

## Урок 4. Базовые вычисления в R

С некоторыми примерами вычислений R мы уже познакомились в первых уроках, теперь посмотрим на другие полезные операторы и функции.

Возведение в степень в R осуществляется двумя способами. Либо с помощью двойной звездочки, как в Python:

```
5 ** 2
```

```
## [1] 25
```

Либо с помощью крышечки:

```
5 ^ 2
```

```
## [1] 25
```

Если нам нужно возвести число  $e$  в степень, можем это сделать с помощью функции `exp()`:

```
exp(1)
```

```
## [1] 2.718282
```

```
exp(2)
```

```
## [1] 7.389056
```

Если нужен логарифм по произвольному основанию, нам понадобится функция `log()`. Основание нужно указать в виде опции `base`:

```
log(100, base = 10)
```

```
## [1] 2
```

Название опции можно опустить:

```
log(100, 10)
```

```
## [1] 2
```

Для нахождения натурального логарифма (логарифма по основанию  $e$ ) основание можно опустить и использовать функцию `log()` без дополнительных опций:

```
log(100)
```

```
## [1] 4.60517
```

Отметим заодно, что, так же как и в Python, в качестве десятичного разделителя в R используется точка, не запятая.

Посмотрим на округление. Стандартное арифметическое округление реализуется с помощью функции `round()`:

```
round(1.67)
```

```
## [1] 2
```

При необходимости можно указать число знаков после запятой (до какого разряда округляем):

```
round(1.67, 1)
```

```
## [1] 1.7
```

Для округления в меньшую сторону используется функция `floor()` («пол»):

```
floor(1.67)
```

```
## [1] 1
```

В большую сторону – функция `ceiling()` («потолок»):

```
ceiling(1.67)
```

```
## [1] 2
```

В целом, все операции схожи по своему исполнению с Python, главное отличие в том, что здесь нам не пришлось импортировать отдельный модуль для вычислений.

Теперь вернемся к числовым переменным и рассмотрим такую задачу.

*В некотором банке есть фантастический вклад, по которому остаток на счете удваивается каждый год. У клиента на счете лежит 200 тысяч. Сколько денег будет на его счете через три года?*

Напишем универсальный код, который позволит посчитать итоговую сумму на счете для любого клиента через любое число лет. Заодно на примере этой задачи посмотрим, какие названия переменных допустимы в R.

```
start.sum <- 200 # стартовая сумма  
Years <- 3 # число лет
```

Чтобы узнать сумму на счете через один год, мы домножим значение `start.sum` на два, через два года — еще на два и так далее. Получаем возведение в степень, равную числу лет:

```
end.sum <- start.sum * 2^Years  
end.sum # результат
```

```
## [1] 1600
```

Естественно, теперь мы можем обновлять значения переменных и решать любые задачи такого типа.

Ряд комментариев по названиям:

- 1) названия переменных в R могут содержать точки и знаки подчеркивания, но не могут содержать пробелов;
- 2) название переменной не должно начинаться с цифры, но оно может включать цифры; оно также может включать заглавные буквы;
- 3) названия переменных не должны совпадать с названиями операторов.