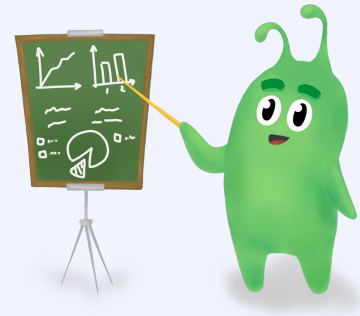


Создание Bash-скриптов



План занятия

- ❖ Структура bash-скриптов
- ❖ Переменные
- ❖ Условия
- ❖ Циклы
- ❖ Функции
- ❖ Обработка ошибок



Bash-скрипт

- ❖ Набор команд, объединенных общей целью
- ❖ Может быть записан в файле или в командной строке
- ❖ Файл скрипта начинается с `#!/bin/bash` (sha-bang)

Для других скриптовых языков sha-bang лучше указывать через `/usr/bin/env`

```
#!/usr/bin/env python3
```

Переменные в скриптах

- ❖ Задаются в виде:
`переменная=значение`
- ❖ Обращение к переменным:
`echo $переменная`

или

`echo ${переменная}`

Кавычки

- ❖ `echo "echo $var"`
выведет строку "echo" + содержимое переменной `var`
- ❖ `echo 'echo $var'`
выведет строку "echo \$var"
- ❖ `echo `echo $var``
выведет содержимое переменной `var`

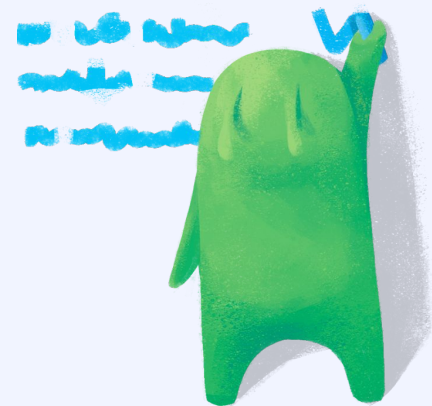
Арифметические операции

- ❖ Команда `let`

```
let "a=2+2"
```

- ❖ Двойные круглые скобки

```
(( b=2+3 ))
```



Аргументы командной строки

При запуске скрипта

```
./test.sh abc def qwe
```

В переменной **\$1** внутри скрипта будет значение “abc”

В переменной **\$2** – “def”

В переменной **\$3** – “qwe”

Аргументы после **\$9** следует заключать в **\${ } – \${10}**

Условия

В bash для проверки условий используются команды

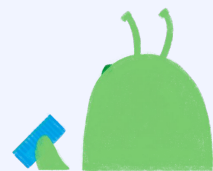
- ❖ `test`
оператор проверки условий
- ❖ `if`
условный оператор проверки результата условий

test

- ❖ -z — пустая ли переменная
- ❖ -e — существует ли файл
- ❖ -eq — проверка на равенство
- ❖ -gt — больше
-lt — меньше
- ❖ -ge — больше или равно

if / then / else

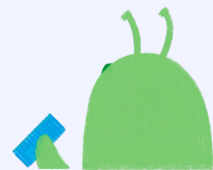
```
if <условие>  
then  
    <действие при успехе>  
else  
    <действие при неудаче>  
fi
```



Цикл for

Выполнять действия для каждого элемента списка

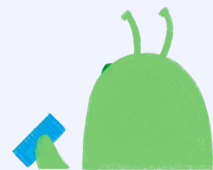
```
for <переменная> in <список>
do
    <действие для каждого элемента>
done
```



Цикл while

Выполнять действия, пока выполняется условие

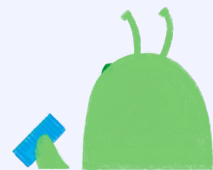
```
while <условие>  
do  
    <действие для каждой итерации>  
done
```



Цикл until

Выполнять действия, пока не выполнится условие

```
until <условие>  
do  
    <действие для каждой итерации>  
done
```



Функции

Структура:

```
<имя функции>()  
{  
    <команды>  
    return <код возврата> — опционально  
}
```

В следующей серии

systemd и системные сервисы



To Be Continued

