарных кавычках будет показана далее в курсе.

Ещё примеры создания переменных:

```
#!/bin/bash

# Здесь вполне допустимо писать без кавычек
n=256
s=Something

# Переменная "пустая"
empty=
empty="" # Тоже самое

# Можно сразу создать несколько переменных,
# но это не рекомендуется из-за проблем с читаемостью.
x=5 y=10
```

Если же строка содержит пробелы, то кавычки необходимы:

```
#!/bin/bash

# Будет ошибка
msg=Hello World

# OK
msg="Hello World"
```

Лично я предпочитаю всегда ставить кавычки, так как это позволяет избежать неожиданного поведения скрипта (эту мысль я повторяю ещё не один раз в течение курса).

В отличие от большинства других языков программирования, в Bash переменные не имеют "типов". По сути, переменные в Bash всегда являются строковыми, но, в зависимости от контекста, с ними допускается целочисленная арифметика. Определяющим фактором здесь служит содержимое переменных. Так, если строка состоит только из цифр она может рассматриваться как целое число, и её допускается использовать при выполнении арифметических выражений. Дробные числа, а также логические значения в Bash полностью отсутствуют.