

# Documentação de Código em Python

Eduardo Ribeiro Silva de Oliveira

07 de Outubro de 2024

# O que é documentação de código?

- ▶ Registrar funcionalidades, finalidades, entradas, saídas e dependências.
- ▶ Ajudar na manutenção e compreensão do código.

# Por que é importante?

- ▶ Facilita manutenção e colaboração.
- ▶ Reduz erros ao melhorar a compreensão do fluxo de dados.

# Como otimizar a documentação?

- ▶ Escrever códigos intuitivos.
- ▶ Seguir um padrão.
- ▶ Utilizar ferramentas automatizadas.

# Auto-documentação por Valdemar W. Setzer

- ▶ Proposta para adicionar informações detalhadas diretamente no código.
- ▶ Facilita a consulta de documentação sem outros documentos.
- ▶ Mais informações: <https://www.ime.usp.br/~vwsetzer/Autodoc-de-programas.pdf>
- ▶ Nível 1: Informações gerais dos procedimentos.
- ▶ Nível 2: Documentação de nível 1 e os pseudo-códigos.
- ▶ Nível 3: Todo o código fonte do programa.

# Exemplo de Cabeçalho

```
#1[  
#1 TITULO: NOME DO PROGRAMA OU NOME DO ARQUIVO  
#1 AUTOR: AUTOR DO ARQUIVO  
#1 DATA: DATA DA VERSÃO ATUAL  
#1 VERSAO: NÚMERO DA VERSÃO ATUAL  
#1 FINALIDADE: EXPLICAÇÃO DO QUE O PROGRAMA FAZ  
#1 ENTRADAS: INDICA AS ENTRADAS NECESSÁRIAS PARA A EXECUÇÃO  
#1 SAIDAS: INDICA A SAÍDA DO PROGRAMA  
#1 ROTINAS CHAMADAS: INDICA AS ROTINAS USADAS  
#1]
```

# Exemplo de Documentação de Função

```
#1[
#1 ROTINA: __INIT__
#1 FINALIDADE: INICIALIZA AS VARIÁVEIS DA CLASSE EXTRACTOR
#1 ENTRADAS: CAMINHO DO ARQUIVO/DIRETÓRIO DE ENTRADA, CAMINHO DO
#1 ARQUIVO DE SAÍDA, MODO DE EXECUÇÃO
#1 DEPENDENCIAS: OS
#1 CHAMADO POR: EXTRACTOR
#1 CHAMA: _GET_ALL_FILES (SE O CAMINHO FOR UM DIRETÓRIO)
#1]
#2[
#2 PSEUDOCODIGO DE: __init__
def __init__(self, input_path: str, output_path: str, execution_mode: int = 3):
    #2 ARMAZENA O CAMINHO DE ENTRADA E O CAMINHO DE SAÍDA
    self.input_path = input_path
    self.output_path = output_path
    #2 DEFINE O MODO DE EXECUÇÃO PADRÃO COMO 3
    self.execution_mode = execution_mode
    #2 VERIFICA SE O CAMINHO DE ENTRADA É UM ARQUIVO ÚNICO
    self.one_file = os.path.isfile(input_path)
    #2 GERA UMA LISTA DE ARQUIVOS A SEREM PROCESSADOS (SE FOR UM DIRETÓRIO)
    self.file_path_list = [input_path] if self.one_file else self._get_all_files(input_path)
    #2 INICIALIZA UM DICIONÁRIO PARA ARMAZENAR A DOCUMENTAÇÃO
    self.documentation = {}
    #2 INICIALIZA UMA STRING VAZIA PARA O CONTEÚDO HTML
    self.html_content = '' # Armazenará o HTML gerado
#2]
```

# Imagens de Exemplos

Nome do Arquivo
text_handler.py
Documentação
<pre>#1[ #1 TITULO: TEXTHANDLER #1 AUTOR: EDUARDO RIBEIRO SILVA DE OLIVEIRA #1 DATA: 07/10/2024 #1 VERSAO: 1 #1 FINALIDADE: REALIZAR EXTRAÇÃO E FILTRAGEM DE TEXTO BRUTO A PARTIR DE CONTEÚDO HTML #1 ENTRADAS: NOME DO DIRETÓRIO (NO CONSTRUTOR), CONTEÚDO HTML (NA FUNÇÃO EXTRACT_RAW_TEXT) #1 SAIDAS: TEXTO LIMPO EM MINÚSCULAS EXTRAÍDO DO CONTEÚDO HTML #1 ROTINAS CHAMADAS: EXTRACT_RAW_TEXT, _FILTER_PORTUGUESE_TEXT #1]  #1[ #1 ROTINA: __INIT__ #1 FINALIDADE: INICIALIZA A CLASSE TEXTHANDLER E CONFIGURA O DIRETÓRIO #1 ENTRADAS: NOME DO DIRETÓRIO (STRING) #1 DEPENDENCIAS: LOGGINGHANDLER #1 CHAMADO POR: TEXTHANDLER #1 CHAMA: LOGGINGHANDLER.__INIT__ #1]  #1[ #1 ROTINA: EXTRACT_RAW_TEXT #1 FINALIDADE: EXTRAÍ O TEXTO BRUTO DO CONTEÚDO HTML E O FILTRA PARA REMOVER CARACTERES INDESEJADOS #1 ENTRADAS: CONTEÚDO HTML (STRING) #1 DEPENDENCIAS: BEAUTIFULSOUP, RE #1 CHAMADO POR: USUÁRIO #1 CHAMA: _FILTER_PORTUGUESE_TEXT #1]  #1[ #1 ROTINA: _FILTER_PORTUGUESE_TEXT #1 FINALIDADE: FILTRA O TEXTO PARA REMOVER CARACTERES QUE NÃO SÃO DO</pre>



