

# Fundamentos de Algoritmos

## Projeto de funções

Marco A L Barbosa

[malbarbo.pro.br](http://malbarbo.pro.br)

### Começando

- 1) Leia o material de projeto de funções e responda:
  - a) Descreva com suas palavras o objetivo de cada etapa do processo de projeto de funções.
  - b) Dê um exemplo de informação irrelevante e de uma ambiguidade que poderiam aparecer em um enunciado de problema.
  - c) Por que é importante definir os tipos de dados antes da especificação?
  - d) Explique o papel de cada parte da especificação (assinatura, propósito e exemplos) no projeto de uma função.
  - e) Qual é a principal propriedade que uma especificação deve ter para ser considerada adequada?
  - f) O que é a assinatura de uma função?
  - g) Qual é o objetivo inicial dos exemplos no projeto de uma função? E os demais objetivos?
  - h) Se não forem encontrados erros na verificação para os exemplos da especificação, é possível afirmar que a função está isenta de erros? Explique.
  - i) A implementação é a fase mais importante do projeto de funções, verdadeiro ou falso? Explique.
  - j) Como proceder quando um teste falha?
  - k) Por que a verificação deve ser feita novamente após a revisão?
  - l) Qual é a diferença do resultado da análise e a descrição do propósito da função?
- 2) Faça o download dos exemplos de projeto de programas na página da disciplina e para cada arquivo
  - a) Observe a forma como os propósitos foram escritos.
  - b) Observe a forma como os exemplos foram escritos.
  - c) Observe como as implementações foram feitas.
  - d) Faça a verificação clicando em Run no Spython (verifica tipos e exemplos).

### Praticando

- 3) Implemente a função de acordo com a especificação a seguir. Corrija a especificação se necessário.

```
def isento_tarifa(idade: int) -> bool:
    """
    Produz True se uma pessoa de *idade* anos é isento da tarifa
    de transporte público, isto é, tem menos que 18 anos ou 65
    ou mais. Produz False caso contrário.

    Exemplos
    >>> isento_tarifa(17)
```

```

True
>>> isento_tarifa(18)
True
>>> isento_tarifa(50)
False
>>> isento_tarifa(65)
True
>>> isento_tarifa(70)
True
'''
return False

```

- 4) Escreva os exemplos e implemente a função de acordo com a assinatura e o propósito a seguir.

```

def dma_para_amd(data: str) -> str:
    '''
    Transforma *data*, que deve estar no formato "dia/mes/ano",
    onde dia e mes tem dois dígitos e ano tem quatro dígitos,
    para o formato "ano/mes/dia".
    '''
    return ''

```

- 5) Escreva a especificação para a seguinte implementação de função. Observe que a especificação sozinha deve ser suficiente para um desenvolvedor fazer uma nova implementação.

```

def numero_digitos(n: int) -> int:
    return len(str(abs(n)))

```

- 6) Para cada função da lista de exercícios de conceitos básicos que você implementou, complete o projeto da função adicionando a especificação (propósito e exemplos). Faça a verificação com o Spython.

## Avançando

- 7) Você está fazendo um programa e precisa verificar se um texto digitado pelo usuário está de acordo com algumas regras. A regra “sem espaços extras” requer que o texto não comece e não termine com espaços. Projete uma função que verifique se um texto qualquer está de acordo com a regra “sem espaços extras”.
- 8) Rotacionar uma string  $n$  posições a direita significa mover os últimos  $n$  caracteres da string para as primeiras  $n$  posições da string. Por exemplo, rotacionar a string "marcelio" 5 posições a direita produz a string "celiomar". Projete uma função que receba como entrada uma string e um número  $n$  e produza uma nova string rotacionando a string de entrada  $n$  posições a direita.

## Desafios

- 9) O CPF (Cadastro de Pessoa Física) pode ser representado de duas formas: sem formatação, como '12345678901', ou com formatação, como '123.456.789-01'. Projete duas funções, uma que receba um CPF sem formatação e produza o CPF formatado, e outra que receba um CPF formatado e produza o CPF sem formatação.