

AULA 1 - INTRODUÇÃO

Na oficina cita dois programas: **Jupyter Notebook** e **Colab Research Google**.
No **Jupyter Notebook** você pode ir em **Run** ou **CTRL + Enter** para executar.

AULA 2 – OPERAÇÕES BÁSICAS E CONDIÇÕES

“#” o símbolo de sharp representa o comentário de uma linha.

```
In [2]: 1 # Lê um dado da tela e imprime o resultado
2 print('Primeiros passos em python')
3 print()
4 nome = input("Entre com o seu nome:")
5 print('Olá', nome)
```

Para pular linha

Recebe o que o usuário digita e ao lado é a frase que vai aparecer para o usuário digitar.

Conteúdo da variável

```
numero1 = 10
numero2 = 3
soma = numero1+numero2
subtração = numero1-numero2
multiplicação = numero1*numero2
divisão = numero1/numero2
divisão_inteira = numero1//numero2
print('O valor da soma é:', soma)
print('O valor da subtração é:', subtração)
print('O valor da multiplicação é:', multiplicação)
print('O valor da divisão é:', divisão)
print('O valor da divisão inteira é:', divisão_inteira)
```

Ele descarta o resto da divisão, e é mostrado somente o valor inteiro

Operadores de comparação

Operator	Description
==	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

Operadores lógicos

Operator	Description
and	Both conditions must be met
or	Either condition must be met

```
In [5]: 1 temperatura = 10
2 if temperatura > 10:
3     print('Temperatura maior que 10')
4 elif temperatura == 10:
5     print('Temperatura igual a 10')
6 else:
7     print('Temperatura é menor que 10')
8 RESULTADO = "O RESULTADO É:"
9 print(RESULTADO, temperatura)
10 resultado = str.lower(RESULTADO)
11 print(resultado)
```

Automaticamente ele já identifica como um bloco de condicional

"senão se"

Se não for nenhum e nem outro.

Automaticamente identifica como uma string, a variável RESULTADO irá imprimir o que está escrito entre "".

Função para deixar o conteúdo dentro da variável em letra minúscula.

Temperatura igual a 10
O RESULTADO É: 10
o resultado é:

AULA 3 – STRINGS E FUNÇÕES MATEMÁTICAS

Strings:

No python, quando o usuário entra com um número no **input** o python interpreta como se fosse uma string, então você deve converter esse **input** para int ou float.

```
1 nome1 = "zezinho"
2 print(nome1)
3 nome2 = 'huguinho'
4 numero = input('Entre com um número:')
5 resultado = int(numero) + 10
6 print(resultado)
```

Você pode fazer a **concatenação**, como exemplo, o número de uma casa como endereço. Você consegue pegar o número armazenado na variável e imprimi-lo na tela.

```
7 numero_casa = 123
8 endereco = 'Av. Brasil,' + str(numero_casa) + '\n' + "Maringá, Pr"
9 print(endereco)
```

→ PULA LINHA

→ IMPRIME O NÚMERO EM FORMA DE STRING

Você pode obter o **tamanho de uma variável**, utilizando a função **len** e usando o **print** para imprimir na tela. É possível também aumentar ou diminuir os caracteres para maiúsculo/minúsculo utilizando a função **str.upper/str.lower** e **print** para imprimir na tela.

```
10 comprimento_nome1 = len(nome1)
11 print(comprimento_nome1)
12 nome1_maiusculo = str.upper(nome1)
13 print(nome1_maiusculo)
```

→ OBTÉM O TAMANHO DE UMA VARIÁVEL

→ DEIXA OS CARACTERES EM LETRA MAIUSCULA, SE ESTIVER PEQUENO.

