

ALGORITMOS E LÓGICA DE PROGRAMAÇÃO

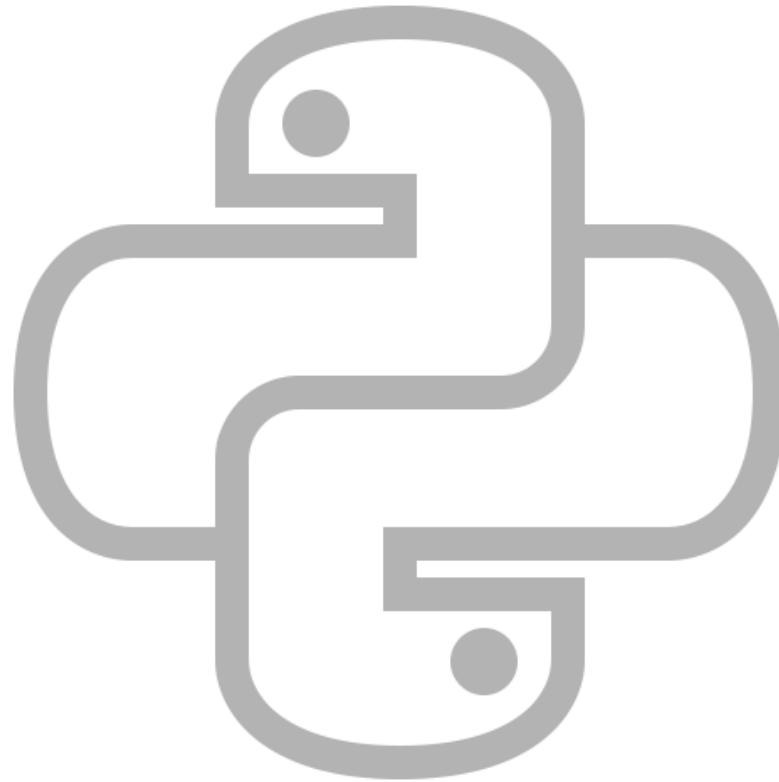
AULA 4: ESTRUTURAS DE SELEÇÃO ANINHADAS E ENCADEADAS

DO QUE VAMOS FALAR

- 1. Estruturas de seleção encaixadas
- 2. Estruturas de seleção aninhadas
- 3. Estruturas de seleção encadeadas
- 4. Estruturas de seleção com seleções aninhadas e encadeadas
- 5. Exercícios extras



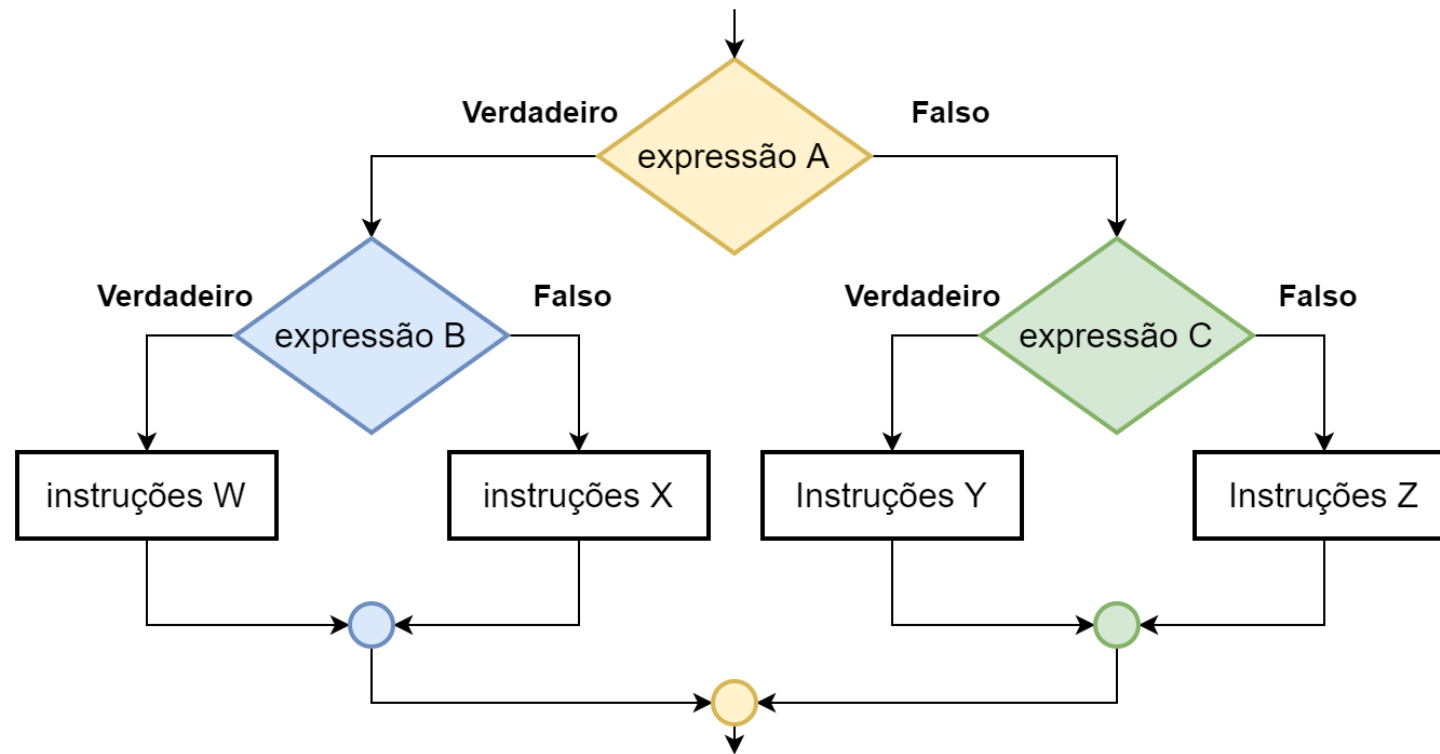
1. ESTRUTURAS DE SELEÇÃO ENCAIXADAS



ESTRUTURAS DE SELEÇÃO ENCAIXADAS

É possível que uma estrutura de seleção esteja **contida em outra**, isto é, que uma estrutura de seleção esteja no bloco de instruções de outra.

