Seu primeiro script

**Repositório**

Antes de começar a programar seu primeiro script será necessário criar um repositório no github, neste local vamos armazenar e controlar o histórico de alterações do programa.

Para um criar o repositório, após fazer login no github acesse https://github.com/new basta preencher o nome com python-base e clicar em create repository.

O seu novo repositório estará acessível em:

https://github.com/USERNAME/python-base

**USERNAME** deve ser substituido pelo seu nome de usuário do github.

Configure seu github

O primeiro passo é certificar que localmente no seu computador ou no terminal em que você está usando para o treinamento você possui o git e ele esteja configurado para seu usuário, basta digitar os comandos:

Substitua "Seu Nome" pelo seu nome completo

git config --global user.name "Seu Nome"

Substitua "email@domain" pelo seu verdadeiro e-mail, o mesmo utilizado no github

git config --global user.email "email@domain"

Opcionalmente, você pode dizer ao git qual seu editor de código favorito

Substitua "micro" pelo seu editor de escolha ex: vim, nano, code, sublime

git config --global core.editor "micro"

Clone

Agora que seu git está configurado, vamos fazer o clone do seu repositório este processo vai criar uma cópia do repositório na sua máquina local.

Para fazer o clone existem alguns detalhes:

1. Escolha o caminho onde vai armazenar seu projeto, eu por exemplo coloco na minha pastas de usuário em um subdiretório chamado Projects ex: ~/Projects acesse esta pasta antes de fazer o clone, você pode escolher o local que preferir.

mkdir ~/Projects # caso ainda não exista

cd ~/Projects

1. Agora você precisa escolher entre 2 modos de autenticação, o github suporta https e ssh, o ssh é o mais seguro e recomendado, mas exige que você tenha suas chaves SSH configuradas, você pode ver como fazer isso aqui: https://docs.github.com/pt/authentication/connecting-to-github-with-ssh
2. Efetue o clone

Por https (não precisa de SSH, só tem que informar usuario e senha quando for alterar o repositório)

git clone https://github.com/USERNAME/python-base

por SSH (vantagem é não precisar informar usuário e senha quando alterarmos o repositório)

git clone git@github.com:USERNAME/python-base.git

Acesse seu repositório

cd python-base

Durante o treinamento usaremos com frequencia os seguintes comandos do git:

* git status para verificar o estado dos arquivos e se existem mudanças a aplicar
* git add para adicionar arquivos novos e adicionar mudanças em arquivos existentes
* git commit para confirmar uma alteração e marcar um **checkpoint** no histórico do repositório
* git push para enviar as mudanças locais para o github
* git pull para baixar mudanças do github para o local
* git commit -p para o commit em modo interativo

**Seu Primeiro Script :)**

Obrigatoriamente seu primeiro script deve ser um programa que imprime na tela a frase "Hello, World!"

Um script nada mais é do que o conjunto de comandos que enviamos individualmente ao interpretador mas de uma forma organizada em um único arquivo.

Da mesma forma que é possível executar

python -c "print('Hello, World!')"

python -c "print(1 + 1)"

VocÊ pode criar um arquivo e colocar os comandos em sequencia, linha a linha dentro dele.

utilize o editor de sua preferencia para criar um arquivo

micro hello.py

Com o editor aberto coloque o conteúdo do seu script.

print("Hello, World!")

Salve o script ctrl + S na maioria dos editores e em outro terminal execute

python3 hello.py

Hello, World!

repare que usamos python3 para ser especifico em relação a versão do python utilizada para executar.

Ao escrever uma nova parte do código é muito importante após efetuar os testes fazer o commit para salvar o código.

# exibe o status do repositório

git status

# adiciona o arquivo novo ou alterado no histórico do git

git add hello.py

# Efetua um commit marcando a alteração\

git commit -m "Criado meu primeiro hello world"

# envia as mudanças para o Github

git push # sua senha pode ser pedida aqui caso tenha usado https

**Comentários no código**

Em python é possível adicionar partes do código que serão ignoradas pelo interpretador, essas linhas são úteis para adicionar comentários, lembretes e metadados do programa.

Exemplos

# Comentário de linha

print("Hello") # Comentário de final de linha

"""

comentário

multi

linhas

"""

As linhas iniciadas em # são ignoradas pelo Python assim como tudo que estiver após # em uma linha de código e também todo o contéudo dentro de """

**Shebang**

Em ambientes Linux é muito importante definir o comentário especial Shebang, nele especificamos qual interpretador será usado para executar o programa

#!/usr/bin/env python3

print("Hello, World!")

A primeira linha informa o terminal que aquele programa roda com o Python3 da env em execução, esta forma é possivel omitir o interpretador e executar o script diretamente pelo seu nome.

# primeiro damos parmissão de execução ao script

chmod +x hello.py

Agora podemos executar de 2 formas

# especificando o interpretador na linha de comando

python3 hello.py

# usando o interpretador especificado na linha `#!/usr/bin/env python`

./hello.py

A vantagem da segunda forma é que podemos mudar a extensão de .py para .exe por exemplo, ou podemos até remover a extensão e executar ./hello