**NUMBERS**

Sekarang setelah Anda menyelesaikan semuanya, ayo tulis sebuah program! Buka editor teks favorit Anda dan ketik yang berikut ini:

*puts 1+2*

Simpan program Anda (ya, itu sebuah program!) Sebagai *calc.rb* (therr adalah apa yang biasanya kita tulis di akhir program yang ditulis di Ruby). Sekarang jalankan program anda dengan mengetik ruby ​*​calc.rb* ke command line anda. Seharusnya memasang 3 di layar Anda. Lihat, pemrogramannya tidak begitu sulit, kan?

**INTRODUCTION TO PUTS**

Jadi apa yang terjadi dalam program itu? Saya yakin Anda bisa menebak apa yang 1 + 2 tidak; Program kami pada dasarnya sama seperti:

*puts 3*

menempatkan hanya menulis ke layar apa pun yang datang setelah itu.

**INTEGER AND FLOAT**

Dalam kebanyakan bahasa pemrograman (dan Ruby tidak terkecuali) angka tanpa titik desimal disebut bilangan bulat, dan angka dengan titik desimal biasanya disebut bilangan floating-point, atau lebih sederhana lagi, mengapung. Berikut adalah beberapa bilangan bulat :

*5*

*-205*

*9999999999999999999999999*

*0*

Dan berikut contoh Float :

*54.321*

*0.001*

*-205.3884*

*0.0*

Dalam prakteknya, sebagian besar program tidak menggunakan Float; hanya bilangan bulat (Lagi pula, tidak ada yang mau melihat 7,4 email, atau melihat-lihat 1,8 halaman web, atau mendengarkan 5.24 lagu favorit mereka ...) Float lebih banyak digunakan untuk tujuan akademis (eksperimen fisika dan sejenisnya) dan untuk grafis 3D. Bahkan kebanyakan program uang menggunakan bilangan bulat; mereka hanya melacak jumlah uang!

**SIMPLE ARITHMETIC**

Sejauh ini, kita punya semua pembuatan kalkulator sederhana. (Kalkulator selalu menggunakan pelampung, jadi jika Anda ingin komputer Anda bertindak seperti kalkulator, Anda juga harus menggunakan pelampung.) Untuk penambahan dan pengurangan, kami menggunakan + dan -, seperti yang kami lihat. Untuk multiplikasi, kita menggunakan \*, dan untuk pembagian yang kita gunakan /. Sebagian besar keyboard memiliki tombol ini di keypad numerik di sisi kanan. Jika Anda memiliki keyboard atau laptop yang lebih kecil, Anda bisa menggunakan Shift 8 dan / (tombol yang sama dengan tombol?). Mari kita coba untuk memperluas program calc.rb kita sedikit. Ketik berikut ini dan kemudian jalankan.

*puts 1.0 + 2.0*

*puts 2.0 \* 3.0*

*puts 5.0 - 8.0*

*puts 9.0 / 2.0*

(Ruang dalam program tidak penting, mereka hanya membuat kode lebih mudah dibaca.) Nah, itu tidak terlalu mengejutkan. Sekarang mari kita coba dengan bilangan bulat:

*puts 1+2*

*puts 2\*3*

*puts 5-8*

*puts 9/2*

Kecuali yang terakhir! Tapi ketika Anda melakukan aritmatika dengan bilangan bulat, Anda akan mendapatkan jawaban bilangan bulat. Bila komputer Anda tidak bisa mendapatkan jawaban "benar", itu selalu berputar ke bawah. (Tentu saja, 4 adalah jawaban yang benar dalam bilangan integer aritmatika untuk 9/2, mungkin bukan jawaban yang Anda harapkan.)

Mungkin Anda bertanya-tanya apa pembagian bilangan bulat itu bagus. Nah, katakan saja Anda akan pergi ke bioskop, tapi Anda hanya punya $ 9. Di Portland sini, Anda bisa melihat film di Bagdad seharga 2 dolar. Berapa banyak film yang bisa Anda lihat di sana? 9/2 ... 4 film. 4.5 jelas bukan jawaban yang tepat dalam kasus ini; mereka tidak akan membiarkan Anda menonton setengah dari sebuah film, atau membiarkan setengah dari Anda untuk melihat keseluruhan film ... beberapa hal tidak dapat dibagi.

Jadi sekarang bereksperimen dengan beberapa program Anda sendiri! Jika Anda ingin menulis ekspresi yang lebih kompleks, Anda bisa menggunakan tanda kurung. Sebagai contoh:

*puts 5 \* (12-8) + -15*

*puts 98 + (59872 / (13\*8)) \* -52*