Como documentar código fonte?

Me. Nonato Sales

Por que documentar o código fonte?

- Quem desenvolve sistemas profissionalmente, sabe ou pelo menos deveria saber da importância de se documentar o código fonte. É preciso deixar claro que, por mais que utilizemos melhores práticas em nossos projetos, se não documentarmos de forma eficiente os arquivos de código envolvidos, estaremos criando dificultadores para aqueles que receberem a incumbência de efetuar manutenções corretivas e evolutivas em nosso código no futuro.

Por que documentar o código fonte?

- Padronização do nome das variáveis e parâmetros;
- Documentação de pontos importantes no decorrer do código;
- Principais metodologias de programação existentes atualmente indicam que cerca de 50% do código fonte deve estar documentado;
- Quando desenvolvemos precisamos pensar no próximo;

Bibliotecas

Algumas bibliotecas

- Pydoc (https://docs.python.org/pt-br/3/library/pydoc.html) ;
- Sphinx - (https://docs.readthedocs.io/en/stable/intro/getting-started-with-sphinx.html);
- Doxygen - (https://www.doxygen.nl/index.html);

Docstrings vs Comentários

- Docstrings:
  - São semelhantes aos comentários, mas são uma versão aprimorada, mais lógica e útil dos comentários. Docstrings atuam como documentação para a **classe**, **módulo** e **pacotes**;
  - As docstrings são representadas com aspas de abertura e fechamento e pode ser simples ou duplas.

- Comentários:
  - São usados principalmente para explicar partes não óbvias do código e podem ser úteis para comentários sobre correção de bugs e tarefas que precisam ser realizadas;
  - enquanto os comentários começam com um **# no início**;
  - Os comentários não podem ser acessados com o atributo doc integrado e a função de ajuda.

Padrões de docstrings

Google

```
class Vehicles(object):
    '''
    The Vehicle object contains a lot of vehicles

    Args:
        arg (str): The arg is used for...
        *args: The variable arguments are used for...
        **kwargs: The keyword arguments are used for...

    Attributes:
        arg (str): This is where we store arg,
    '''
    def __init__(self, arg, *args, **kwargs):
        self.arg = arg

    def cars(self, distance,destination):
        '''We can't travel distance in vehicles without fuels, so here is the fuels

        Args:
            distance (int): The amount of distance traveled
            destination (bool): Should the fuels refilled to cover the distance?

        Raises:
            RuntimeError: Out of fuel

        Returns:
            cars: A car mileage
        '''
        pass
```

Numpy

```
class Photo(ndarray):
    """
    Array with associated photographic information.

    ...

    Attributes
    ----------
    exposure : float
        Exposure in seconds.

    Methods
    -------
    colorspace(c='rgb')
        Represent the photo in the given colorspace.
    gamma(n=1.0)
        Change the photo's gamma exposure.

    """
```

Biblioteca **Pydoc**

Pydoc

- Pydoc é um módulo que gera automaticamente a documentação dos módulos do Python;
- Para módulos, classes, funções e métodos, a documentação exibida é derivada da **docstring** (**__doc__.py**);
- O PyDOC utiliza as DocStrings dos módulos, para gerar a documentação;
- Caso não existir uma docstring, o pydoc tenta obter uma descrição do bloco de linhas de comentário logo acima da definição da classe, função ou método no arquivo de origem, ou na parte superior do módulo;

Pydoc

- A função **help ()** chama o sistema de ajuda on-line no interpretador interativo, que usa o pydoc para gerar sua documentação como texto no **console**.
- A mesma documentação de texto também pode ser vista de fora do interpretador Python, executando o pydoc como um script no prompt de comando do sistema operacional. Por exemplo, execute no linux: **pydoc sys**

Pydoc

- Especificar um sinalizador -w antes do argumento fará com que a documentação em HTML seja gravada em um arquivo no diretório atual, em vez de exibir texto no console;
- Para iniciar um servidor **HTTP** na máquina local, que servirá a documentação para os navegadores Web (**pydoc -p 1234**);
- Especificar **0** como o número da porta, será selecionada uma porta não utilizada arbitrariamente;
- Para gerar documentação na forma de páginas html ou até mesmo no console (**python pydoc -w arquivo**).

Atividade

Atividade

- Como você já entendeu como documentar um código e gerar a documentação utilizando o PYDOC. Cria uma classe e empregue todo o conhecimento adquirido nessa aula e utilizando as outras bibliotecas para gerar a documentação do seu código usando a: **Sphinx** e/ou **Doxygen.**

Referências

- (TUTORIAL) DOCSTRINGS IN PYTHON. 10 abr. 2020. Disponível em: https://www.datacamp.com/community/tutorials/docstrings-python. Acesso em: 15 maio 2021.

- (TUTORIAL) DOCSTRINGS IN PYTHON. 10 abr. 2020. Disponível em: https://www.datacamp.com/community/tutorials/docstrings-python. Acesso em: 15 maio 2021.

- (PYDOC — GERADOR DE DOCUMENTAÇÃO E SISTEMA DE AJUDA ONLINE — DOCUMENTAÇÃO PYTHON 3.9.5, [s. d.])PYDOC — GERADOR DE DOCUMENTAÇÃO E SISTEMA DE AJUDA ONLINE — DOCUMENTAÇÃO PYTHON 3.9.5. [s. d.]. Disponível em: https://docs.python.org/pt-br/3/library/pydoc.html. Acesso em: 15 maio 2021.

- STYLEGUIDE. [s. d.]. Disponível em: https://google.github.io/styleguide/pyguide.html. Acesso em: 15 maio 2021a.