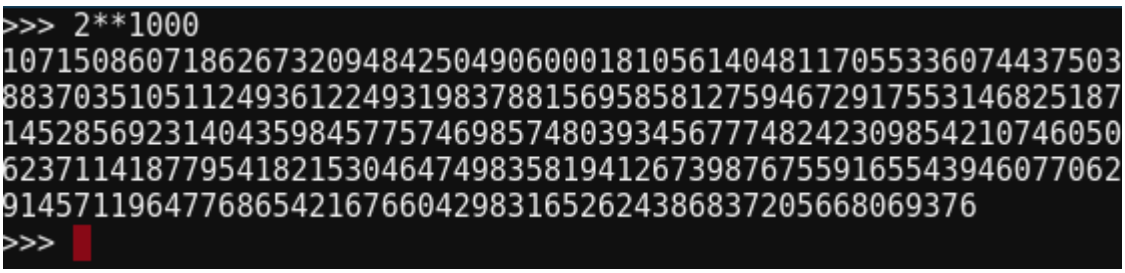
10

Para saber mais: Inteiros de precisão arbitrária

Vimos que podemos representar números muito grandes usando a representação de ponto flutuante. Essa representação tem um grande ganho em eficiência mas causa erros de arredondamento.

Mesmo assim, se você programa com a linguagem Python, verá que você pode representar números gigantescos, como dois elevado a mil, sem erros de arredondamento. (Você pode testar isso no [interpretador online de Python (https://www.python.org/shell/)](https://www.python.org/shell/).)



Como, então, ele consegue armazenar números tão grandes? Ele usa mais de 1000 bits para armazenar todos os seus números? A resposta para isso é: **Inteiros de precisão arbitrária**.

Essa técnica, também chamada de *BigInteger* ou *Bignum*, armazena números sem restrição de posição. A ideia é, em vez de fixar a mesma quantidade de bits para representar todos os números, utilizar uma quantidade variável.

Isso pode ser feito guardando cada dígito do número numa lista e as contas são realizadas dígito a dígito, como aprendemos na escola. Para otimizar, as implementações não usam a nossa base decimal (dígitos de 0 a 9) mas uma base de 2^30 (“dígitos” de 0 a 2^30 - 1). Com isso, as operações dígito a dígito são feita como operações de inteiros convencionais.

Dessa forma, não há uma restrição para o tamanho que esses números podem ter (além do limite de memória que o seu computador deve ter para armazenar tudo isso). Por isso, é dito que esses números tem precisão arbitrária. Várias linguagens (https://en.wikipedia.org/wiki/List_of_arbitrary-precision_arithmetic_software) possuem implementações desse tipo de número mas, diferente do Python, costumam ser bibliotecas auxiliares da linguagem.

De qualquer forma, se um *Bignum* tem precisão infinita por que não usamos ele como o padrão para tudo? Isso acontece porque fazer contas com ele é muito lento. Quanto mais dígitos o número tiver, mais computações vai demorar para fazer todos os cálculos. Além disso, não precisamos de uma precisão tão extrema na maioria das aplicações práticas. Por isso, os números de precisão arbitrária são bem importantes mas não podem ser aplicados em qualquer situação.

Se você quer saber mais sobre o assunto, você pode ler o artigo, um pouco avançado, do Arten Golubin (https://rushter.com/blog/python-integer-implementation/) (em inglês) sobre a implementação do Bignum em Python. Lá tem vários detalhes de implementação e é destrinchado como algumas contas são feitas.