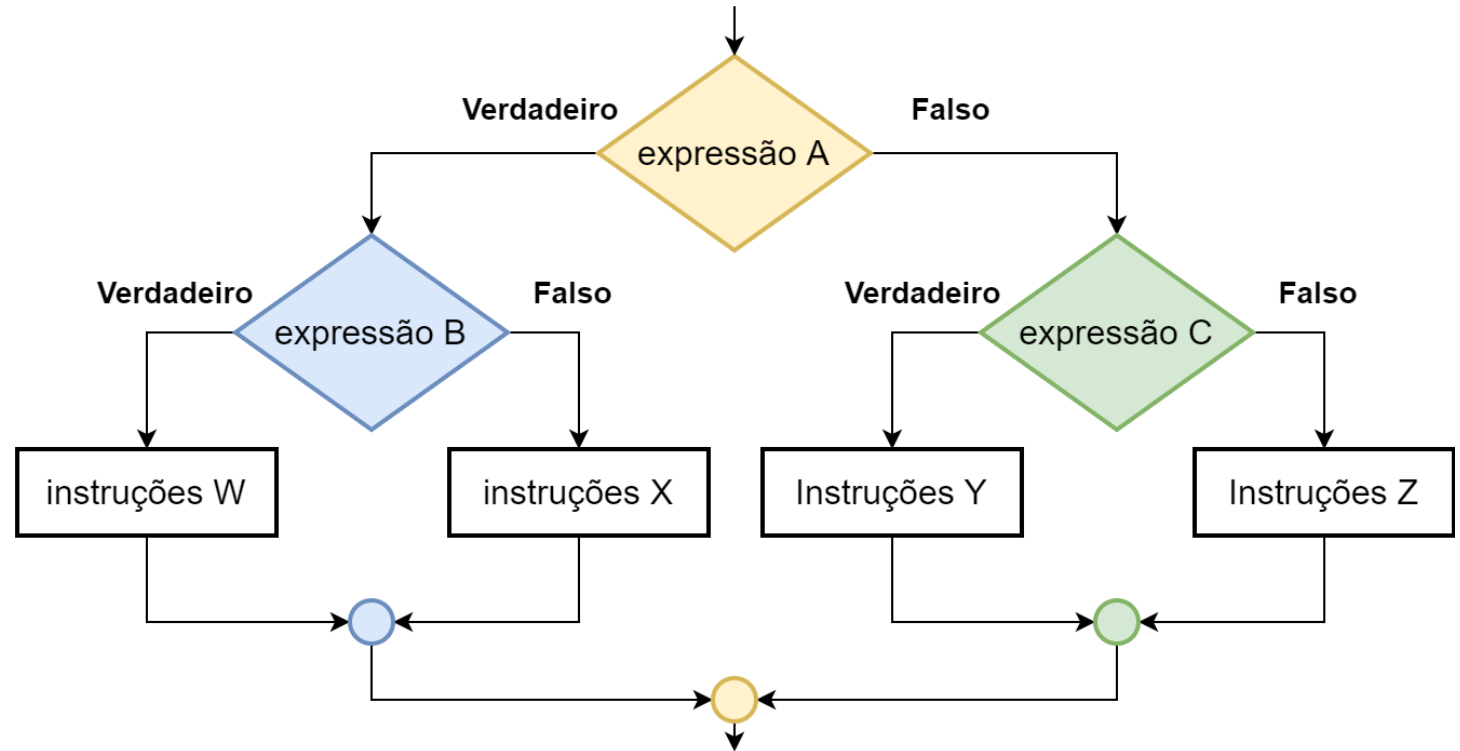
Na ilustração, temos duas estruturas de seleção encaixadas, uma com a **expressão B** e outra com a **expressão C**.



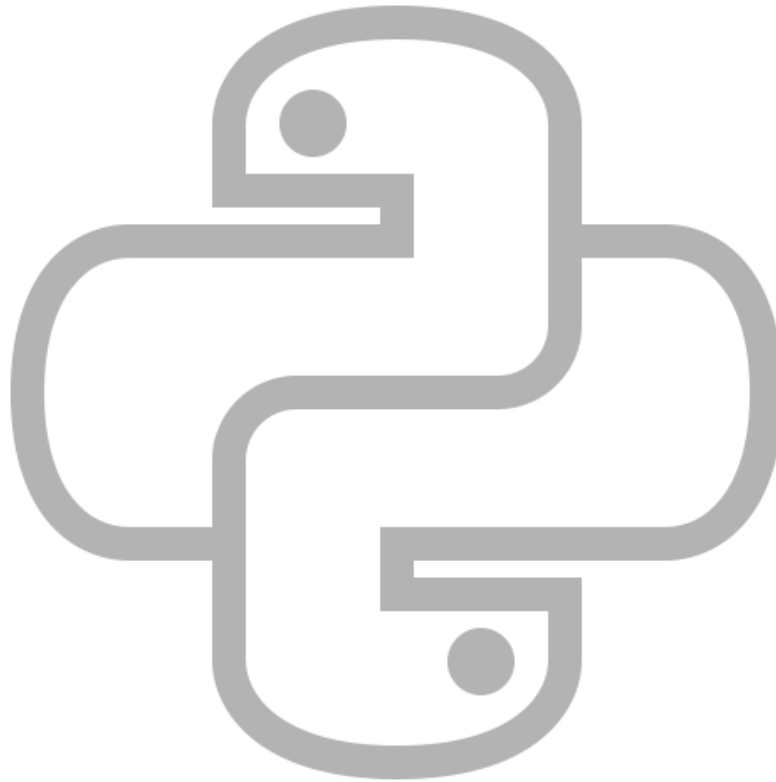
ESTRUTURAS DE SELEÇÃO ENCAIXADAS

...

```
if expressão A:  
    if expressão B:  
        instruções W  
    else:  
        instruções X  
else:  
    if expressão C:  
        instruções Y  
    else:  
        instruções Z  
...
```



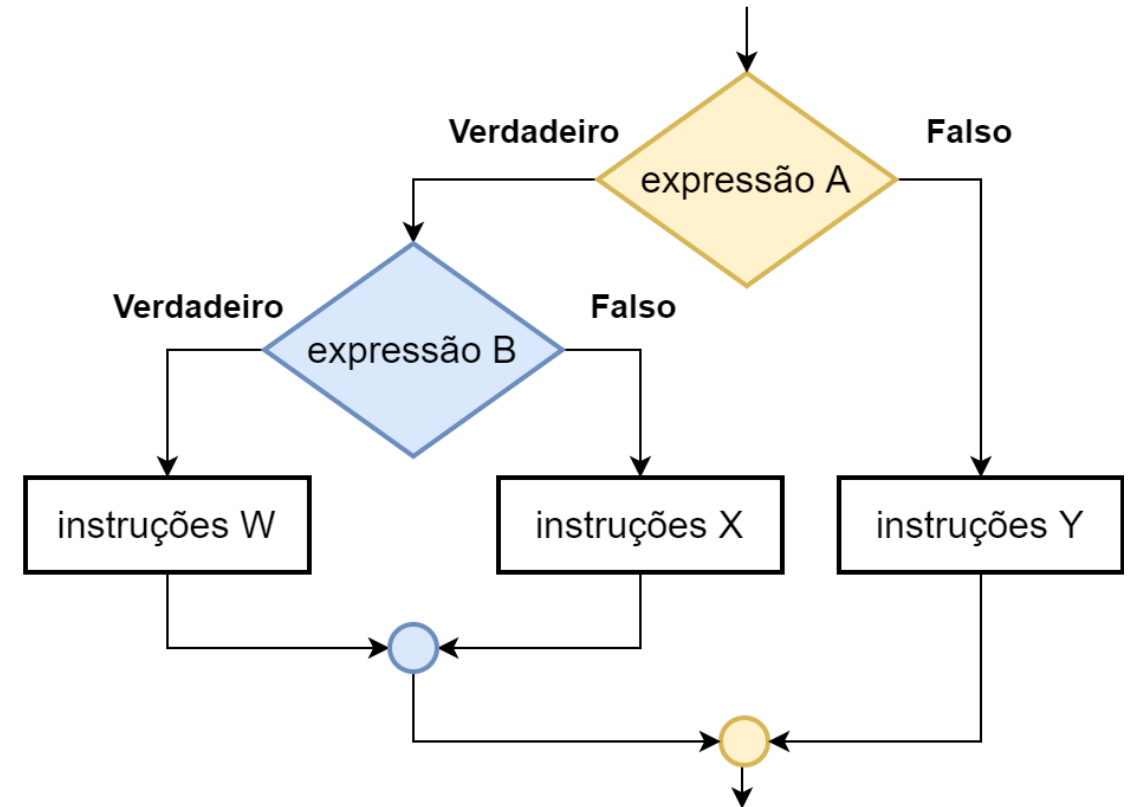
2. ESTRUTURA DE SELEÇÃO ANINHADA



ESTRUTURA DE SELEÇÃO ANINHADA

Quando uma estrutura de **seleção B** está contida no bloco de instruções do fluxo 'verdadeiro' de uma estrutura de **seleção A**, dizemos que **B está aninhada** em relação à A.

