

RS-20182-FSPOA-ALG2-PRE-ADS3N18-2 Algoritmos e Programação II
(ADS3N18-2)

| Conteúdos > Exercícios > Aula_1

AULA_1



Aprendendo Python

Objetivo:

Aprender o básico da linguagem Python, realizando o curso de Python do CodeAcademy acessível gratuitamente em <https://www.codecademy.com/learn/learn-python>

Tempo estimado: 10h



Exercicio sobre o uso do Git com o Github

Objetivo

Desenvolver um projeto, utilizando a linguagem Python, e controlar o desenvolvimento utilizando o sistema de controle de versão Git.

Ferramentas

Para este exercício você precisará do [Git](#) instalado, e de uma forma de criar e gerenciar um projeto de software na plataforma que você escolher.

Sugere-se a utilização do Git por meio da linha de comandos. Você pode obter a ferramenta na [página de downloads do Git](#). No Windows™ você pode utilizar o programa [Git Bash](#). No Linux, você deve utilizar o sistema de pacotes da distribuição que você utiliza para instalar o Git. No macOS™ você pode utilizar o Xcode™ e instalar as ferramentas de linha de comando (o que é recomendado), ou instalar o Git a partir de um gerenciador de pacotes como o [Homebrew](#).

Você pode utilizar também o programa Github Desktop, porém, lembre-se que este programa pode não estar disponível (por exemplo, na faculdade), e funciona apenas em conjunto com o Github, excluindo outros provedores de serviço.

Tarefas

As tarefas devem ser criadas na ordem em que foram especificadas.

1. Implemente, dentro do diretório **python**, um programa que calcule a média de dois valores, com o nome de **media.py**.
2. Abra o terminal (no Windows™, utilize o Git Bash), e vá até o diretório que você criou.
3. Uma vez *dentro* do diretório do projeto, inicialize o repositório (**git init**).
4. Crie um arquivo **.gitignore**, e neste arquivo adicione o seguinte conteúdo:

```
__pycache__/  
*.pyc  
*.pyo  
*.tmp  
*.~  
*.swp
```

1. O arquivo **.gitignore** deve ser tratado como um arquivo fonte, e deve ser adicionado e gravado no repositório git.
6. Após inicializar e configurar o repositório verifique o estado dos arquivos locais (**git status**) e adicione e grave no repositório os arquivos criados.

7. Modifique o arquivo **media.py**, e adicione um novo método que calcula o maior de três números.
8. Adicione as alterações ao repositório.
9. Crie um repositório no [Github](#), com o nome de "algoritmos2-2018-2". (Caso você ainda não tenha um usuário, você deverá criar um.)
 1. Crie um repositório público, pois os privados são pagos, e você não precisará de um repositório pago.
 2. Crie um repositório vazio, sem adicionar arquivos de licença, README, ou gitignore.
10. Configure, de acordo com as instruções no site Github.com, o seu repositório de forma que este utilize o repositório criado no Github como repositório remoto.
11. Sincronize o repositório local com o repositório no Github, enviando as modificações para o Github. Verifique, no site do Github, se seu código está aparecendo lá.
12. Clone o repositório do Github em um outro diretório do seu computador, para verificar que tudo está funcionando normalmente.
13. Vá ao site do Github, e crie, a partir do site, um arquivo **README.md**. Adicione este arquivo ao repositório.
14. Sincronize o seu repositório local com o Github, trazendo, do Github, as modificações. Você agora deve ter um arquivo **README.md** no seu repositório local.

Material de Apoio

- [Tutorial sobre o Git](#)
- [Curso de Git no Code Academy](#)
- [Curso de Git na Udacity](#)
- [Curso Introdutório de comandos Linux da Udacity](#) (útil para o Git Bash)



Configuração do Atom

Objetivos:

Configurar o editor de textos **Atom** (<https://atom.io>) com *plugins* que auxiliarão na disciplina.

Tarefas:

Instalar o editor de textos Atom e os seguintes *plugins*:

- behave-step
- language-gherkin
- language-gherkin-i18n
- linter
- linter-flake8
- platformio-ide-terminal