

Algoritmos e estrutura de dados I



Funções

Introdução

Códigos que são executados frequentemente durante o programa, podem ser inseridos em uma função

A função começa com a palavra reservada `def`

```
>>>def soma(a, b):  
>>>    print(a + b)
```

A função pode realizar uma série de comandos e devolver uma variável

```
>>>def soma(a, b):  
>>>    return a+b
```

Escopo

Quando as variáveis são criadas, elas ficam limitadas a um escopo

Se a variável é declarada fora da função, a mesma pode ser acessada em qualquer parte do programa

Se a variável for declarada dentro da função ela possui apenas o escopo local, ou seja, é acessada apenas na função

```
>>>empresa = "Teste"
```

```
>>>def imprime():
```

```
>>>    print(empresa)
```

Função para Validação

É possível usar uma função para validar algum dado

Ex.:

```
>>>while True:
>>>  v = int(input("Digite um valor entre 0 e 5:"))
>>>  if v < 0 or v > 5:
>>>    print("Valor inválido")
>>>  else:
>>>    break
```

Parâmetros opcionais

Uma função pode receber um parâmetro, esse valor pode ser definido como opcional

```
>>> def soma(a, b=0):  
>>>     return a+b
```

Função recursiva

Chamamos de função recursiva a função que chama a si mesma

O problema do fatorial pode ser definido utilizando uma função com laço de repetição ou então com função recursiva

Sabemos que o fatorial de um número é o próprio valor multiplicado pelo fatorial do seu antecessor

Ex.: $4! = 4 \cdot 3!$

Para que função recursiva funcione corretamente temos que atingir uma condição que o resultado é conhecido

Ex.: $1! = 1$ ou $0! = 1$

Função recursiva fatorial

```
>>>def fatorial(n):  
>>>  if n==0 or n==1:  
>>>    return 1  
>>>  else:  
>>>    return n * fatorial(n - 1)
```