

# Programação em Python

## Arquivos

Prof. Daniel Di Domenico

[https://github.com/danidomenico/gex003\\_algprog](https://github.com/danidomenico/gex003_algprog)

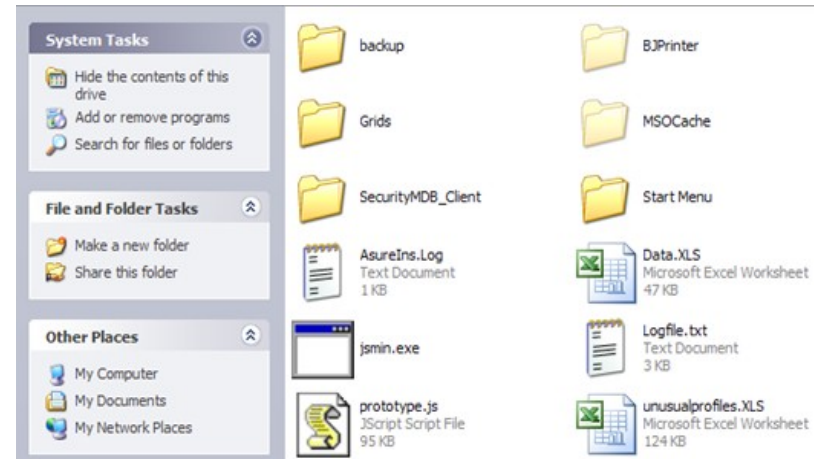
Slides cortesia da profa. Andrea Charão (UFSM) e do prof. João V. F. Lima (UFSM)

# Operações com arquivos



# Operações com arquivos

- São realizada com funções;
- Não é preciso usar “import” no caso de funções básicas;
- Algumas funções básicas:
  - **open**: abre arquivo;
  - **close**: fecha o arquivo;



## ATENÇÃO!

- Abra o arquivo (open) antes de qualquer operação;
- Feche o arquivo (close) após as operações.

# Abrir e fechar arquivo

```
#Abre o arquivo numeros.txt em modo leitura  
arquivo = open("numeros.txt", "r")  
  
#Fecha o arquivo  
arquivo.close()
```

numeros.txt



5  
6  
7  
8.8  
5

- Modo de abertura do arquivo:
  - **r** = abre o arquivo somente para leitura (padrão);
  - **w** = abre o arquivo somente para escrita, apagando seu conteúdo antigo;
  - **a** = abre o arquivo para escrita, mantendo seu conteúdo e adicionando novas informações ao seu final.

# Leitura de um arquivo

```
arquivo = open("numeros.txt", "r")
print(arquivo.read()) #Imprime todo o arquivo

arquivo.seek(0) #Retorna o cursor ao início do arquivo
print(arquivo.readline()) #Imprime primeira linha
5\n

arquivo.seek(0)
lista = arquivo.readlines() #Cada linha do arquivo torna-se
print(lista)                #um elemento de uma lista
['5\n', '6\n', '7\n', '8.8\n', '5\n']

arquivo.close()
```

numeros.txt

5  
6  
7  
8.8  
5

Leitura dos  
dados como  
string

- Atenção ao **cursor** do arquivo:
  - Ao abrir o arquivo, o cursor é setado para seu início;
  - Após ler alguma informação do arquivo, o cursor permanece na última posição lida;
  - Utilize o método `seek(0)` para retornar ao início.

# Leitura de um arquivo

```
arquivo = open("numeros.txt", "r")

print(arquivo.readline().strip()) #strip para remover \n
5

arquivo.seek(0)
#Cada linha do arquivo torna-se um elemento de uma lista
lista = arquivo.read().splitlines() #Desta forma, não inclui \n
print(lista)
['5', '6', '7', '8.8', '5']

arquivo.close()
```

numeros.txt



5  
6  
7  
8.8  
5

- Ler dados sem `\n` ao final da linha:
  - Para ler linha a linha, utilize o método `strip()`;
  - Para ler o arquivo como uma lista, utilize `read().splitlines()`.

