

Expressões

1. Calcule o resultado das seguintes expressões. **Primeiramente, calcule o valor manualmente, sem o auxílio do computador.** Fique atento tanto ao valor resultante quanto ao tipo de dado desse valor! Depois, calcule o valor no Python, utilizando o modo interativo (shell). Caso o valor esperado seja diferente do valor obtido, explique a razão. O primeiro e o terceiro resultados são exemplos de como resolver.

Expressão	Valor esperado	Valor obtido	Explicação
$2 + 3$	5	5	
$3 * 4 + 2$	14	14	
$3 * 4 + 2.0$	14	14.0	Como tem a soma de um número de ponto flutuante na expressão, a resposta também foi um ponto flutuante.
$3 (4 + 2)$	18	TypeError: 'int' object is not callable	Na expressão está faltando um asterisco na frente do 3, para que assim, o Python possa ler como uma equação.
$5 / 10$	0.5	0.5	
$5 / 10.0$	0.5	0.5	
$5 // 10$	0	0	
$5 // 10.0$	0.0	0.0	
$5 // -10.0$	0.0	-1.0	O // traz o número mais próximo do piso do número dividido, logo, o número mais próximo e menor que zero seria o -1.
$5 / 9$	0.55	0.5555555555555556	
$2 ** 4$	16	16	
$4 ** 0.5$	2	2.0	Como um dos números era de ponto flutuante, o resultado também será de ponto flutuante.
$3 ** 2 ** 3$	729	6561	O Python fez a conta em pares e da direita para esquerda, então ficaria; 1°- $3 ** (2 ** 3)$; 2°- $3 ** 8$; 3°- 6561.
$- - 2$	2	2	
$+ - + 3$	-3	-3	
$- 3 * - 4$	12	12	
$/ 7$	Erro	SyntaxError: invalid syntax	
"Olá" + "mundo"	Olá mundo	'Olá mundo'	
"Olá" + 3	Erro	TypeError: can only concatenate str (not "int") to str	
"Olá" + True	Erro	TypeError: can only concatenate str (not "bool") to str	

True + 3	Erro	4	O True é tratado como o número 1.
False + 3	3	3	

2. Mostre duas formas em Python para inverter o sinal de um número, seja ele inteiro ou de ponto flutuante.

R: $2 / -1$ e $-2 * -1$.

3. Em tese, seria possível inverter o sinal de um número com a operação de divisão por -1. No entanto, na prática isso possui um efeito indesejável. Realize essa operação com um número de ponto flutuante e, depois, com um número inteiro, observando o resultado. O que aconteceu com o valor e o tipo de dado resultante?

R: Ambos os valores resultaram em números de ponto flutuante. A divisão transforma o número de forma automática em float.

4. Como seriam as expressões aritméticas para calcular o diâmetro, a circunferência e a área de um círculo? Podem considerar um círculo de raio igual a 7.

R: Diâmetro: $7 * 2$, Comprimento: $3.14 * 7 * 2$ e Área: $3.14 * 7 ** 2$

5. Como você poderia realizar a operação de módulo sem utilizar o operador de módulo? Mostre uma expressão aritmética equivalente. Considere os seguintes dados como exemplo:

Número 1	Número 2	Número 1 % Número 2
3	4	3
5	4	1

R: $3 - (3 // 4) * 4$ e $5 - (5 // 4) * 4$

6. Considere uma disciplina em que a nota final seja calculada a partir das seguintes avaliações: duas provas individuais (P1 e P2), um trabalho em grupo (Trabalho) e um conjunto de listas de exercícios (Listas). Cada prova, trabalho e lista de exercícios possui nota de 0,0 a 10,0. Cada prova tem peso 3 e a lista e trabalhos possuem peso 2 cada. A nota de Listas é a média harmônica das listas de exercícios, ou seja, a razão entre a quantidade n de notas e a soma do inverso de cada nota. A nota final será a média ponderada de p1, p2, trabalho e listas.

Considere que temos, por enquanto, três listas de exercícios apenas. Dessa forma, o cálculo da média geométrica seguirá a seguinte fórmula:

$$Listas = \frac{3}{\frac{1}{L1} + \frac{1}{L2} + \frac{1}{L3}}$$

Tudo isso exposto, construa uma expressão aritmética que calcule a média, considerando as seguintes notas: 5, 5 para P1, 6, 4 para P2, 5, 5 para Trabalho, 10, 0 para Lista 1, 3, 0 para Lista 2 e 3, 5 para Lista 3.

R: $((5.5*3) + (6.4*3) + (5.5*2) + ((3/((1/10) + (1/3) + (1/3.5)))*2))/10$

7. Tente calcular uma expressão que resulte em um número inteiro muito grande em Python, por exemplo, 123^{456} . Existe um limite quanto aos números inteiros que o Python consegue trabalhar?

R: Sim, existe um limite dependendo da memória RAM do computador.

8. Tente calcular uma expressão que resulte em um número de ponto flutuante muito grande, por exemplo $123 \cdot 0^{456}$. Existe um limite quanto aos números de ponto flutuante que o Python consegue trabalhar?

R: Sim, existe esse número aproximadamente 1.8×10^{308} .

9. Em um sistema de arquivo típico do Linux, um arquivo possui regras quanto às permissões para leitura (read), escrita (write) e execução (execution). Essas permissões podem ser configuradas para o proprietário

do arquivo (owner), para um grupo de usuários vinculado ao arquivo (group) e para todo mundo (all). As permissões para cada um desses elementos é definida por um número de 0 à 7, o que requer 3 bits. De modo geral temos o seguinte significado para cada bit:

Bit 0	Execução
Bit 1	Escrita
Bit 2	Leitura

Por exemplo, para o número 6, cuja representação binária é 110, teríamos que foi dada permissão de leitura (bit 2 está ativado) e escrita (bit 1 está ativado), mas não foi dada permissão de execução (afinal o bit 0 está desativado). Como cada arquivo tem três elementos (owner, group e all) com permissões possivelmente distintas, nós representamos as permissões numericamente com três números. Por exemplo, um arquivo com permissão 777 seria equivalente a permitir a leitura, escrita e execução por qualquer pessoa, grupo e proprietário. No entanto, de modo geral as permissões são mais restritivas, sendo vedada a permissão para todos para qualquer operação, permitindo-se a leitura e execução para o grupo e se permitindo qualquer operação pelo proprietário. Isso seria equivalente à 750.

Agora suponha que você deseja habilitar a leitura do arquivo para qualquer usuário. Uma maneira de fazer isso seria atribuir o valor 1 para o bit 2 da respectiva permissão. Por exemplo, poderíamos atribuir o número 754. No entanto, isso poderia alterar, de forma indesejada, as outras permissões. Caso um arquivo estivesse configurado apenas para escrita para qualquer pessoa (222), alterar para 754 não seria o correto, porque o grupo e os demais usuários perderiam a permissão para escrita.

Assim, escreva para cada item abaixo uma expressão que utiliza operações bit a bit para alterar apenas o suficiente para cumprir o objetivo da alteração de permissão, preservando as demais permissões. Veja que isso independe do valor original das permissões do arquivo. Considere os seguintes exemplos:

Leiametext, 443

Calendário.png, 352

a) Permita que um usuário qualquer leia e execute um arquivo.

R:

#Leiametext

443 or 447

100 100 011 or 100 100 111

#Calendário.png

352 or 357

011 101 010 or 011 101 111

b) Não permita que o grupo consiga ler um arquivo.

R:

#Leiametext

443 and 403

100 100 011 and 100 000 011

#Calendário.png

352 and 312

011 101 010 and 011 001 010