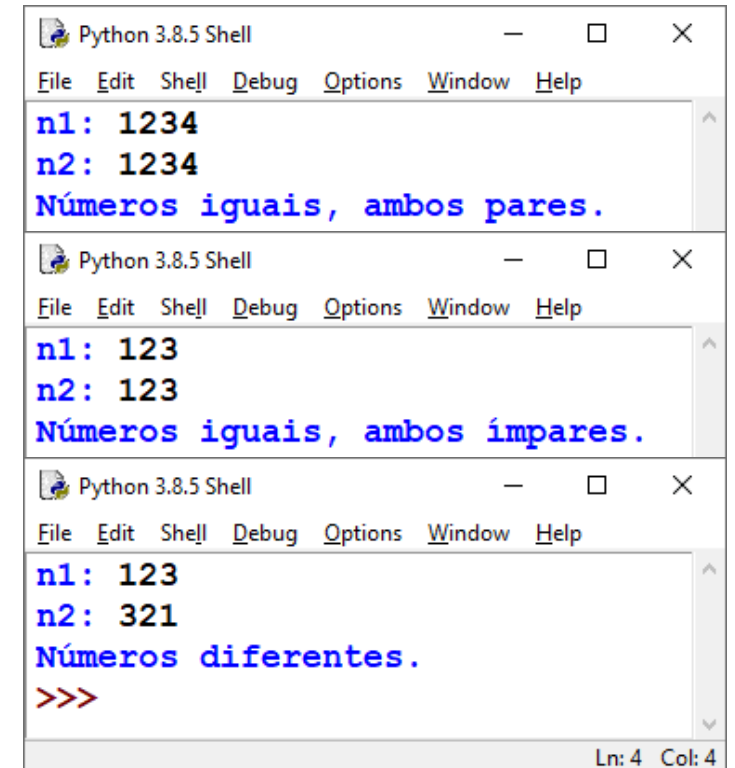
```
...  
if expressão A:  
    if expressão B:  
        instruções W  
    else:  
        instruções X  
else:  
    instruções Y  
...
```



ESTRUTURA DE SELEÇÃO ANINHADA

[EXEMPLO 1] Crie um programa que receba dois números inteiros e exiba uma mensagem indicando se os números são iguais ou diferentes e, caso sejam iguais, se são pares ou ímpares.

```
n1 = int(input('n1: '))
n2 = int(input('n2: '))
if n1==n2:
    print('Números iguais,', end=' ')
    if n1%2==0:
        print('ambos pares.')
    else:
        print('ambos ímpares.')
else:
    print('Números diferentes.')
```



The image shows three screenshots of a Python 3.8.5 Shell window, demonstrating the execution of the program for different inputs. Each window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The first window shows the input 'n1: 1234' and 'n2: 1234', resulting in the output 'Números iguais, ambos pares.'. The second window shows the input 'n1: 123' and 'n2: 123', resulting in the output 'Números iguais, ambos ímpares.'. The third window shows the input 'n1: 123' and 'n2: 321', resulting in the output 'Números diferentes.'. The status bar at the bottom of the third window indicates 'Ln: 4 Col: 4'.

```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
n1: 1234
n2: 1234
Números iguais, ambos pares.

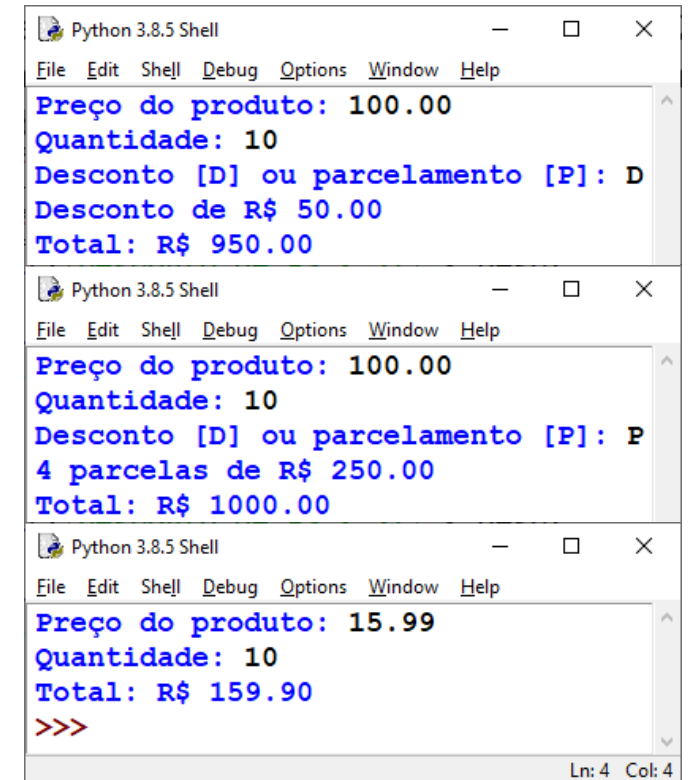
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
n1: 123
n2: 123
Números iguais, ambos ímpares.

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
n1: 123
n2: 321
Números diferentes.
>>>
Ln: 4 Col: 4
```


ESTRUTURA DE SELEÇÃO ANINHADA

[EXEMPLO 2] Crie um programa que receba o preço de um produto e a quantidade comprada. Se o total for maior do que R\$ 200,00 o cliente optará por um desconto de 5% ou parcelamento em 4 vezes sem juros. O programa deverá exibir o valor do desconto ou das parcelas, se aplicável, e o total final.

```
preco = float(input('Preço do produto: '))
qtd = int(input('Quantidade: '))
total = preco * qtd
if total > 200.00:
    opcao = input('Desconto [D] ou parcelamento [P]: ')
    if opcao=='D':
        desc = 0.05*total
        total -= desc
        print('Desconto de R$ %.2f' % desc)
    else:
        print('4 parcelas de R$ %.2f' % (total/4))
print('Total: R$ %.2f' % total)
```



The image shows three sequential screenshots of a Python 3.8.5 Shell window, demonstrating the program's execution for different inputs. Each window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The first screenshot shows the user inputting a price of 100.00 and a quantity of 10, selecting option 'D' for a 5% discount, resulting in a total of 950.00. The second screenshot shows the same inputs but selecting option 'P' for 4 installments, resulting in a total of 1000.00. The third screenshot shows a price of 15.99 and a quantity of 10, with the program outputting the installment amount and the total of 159.90. The prompt '>>>' is visible at the bottom of the third window.

```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Preço do produto: 100.00
Quantidade: 10
Desconto [D] ou parcelamento [P]: D
Desconto de R$ 50.00
Total: R$ 950.00

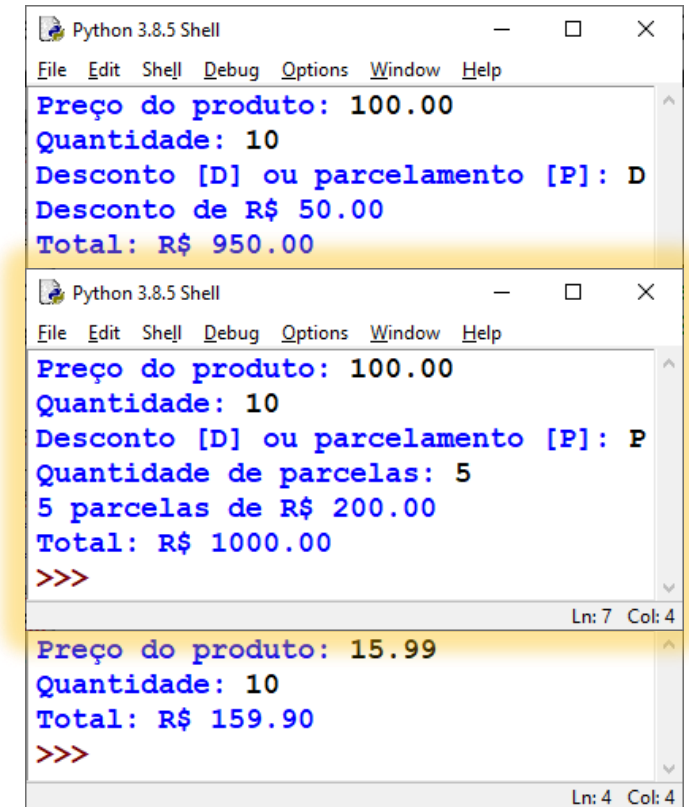
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Preço do produto: 100.00
Quantidade: 10
Desconto [D] ou parcelamento [P]: P
4 parcelas de R$ 250.00
Total: R$ 1000.00

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Preço do produto: 15.99
Quantidade: 10
Total: R$ 159.90
>>>
Ln: 4 Col: 4
```

