

# AC1 - Linguagem de Programação

Total de pontos 6/10

As questões contidas nessa atividade estão relacionadas aos conteúdos das Partes 01, 02, 03, 04 e 05.

O e-mail do participante (**fabiana.campanari@aluno.faculdadeimpacta.com.br**) foi registrado durante o envio deste formulário.

Tipos de dados e identificadores

1 de 2 pontos



✗ A respeito dos tipos de dados básicos da linguagem de programação Python, marque a alternativa FALSA. \*0/1

- ☒ É possível realizar operações aritméticas e relacionais entre os tipos int e float sem a necessidade de conversão explícita. ✗
- ☐ O tipo str é usado para representar textos, e para ser concatenado com um número, é preciso explicitamente converter o número para string.
- ☐ O tipo bool é usado para representar valores lógicos (True ou False)
- ☐ O tipo int é usado para representar os números positivos e o tipo float os números decimais.
- ☐ O tipo int é usado para representar números inteiros e o tipo float os números reais

#### Resposta correta

- ☒ O tipo int é usado para representar os números positivos e o tipo float os números decimais.

#### Feedback

*O tipo int representa todos os números inteiros, não apenas os positivos. (conteúdo abordado no capítulo 2 da Unidade 1 da apostila)*



✓ Marque o único identificador válido, seguindo as regras de sintaxe do Python 3. \*1/1

- ☐ nome-estudante
- ☐ nome+sobrenome
- ☐ megan fox
- ☒ \_\_user\_name
- ☐ 2nd\_player



#### Feedback

As regras são:

\* conter apenas letras, números e sublinhados, sem nenhum outro caractere especial.

\* não começar com número

\* não ser uma palavra reservada, veja na shell >>> help('keywords')

Conteúdo abordado no Capítulo 2 da Unidade 1 da apostila.

Expressões

3 de 4 pontos



✓ Marque a alternativa que apresenta a definição mais apropriada, em ordem, para os operadores aritméticos binários a seguir: +, -, \*, /, //, % e \*\*.

\*1/1

- ☐ adição, subtração, multiplicação, divisão real, divisão inteira, exponenciação e resto, respectivamente.
- ☐ adição, subtração, multiplicação, divisão inteira, divisão real, resto e exponenciação, respectivamente.
- ☐ adição, subtração, multiplicação, resto, divisão real, divisão inteira e exponenciação, respectivamente.
- ☐ adição, subtração, exponenciação, divisão real, divisão inteira, resto e multiplicação, respectivamente.
- ☒ adição, subtração, multiplicação, divisão real, divisão inteira, resto e exponenciação, respectivamente. ✓

#### Feedback

*Conteúdo abordado no Capítulo 3 da Unidade 1 da apostila.*



✓ Na expressão da imagem a seguir, há quantos operadores binários? \* 1/1

$(-2 + 3 * 24 // 5) ** 2 / -3$

☐ 7

☐ 3

☐ 6

☒ 5

☐ 4



#### Feedback

Operador binário é aquele que atua sobre dois operandos. Nesta expressão há 5 operadores deste tipo: + ; \* ; // ; \*\* e / ;  
Os operadores (-) são nesta expressão operadores unários, pois atuam sobre apenas um operando (invertendo seu sinal).

Conteúdo abordado no Capítulo 3 da Unidade 1 da apostila.



✗ Considerando as regras de precedência de operadores em Python, qual alternativa tem como resultado um número natural primo. \*0/1

Observação: números naturais primos são os números naturais maiores do que 1 e que têm apenas dois divisores naturais, o 1 e o próprio número.

