PYTHON CHEAT SHEET

# GIT

## - Criar Repositório no GitHub

* <https://github.com/new>
* Preencher o nome do repositório
* exemplo de repositório criado **https://github.com/username/nome\_do\_repositorio**

## - Configurar seu github no terminal

* git config --global user.name "Seu Nome"
* git config --global user.email "[email@domain](mailto:email@domain)"

## - Clonar seu github

* Escolha a pasta em seu computador onde está o projeto
* Clone com https: **git clone https://github.com/username/nome\_do\_repositorio**
* Clone com SSH: **git clone** [**git@github.com**](mailto:git@github.com)**:username/nome\_do\_repositorio.git**

## - Comandos básicos

* **git status** para verificar o estado dos arquivos e se existem mudanças a aplicar
* **git pull** para baixar mudanças do github para o local
* **git add** **.** (para todos os arquivos) ou **git add nome\_do\_arquivo.extensao** para adicionar arquivos novos e adicionar mudanças em arquivos existentes
* **git commit -m** para o comentar as mudanças realizadas e marcar um checkpoint no histórico do repositório
* **git push** para enviar as mudanças locais para o github

# SCRIPT

## - Shebang

O comentário especial **Shebang** especifica qual interpretador será usado para executar o programa. Desta forma é possível omitir o interpretador e executar o script diretamente pelo seu nome.

* No arquivo Python:
* No terminal:

- Permissão para executar o script:

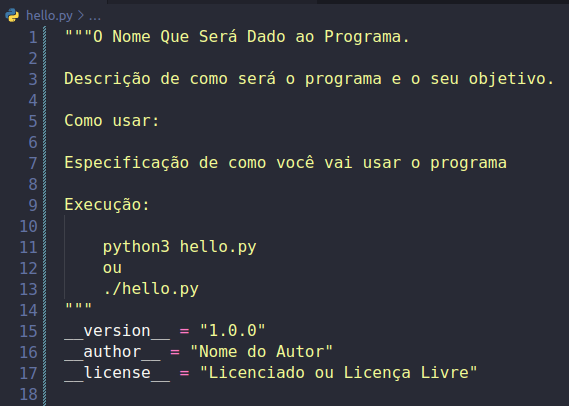
- As duas formas de executar o script:

Figura 1: Especificando o interpretador

Figura 2: Usando o Shebang

## - Docstring e metadados dunder

Em todo script Python é uma boa prática incluir um comentário de multi linhas logo nas primeiras linhas do script explicando o objetivo do script e provendo documentação para o usuário.



# LINUX

## - Comandos Básicos

No terminal:

* **Touch**: O comando touch do Linux cria um arquivo vazio. Ex.: touch .arquivo
* **Echo**: O como echo adiciona o texto para um determinado arquivo. Ex.: echo “texto” >> .arquivo

# AMBIENTE VIRTUAL

## - Venv

No terminal:

* **Criando o ambiente virtual**: Dentro da pasta que deseja criar, *python3 -m venv .nomedavenv*(obs.: Ao digitar o nome da venv e inserir “.” antes do nome, ele oculta a pasta).
* **Ativando o ambiente virtual**: A pasta criada da venv tem o arquivo “activate” para ativar o ambiente você deve utilizar o comando *source*. Ex.: source .venv/bin/activate