Арифметика

Арифметика

Синтаксис:

let “переменная = выражение”

Арифметика

Синтаксис:

let “переменная = выражение”

Пример:

```
let “c = 1 + 1”
let “c = a + b”
```

Арифметика

Синтаксис:

let “переменная = выражение”

Пример:

```
let “c = 1 + 1”
let “c = a + b”
```

Операции:

+, -, /, *	стандартные
%	остаток от деления
**	возведение в степень

Арифметика

Синтаксис:

let “переменная = выражение”

Пример:

```
let “c = 1 + 1”
let “c = a + b”
```

Операции:

+, -, /, *	стандартные
%	остаток от деления
**	возведение в степень

Сокращение:

let “a**=**a**+**b” эквивалентно let “a**+=**b”

Внешние программы

Внешние программы

Синтаксис:

```
переменная=`программа`
```

Пример:

```
a=`echo “test”`
files=`ls ~`
```

Внешние программы

Код возврата:

0 корректное завершение

не 0 в процессе работы были ошибки

Узнать код:

```
$?
```

Выйти с кодом:

```
exit код
```

Пример:

```
touch file.txt
echo $?
```

Внешние программы

Проверка кода возврата:

```
if `программа`
then
  # действия, если код 0
else
  # действия, если код не 0
fi
```

Функции

Функции

Задаем функцию:

```
имя_функции ()
{
   # действия
}
```

Функции

Задаем функцию:

```
имя_функции ()
{
   # действия
}
```

Используем функцию:

```
...
имя_функции
...
```

Функции

Функции с параметрами:

```
имя_функции ()
{
  # действия с $1, $2, … , $#
}
```

Используем функцию:

```
...
имя_функции аргумент1 аргумент2 ...
...
```

Функции

Переменные:

```
имя_функции ()
{
  var_global=1
  local var_local=1
}
```

Используем:

```
имя_функции
echo $var_global # выведет 1
echo $var_local  # ничего не выведет
```

Функции

Компактная запись:

```
имя_функции () { действ1; действ2; }
```

Функции

Компактная запись:

```
имя_функции () { действ1; действ2; }
```

Актуально и в других конструкциях:

```
if [[ $var=="test "]]; then
...
for i in 1 2 3 4 5; do
...
```