☐ 100 / 30

☒ 1 + 4 \* 5 + 4

✗

☐ 40 // 5 \* (19 % 5)

☐ 3 \*\* 2 \* 3 - 4

☐ 5 + 2 \* 3 + 10

Resposta correta

☒ 3 \*\* 2 \* 3 - 4

#### Feedback

$1 + 4 * 5 + 4 \rightarrow 25$  (não é primo, tem mais de dois divisores naturais)

$5 + 2 * 3 + 10 \rightarrow 21$  (não é primo, tem mais de dois divisores naturais)

$40 // 5 * (19 \% 5) \rightarrow 32$  (não é primo, tem mais de dois divisores naturais)

$1 + 2 * 3 ** 2 \rightarrow 23$  (é primo, tem apenas dois divisores naturais: 1 e 23)

$100 / 30 \rightarrow 3.33$  (não é um número natural, portanto não pode ser um natural primo)



✓ Qual o tratamento dado pelo Python para o encadeamento de operadores relacionais ( $x < y < z$ ) \*1/1

- ☐ Tal encadeamento não é permitido em Python, pois os operadores relacionais são não associativos, de modo que a expressão do enunciado resultaria em um erro de sintaxe.
- ☒ A expressão é traduzida para um encadeamento de operações binárias, repetindo o operando intermediário e unindo as expressões com o operador lógico and ( $x < y$  and  $y < z$ ). ✓
- ☐ A expressão é avaliada da esquerda para a direita, pois os operadores relacionais possuem a mesma ordem de prioridade ( $(x < y) < z$ ).
- ☐ A expressão é avaliada da direita para a esquerda, pois os operadores relacionais possuem a mesma ordem de prioridade ( $x < (y < z)$ ).
- ☐ O interpretador do Python irá abrir uma janela para perguntar ao usuário se a expressão deve ser avaliada da esquerda para a direita ou da direita para a esquerda.

**Feedback**

*Conteúdo abordado no Capítulo 4 da Unidade 2 da apostila.*

Estruturas de seleção

2 de 4 pontos



- ✓ ANULADA - O programa a seguir deve calcular a alíquota do seguro de um veículo alugado com base na idade e no tempo de carta do motorista. A política da empresa é cobrar uma taxa base de 5%, adicionando 3% caso o motorista tenha menos de 25 anos, e outros 2% caso o tempo de habilitação seja de 2 anos ou menos. Marque a alternativa que preenche corretamente as lacunas do código. \*1/1

```
idade = int(input('sua idade: '))
tempo_cnh = int(input('Tempo de carta: '))

aliquota = 0.05

if _____:
    aliquota += 0.03

if _____:
    aliquota += 0.02

print(f'Sua aliquota é: {aliquota:.1%}')
```

- ☐ tempo\_cnh <=2 ; idade < 24 ; respectivamente
- ☐ idade <= 25 ; tempo\_cnh <= 3 ; respectivamente
- ☐ idade < 24 ; tempo\_cnh <= 2 ; respectivamente
- ☐ tempo\_cnh < 3 ; idade < 25 ; respectivamente
- ☒ idade < 25 ; tempo\_cnh <= 2 ; respectivamente



#### Feedback

A condição do primeiro if deve ser idade < 25, ou então idade <= 24;  
A condição do segundo if deve ser tempo\_cnh < 3, ou então tempo\_cnh <= 2;

A duas condições em cada caso são equivalentes, pois estamos falando de valores do tipo inteiro, portanto dizer que  $x < 10$  ou que  $x \leq 9$  tem o mesmo efeito.





- ✗ Para verificar se um número natural  $n$  é par, basta dividi-lo por dois e verificar se o resto da divisão é zero, caso afirmativo, isso significa que o número é divisível por dois e, evidentemente, é par. Para completar adequadamente o código ilustrado na imagem a seguir, pode-se preencher a lacuna com a expressão  $(n \% 2 == 0)$ , ou com outra expressão equivalente. Analise as alternativas a seguir e selecione a expressão que NÃO é uma expressão equivalente válida para completar o código da imagem. \*0/1

Observação: Lembre-se que no resto da divisão de um número natural por 2 só há duas possibilidades de resultado: resto igual a 0 (zero) ou resto igual a 1 (um).

```
n = int(input('Dígite um número: '))

if _____ :
    print(f'{n} é par!')
else:
    print(f'{n} é ímpar!')
```

- ☒ not ( $n \% 2 != 0$ ) ✗
- ☐ not not ( $n \% 2 == 0$ )
- ☐ not ( $n \% 2 == 1$ )
- ☐  $n \% 2 != 1$
- ☐  $n \% 2 == 1$

Resposta correta

- ☒  $n \% 2 == 1$

### Feedback

Conteúdo abordado no Capítulo 5 da Unidade 2 da apostila.