# Imagens de Exemplos - Nível 2

Nome do Arquivo
text_handler.py
Documentação
<pre>#1[ #1 TITULO: TEXTHANDLER #1 AUTOR: EDUARDO RIBEIRO SILVA DE OLIVEIRA #1 DATA: 07/10/2024 #1 VERSAO: 1 #1 FINALIDADE: REALIZAR EXTRAÇÃO E FILTRAGEM DE TEXTO BRUTO A PARTIR DE CONTEÚDO HTML #1 ENTRADAS: NOME DO DIRETÓRIO (NO CONSTRUTOR), CONTEÚDO HTML (NA FUNÇÃO EXTRACT_RAW_TEXT) #1 SAIDAS: TEXTO LIMPO EM MINÚSCULAS EXTRAÍDO DO CONTEÚDO HTML #1 ROTINAS CHAMADAS: EXTRACT_RAW_TEXT, _FILTER_PORTUGUESE_TEXT #1]  #1[ #1 ROTINA: __INIT__ #1 FINALIDADE: INICIALIZA A CLASSE TEXTHANDLER E CONFIGURA O DIRETÓRIO #1 ENTRADAS: NOME DO DIRETÓRIO (STRING) #1 DEPENDENCIAS: LOGGINGHANDLER #1 CHAMADO POR: TEXTHANDLER #1 CHAMA: LOGGINGHANDLER.__INIT__ #1] #2[ #2 PSEUDOCODIGO DE: __init__ #2 CHAMA O CONSTRUTOR DA CLASSE PAI E INICIALIZA O DIRETÓRIO #2 ARMAZENA O NOME DO DIRETÓRIO #2] #1[ #1 ROTINA: EXTRACT_RAW_TEXT #1 FINALIDADE: EXTRAÍ O TEXTO BRUTO DO CONTEÚDO HTML E O FILTRA PARA REMOVER CARACTERES INDESEJADOS #1 ENTRADAS: CONTEÚDO HTML (STRING) #1 DEPENDENCIAS: BEAUTIFULSOUP, RE #1 CHAMADO POR: USUÁRIO</pre>

# Imagens de Exemplos - Nível 3

Nome do Arquivo
text_handler.py
Documentação
<pre>#1[ #1 TITULO: TEXTHANDLER #1 AUTOR: EDUARDO RIBEIRO SILVA DE OLIVEIRA #1 DATA: 07/10/2024 #1 VERSAO: 1 #1 FINALIDADE: REALIZAR EXTRAÇÃO E FILTRAGEM DE TEXTO BRUTO A PARTIR DE CONTEÚDO HTML #1 ENTRADAS: NOME DO DIRETÓRIO (NO CONSTRUTOR), CONTEÚDO HTML (NA FUNÇÃO EXTRACT_RAW TEXT) #1 SAÍDAS: TEXTO LIMPO EM MINÚSCULAS EXTRAÍDO DO CONTEÚDO HTML #1 ROTINAS CHAMADAS: EXTRACT_RAW_TEXT, _FILTER_PORTUGUESE_TEXT #1]  import re from bs4 import BeautifulSoup from logging_logic.logging_handler import LoggingHandler  class TextHandler(LoggingHandler):     #1[     #1 ROTINA: __INIT__     #1 FINALIDADE: INICIALIZA A CLASSE TEXTHANDLER E CONFIGURA O DIRETÓRIO     #1 ENTRADAS: NOME DO DIRETÓRIO (STRING)     #1 DEPENDENCIAS: LOGGINGHANDLER     #1 CHAMADO POR: TEXTHANDLER     #1 CHAMA: LOGGINGHANDLER.__INIT__     #1]     #2[     #2 PSEUDOCODIGO DE: __init__     def __init__(self, directory_name):         #2 CHAMA O CONSTRUTOR DA CLASSE PAI E INICIALIZA O DIRETÓRIO         super().__init__(directory_name)         #2 ARMAZENA O NOME DO DIRETÓRIO         self.directory_name = directory_name</pre>