

# Capítulo 2 - Solução

November 20, 2020

## 0.1 Exercício 1: Lista de compras

Crie um código, que faça uma lista de estoque formatada com largura máxima de cada linha de 50 caracteres, ela deverá informar o nome do item e a sua quantidade disponível. O código deverá pedir o usuário que entre com o nome e quantidade de pelo menos 3 itens. Considere que a quantidade máxima de cada item seja 999

**Solução:**

```
[1]: print("Bem-vindo ao pylist")

item_1_nome = input("Nome do item: ").title()
item_1_quantidade = int(
    input("Quantidade de {}: ".format(item_1_nome))
)
item_2_nome = input("Nome do item: ").title()
item_2_quantidade = int(
    input("Quantidade de {}: ".format(item_2_nome))
)
item_3_nome = input("Nome do item: ").title()
item_3_quantidade = int(
    input("Quantidade de {}: ".format(item_3_nome))
)

item_1_nome = "Item 1: "+item_1_nome
item_2_nome = "Item 2: "+item_2_nome
item_3_nome = "Item 3: "+item_3_nome

print("{:_^50}".format('Lista de estoque'))
print("{:-<45}{}".format("Item", "Quant"))
print("{:-<45}{}".format(item_1_nome, item_1_quantidade))
print("{:-<45}{}".format(item_2_nome, item_2_quantidade))
print("{:-<45}{}".format(item_3_nome, item_3_quantidade))
```

Bem-vindo ao pylist

Nome do item: Arroz

Quantidade de Arroz: 2

Nome do item: Feijão

Quantidade de Feijão: 3

Nome do item: Carne  
Quantidade de Carne: 1

```
-----Lista de estoque-----  
Item-----Quant  
Item 1: Arroz-----2  
Item 2: Feijão-----3  
Item 3: Carne-----1
```

## 0.2 Exercício 2: Sentenças lógicas

Escreva o resultado de cada uma das seguintes expressões em Markdown:

```
(5 > 4) and (3 == 5)  
not (5 > 4)  
(5 > 4) or (3 == 5)  
not ((5 > 4) or (3 == 5))  
(True and True) and (True == False)  
(not False) or (not True)
```

### Resolução:

```
(5 > 4) and (3 == 5) -> True and False -> False  
not (5 > 4) -> not (True) -> False  
(5 > 4) or (3 == 5) -> True or False -> True  
not ((5 > 4) or (3 == 5)) -> not (True or False) -> False  
(True and True) and (True == False) -> True and False -> False  
(not False) or (not True) -> True or False -> True
```

## 0.3 Exercício 3: Semáforo

Um semáforo comum costuma ter de 2 a 4 fases e mais uma fase para os pedestres. Crie uma expressão que represente um semáforo, as setas abaixo indicam o fluxo.

```
  N  
L  0  
  S
```

Crie uma expressão para cada uma das fases deste cruzamento.

### Resolução:

```
Norte and not (Leste or Oeste or Sul)  
Leste and not (Norte or Sul or Oeste)  
Oeste and not (Norte or Sul or Oeste)  
Sul and not (Leste or Oeste or Norte)
```

## 0.4 Exercício 4: Ano bissexto

Escreva uma expressão (em função de uma incógnita ano) que resulte `True` caso o ano seja bissexto e `False` caso contrário.

**Resolução:**

```
[2]: # Fórmula dos anos bissextos:  
# Todos os anos múltiplos de 400 ou  
# Todos os anos múltiplos de 4 e não múltiplos de 100.  
print("Bem-vindo ao pybisexto, diga um ano e direi se é bisexto ou não")  
ano = int(input("Qual o Ano Desejado ? "))  
resposta = (ano % 400 == 0) or ((ano % 4 == 0) and (ano % 100 != 0))  
print(resposta)
```

Bem-vindo ao pybisexto, diga um ano e direi se é bisexto ou não

Qual o Ano Desejado ? 2041

False