ESTRUTURA DE SELEÇÃO ANINHADA

[EXERCÍCIO 1] Altere o programa do Exemplo 2 de modo que o cliente possa escolher a quantidade de parcelas. O programa deverá funcionar idêntico ao ilustrado nas figuras.

```
preco = float(input('Preço do produto: '))
qtd = int(input('Quantidade: '))
total = preco * qtd
if total >= 200.00:
    opcao = input('Desconto [D] ou parcelamento [P]: ')
    if opcao=='D':
        desc = 0.05*total
        total -= desc
        print('Desconto de R$ %.2f' % desc)
    else:
        p = int(input('Quantidade de parcelas: '))
        print('%d parcelas de R$ %.2f' % (p, total/p))
print('Total: R$ %.2f' % total)
```



```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Preço do produto: 100.00
Quantidade: 10
Desconto [D] ou parcelamento [P]: D
Desconto de R$ 50.00
Total: R$ 950.00

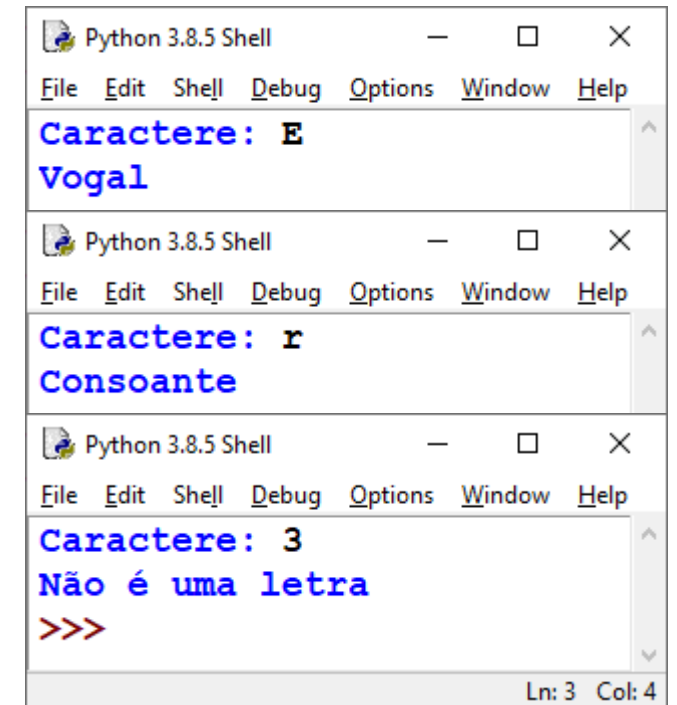
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Preço do produto: 100.00
Quantidade: 10
Desconto [D] ou parcelamento [P]: P
Quantidade de parcelas: 5
5 parcelas de R$ 200.00
Total: R$ 1000.00
>>>
Ln: 7 Col: 4

Preço do produto: 15.99
Quantidade: 10
Total: R$ 159.90
>>>
Ln: 4 Col: 4
```

ESTRUTURA DE SELEÇÃO ANINHADA

[EXERCÍCIO 2] Crie um programa que receba um caractere `c` e exiba uma mensagem indicando se `c` é uma vogal, uma consoante ou se não é uma letra. Considere apenas letras de nosso alfabeto ('A'...'Z' e 'a'...'z') e sem acentuação.

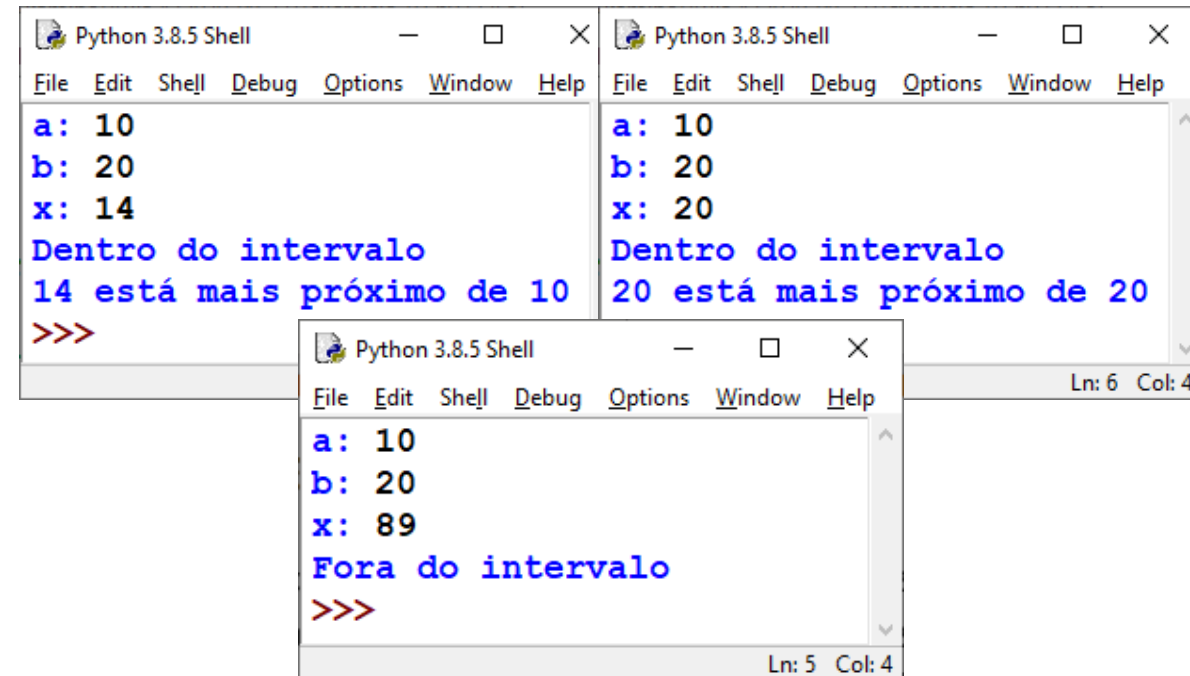
```
c = input('Caractere: ')
if 'a'<=c<='z' or 'A'<=c<='Z':
    if c=='a' or c=='e' or c=='i' or c=='o' or c=='u' or \
       c=='A' or c=='E' or c=='I' or c=='O' or c=='U':
        print('Vogal')
    else:
        print('Consoante')
else:
    print('Não é uma letra')
```



ESTRUTURA DE SELEÇÃO ANINHADA

[EXERCÍCIO 3] Crie um programa que receba três inteiros a , b e x , onde a e b são os limites inferior e superior de um intervalo de inteiros, respectivamente, com $a < b$. O programa exibirá uma mensagem indicando se x está no intervalo $[a..b]$ e, caso esteja, indicar se x está mais próximo de a ou de b (adote que jamais dará empate).

