

### Instruções

- Esta décima primeira lista é uma lista sobre manipulação de arquivos. Essa é uma lista curta. O objetivo é que você tenha uma visão geral sobre manipulação de arquivos.
- Utilize a linguagem python para a execução de todos os exercícios na lista.
- **Evite o uso de funções prontas da linguagem.** Tente fazer você mesmo para entender melhor a lógica de programação. Nem todas as linguagens oferecem as mesmas funcionalidades que o python.

### Exemplo

Enunciado: Faça um programa que escreve um arquivo com números em ordem crescente de 1 a 150. Cada número deve ser separado por uma vírgula. O arquivo deve se chamar “ordenado.txt”.

Solução:

```
arquivo1 = open("ordenado.txt", "w")

for i in range(1, 150):
    arquivo1.write(str(i)+", ")

arquivo1.write("150\n")
arquivo1.close()
```

Comentários:

1. Primeiro criamos o arquivo nomeado ordenado.txt e abrimos ele no modo escrita: “w”. Depois preenchemos o arquivo com os valores de 1 a 149 escrevendo uma vírgula após o número. Por fim, escrevemos o valor 150 e não colocamos uma vírgula depois, mas sim uma quebra de linha. Então fechamos o arquivo.
2. A saída desse programa ficará salvo em um arquivo com nome ordenado.txt e estará salvo na pasta onde se encontra o programa.
3. Há outras formas de fazer esse mesmo exercício. Não se prenda a essa! Encontre o seu jeito de fazer!

## Lista 11

1. Faça um programa que escreva um arquivo com números em ordem decrescente de 200 a 50. Cada número deve estar em uma linha. Dê o nome que quiser para o arquivo.
2. Faça um programa que leia um número N e gere um arquivo com N nomes e idades aleatórios:
  - Faça uso de duas listas criadas na mão: uma que contenha 20 nomes e outra que contenha 20 sobrenomes;
  - Cada linha do arquivo resultante deve conter um nome completo e a sua idade;
  - Todos os dados podem ser gerados no próprio código sem a necessidade de fazer a leitura por meio de um arquivo.
3. Estenda o exemplo do cadastro para considerar também a altura da pessoa
4. Escreva um programa que receba dois nomes de arquivos e copie o conteúdo do primeiro arquivo para o segundo arquivo. Considere que o conteúdo do arquivo de origem é um texto. Seu programa não deve copiar linhas comentadas (que começam com //)
5. Faça um programa que recebe os nomes de dois arquivos. O primeiro arquivo contém nomes de alunos e o segundo arquivo contém as notas dos alunos. No primeiro arquivo, cada linha corresponde ao nome de um aluno e no segundo arquivo, cada linha corresponde às notas dos alunos (uma ou mais). Assuma que as notas foram armazenadas como strings, e estão separadas umas das outras por espaços em branco. Leia os dois arquivos e gere um terceiro arquivo que contém o nome do aluno seguido da média de suas notas.
6. Faça um programa para alterar uma das notas de um aluno (usando os arquivos do exercício anterior). O programa deve receber o nome do aluno, a nota velha e a nova nota.
7. Faça um programa que leia um arquivo texto contendo uma lista de endereços IP e gere dois outros arquivos, um contendo os endereços IP válidos e outro contendo os endereços inválidos. O formato de um endereço IP é num1.num.num.num, onde num1 vai de 1 a 255 e num vai de 0 a 255.