# Sequência projeto trigonometria

- **1.** Crie uma pasta para o projeto (**3-py-interface-trigonometria**). E abra com VSCode.
- 2. Crie um ambiente virtual (opcional, mas recomendado) abra um terminal e digite:

#### python -m venv venv

Ative o ambiente virtual:

• No Windows (No terminal digite):

#### venv\Scripts\activate

Instale as dependências necessárias:pip install Pillow

- 4. Criar um arquivo chamado **interface3.py** no diretório do projeto.
- 5. Digite a codificação (Imagem em anexo).
- 6. Execute o programa: python interface3.py

# Por que usar ambiente virtual?

Recomendo o uso de um ambiente virtual por várias razões importantes:

- 1. Isolamento de dependências: Cada projeto pode ter suas próprias versões de bibliotecas, evitando conflitos entre projetos diferentes.
- 2. Reprodutibilidade: Facilita a recriação exata do ambiente de desenvolvimento em outras máquinas ou em produção.
- 3. Limpeza do sistema: Evita a poluição do Python global do sistema com múltiplas bibliotecas.
- 4. Segurança: Reduz riscos de segurança ao isolar as dependências de cada projeto.
- 5. Facilidade de gerenciamento: Permite gerenciar facilmente as dependências específicas de cada projeto.

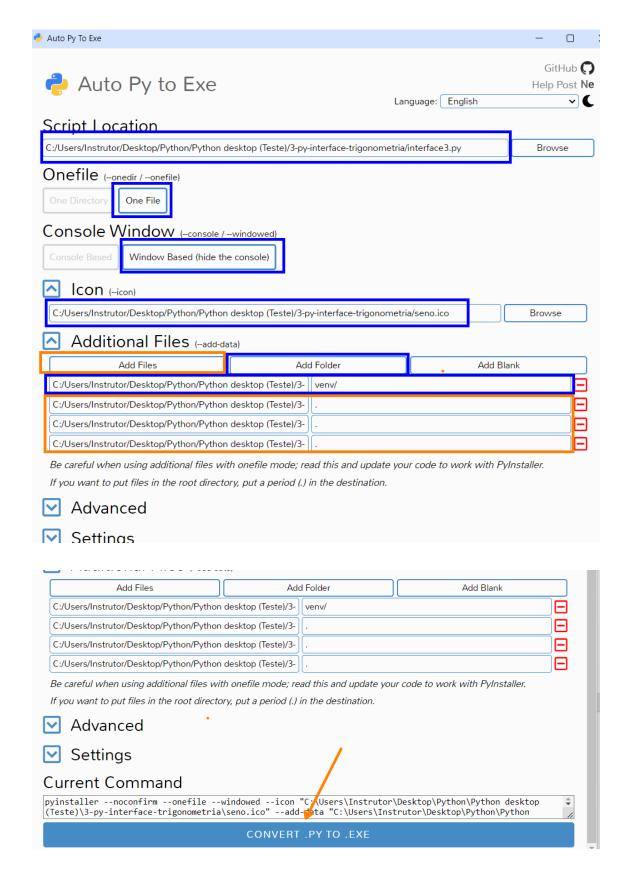
- 6. Portabilidade: Torna o projeto mais portátil entre diferentes sistemas e desenvolvedores.
- 7. Testes de diferentes versões: Permite testar o projeto com diferentes versões de Python ou bibliotecas.
- 8. Facilita a colaboração: Outros desenvolvedores podem replicar facilmente o ambiente do projeto.
- 9. Melhores práticas: É considerado uma boa prática no desenvolvimento Python profissional.
- 10. Fácil limpeza: Se algo der errado, você pode simplesmente deletar o ambiente virtual e criar um novo, sem afetar o resto do sistema.

Usar um ambiente virtual é especialmente útil em projetos maiores ou quando você trabalha em múltiplos projetos com diferentes requisitos. Para projetos pequenos ou experimentos rápidos, pode não ser estritamente necessário, mas ainda é uma boa prática a se adotar.

### Gerar exe usando auto-py-to-exe

1. Rodar (Interface): auto-py-to-exe

**Obs:** Adicionar todas imagens e pasta com ambiente virtual para correta compilação.



#### 2. Conferir executável.