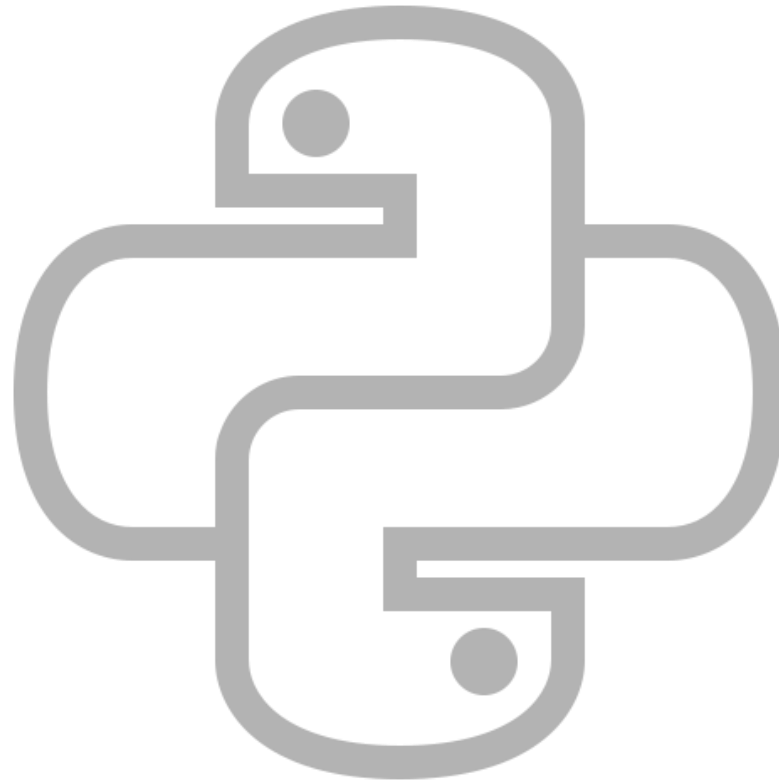
```
a = int(input('a: '))
b = int(input('b: '))
x = int(input('x: '))
if a<=x<=b:
    print('Dentro do intervalo')
    if x-a < b-x:
        print(x, 'está mais próximo de', a)
    else:
        print(x, 'está mais próximo de', b)
else:
    print('Fora do intervalo')
```



The image shows three screenshots of a Python 3.8.5 Shell window, illustrating the program's output for different inputs. The first two screenshots are side-by-side at the top, and the third is centered below them.

- Top Left Screenshot:** Inputs are $a=10$, $b=20$, and $x=14$. The output is "Dentro do intervalo" and "14 está mais próximo de 10".
- Top Right Screenshot:** Inputs are $a=10$, $b=20$, and $x=20$. The output is "Dentro do intervalo" and "20 está mais próximo de 20".
- Bottom Center Screenshot:** Inputs are $a=10$, $b=20$, and $x=89$. The output is "Fora do intervalo".

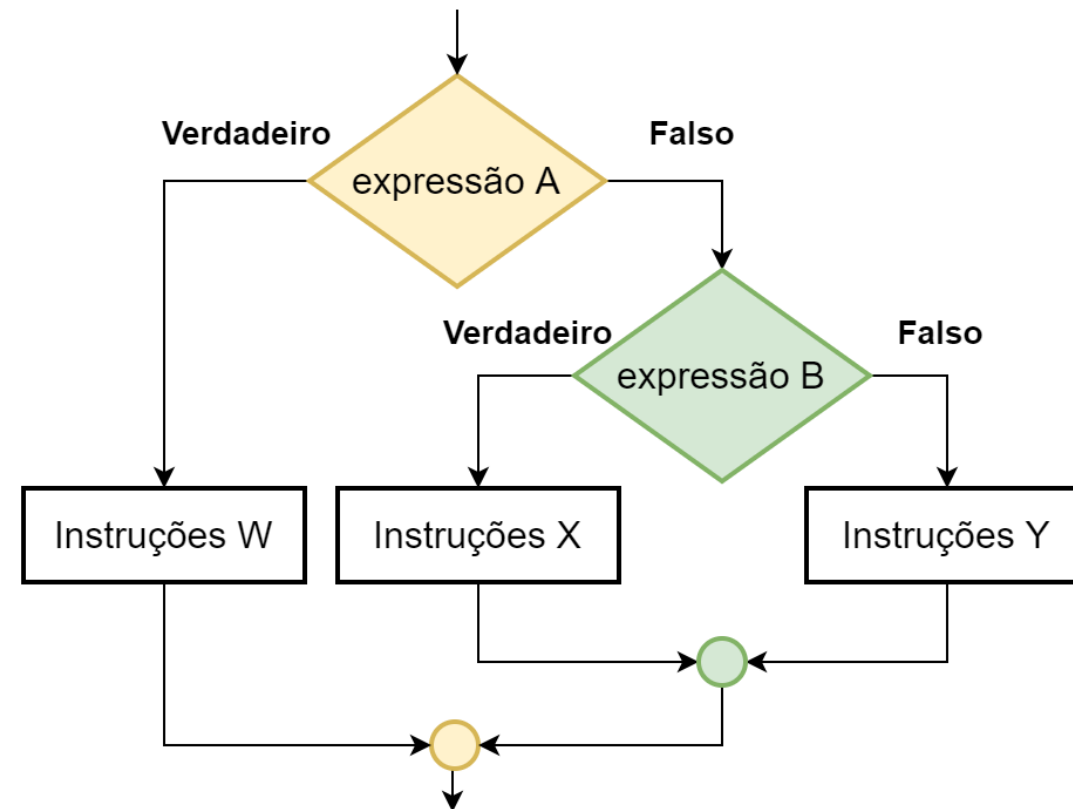
3. ESTRUTURA DE SELEÇÃO ENCADEADA



ESTRUTURA DE SELEÇÃO ENCADEADA

Quando uma estrutura de **seleção B** está contida no bloco de instruções do fluxo 'falso' de uma estrutura de **seleção A**, dizemos que **B está encadeada** em relação à A.

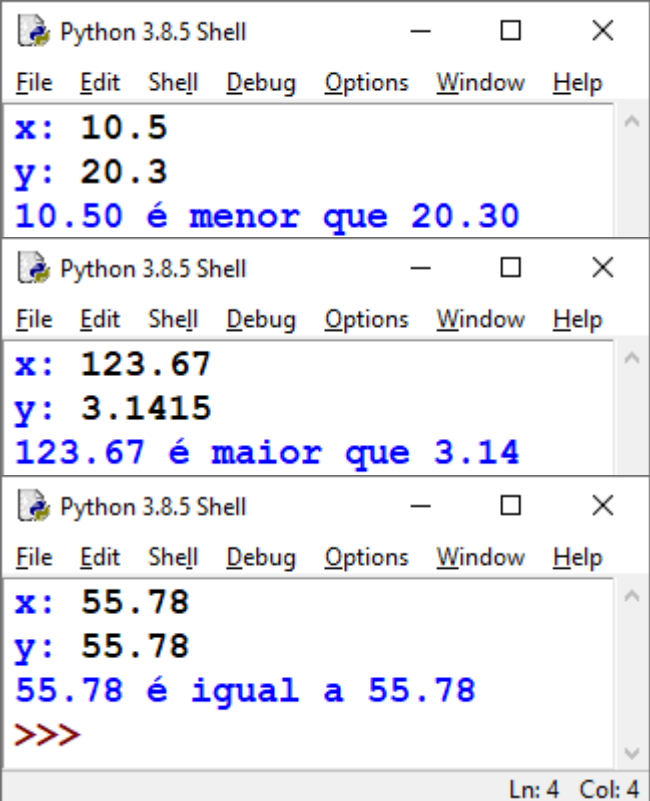
```
...  
if expressão A:  
    instruções W  
else:  
    if expressão B:  
        instruções X  
    else:  
        instruções Y  
...
```



ESTRUTURA DE SELEÇÃO ENCADEADA

[EXEMPLO 3] Crie um programa que receba dois reais x e y e exiba uma mensagem indicando se x é menor que y , ou se x é maior que y ou se ambos são iguais.

```
x = float(input('x: '))
y = float(input('y: '))
if x < y:
    print('%.2f é menor que %.2f' % (x, y))
else:
    if x > y:
        print('%.2f é maior que %.2f' % (x, y))
    else:
        print('%.2f é igual a %.2f' % (x, y))
```



The image displays three sequential screenshots of a Python 3.8.5 Shell window, demonstrating the execution of the program for different input values. Each window shows the input of x and y, followed by the program's output message.

