

Interpretador Python – elementos básicos

SSC0301
Prof Delamaro

Elementos básicos

- Números
- Variáveis
- Strings
- Funções

Números

- Inteiro (*int*)
- Ponto flutuante (*float*)
- Tipo diz respeito à representação e não ao valor
- 3.0 é float mas 3 é int

Operações

- Binárias (2 operandos)
 - + - * /
 - ** (potência)
- Unárias (1 operando)
 - + -

Operações

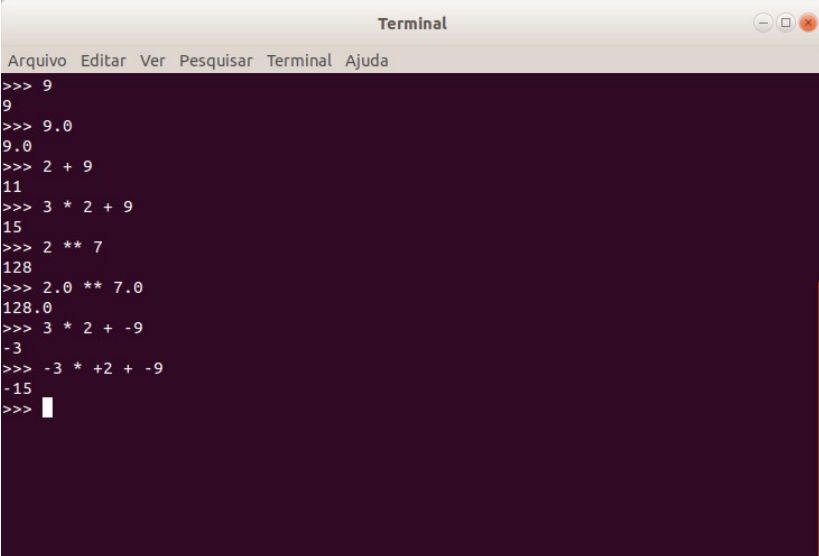
- Binárias (2 operandos)

- + - * /

- ** (potência)

- Unárias (1 operando)

- + -

A terminal window titled "Terminal" with a menu bar containing "Arquivo", "Editar", "Ver", "Pesquisar", "Terminal", and "Ajuda". The terminal shows a series of arithmetic operations and their results:

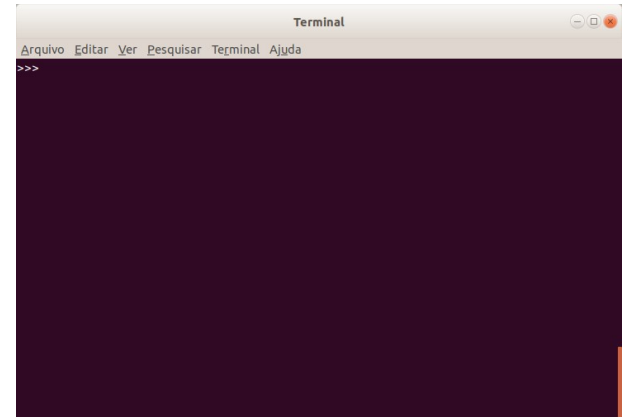
```
>>> 9
9
>>> 9.0
9.0
>>> 2 + 9
11
>>> 3 * 2 + 9
15
>>> 2 ** 7
128
>>> 2.0 ** 7.0
128.0
>>> 3 * 2 + -9
-3
>>> -3 * +2 + -9
-15
>>> |
```

Operações

- Tipos de números podem ser combinados
 - Dois números com mesmo tipo
 - Tipos diferentes → resultado é float
 - Divisão sempre resultado é float
- Existe uma precedência entre os operadores
 - **
 - Operadores unários
 - * e /
 - + e -

Operações

- Tipos de números podem ser combinados
 - Dois números com mesmo tipo
 - Tipos diferentes → resultado é float
 - Divisão sempre resultado é float
- Existe uma precedência entre os operadores
 - **
 - Operadores unários
 - * e /
 - + e -

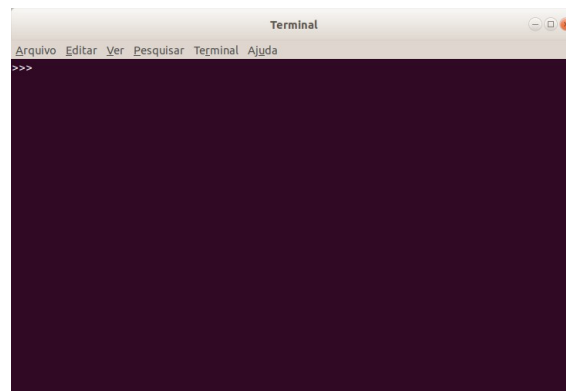


Operações inteiras

- Divisão inteira (//)
 - divide o primeiro operando pelo segundo, obtendo o quociente;
 - pega o mais próximo valor inteiro, que seja menor do que o quociente;
 - $13 // 4 \rightarrow 3$ $-13 // 4 \rightarrow -4$ $(13 / 4 \rightarrow 3.25)$
- Resto da divisão inteira (%)
 - $13 \% 4 \rightarrow 1$
 - $(A // B) * B + A \% B = A$

Operações inteiras

- Divisão inteira (//)
 - divide o primeiro operando pelo segundo, obtendo o quociente;
 - pega o mais próximo valor inteiro, que seja menor do que o quociente;
 - $13 // 4 \rightarrow 3$ $-13 // 4 \rightarrow -4$ $(13 / 4 \rightarrow 3.25)$
- Resto da divisão inteira (%)
 - $13 \% 4 \rightarrow 1$
 - $(A // B) * B + A \% B = A$



Precedência

Maior
precedência

Menor
precedência

-, +

***, /, //, %**

-, +

Exercícios

- Seção 8.2.1 do texto
- Apenas para se familiarizar com o interpretador

Variáveis

- São “locais” aonde podemos guardar valor (números, por enquanto)
- Variável não tem um tipo, mas o valor armazenado nela tem
- Para guardar um valor numa variável, usa-se o símbolo =
- $x = 10.0$
- $y = x ** 2$
- $y = y + 2$

Nomes de variáveis

- O que mais se usa: letras, dígitos e _
 - X, y, z k, delta, x2, x2, X1, X2, data_de_validade, imc
 - 1x, pes@, data-de-validade
- Nomes que não podem ser usados (palavras reservadas)
 - and, def, exec, if, not, return, assert, del, finally import, or, try, break, elif, for, in, pass, while, class, else, from, is, print, yield, continue, except, global, lambda, raise

Exercícios

- Seção 8.3.1

- Seção 8.6 – Método de Bhaskara

Bônus – expressão lambda

- Podemos guardar em uma variável, uma expressão
- Por exemplo, uma expressão que vai calcular a soma de dois números:
 - soma = lambda a, b: a + b
 - soma(10,4) → 14
- cubo = lambda x: x ** 3
- f = lambda x : 18 * x ** 2 - 4 * x + 13

Exercícios

- Agora, para vocês fazerem: Seção 8.8