Estrutura Python

Quando um utilizador ou colaborador em acede à página do seu repositório, ele vê algumas coisas:

* Nome do Projeto
* Descrição do Projeto
* Arquivos e pastas de arquivos

Estrutura do Projeto:

* **Nossos ficheiros de código**
* **Licença** **-** O texto completo da licença e as reivindicações de direitos autorais devem constar neste arquivo -> <https://choosealicense.com/>
* **Setup.py** **-** É um arquivo python, cuja presença é uma indicação de que o módulo ou pacote que estamos prestes a instalar provavelmente foi empacotado e distribuído com Distutils(Pacote que fornece suporte para construir e instalar módulos adicionais em uma instalação Python), que é o padrão para a distribuição de módulos Python. Isso permite que instalemos facilmente os pacotes Python. Muitas vezes é o suficiente para escrever o comando no Cmd.: **pip install**
* **Requirements.txt –** Deve ser colocado na raiz do repositório, especificar as dependências necessárias para contribuir com o projeto: teste, construção e geração de documentação

Se o projeto não tiver dependências de desenvolvimento ou se preferirmos configurar um ambiente de desenvolvimento via setup.py, o ficheiro Requirements.txt pode ser dispensado.

* **Test\_sample.py –** Execução de testes - <https://docs.python-guide.org/writing/tests/>

Algumas regras: Cada teste deve ser composto por uma parte de funcionalidade e provar que está correta. – Ex: Login Correto

Fazer testes que sejam rápidos.

Usar nomes longos e descritivos. – Ex: test\_login\_success()

* **Makefile –** Mantém o projeto atualizado reconstruindo apenas as partes necessárias do nosso código fonte que as bibliotecas ou partes de código importantes estejam desatualizadas. Também pode automatizar a compilação builds e testes ou tornando a rotina parte do projeto com script ou automatizado. **-** <https://stackabuse.com/how-to-write-a-makefile-automating-python-setup-compilation-and-testing/>**,** <https://www.youtube.com/watch?v=X1YW4GeROac>



* **\_\_init\_\_.py –** É um diretório de pacotes python.

Geralmente está vazio(Quando os módulos e subpacotes do pacote não precisam de partilhar nenhum código), mas pode ser usado para exportar partes selecionadas do pacote, com um nome mais conveniente, manter funções de conveniência, …

Reúne todas as definições de todo o pacote.

Permite definir qualquer variável no nível do pacote.



Qualquer diretório com um “\_\_init\_\_.py” é considerado um pacote Python.

Os ficheiros que forem ser chamados em módulos devem:

* Ser nomeados com letras minúsculas
* Evitar usar “.” e “?”
* Usar nomes curtos

 O que não fazer Ex: from my.file.py import function

 O que fazer Ex: from myfile.py import function

Devemos evitar usar o mesmo nome para coisas diferentes.

**Link do Site -** <https://docs.python-guide.org/writing/structure/>