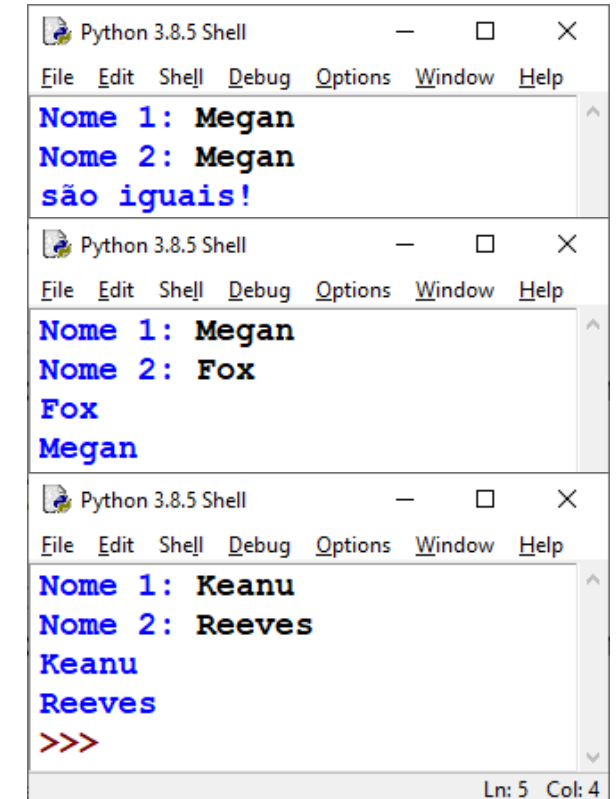
Input x	Input y	Output Message
10.5	20.3	10.50 é menor que 20.30
123.67	3.1415	123.67 é maior que 3.14
55.78	55.78	55.78 é igual a 55.78

The final screenshot shows the prompt `>>>` at the bottom, indicating the end of the program execution.

ESTRUTURA DE SELEÇÃO ENCADEADA

[EXERCÍCIO 4] Crie um programa que receba dois nomes e exiba uma mensagem indicando se são iguais ou, caso sejam diferentes, exiba-os em ordem alfabética.

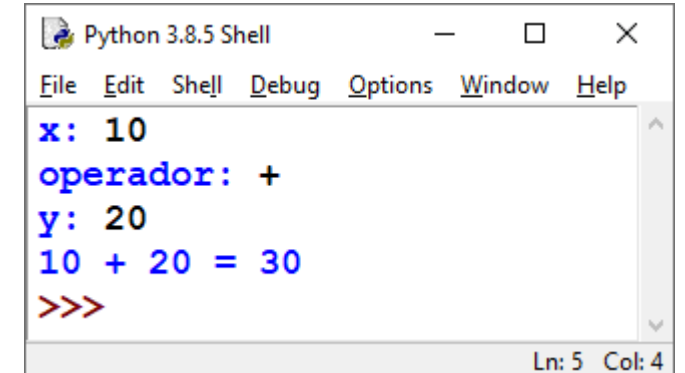
```
n1 = input('Nome 1: ')
n2 = input('Nome 2: ')
if n1 == n2:
    print('são iguais!')
else:
    if n1 < n2:
        print('%s\n%s' % (n1, n2))
    else:
        print('%s\n%s' % (n2, n1))
```



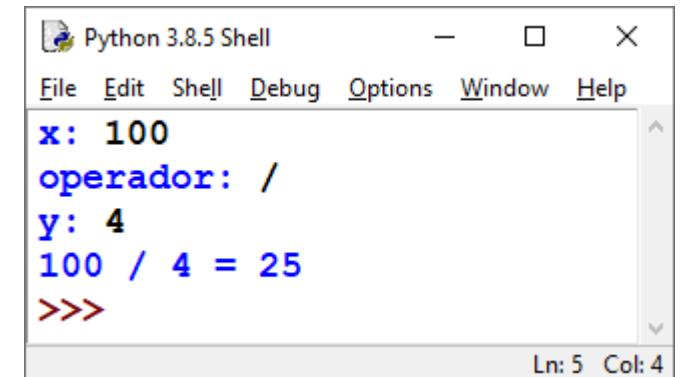
ESTRUTURA DE SELEÇÃO ENCADEADA

[EXEMPLO 4] Crie um programa que receba dois números inteiros x e y e um caractere op representando um operador aritmético (+, −, * ou /). O programa deverá exibir o inteiro resultante da expressão $x \ op \ y$.

```
x = int(input('x: '))
op = input('operador: ')
y = int(input('y: '))
if op == '+':
    print('%d %s %d = %d' % (x, op, y, x+y))
else:
    if op == '-':
        print('%d %s %d = %d' % (x, op, y, x-y))
    else:
        if op == '*':
            print('%d %s %d = %d' % (x, op, y, x*y))
        else:
            print('%d %s %d = %d' % (x, op, y, x/y))
```



```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
x: 10
operador: +
y: 20
10 + 20 = 30
>>>
Ln: 5 Col: 4
```

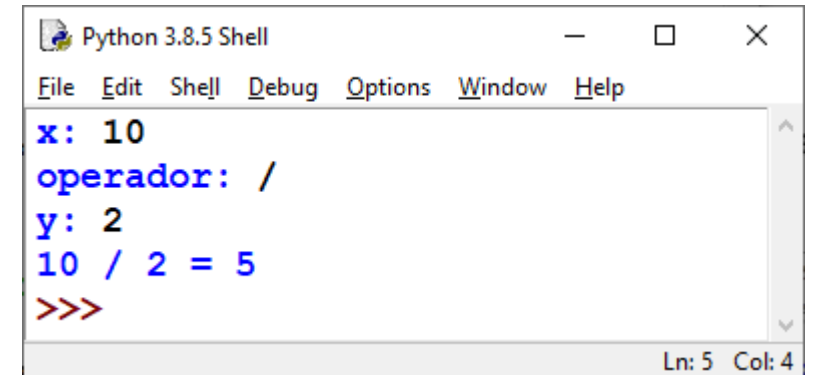


```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
x: 100
operador: /
y: 4
100 / 4 = 25
>>>
Ln: 5 Col: 4
```

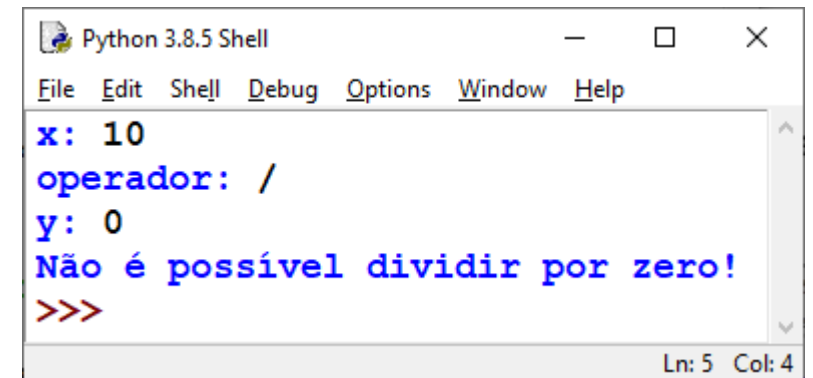
ESTRUTURA DE SELEÇÃO ENCADEADA

[EXERCÍCIO 5] Altere o Exemplo 4, de modo que o usuário seja alertado caso tente fazer uma operação de divisão com divisor zero.

```
x = int(input('x: '))
op = input('operador: ')
y = int(input('y: '))
if op == '+':
    print('%d %s %d = %d' % (x, op, y, x+y))
else:
    if op == '-':
        print('%d %s %d = %d' % (x, op, y, x-y))
    else:
        if op == '*':
            print('%d %s %d = %d' % (x, op, y, x*y))
        else:
            if y==0:
                print('Não é possível dividir por zero!')
            else:
                print('%d %s %d = %d' % (x, op, y, x/y))
```



A screenshot of a Python 3.8.5 Shell window. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area shows the following input and output: 'x: 10', 'operador: /', 'y: 2', and '10 / 2 = 5'. The prompt '>>>' is visible at the bottom. The status bar at the bottom right indicates 'Ln: 5 Col: 4'.



