

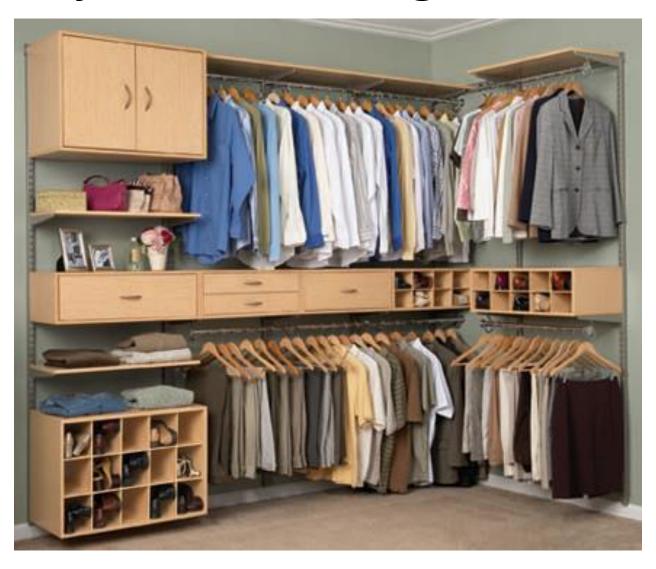
# Revisão Funções

fmasanori@gmail.com

# Sejamos mais organizados



# Sejamos mais organizados



# Sejamos organizados

- Quando os programas crescem, geralmente o código fica mais complexo
- Uma forma de gerenciar essa complexidade é usar funções
- Elas permitem separar ações comuns, assim seu código fica mais fácil de ler e mais fácil de manter

# O Starbuzz está sem grãos

- O diretor da Starbuzz quer uma opção para compra rápida, sem esperar abaixar o preço
- Ao rodar o programa irei perguntar se quer comprar já ou não
- Se o usuário responder sim, então irei pegar o preço atual e comprar
- Caso negativo, irei esperar abaixar para menos que 4.74

# Sugestão de novo programa

```
import urllib.request
import time
opção = input('Deseja comprar já? (S/N)')
if opção == 'S':
   pagina = urllib.request.urlopen(
    'http://beans.itcarlow.ie/prices-loyalty.html')
    texto = pagina.read().decode('utf8')
    onde = texto.find('>$')
    inicio = onde + 2
    fim = inicio + 4
   preço = float(texto[inicio:fim])
   print ('Comprar! Preço: %5.2f' %preço)
else:
   preço = 99.99
   while preço >= 4.74:
        pagina = urllib.request.urlopen(
        'http://beans.itcarlow.ie/prices-loyalty.html')
        texto = pagina.read().decode('utf8')
        onde = texto.find('>$')
        inicio = onde + 2
        fim = inicio + 4
        preço = float(texto[inicio:fim])
        if preço >= 4.74:
            time.sleep(600)
   print ('Comprar! Preço: %5.2f' %preço)
```

#### Programa feio...

- Não <u>duplique</u> seu código...
- Isso leva ao excesso de código, tornando a manutenção do seu código difícil
- Procure <u>reutilizar</u> seu código
- Definindo funções faremos <u>reuso</u> de código
- Qual é a diferença?
- Se eu tiver que mudar algo, irei alterar em apenas um lugar

# Funções

- Funções são códigos compartilháveis
- Defino um nome e chamo a função ao longo do programa
- A função deve ser definida <u>antes</u> de você chamá-la
- Se você quiser retornar um valor para quem chamou a função deverá utilizar o comando return

```
import urllib.request
import time
def pega preço():
    pagina = urllib.request.urlopen(
    'http://beans.itcarlow.ie/prices-loyalty.html')
    texto = pagina.read().decode('utf8')
    onde = texto.find('>$')
    inicio = onde + 2
    fim = inicio + 4
   return float(texto[inicio:fim])
opção = input('Deseja comprar já? (S/N)')
if opção == 'S':
    preço = pega preço()
    print ('Comprar! Preço: %5.2f' %preço)
else:
    preço = 99.99
    while preço >= 4.74:
        preço = pega_preço()
        if preço >= 4.74:
            time.sleep(600)
    print ('Comprar! Preço: %5.2f' %preço)
```

# Não existem perguntas idiotas

- O comando return é igual ao print? Não, print exibe algo na tela, enquanto return devolve um valor para quem chamou a função.
- Se eu não der return dentro da função, o que ela retorna? None
- Return sempre deve vir no final da função?
   Nem sempre, depende da lógica da função
- Uma função pode devolver mais de um valor?
   Sim, inclusive listas ou dicionários.

# ISON killer module



# Java Script Object Notation

- http://json.org/
- · Independente de linguagem
- · Pares nome/valor
- · Lista ordenada
- · Muito adotado atualmente
- · Alternativa para o XML
  - XML mais verbosa
  - -XML menor legibilidade

# Chuck Norris Nerdy Jokes

```
import urllib.request
import json

url = 'http://api.icndb.com/jokes/random?limitTo=[nerdy]'

resp = urllib.request.urlopen(url).read()
data = json.loads(resp.decode('utf-8'))

print (data['value']['joke'])
```

https://gist.github.com/4745061