# Escrita de um arquivo

```
arquivo = open("arq1.txt", "w")
```

```
arquivo.write("1")
```

```
frase = "abcde"
```

```
arquivo.write(frase)
```

```
#Para quebrar linha, é preciso utilizar \n
```

```
arquivo.write("nova linha\n")
```

```
#Escrita de número (converter para string)
```

```
arquivo.write(str(1) + "\n")
```

```
arquivo.close()
```

arq1.txt

```
1abcde  
nova linha  
1
```

- É possível escrever no arquivo textos e variáveis:
  - Apenas strings podem ser escritas.

# Escrita de um arquivo

```
arquivo = open("arq1.txt", "w")

print("1", file=arquivo)
frase = "abcde"
print(frase, file=arquivo)

#Escrita de número sem conversão
print(123, file=arquivo)

arquivo.close()
```

arq1.txt

```
1
abcde
123
```

- É possível utilizar o comando *print*:
  - Trocar o parâmetro **file** para apontar para o arquivo;
  - Podem ser utilizados os parâmetros **end** e **sep**;



# Exemplo: somar arquivo

```
arquivo = open("numeros.txt", "r")

soma = 0
linha = arquivo.readline() #Lê primeira linha
while linha != "": #Se chegar ao fim do arquivo, linha == ""
    soma = soma + float(linha) #Converte para float
    linha = arquivo.readline() #Lê próxima linha

arquivo.close()
print("A soma é:", soma)
A soma é: 31.8
```

numeros.txt

5  
6  
7  
8.8  
5

- Leitura de um arquivo inteiro:
  - Lê linha a linha através do método *readline()*;
  - Converte o conteúdo da linha (número) para **float**;
  - Caso não existam mais linhas, *readline()* retorna "";

# Exemplo: somar arquivo

```
arquivo = open("numeros.txt", "r")

soma = 0
linhas = arquivo.read().splitlines()
for lin in linhas:
    soma = soma + float(lin) #Converte para float

arquivo.close()
print("A soma é:", soma)
A soma é: 31.8
```

numeros.txt

5  
6  
7  
8.8  
5

- Leitura de um arquivo inteiro:
  - Lê todas as linhas através do método *read().splitlines()*:
    - Retorna uma lista com cada linha sendo um elemento;
  - Converte o conteúdo da linha (número) para **float**.

# Exercícios

- 1) Escreva um programa que calcule a média dos valores contidos no arquivo `numeros.txt`.
- 2) Faça um programa que leia o nome e duas notas de 5 alunos e mostre a média final de cada estudante. O cálculo da média deve ser realizado em uma função. Após calcular a média, escreva o nome do aluno e sua média no arquivo `medias.txt`.

# Arquivos CSV

- É um formato para arquivos de dados;
- Usado em planilhas e bancos de dados;
- **CSV = Comma-separated values:**
  - Dados separados por vírgula (ou outro caracter)

- Exemplos:

```
3.5, 5.6  
4.1, 4.3  
2.1, 3.5  
3.8, 4.7
```

```
"Fulano", 5.6  
"Beltrano", 8.3  
"Sicrano", 7.0
```

```
10,13,11,10,"Sim, um pouco","Sim, um pouco",Feminino,39  
9,14,18,16,Não,"Sim, um pouco",Masculino,30  
15,18,14,18,"Sim, um pouco",Não,Feminino,20  
5,15,12,8,Não,"Sim, um pouco",Feminino,20  
9,20,14,14,"Sim, um pouco","Sim, um pouco",Feminino,28
```

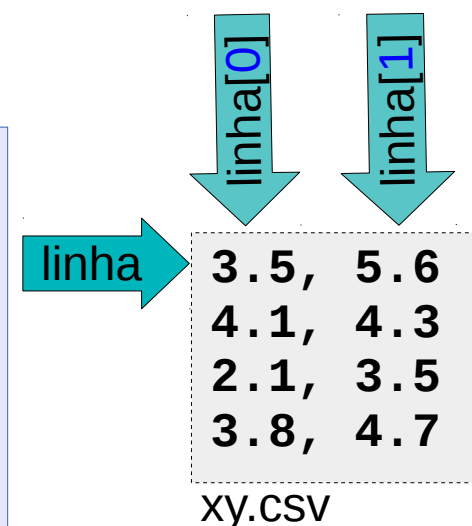
# Arquivos CSV e listas

- Função para criar uma lista com dados de arquivo CSV: `csv.reader(arquivo)`;
- Para usar, importar a biblioteca **csv**:

```
import csv
arquivo = open("xy.csv")
leitorcsv = csv.reader(arquivo)

#Laço para percorrer todas as linhas do arquivo
for linha in leitorcsv:
    x = float(linha[0])
    y = float(linha[1])
    print(x, y)

arquivo.close()
```