A screenshot of a Python 3.8.5 Shell window. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area shows the following input and output: 'x: 10', 'operador: /', 'y: 0', and 'Não é possível dividir por zero!'. The prompt '>>>' is visible at the bottom. The status bar at the bottom right indicates 'Ln: 5 Col: 4'.

ESTRUTURA DE SELEÇÃO ENCADEADA

Note que o excesso de estruturas de seleção aninhadas ou encadeadas pode comprometer a legibilidade do código, afinal quanto mais níveis de estruturas internas, mais o código se expandirá na horizontal por causa da indentação.

Uma forma de eliminar esse comportamento em estruturas de seleção encadeadas é com o uso do comando **elif** (**else** + **if**).

```
...
if    expressão A:
    ...
elif  expressão B:
    ...
elif  expressão C:
    ...
else:
    ...
...
```

OBSERVAÇÕES:

- 1) Em uma estrutura de seleção `if` é obrigatório;
- 2) Você pode usar a quantidade de `elif` que precisar;
- 3) O `else` é opcional, mas se usado deve ficar no final da estrutura de seleção;
- 4) As expressões são sempre avaliadas em sequência, de cima para baixo;
- 5) As avaliações se encerrarão no momento que uma expressão resultar verdadeiro;
- 6) Caso nenhuma expressão resulte verdadeiro, o bloco do `else` será executado.

ESTRUTURA DE SELEÇÃO ENCADEADA

[EXERCÍCIO 6] Altere todos os exemplos e exercícios de estruturas de seleção encadeadas de modo que usem **elif**.

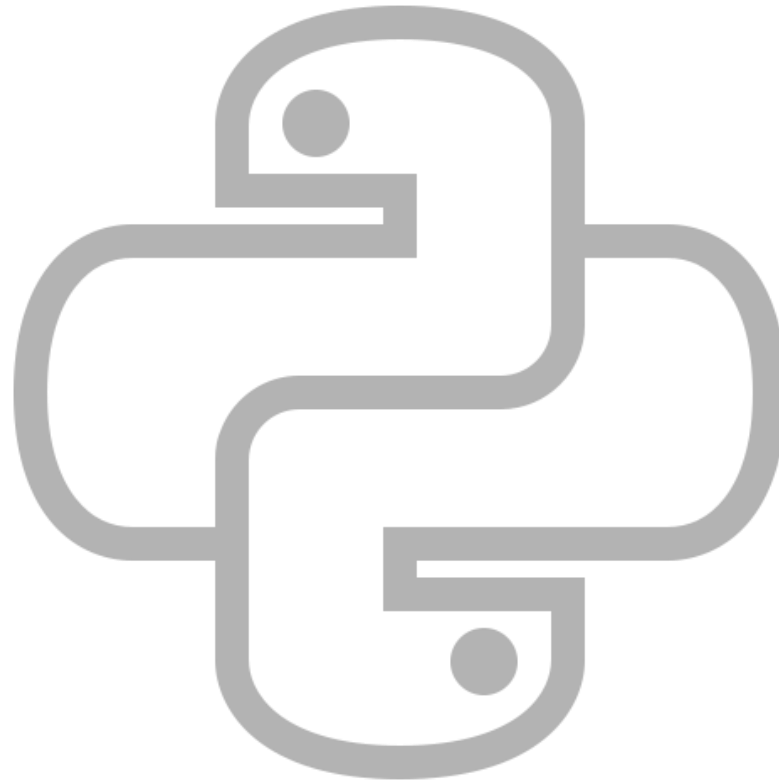
```
x = float(input('x: '))
y = float(input('y: '))
if x < y:
    print('%.2f é menor que %.2f' % (x, y))
elif x > y:
    print('%.2f é maior que %.2f' % (x, y))
else:
    print('%.2f é igual a %.2f' % (x, y))
```

```
n1 = input('Nome 1: ')
n2 = input('Nome 2: ')
if n1 == n2:
    print('São iguais!')
elif n1 < n2:
    print('%s\n%s' % (n1, n2))
else:
    print('%s\n%s' % (n2, n1))
```

```
x = int(input('x: '))
op = input('operador: ')
y = int(input('y: '))
if op == '+':
    print('%d %s %d = %d' % (x, op, y, x+y))
elif op == '-':
    print('%d %s %d = %d' % (x, op, y, x-y))
elif op == '*':
    print('%d %s %d = %d' % (x, op, y, x*y))
else:
    print('%d %s %d = %d' % (x, op, y, x/y))
```

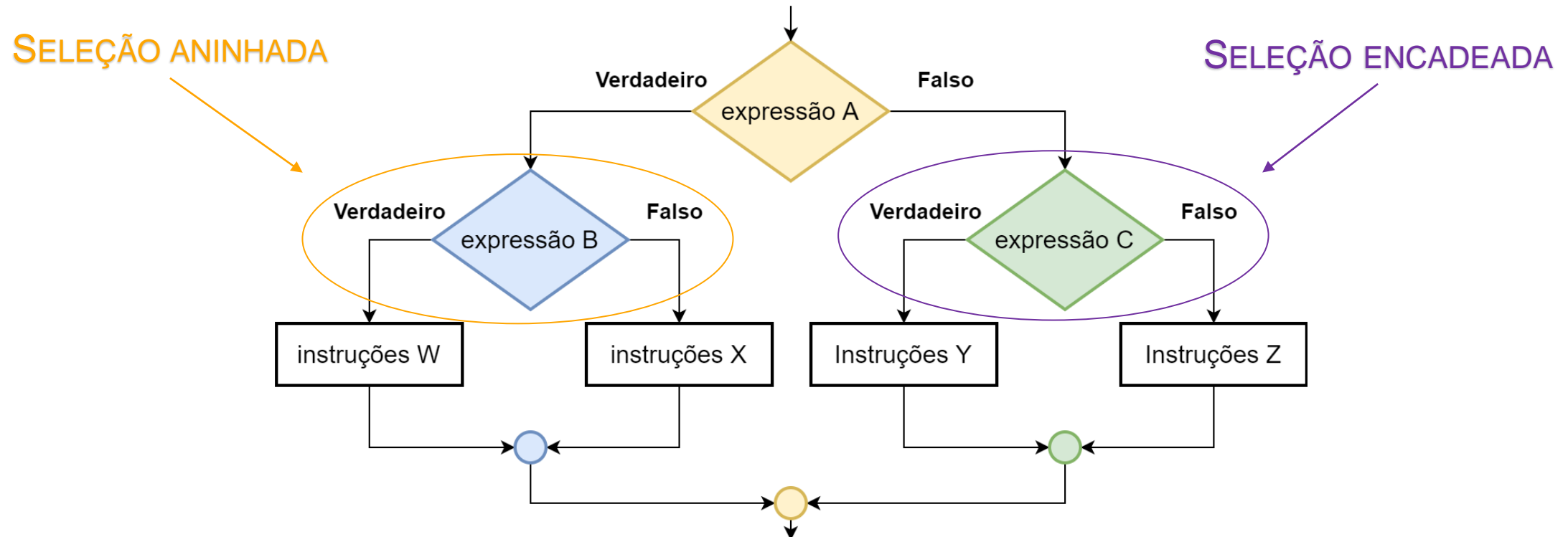
```
x = int(input('x: '))
op = input('operador: ')
y = int(input('y: '))
if op == '+':
    print('%d %s %d = %d' % (x, op, y, x+y))
elif op == '-':
    print('%d %s %d = %d' % (x, op, y, x-y))
elif op == '*':
    print('%d %s %d = %d' % (x, op, y, x*y))
elif y == 0:
    print('Não é possível dividir por zero!')
else:
    print('%d %s %d = %d' % (x, op, y, x/y))
```

