4.2. Присваивание значений переменным

=

оператор присваивания (*пробельные символы до и после оператора -- недопустимы*)

Не путайте с операторами сравнения = и -eq!

Обратите внимание: символ = может использоваться как в качестве оператора присваивания, так и в качестве оператора сравнения, конкретная интерпретация зависит от контекста применения.

Пример 4-2. Простое присваивание

```
#!/bin/bash
# Явные переменные

echo

# Когда перед именем переменной не употребляется символ '$'?
# В операциях присваивания.

# Присваивание
a=879
echo "Значение переменной \"a\" -- $a."

# Присваивание с помощью ключевого слова 'let'
let a=16+5
echo "Значение переменной \"a\" теперь стало равным: $a."

echo

# В заголовке цикла 'for' (своего рода неявное присваивание)
echo -n "Значения переменной \"a\" в цикле: "
for a in 7 8 9 11
do
  echo -n "$a "
done
```

```
echo
echo

# При использовании инструкции 'read' (тоже одна из разновидностей
присваивания)
echo -n "Введите значение переменной \"a\" "
read a
echo "Значение переменной \"a\" теперь стало равным: $a."

echo

exit 0
```

Пример 4-3. Присваивание значений переменным простое и замаскированное

```
#!/bin/bash

a=23              # Простейший случай
echo $a
b=$a
echo $b

# Теперь немного более сложный вариант (подстановка команд).

a=`echo Hello!`   # В переменную 'a' попадает результат работы команды
'echo'
echo $a
#  Обратите внимание на восклицательный знак (!) в подстанавливаемой
команде
#+ этот вариант не будет работать при наборе в командной строке,
#+ поскольку здесь используется механизм "истории команд" BASH
#  Однако, в сценариях, механизм истории команд запрещен.

a=`ls -l`         # В переменную 'a' записывается результат работы
команды 'ls -l'
echo $a           # Кавычки отсутствуют, удаляются лишние пробелы и
пустые строки.
echo
echo "$a"         # Переменная в кавычках, все пробелы и пустые строки
сохраняются.
                  # (См. главу "Кавычки.")

exit 0
```

Присваивание переменных с использованием $(...) (более современный метод, по сравнению с обратными кавычками)

```
# Взято из /etc/rc.d/rc.local
R=$(cat /etc/redhat-release)
arch=$(uname -m)
```

Переменные и параметры. Введение. Наверх Переменные Bash не имеют типа