

Операторы условий Powershell

 newadmin.ru/operatory-uslovij-powershell

23 сентября, 2020



Powershell

 **admin**

Операторы условий (они же **условные операторы**) – используются для условного выполнения кода. Проще говоря, в момент когда необходимо принять решение, будут задействованы операторы условий. Почти всегда используются совместно с операторами сравнения, рассмотренными ранее. Давайте взглянем поближе на операторы условий Powershell.

Оператор IF

IF (если) – вычисляет значение заданное в круглых скобках (). Если результатом вычисления является логическое **True**, то выполняется конструкция команд в фигурных скобках {}. Если результат вычисления будет **False** выполнятся команды следующие за фигурными скобками. Давайте рассмотрим на примере.

```
1 $test_operator=$True
2 if ($test_operator)
3 {
4 Write-Host "УРА. Оператор имеет значение TRUE"
5 }
```

Результатом выполнения данного скрипта будет вывод в консоль фразы: **УРА**.
Оператор имеет значение TRUE. Если переменная **\$test_operator** будет иметь значение **FALSE** фразы указанной выше не появится, т.к. командлет выводящий данную фразу на экран находится в **{}** скобках.

Оператор ELSE

ELSE (*иначе*) – используется в связке с оператором **IF**. Оператор **ELSE** позволяет определить что делать в случае если оператор **IF** не равен **TRUE**. Написание конструкции состоит из оператора **ELSE**, далее идут фигурные скобки. Разберем на примере для лучшего понимания.

```
1 $proc=(Get-Process).ProcessName
2 if ($proc -like "notepad")
3 {
4 Write-Host "Блокнот запущен."
5 }
6 else
7 {start notepad}
```

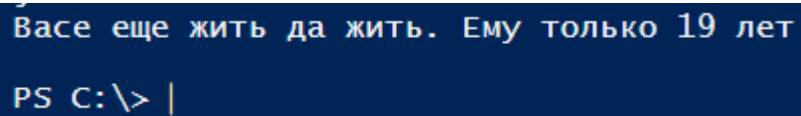
В переменную **\$proc** мы сохраняем список запущенных процессов с указанием вывода по полю **ProcessName**. Далее оператором **if** сравниваем, есть ли в переменной **\$proc** запущенный процесс с именем **notepad**. Если есть, то выводим на экран сообщение **Блокнот запущен**. Если такого процесса нет в ход идет конструкция **else** и выполняет команды в фигурных скобках, то есть запускает **notepad**.

Данный скрипт конечно в простейшем варианте, но тоже пригоден и часто используется системными администраторами. Очень удобно контролировать запущен ли конкретный процесс и если нет запускать его.

Оператор ELSEIF

ELSEIF – объединяет **IF** и **ELSE**. Используется после оператора **IF** в случае когда необходимо сделать дополнительное сравнение. Давайте рассмотрим пример в Васей. Определим переменную **\$vasya_years** и укажем возраст. Далее через конструкцию **IF, ELSEIF** и **ELSE** выясним насколько Вася старый.

```
1 $vasya_years=19
2 if ($vasya_years -gt 60)
3 {Write-Host "Вася уже старенький. Ему" $vasya_years "лет"}
4 elseif ($vasya_years -ge 40)
5 {Write-Host "Вася еще молод. Ему" $vasya_years "лет"}
6 else {
7 Write-Host "Васе еще жить да жить. Ему только" $vasya_years "лет"
8 }
```



```
Васе еще жить да жить. Ему только 19 лет
PS C:\> |
```

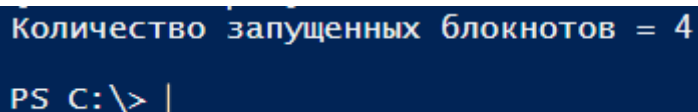
В переменной мы указали что Васе 19 лет. Первый условный оператор **if** проверил что Васе не больше 60 лет и передал следующему оператору **elseif**. В результате проверки выяснилось что Васе меньше 40 лет и следующий этап за оператором **else**. Который принял решение, что Вася не попадает под первые два оператора, и он явно еще молодой.

Операторы условий Powershell могут содержать в себе и другие условные операторы. Бывает необходимо для более сложной оценки. Посмотрим на примере с блокнотами.

```
1 $proc=(Get-Process).ProcessName
2 $len=($proc -like "notepad").Length
3 if ($proc -like "notepad")
4 {
5     if ($len -ge 2)
6     {Write-Host "Количество запущенных блокнотов =" $len}
7     else {
8         Write-Host "Блокнот запущен."}
9     }
10 else
11 {start notepad}
```

В первом условии **if** мы сравниваем переменную **\$proc** и в случае положительного результата переходим в вложенному **if**. Во втором

(вложенном) **if** мы сравниваем переменную **\$len** с двойкой. Если запущено 2 и более процесса **notepad** будет выведено сообщение с указанием количества запущенных блокнотов. В случае 1 запущенного блокнота будет сообщение Блокнот запущен. Если блокнот не запущен, в последнем операторе **else** скрипт его запустит.



```
Количество запущенных блокнотов = 4
PS C:\> |
```

Оператор SWITCH

SWITCH – использует немного другой синтаксис для сравнения значений. Его еще называют оператор переключения. В круглых скобках указывается выражение, результат которого сравнивается со значениями в фигурных скобках. Перейдем к примеру

```
1  $names = 'Даша'
2  switch ($names)
3  {
4      'Даша'
5      {
6          'Поздравляю вы выбрали Дарью'
7      }
8      'Машу'
9      {
10         'Вы выбрали Марию'
11     }
12     'Глафира'
13     {
14         'Вы выбрали Глафиру'
15     }
16     default {"Вы никого не выбрали получите Колю"}
17 }
```

В переменной **\$names** мы указали имя

Даша. Оно последовательно

сравнивается с каждым именем в

фигурных скобках. Если не одно из имен

не подошло, выполняется действие указанное в параметре **Default**. Параметр

default не обязательный можно его не указывать. В случае отсутствия совпадения скрипт просто завершит работу.

Поздравляю вы выбрали Дарью

Рекомендую к прочтению:

- [Powershell скрипты](#)
- [Переменные](#)
- [Операторы сравнения](#)
- [Циклы](#)