4. ESTRUTURAS DE SELEÇÃO COM SELEÇÕES ANINHADAS E ENCADEADAS



ESTRUTURAS DE SELEÇÃO COM SELEÇÕES ANINHADAS E ENCADEADAS

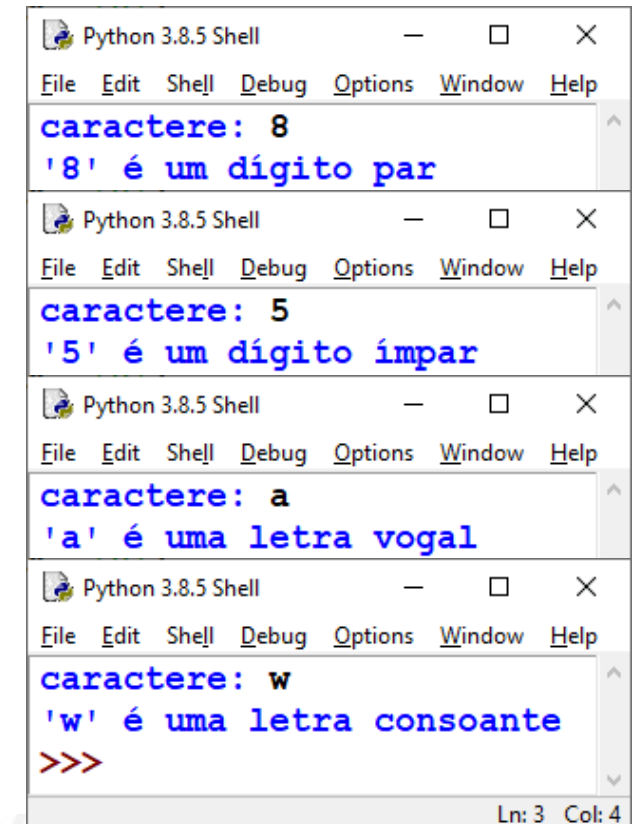
É possível que uma estrutura de seleção contenha simultaneamente estruturas de seleção aninhadas e estruturas de seleção encadeadas.



ESTRUTURAS DE SELEÇÃO COM SELEÇÕES ANINHADAS E ENCADEADAS

[EXEMPLO 5] Crie um programa que leia um caractere x e indique se x é um dígito ou uma letra. Se x é dígito, indique se representa um par ou ímpar, se letra, se representa uma vogal ou consoante. Obs.: x será apenas dígito ou minúscula.

```
x = input('caractere: ')
if '0' <= x <= '9':
    print('\'%s\' é um dígito' % x, end=' ')
    if x=='0' or x=='2' or x=='4' or x=='6' or x=='8':
        print('par')
    else:
        print('ímpar')
else:
    print('\'%s\' é uma letra' % x, end=' ')
    if x=='a' or x=='e' or x=='i' or x=='o' or x=='u':
        print('vogal')
    else:
        print('consoante')
```



The image shows four sequential screenshots of a Python 3.8.5 Shell window, demonstrating the program's output for different inputs. Each window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The first screenshot shows the input '8' resulting in the output '8 é um dígito par'. The second screenshot shows the input '5' resulting in '5 é um dígito ímpar'. The third screenshot shows the input 'a' resulting in 'a é uma letra vogal'. The fourth screenshot shows the input 'w' resulting in 'w é uma letra consoante', followed by the prompt '>>>'.

```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
caractere: 8
'8' é um dígito par

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
caractere: 5
'5' é um dígito ímpar

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
caractere: a
'a' é uma letra vogal

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
caractere: w
'w' é uma letra consoante
>>>
Ln: 3 Col: 4
```

ESTRUTURAS DE SELEÇÃO COM SELEÇÕES ANINHADAS E ENCADEADAS

[EXEMPLO 6] Com base no Exemplo 5, crie uma versão em que o caractere `x` também possa ser uma letra maiúscula e tenha a respectiva indicação.

```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
caractere: 8
'8' é um dígito par
>>>
```

Ln: 3 Col: 4

```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
caractere: a
'a' é uma letra vogal minúscula
>>>
```

Ln: 3 Col: 4

```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
caractere: A
'A' é uma letra vogal maiúscula
>>>
```

Ln: 3 Col: 4

```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
caractere: 5
'5' é um dígito ímpar
>>>
```

Ln: 3 Col: 4

```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
caractere: w
'w' é uma letra consoante minúscula
>>>
```

Ln: 3 Col: 4

```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
caractere: R
'R' é uma letra consoante maiúscula
>>>
```

Ln: 3 Col: 4

ESTRUTURAS DE SELEÇÃO COM SELEÇÕES ANINHADAS E ENCADEADAS

```
x = input('caractere: ')
if '0' <= x <= '9':
    print('\'%s\' é um dígito' % x, end=' ')
    if x=='0' or x=='2' or x=='4' or x=='6' or x=='8':
        print('par')
    else:
        print('ímpar')
else:
    print('\'%s\' é uma letra' % x, end=' ')
    if 'a' <= x <= 'z':
        if x=='a' or x=='e' or x=='i' or x=='o' or x=='u':
            print('vogal minúscula')
        else:
            print('consoante minúscula')
    else:
        if x=='A' or x=='E' or x=='I' or x=='O' or x=='U':
            print('vogal maiúscula')
        else:
            print('consoante maiúscula')
```

ESTRUTURAS DE SELEÇÃO COM SELEÇÕES ANINHADAS E ENCADEADAS

[EXERCÍCIO 7] Com base no Exemplo 6, crie uma versão em que não existam estruturas de seleção aninhadas. Obs.: use **elif**.

```
x = input('caractere: ')
if x=='0' or x=='2' or x=='4' or x=='6' or x=='8':
    print('\'%s\' é um dígito par' % x)
elif x=='1' or x=='3' or x=='5' or x=='7' or x=='9':
    print('\'%s\' é um dígito ímpar' % x)
elif x=='a' or x=='e' or x=='i' or x=='o' or x=='u':
    print('\'%s\' é uma letra vogal minúscula' % x)
elif 'a'<=x<='z' and (x!='a' and x!='e' and x!='i' and x!='o' and x!='u'):
    print('\'%s\' é uma letra consoante minúscula' % x)
elif x=='A' or x=='E' or x=='I' or x=='O' or x=='U':
    print('\'%s\' é uma letra vogal maiúscula' % x)
else:
    print('\'%s\' é uma letra consoante maiúscula' % x)
```

ESTRUTURAS DE SELEÇÃO COM SELEÇÕES ANINHADAS E ENCADEADAS

[EXERCÍCIO 8] Altere a solução do Exercício 7 de modo que se `x` não for dígito, letra minúscula ou letra maiúscula o programa exiba 'caractere desconhecido'.

```
x = input('caractere: ')
if x=='0' or x=='2' or x=='4' or x=='6' or x=='8':
    print('\'%s\' é um dígito par' % x)
elif x=='1' or x=='3' or x=='5' or x=='7' or x=='9':
    print('\'%s\' é um dígito ímpar' % x)
elif x=='a' or x=='e' or x=='i' or x=='o' or x=='u':
    print('\'%s\' é uma letra vogal minúscula' % x)
elif 'a'<=x<='z' and (x!='a' and x!='e' and x!='i' and x!='o' and x!='u'):
    print('\'%s\' é uma letra consoante minúscula' % x)
elif x=='A' or x=='E' or x=='I' or x=='O' or x=='U':
    print('\'%s\' é uma letra vogal maiúscula' % x)
elif 'A'<=x<='Z' and (x!='A' and x!='E' and x!='I' and x!='O' and x!='U'):
    print('\'%s\' é uma letra consoante maiúscula' % x)
else:
    print('caractere desconhecido')
```

5. EXERCÍCIOS EXTRAS