Você pode também obter **um determinado trecho de um texto**,

```
14 texto = 'Este é um texto de teste'
15 print(texto[2:8])
```

→ IMPRIMI DA POSIÇÃO INICIAL '2' ATÉ A FINAL '8'

IMPRIMIUI: "te é u"

Você pode usando strings, **fazer que o usuário escolha uma opção**, utilizando input para que o usuário digite e logo após impor condições com **if**.

```
16 opcao = input("Entre com a opção: triângulo ou retângulo:")
17 if opcao == 'retângulo':
18     print('A opção selecionada é um retângulo')
19 else:
20     print('A opção selecionada é um triângulo')
```

→ O USUÁRIO DIGITA UM OU OUTRO, O PYTHON RECONHECE A OPÇÃO NAS CONDIÇÕES NO 'IF'

Funções matemáticas:

Você pode chamar a biblioteca **math** usando **import math**

```
1 import math
2 numero = 10
3 raiz = math.sqrt(numero)
4 print(raiz)
```

CHAMA A BIBLIOTECA

FUNÇÃO PARA RAIZ QUADRADA

math.sqrt = raiz quadrada

math.pi = constante pi (3,14)

etc...

Você pode usar **"**"** para elevar algum número.

```
1 import math
2 numero = 10
3 raiz = math.sqrt(numero)
4 print(raiz)
5 PI = math.pi
6 print(PI)
7 dez_elevado_dois = numero**2
8
```

PARA ELEVAR AO EXPOENTE ALGUM NÚMERO

Você pode usar **"%"** para calcular o resto da divisão.

```
9 resto = 10%3
10 print(resto)
```

RESTO DA DIVISÃO POR 3 É IGUAL A 1

EXEMPLO:

```
11 base = input("Entre com a base:")
12 altura = input("Entre com a altura:")
13 base_float = float(base)
14 altura_float = float(altura)
15 retangulo = base_float*altura_float
16 print(retangulo)
```

USUÁRIO ENTRA COM OS DADOS

CONVERSÃO DE STRING PARA FLOAT

CÁLCULO DE ÁREA DE UM RETÂNGULO

VALOR = 200


AULA 4 – LOOPS E RANDOM

Principais utilidades usando FOR

É um laço de repetição

For para incremento:

```
In [11]: 1 # Loop for
          2 #for i in range(2,10):
          3 #     print(i)
          4
```




Incremento +2:

```
#for i in range(1, 10, 2):
#     print(i)
```



For para decremento:

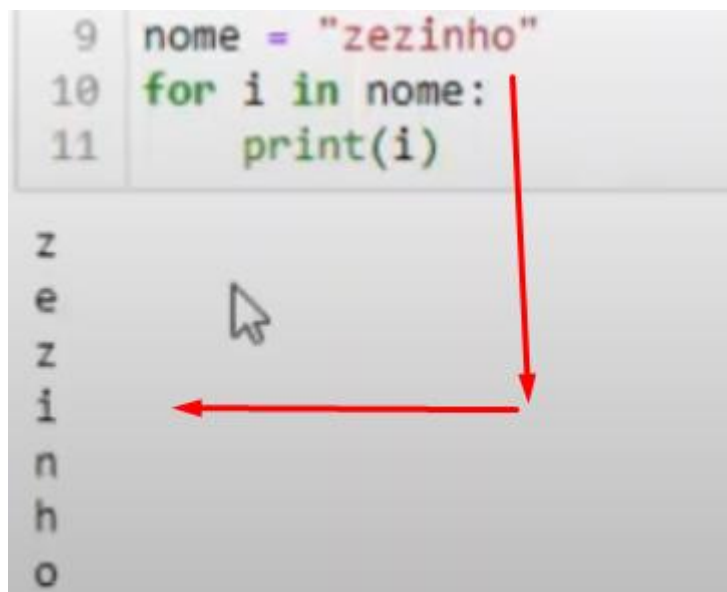
```
7 for i in range(10, 1, -2):
8     print(i)
```



For para string:

Percorre cada elemento da variável nome "Zezinho" e imprime na tela.

```
9 nome = "zezinho"
10 for i in nome:
11     print(i)
```