Basta analisar a codificação e verificar que a condição da seleção deve resultar em verdadeiro (True) quando o número natural  $n$  for par e em falso (False) quando  $n$  for ímpar.

Dentre as alternativas, a única que não resulta em True quando  $n$  é par é aquela com expressão  $(n \% 2 == 1)$ , que justamente resulta em True quando  $n$  é ímpar e, evidentemente, em False quando  $n$  é par.



- ✓ O código a seguir deve ler um número inteiro e exibir se ele é ou não divisível por 2 e por 7 simultaneamente (por exemplo, 14 é divisível por 2 e por 7; 20 é divisível por 2, mas não por 7; e 21 é divisível por 7, mas não por 2). Marque a alternativa que preenche corretamente as lacunas do código a seguir. \*1/1

```
n = _____(_____('Digite um número inteiro: '))  
  
if n % 2 == 0 and n % 7 == 0:  
    print(_____)  
else:  
    print(_____)
```

- ☐ input ; int ; 'não é divisível' ; 'é divisível'
- ☐ int ; input ; 'não é divisível' ; 'é divisível'
- ☒ int ; input ; 'é divisível' ; 'não é divisível' ✓
- ☐ input ; int ; 'é divisível' ; 'não é divisível'
- ☐ float ; int ; 'é divisível' ; 'não é divisível'

#### Feedback

A primeira lacuna deve receber a função `int` e a segunda lacuna recebe a função `input`. O texto a ser exibido no bloco do `if` é "é divisível", e o texto a ser exibido no bloco do `else` é: "não é divisível".



- ✗ O programa da imagem a seguir deverá receber como entradas o preço unitário de um produto e a quantidade comprada e, em seguida, verificar se o cliente tem direito ao desconto, calculando o novo valor final em caso afirmativo. Regras para o desconto: é oferecido um desconto de 8% quando (i) o valor final da compra é superior a 200 reais ou (ii) a quantidade comprada é de pelo menos 25 unidades do produto. Marque a alternativa que completa corretamente as lacunas do código para atender as condições do desconto. \*0/1

```
valor = float(input('Preço unitário: '))
qtd = int(input('Quantidade comprada: '))

total = valor * qtd

if _____:
    desconto = _____
    total ____ desconto

print(f'Valor final: R$ {_____:.2f}')
```

- ☒ qtd >= 25 or valor > 200 ; 0.08 \* total ; -= ; total ✗
- ☐ qtd >= 25 or total > 200 ; 0.08 \* total ; -= ; total
- ☐ qtd >= 25 and valor > 200 ; 0.08 \* total ; -= ; total
- ☐ qtd > 25 or total > 200 ; 0.8 \* total ; -= ; desconto
- ☐ qtd > 25 and total > 200 ; 0.08 \* total ; += ; desconto

Resposta correta

- ☒ qtd >= 25 or total > 200 ; 0.08 \* total ; -= ; total

#### Feedback

A quantidade deve ser pelo menos 25, portanto:  $qtd \geq 25$   
O total deve ser superior a 200, portanto:  $total > 200$   
Basta que uma das condições seja verdade para que o desconto seja oferecido, portanto devem ser unidas com o operador: or

O desconto é de 8%, portanto  $(8/100) * total \rightarrow 0.08 * total$   
A atualização do valor total deve ser feita com o operador de atribuição composto "menos



*igual": -=*

*O valor final da compra, a ser exibido na tela, está na variável: total*

Este formulário foi criado em FACULDADE IMPACTA DE TECNOLOGIA - FIT.

Google Formulários



