

1. ЧИСЛА

А сейчас, когда вы всё установили, давайте напомним программу! Откройте ваш любимый текстовый редактор и наберите на клавиатуре следующее:

```
puts 1+2
```

Сохраните вашу программу (да, это программа!) под именем `calc.rb` (`.rb` — это то, что мы обычно пишем в конце имени программы, написанной на Ruby). А теперь выполните вашу программу, набрав `ruby calc.rb` в командной строке. Она должна вывести на экране число 3. Видите, программировать не так уж трудно, не правда ли?

3. НАКОМСТВО С PUTS

Так что же происходит в этой программе? Я уверен, что вы смогли догадаться, что делает **1+2**, и наша программа, в принципе, такая же, как эта:

```
puts 3
```

Слово **puts** просто выводит на экран всё, что следует за ним.

ЦЕЛЫЕ И ПЛАВАЮЩИЕ

В большинстве языков программирования (и Ruby не является исключением) числа без десятичной точки называются целыми, а числа с десятичной точкой обычно называются числами с плавающей точкой или проще — плавающими. [По сути, это дробные десятичные числа, представленные в компьютере в двоичном виде. — Прим. перев.]

Вот несколько целых чисел:

A horizontal number line with arrows at both ends. There are six major tick marks labeled 0, 1, 2, 3, 4, and 5 from left to right. Below the tick mark for 0, the number -205 is written.

А вот несколько чисел с плавающей точкой:

$$\begin{array}{r} 54.321 \\ 0.001 \\ -205.3884 \\ 0.0 \end{array}$$

На практике в большинстве программ плавающие числа не используются, только целые. (В конце концов, никто не хочет прочитать 7.4 сообщений электронной почты, просмотреть 1.8 web-страниц или послушать 5.24 любимых песен...) Числа с плавающей точкой больше используются для научных целей (физических

экспериментов и тому подобное) и для 3D-графики. Даже большинство программ для денежных расчётов применяют целые числа: в них просто учитываются суммы в копейках! [в оригинале: "в пенсах"— *Прим. перев.*]

ПРОСТАЯ АРИФМЕТИКА

К этому моменту у нас есть всё необходимое, чтобы сделать простой калькулятор. (В калькуляторах всегда используются плавающие числа, поэтому если вы хотите, чтобы ваш компьютер просто работал как калькулятор, вы также должны использовать числа с плавающей точкой.) Для сложения и вычитания мы используем `+` и `-`, что мы уже видели в программе. Для умножения мы используем `*`, а для деления мы используем `/`. На большинстве клавиатур эти клавиши располагаются среди цифровых клавиш в правой её части. Если у вас уменьшенная клавиатура или ноутбук, то вы можете просто нажать клавиши `Shift` и `8` или `/` (та же клавиша, что и клавиша `?`). Давайте попробуем немного расширить нашу программу `calc.rb`. Наберите на клавиатуре, а затем выполните следующее:

```
puts 1.0 + 2.0
puts 2.0 * 3.0
puts 5.0 - 8.0
puts 9.0 / 2.0
```

Вот что возвращает эта программа:

```
3.0
6.0
-3.0
4.5
```

(Пробелы в программе не важны; они просто делают код легче для чтения.) Что ж, в том, что мы увидели, нет ничего удивительного. А теперь давайте попробуем на целых числах:

```
puts 1+2
puts 2*3
puts 5-8
puts 9/2
```

В основном, то же самое, правда?

```
3
6
-3
4
```

Хм... кроме последнего! Но когда вы выполняете арифметические действия с целыми числами, вы получаете целочисленные ответы. Когда ваш компьютер не может получить "правильный" ответ, он всегда округляет его. (Конечно, 4 и *есть* правильный ответ в целочисленной арифметике для выражения `9/2`; хотя, возможно, это не тот ответ, который вы ожидали.)

Возможно, вы недоумеваете, зачем может понадобиться целочисленное деление.

Ну, скажем, вы собираетесь в кино, но у вас только 9 долларов. У нас в Портленде вы можете посмотреть фильм в кинотеатре "Багдад" за 2 бакса. Сколько фильмов вы сможете посмотреть там? $9/2$... 4 фильма. 4.5 — это в данном случае, конечно, *неправильный* ответ; вам не позволят посмотреть полфильма и не позволят половине вас посмотреть целый фильм... Некоторые вещи нельзя разделить.

А теперь поэкспериментируйте сами с несколькими программами! Если вы хотите применить более сложные выражения, можете использовать скобки. Например:

```
puts 5 * (12-8) + -15
puts 98 + (59872 / (13*8)) * -52
```

5
-29802

ПОПРОБУЙТЕ ЕЩЁ КОЕ-ЧТО

Напишите программу, которая сообщит вам:

- сколько часов в году?
- сколько минут в десятилетии?
- ваш возраст в секундах?
- сколько шоколадок вы надеетесь съесть за свою жизнь?
***Предупреждение:** Вычисления в этой части программы могут потребовать очень много времени!*
- А вот задание потруднее: Если я прожил 943 миллиона секунд, то каков мой возраст?

Когда вы закончите развлекаться с числами, давайте взглянем на буквы.