# Exemplo: arquivo CSV

- Análise dos dados coletados:

```
import csv
import matplotlib.pyplot as pyplot

arquivo = open("form-google-coleta.csv")
leitorcsv = csv.reader(arquivo)
conta_pontos = [0,0,0,0]

for linha in leitorcsv: #conta pontos das colunas 0 e 1
    conta_pontos[0] = conta_pontos[0] + int(linha[0])
    conta_pontos[1] = conta_pontos[1] + int(linha[1])

posicao=1 #gera o gráfico através da biblioteca pyplot
for item in conta_pontos:
    pyplot.bar(posicao, item)
    posicao = posicao + 1
pyplot.show()
arquivo.close()
```

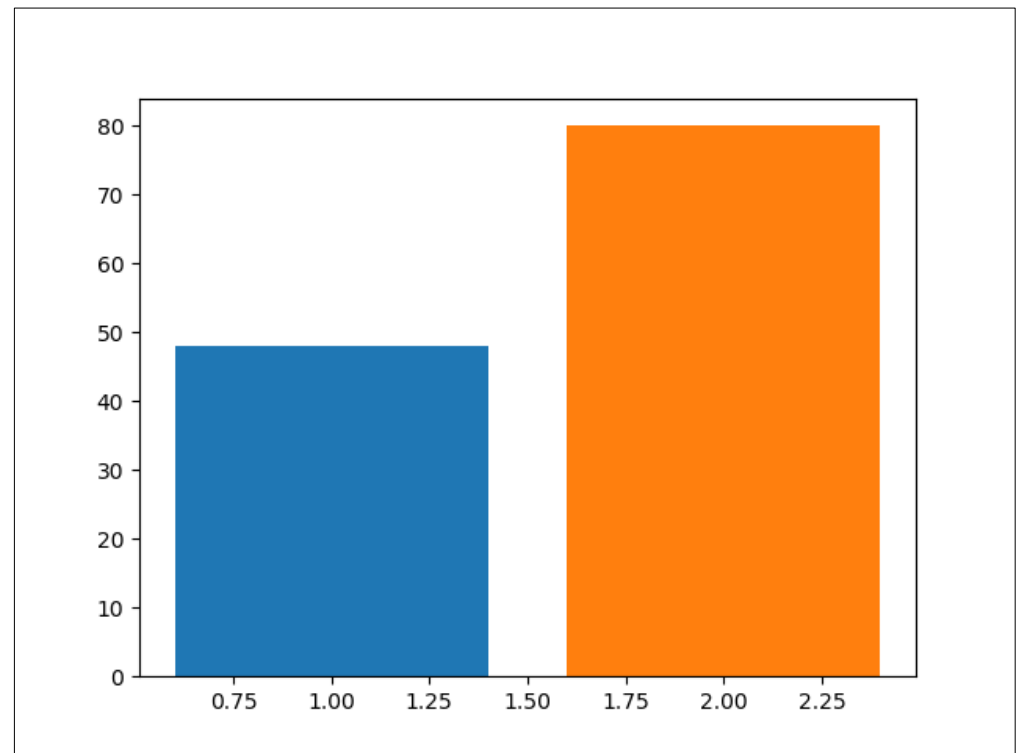
# Exemplo: arquivo CSV

- Análise dos dados coletados:

```
10,13,11,10,"Sim, um pouco","Sim, um pouco",Feminino,39  
9,14,18,16,Não,"Sim, um pouco",Masculino,30  
15,18,14,18,"Sim, um pouco",Não,Feminino,20  
5,15,12,8,Não,"Sim, um pouco",Feminino,20  
9,20,14,14,"Sim, um pouco","Sim, um pouco",Feminino,28
```

dados coluna azul

dados coluna laranja



# Exercícios

- 1) Baixe o arquivo de dados `form-google-coleta.csv` e complete o programa do slide 14 de modo a contar os pontos das colunas 2 e 3 do arquivo, gerando o gráfico abaixo.