Principais utilidades usando WHILE

É um laço que não conhecemos a quantidade de interações

while é praticamente, se pensar da seguinte maneira: enquanto **while** for verdade, executa! Se falso, não executa!

```
13 situação = "não"
14 acumulador = 0
15 while situação == "não":
16     numero = input("Entre com um número:")
17     numero_float = float(numero)
18     acumulador = acumulador + numero_float
19     print("O valor do acumulador é:", acumulador)
20     situação = input("Deseja sair:")
21
```

ENQUANTO O USUÁRIO DIGITAR "NÃO", IRÁ EXECUTAR O CÓDIGO DO ESCOPO

ATUALIZA O ACUMULADOR SOMANDO OS NÚMEROS

SE DIGITAR ALGO DIFERENTE DE "NÃO", O PROGRAMA IRÁ FECHAR.

```
Entre com um número:10
O valor do acumulador é: 10.0
Deseja sair:não
Entre com um número:20
O valor do acumulador é: 30.0
Deseja sair:não
Entre com um número:20
O valor do acumulador é: 50.0
Deseja sair:sim
```

Utilizando a função random

Você deve importar a biblioteca random "**import random**", ele gera um número aleatório entre 0 e 1.

```
1 import random
2 #gera um número aleatório entre 0 e 1.
3 numero = random.random()
4 print(numero)
```

CHAMA A BIBLIOTECA RANDOM E A FUNÇÃO RANDOM

Você pode imprimir um número maior do que 0 e 1, manipulando o número aleatório recebido *100:

```
1 import random
2 #gera um número aleatório
3 numero = random.random()
4 print(numero)
5 numero_maior = numero*100
6 print(numero_maior)
```

Você pode gerar um número aleatório dentro de um espaço, utilizando a função `random.randint ()`:

```
7 numero_int = random.randint(0,9)
8 print(numero_int)
```

IMPRIMI UM NÚMERO ALEATÓRIO ENTRE 0 E 9

Você pode escolher de forma aleatória o item de uma lista, utilizando a função `random.choice ([" ", " "])`:

```
9 item = random.choice(["lápis", "borracha", "caneta"])
10 print(item)
```

borracha

AULA 5 – BIBLIOTECA TURTLE

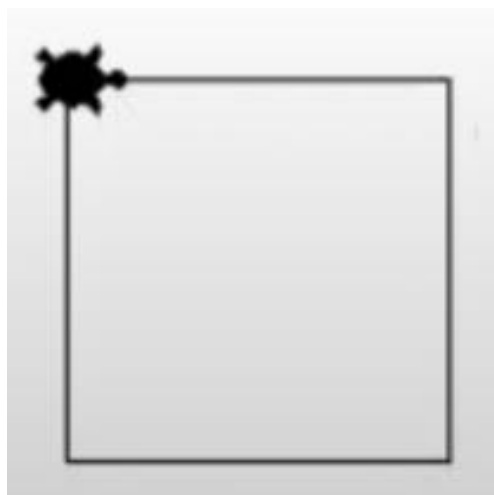
Essa biblioteca permite desenhar formas geométricas, para utilizar basta **import** turtle. Os formatos podem ser: turtle (tartaruga), arrow (seta), circle (círculo).

Você define a forma usando `turtle.shape("turtle")`, ou seja, é responsável pelo traço, uma tartaruga:

```
In [ ]: 1 import turtle
        2 #formato turtle, arrow, circle
        3 turtle.shape("turtle")
        4 for i in range (0,4):
        5     turtle.speed(1)
        6     turtle.forward(100)
        7     turtle.right(90)
        8     turtle.exitonclick()
```

DEFINE A FORMA, O RESPONSÁVEL PELO TRAÇO
REPETE O ESCOPO 4 VEZES, OU SEJA, VAI DO 0 AO 3 (0,4)
VELOCIDADE
ANDAR EM FRENTE
VIRAR A DIREITA EM 90 GRAUS
AO CLICAR NO DESENHO, ELE SAI

RESULTADO



```
import turtle
turtle.title("Vamos colocar mais emoção")
turtle.bgcolor("green")
turtle.shape("turtle")
turtle.pensize(5)
#mudar a cor da turtle
turtle.fillcolor("red")
#mudar a cor do traço
turtle.pencolor("blue")
for i in range(0,10):
    turtle.right(36)
    for i in range(0,5):
        turtle.forward(100)
        turtle.right(72)
turtle.exitonclick()
```

ADICIONA UM TÍTULO A JANELA

COR DA TELA DE FUNDO (BACKGROUND)

RESPONSÁVEL PELO TRAÇO, FORMA

TAMANHO DA CANETA

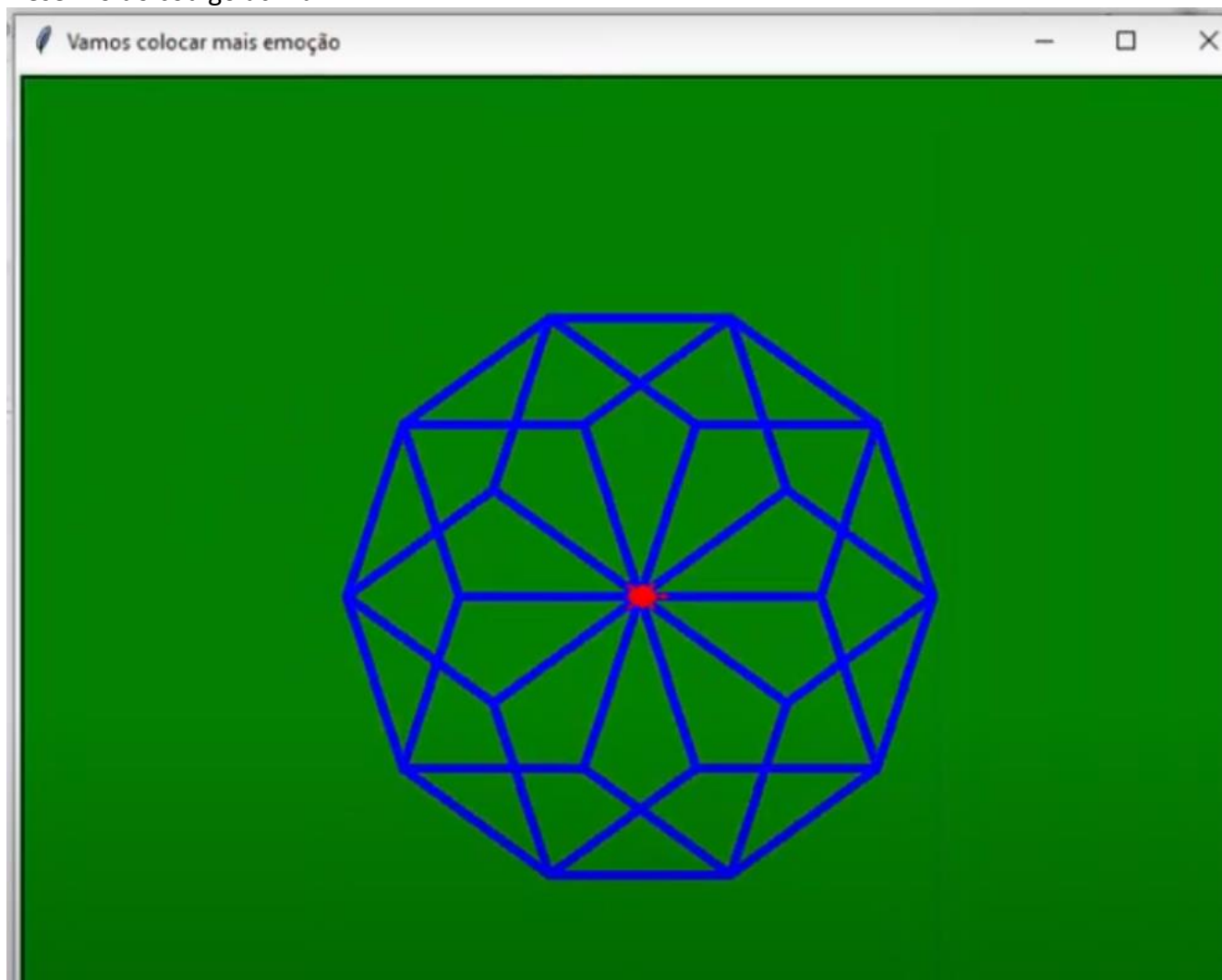
MUDAR A COR DA TARTARUGA

MUDAR A COR DO TRAÇO

TURTLE VIRA A DIREITA 36 GRAUS POR 10 VEZES
E ENQUANTO VIRA A DIREITA... A TURTLE VAI EM FRENTE 100 POSIÇÕES
E VIRA A DIREITA 72 GRAUS

Observação: o **for interno** está desenhando um pentágono, e o **for externo** está executando o pentágono 10 vezes indo 36 graus a direita.

Desenho do código acima:



AULA 6 – CRIAÇÃO DE FUNÇÕES

Para definir uma função utiliza a palavra **def**.

Exemplo: Soma de valores

```
1 def somar(x, y):  
2     z = float(x) + float(y);  
3     return z
```

FUNÇÃO CRIADA PARA SOMAR ELEMENTOS,
OS VALORES DE X,Y ESTÃO SENDO PASSADOS POR PARÂMETRO
E CONVERTIDOS EM FLOAT.

```
5 valor1 = input("Entre com o 1o número:")  
6 valor2 = input("Entre com o 2o número:")  
7 resultado = somar(valor1, valor2)  
8 print("O resultado da soma é:", resultado)
```

CHAMA A FUNÇÃO 'SOMAR' PARA A VARIÁVEL
E PASSA OS VALORES 1 E 2.

Entre com o 1o número:10

Entre com o 2o número:20

O resultado da soma é: 30.0

Você pode fazer uma função para se obter dados e armazenar em uma lista:

```
1 def obter_dados():  
2     nome=input("Entre com o nome:")  
3     idade = input("Entre com a idade:")  
4     lista_dados = [nome, idade]  
5     return lista_dados
```

ELEMENTOS DENTRO DA VARIÁVEL

RETORNA PARA A FUNÇÃO O CONTEÚDO DOS ELEMENTOS
DA VARIÁVEL

```
7 lista_retornada = obter_dados()  
8 print("A lista é:", lista_retornada)  
9 print("Nome:", lista_retornada[0])  
10 print("Idade:", lista_retornada[1])  
11 |
```

POSIÇÃO 1/PRIMEIRO ITEM DA LISTA

POSIÇÃO 2 / SEGUNDO ITEM DA LISTA

Entre com o nome:zezinho

Entre com a idade:14

A lista é: ['zezinho', '14']

Nome: zezinho

Idade: 14

Você pode usar valores para fazer uma comparação, como exemplo, a proporção entre gasolina e álcool:

```
1 def comparar_valores(a, g):  
2     proporcao = float(a)/float(g)  
3     print("O valor da proporcao é:", proporcao)  
4     if proporcao > 0.7:  
5         print("Compensa usar gasolina")  
6     else:  
7         print("Compensa usar álcool")  
8  
9     alcool = input("Entre com o valor do álcool: ")  
10    gasolina = input("Entre com o valor da gasolina:")  
11    comparar_valores(alcool, gasolina)
```

RECEBE OS PARÂMETROS DO ALCOOL E GASOLINA
TRANSFORMA OS VALORES EM FLOAT E FAZ UMA DIVISÃO
ESCOPO DE CÓDIGO DA FUNÇÃO CRIADA, É DADA UMA CONDIÇÃO
ARMAZENA O VALOR DE 'A'
ARMAZENA O VALOR DE 'B'
CHAMA A FUNÇÃO QUE RETORNA O 'PRINT'

Entre com o valor do álcool: 3.1
Entre com o valor da gasolina:4.5
O valor da proporcao é: 0.6888888888888889
Compensa usar álcool

AULA 7 – TUPLAS, LISTAS E DICIONÁRIOS

O conteúdo da Tupla não pode ser alterado.

```
1 #conteúdo da tupla não pode ser alterado  
2 tupla_marcas=("tesla", "audi", "gm", "lada", "gurgel")  
3 print(tupla_marcas)  
4 print("Marca selecionada:", tupla_marcas[0])
```

A **Tesla** foi selecionada.

As listas podem ser manipuladas com os códigos:

del [posição] →

Deleta um item pelo índice.

variável.**append** (input ("nome")) →

Adiciona um nome a uma lista.

variável.**sort** () →

Ordena os elementos de uma lista.

variável.**insert** (posição, "nome") →

Adiciona um nome a uma determina posição.

variável.**remove** ("nome") →

Exclui um item pelo o seu nome.

len (nome da variável) →

Verifica o tamanho da variável.

Exemplo abaixo:

```
#Listas
lista_candidatos=["Robert","Daenerys","Cersei"]
print(lista_candidatos)
#excluir item pelo índice
del lista_candidatos[0]
print("Lista atualizada:", lista_candidatos)
lista_candidatos.append(input("Quem vc acha que deve entrar na lista?"))
print("Lista com opinião:", lista_candidatos)
lista_candidatos.sort()
print("Lista ordenada:", lista_candidatos)
print("O tamanho da lista é:", len(lista_candidatos))
print("Os candidatos da posição 1 e 2 são:", lista_candidatos[1:3])
lista_candidatos.insert(2, "Bran")
print("Lista atualizada:", lista_candidatos)
#para excluir um item pelo nome
lista_candidatos.remove("Cersei")
print("Lista com a Cersei deletada:", lista_candidatos)
```

DELETA UM ITEM PELO ÍNDICE

ADICIONA UM CANDIDATO ESCRREVENDO SEU NOME

DEIXA A LISTA ORDENADA

VERIFICA O TAMANHO DA LISTA

ADICIONA UM NOVO CANDIDATO A POSIÇÃO 2 DA LISTA

EXCLUI UM ITEM PELO NOME

Você Pode criar um dicionário. Existe uma relação entre **chave e valor**:
É obrigatório o uso dos dois pontos entre chave e valor.

```
In [ ]: 1 casas = {"stark":"ned", "lannister":"tyron", }
```

CHAVE

VALOR

É OBRIGATÓRIO O USO DOS DOIS PONTOS

Exemplo de utilização:

```
1 casas = {"stark":"ned", "lannister":"tyron", "targaryen":"Daenerys"}
2 print("Dicionario de casas:", casas)
3 print("Representante da casa startk:", casas['stark'])
4
5 for i in casas:
6     print("Informações obtidas do loop for:", casas[i])
7
8 filhos_ned = {1:"robb", 2:"bran", 3:"rickon"}
9 print("Dicionario dos filhos do ned:", filhos_ned)
10 print("Primogênito:", filhos_ned[1])
11
```

DICIONÁRIO DE CASAS

É COLOCADO O NOME DA CHAVE, E É IMPRIMIDO O SEU VALOR

PERCORRE OS ITENS DO DICIONÁRIO DE CASAS.

DICIONÁRIO DE FILHOS DO NED, UTILIZANDO NUMERAÇÃO

IMPRIMI O PRIMEIRO ELEMENTO DO DICIONÁRIO DE FILHOS

```
Dicionario de casas: {'stark': 'ned', 'lannister': 'tyron', 'targaryen': 'Daenerys'}
Representante da casa startk: ned
Informações obtidas do loop for: ned
Informações obtidas do loop for: tyron
Informações obtidas do loop for: Daenerys
Dicionario dos filhos do ned: {1: 'robb', 2: 'bran', 3: 'rickon'}
Primogênito: robb
```


AULA 8 – LISTAS E DICIONÁRIOS 2D

Tanto lista quanto dicionário em 2D corresponde a uma tabela.

Listas 2D

Aluno	Matemática	Inglês	Português
zezinho	6	7	6
luizinho	4	5	8
huguinho	7	7	9
tio patinhas	10	10	0

```
1 #listas 2D
2 #0= matematica, 1=ingles, 2 = portugues
3 #0 = zezinho, 1=luizinho, 2 = huguinho, 3=tio patinhas
4 notas = [[6,7,6], [4,5,8], [7,7,9], [10,10,0] ]
5 print(notas)
6 for i in notas:
7     print(i)
8 #português do Luizinho
9 print(notas[1][2])
10
```

[[6, 7, 6], [4, 5, 8], [7, 7, 9], [10, 10, 0]]
[6, 7, 6]
[4, 5, 8]
[7, 7, 9]
[10, 10, 0]
8

Diagrama de acesso: notas[1][2] acessa a LINHA 1 (segunda lista) e a COLUNA 2 (terceiro elemento), resultando no valor 8.

Dicionário 2D

Tem relação com chave e valor:

```
1 #dicionario 2d
2 notas={"zezinho":{"mat":6, "ing":7, "por":6},
3        "luizinho":{"mat":4, "ing":5, "por":6},
4        "huguinho":{"mat":7, "ing":7, "por":9},
5        "patinhas":{"mat":10, "ing":10, "por":0}}
6
7 print("Dicionario de notas", notas)
8 print("Nota de portugues do huguinho:", notas["huguinho"]["por"])
9 print("Nota de matematica do patinhas:", notas["patinhas"]["mat"])
```

Dicionario de notas {'zezinho': {'mat': 6, 'ing': 7, 'por': 6}, 'luizinh
7, 'ing': 7, 'por': 9}, 'patinhas': {'mat': 10, 'ing': 10, 'por': 0}}
Nota de portugues do huguinho: 9
Nota de matematica do patinhas: 10

Diagrama de acesso: notas["huguinho"]["por"] acessa a CHAVE "huguinho" e o VALOR {"mat": 7, "ing": 7, "por": 9}, retornando o valor 9. notas["patinhas"]["mat"] acessa a CHAVE "patinhas" e o VALOR {"mat": 10, "ing": 10, "por": 0}, retornando o valor 10.

Aula 9 – Arquivos

CSV = arquivo em que os dados são separados por vírgula.

```
1 import csv
2 notas = [[6,7,6],[4,5,8],[7,7,9], [10,10,0] ]
3 print(notas)
4
5 temp = []
6 for i in notas:
7     temp.append(i)
8 print(temp)
9
```

ADICIONA CADA UMA DAS LINHAS À VARIÁVEL "TEMP" USANDO ".APPEND"

```
[[6, 7, 6], [4, 5, 8], [7, 7, 9], [10, 10, 0]]
[[6, 7, 6], [4, 5, 8], [7, 7, 9], [10, 10, 0]]
```

O conteúdo de “temp” é igual ao conteúdo de “notas”.

```
file = open("teste.csv", "w")
```

ABRE ARQUIVO EM "()"

DEFINIÇÃO DE OPERAÇÃO COMO ESCRITA

File vai ser responsável por gerenciar o arquivo “teste.csv”.

```
10 file = open("teste.csv", "w")
11 x=0
12 for i in temp:
13     newRec = str(temp[x][0])+","+str(temp[x][1])+","+str(temp[x][2]) + "\n"
14     file.write(newRec)
15     x = x+1
16 file.close()
```

O "x" É CONFORME O "i" VARIA

CONTADOR/INICIALIZADOR

COLUNA 1

COLUNA 2

COLUNA 3

GRAVA A LINHA AO ARQUIVO

INCREMENTAR PARA IR À PRÓXIMA LINHA

FUNÇÃO PARA FECHAR O ARQUIVO E CRIA-LO.

Após fechar o arquivo com a função, você abre o arquivo criado e executa. Vai abrir os valores da tabela no EXCELL, exemplo:

	A	B	C	D	E
1	6,7,6				
2	4,5,8				
3	7,7,9				
4	10,10,0				
5					
6					

ABRE O ARQUIVO CRIADO NO EXCELL, E CRIARÁ UMA TABELA COM OS VALORES SEPARADOS POR VÍRGULA.

No EXCELL você seleciona as colunas, depois vai em “**Texto para colunas**” → “**Avançar**” → na área de Delimitadores aperte em “**Vírgula**” → “**Avançar**” → “**Concluir**”.

Passo-a-Passo:

Assistente para conversão de texto em colunas - etapa 1 de 3

O assistente de texto especificou os dados como Delimitado.
Se estiver correto, escolha 'Avançar' ou escolha o tipo que melhor descreva seus dados.

Tipo de dados originais

Escolha o tipo de campo que melhor descreva seus dados:

- ☒ Delimitado - Caracteres como vírgulas ou tabulações separam cada campo.
- ☐ Largura fixa - Campos são alinhados em colunas com espaços entre cada campo.

Visualização dos dados selecionados:

1	6, 7, 6
2	4, 5, 8
3	7, 7, 9
4	10, 10, 0
5	

Assistente para conversão de texto em colunas - etapa 2 de 3

Esta tela permite que você defina os delimitadores contidos em seus dados. Você pode ver como seu texto é afetado na visualização abaixo.

Delimitadores

- ☒ Tabulação
- ☐ Ponto e vírgula
- ☒ Vírgula
- ☐ Espaço
- ☐ Outros:

Considerar delimitadores consecutivos como um só ☐

Qualificador de texto: *

Visualização dos dados

6	7	6
4	5	8
7	7	9
10	10	0

Assistente para conversão de texto em colunas - etapa 3 de 3

Esta tela permite que você selecione cada coluna e defina o formato dos dados.

Formato dos dados da coluna

- ☒ Geral
- ☐ Texto
- ☐ Data: DMA
- ☐ Não importar coluna (ignorar)

'Geral' converte valores numéricos em números, valores de data em datas e todos os valores restantes em texto.

Avançado...

Destino: \$A\$1

Visualização dos dados

Geral	Geral	Geral
6	7	6
4	5	8
7	7	9
10	10	0



Ele retira o operador de vírgula.

	A	B	C	D
1	6	7	6	
2	4	5	8	
3	7	7	9	
4	10	10	0	
5				
6				

Para fazer a leitura de um arquivo

```

17
18 file = open("teste.csv", "r")
19 for row in file:
20     print(row)
21
22

```

"READ". SERVE PARA FAZER A LEITURA DO ARQUIVO.

"CADA LINHA". ELE PERCORRE A LEITURA POR CADA LINHA DO ARQUIVO.

```

[[6, 7, 6], [4, 5, 8], [7, 7, 9], [10, 10, 0]]
[[6, 7, 6], [4, 5, 8], [7, 7, 9], [10, 10, 0]]
6,7,6
4,5,8
7,7,9
10,10,0

```

Para fazer a busca por um termo no arquivo

```
file = open("teste.csv", "r")
busca = input("Entre com o termo procurado no arquivo:")
for row in file:
    if busca in str(row):
        print(row)
```

PARA A VARIÁVEL "BUSCA", O USUÁRIO ENTRA COM O TERMO.

CADA LINHA NO ARQUIVO

CONDIÇÃO.

SE A PALAVRA DIGITADA PELO USUÁRIO ESTIVER EM ALGUMA DAS LINHAS, ELE VAI PRINTAR.

```
Entre com o termo procurado no arquivo:7
6,7,6

7,7,9
```

Se o termo digitado estiver em mais linhas, ele imprime.

OBSERVAÇÕES GERAIS

Não esquecer

Listas: sempre usa colchetes " [" .

Tuplas: sempre usa parênteses e os valores sempre serão constantes. " (" .

Dicionário: sempre usa chaves " { " e sempre tem relação com chave e valor.