

Lição 2

Números inteiros e de ponto flutuante

1. Aritmética inteira

Já conhecemos os seguintes operadores que podem ser aplicados a números: **+**, **-**, ***** e ******. O operador de divisão **/** para inteiros fornece um número real de ponto flutuante (um objeto do tipo **float**). A exponenciação ****** também retorna um **float** quando a potência é negativa:

```
print(17/3) # mostra 5.66666666667
print(2**4) # mostra 16
print(2**-2) # mostra 0.25
```

Há uma operação especial para divisão inteira onde o resto é descartado: **//**. A operação que resulta no resto dessa divisão tem o operador **%**. Ambas as operações sempre resultam num objeto do tipo **int**.

```
print(17/3) # mostra 5.66666666667
print(17//3) # mostra 5
print(17%3) # mostra 2
```

2. Números de ponto flutuante

Quando lemos um valor inteiro, lemos uma linha com **input()** e então convertemos uma *string* (cadeia de caracteres) para inteiro usando **int()**. Quando lemos um número real, precisamos converter a *string* para **float** usando **float()**:

```
x = float(input())
print(x)
```

Números reais de ponto flutuante com valor absoluto muito grande ou muito pequeno podem ser escritos usando uma notação científica. Por exemplo, a distância da Terra ao Sol é **1.496x10¹¹**, ou **1.496e11** em Python. A massa de uma molécula de água é **2.99x10⁻²³**, ou **2.99e-23** em Python.

Pode-se converter objetos **float** em objetos **int** descartando a parte fracionária usando a função **int()**. Esta função demonstra o chamado comportamento de arredondamento para zero:

```
print(int(1.3)) # mostra 1
print(int(1.7)) # mostra 1
print(int(-1.3)) # mostra -1
print(int(-1.7)) # mostra -1
```

Há também uma função **round()** que realiza o arredondamento usual:

```
print(round(1.3))    # mostra 1
print(round(1.7))    # mostra 2
print(round(-1.3))   # mostra -1
print(round(-1.7))   # mostra -2
```

Os números reais de ponto flutuante não podem ser representados com precisão exata devido às limitações do hardware. Isso pode levar a efeitos complicados. Consulte a documentação do Python para obter os detalhes.

```
print(0.1+0.2)  # mostra 0.30000000000000004
```

3. Módulo matemático

Python tem muitas funções auxiliares para cálculos com reais de ponto flutuante. Eles podem ser encontrados no módulo matemático.

Para usar este módulo, primeiro precisamos importá-lo escrevendo a seguinte instrução no início do programa:

```
import math
```

Por exemplo, se quisermos encontrar um valor máximo para **x** - o menor inteiro não menor que **x** - chamamos a função apropriada do módulo matemático: **math.ceil(x)**. A sintaxe para chamar funções de módulos é sempre a mesma: **nome_do_módulo.nome_da_função (argumento_1, argumento_2, ...)**

```
import math

x = math.ceil(4.2)
print(x)
print(math.ceil(1 + 3.8))
```

Há outra maneira de usar funções de módulos: para importar certas funções nomeando-as:

```
from math import ceil

x = 7 / 2
y = ceil(x)
print(y)
```

Algumas das funções que lidam com números - **int()**, **round()** e **abs()** (valor absoluto também conhecido como módulo) - são embutidas e não requerem nenhuma importação.

Todas as funções de qualquer módulo Python padrão estão documentadas no site oficial do Python. Aqui está a descrição do módulo matemático. A descrição de algumas funções é dada:

| Função | Descrição da função |
|----------------------------|---|
| Arredondamento | |
| round(x) | Soma 0.5 ao número real x e retorna a parte inteira da soma. |
| floor(x) | Retorna o piso de x, o maior inteiro menor ou igual a x. |
| ceil(x) | Retorna o teto de x, o menor inteiro maior ou igual a x. |
| Raízes e logaritmos | |
| sqrt(x) | Retorna a raiz quadrada de x |
| log(x) | Com um argumento, retorna o logaritmo natural de x (para a base e). Com dois argumentos, retorna o logaritmo de x para a base fornecida |
| e | A constante matemática $e = 2.71828 \dots$ |
| Trigonometria | |
| sin(x) | Retorna o seno de x radianos |
| asin(x) | Retorna o arco seno de x, em radianos |
| pi | A constante matemática $\pi = 3,1415 \dots$ |
| Outras | |
| abs(x) | Retorna o inteiro x sempre com o sinal positivo |
| int(x) | Se x for real, converte x para um inteiro, perdendo a fração. Sendo x uma cadeia de caracteres (string) representando um inteiro, converte x para um inteiro. |
| float(x) | Se x for inteiro, converte x para um real, com a fração zero. Sendo x uma cadeia de caracteres (string) representando um real, converte x para um real. |