

Вызов функции и передача аргументов

Напишем скрипт, который будет содержать простейшую функцию:

```
#!/bin/bash

hello() {
    echo "Hello, World!"
}
```

Если сейчас запустить этот скрипт, то ничего выведено не будет. Причина в том, что мы только объявили функцию, но не вызвали её. Вызвать собственную функцию можно как и любую другую команду в bash-скрипте, просто указав её имя:

```
#!/bin/bash

hello() {
    echo "Hello, World!"
}

hello
# Выведет: Hello, World!
```

В Bash нужно обязательно определить функцию перед её вызовом. Другими словами, объявление функции должно быть выше по тексту сценария, чем её вызов. В противном случае Вы получите ошибку `command not found` . Обратите внимание на то, что даже в тексте ошибки сказано, что функция — это просто команда.

А как передать параметры в функцию и работать с ними внутри этой функции? Если Вы просмотрели урок о параметрах скрипта, то Вы уже знаете ответ на этот вопрос. Пример:

```
#!/bin/bash

max() {
    if [[ $1 -ge $2 ]]
    then
        echo $1
    else
        echo $2
    fi
}

max 12 5 # 12
max 123 500 # 500
```

Как Вы уже могли заметить, работа с аргументами внутри функции происходит точно так же, как и с аргументами скрипта.

А вот пример функции, которая принимает на вход любое число аргументов и считает их сумму (она аналогична скрипту `summator.sh` из предыдущего урока):

```
#!/bin/bash

calc_sum() {
    summa=0
    for n in $@
    do
        ((summa += n))
    done
    echo "Сумма переданных аргументов: $summa"
}

calc_sum 5 2 3      # Сумма переданных аргументов: 10
calc_sum            # Сумма переданных аргументов: 0
calc_sum 1 2 -5 15  # Сумма переданных аргументов: 13
calc_sum "a" 1 b 3  # Сумма переданных аргументов: 4
```

Последний вызов содержит нечисловые аргументы. Но так как при попытке их добавить к сумме они воспринимаются как 0, ошибка не возникает.

Последнее, что хотел бы добавить по теме аргументов функции: переменная `$0` всегда хранит имя скрипта. Чтобы напечатать имя функции, нужно воспользоваться системной переменной `$FUNCNAME` :

```
#!/bin/bash

my_function() {
    echo "Имя скрипта: $0"
    echo "Имя функции: $FUNCNAME"
}

# Не забываем вызывать функцию
my_function
```

Пример запуска:

