**ANOTAÇÕES DE AULA PYTHON PROF GUSTAVO GUANABARA**

CTRL + SHIFT + F10 =>>> **Run**

print('-=-'\*10)

-=--=--=--=--=--=--=--=--=--=--=--=--=--=--=--=--=--=--=--=-

print('<<<>>>'\*10)

<<<>>><<<>>><<<>>><<<>>><<<>>><<<>>><<<>>><<<

print('-=<>='\*15)

-=<>=-=<>=-=<>=-=<>=-=<>=-=<>=-=<>=-=<>=-=<>=-=<>=-=<>=-=<>=-=<>=-=<>=-=<>=

nome\_completo=('Jacques Araujo dos Santos')

idade=45

peso=78

print(nome\_completo, idade, peso)

nome=input('Qual seu nome completo')

idade=input('Qual sua idade')

peso=input('Qual seu peso')

<<<>>><<<>>><<<>>> **Tipos PRIMITIVOS** <<<>>><<<>>><<<>>>

int() --> Números INTEIROS................................................ 12, 18, 23, 35

float() -> Números de ponto FLUTUANTE........................... 4.7, 2.8, 16.5, -32.6

bool() --> Armazena True ou False....................................... True, False

str() ----> Conjunto de CARACTERES..................................... 'cabeçao', 'Jacques', 'lindao'

type() ---> Indica o tipo primitivo da var - x = 'Sapo Tunado' print(type(x)) logo seu tipo primitivo é string

<<<>>><<<>>><<<>>><<<>>><<<>>><<<>>><<<>>><<<>>><<<>>>

\n =>>> quebra linha

Exemplo: print(f’Olá Jacques \ntudo bem?’)

#Run

Olá Jacques

tudo bem?

\t =>>> tab ou tabulação

Exemplo: print(f’Olá Jacques \ttudo bem?’)

#Run

Olá Jacques tudo bem?

end= ' ') =>>> Printar Python sem pular linha.

Exemplo:

numero = 0

while True:

print(numero, end= ', ') # Isto deve ficar em uma linha.

if numero == 20:

break

numero += 1

#Run

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,

<<<>>><<<>>><<<>>><<<>>><<<>>><<<>>><<<>>><<<>>><<<>>>

Espaços =>>> a.isspace()

Númerico =>>> a.isnumeric()

Alfabético =>>> a.isalpha()}

Alfanumérico =>>> a.isalnum()

Maiúsculas =>>> a.isupper()

Minúsculas =>>> a.islower()

Capitalizada =>>> a.istitle()

<<<>>><<<>>><<<>>> **Operadores ARITMÉTICOS** <<<>>><<<>>><<<>>>

+ =>>> Adição

- =>>> Subtração

\* =>>> Multiplicação

/ =>>> Divisão

\*\* =>>> Potencia

// =>>> Divisão inteira

% =>>> Resto da divisão

<<<>>><<<>>><<<>>> **Ordem de PRECEDÊNCIA** <<<>>><<<>>><<<>>>

1° =>>> ()

2° =>>> \*\*

3° =>>> \* / // %

4° =>>> + -

<<<>>><<<>>><<<>>> **MÓDULOS** <<<>>><<<>>><<<>>>

Import > Importa uma biblioteca - import math

from math import cos -> Importa somente a função cos da biblioteca math

<<<>>><<<>>> **math** =>>> **Biblioteca de operadores aritméticos** <<<>>><<<>>>

sqrt() =>>> Raiz Quadrada de um NÚMERO............ raiz = math.sqrt(numero)

floor() =>>> Arredonda o número para BAIXO........ 8,75 fica 8,00

ceil() =>>> Retorna um valor INTEIRO..................... 9,45 fica 9

hypot() =>>> Retorna a hipotenusa dos catetos...... math.hypot(co, ca)

pow() =>>> Potência de um NÚMERO..................... pow(5, 3) = 5³ = 125

radians()=>>> Converte em graus RADIANOS.......... print(math.radians(180))

cos() =>>> Retorne o COSSENO em radianos........... print(math.cos(math.radians(x)))

sin() =>>> Retorne o SENO em radianos.................. print(math.sin(math.radians(x)))

tan() =>>> Retorne a TANGENTE em radianos......... print(math.tan(math.radians(x)))

<<<>>><<<>>> **random -> Gerar numeros pseudoaleatorios** <<<>>><<<>>>

randint() =>>> Retorna um número inteiro no range................. random.randint(1, 10)

choice() =>>> Retorna um elemento aleatório da sequência..... random.choice(x)

shuffle() =>>> Embaralha a sequência x no lugar ....................... random.shuffle(y)

o legal de usar o sample ao invés do shuffle é q da pra limitar a quantidade

de pessoas q vão apresentar, atribuindo a quantidade no 'k', então se na lista

tiverem 10 alunos inscritos mas o k for igual a 5 então serão escolhidos

só 5 dentre os 10 alunos

sample([aluno1, aluno2, aluno3, aluno4], k=2)

-> serão escolhidos 2 dentre os 4 alunos

<<<>>><<<>>> **Manipulando TEXTOS** <<<>>><<<>>>

frase = 'ESTOU APRENDENDO A PROGRAMAR EM PYHTON'

Fatiamento de String:

frase[9] =>>> Pega os caracteres das posições indicadas ............................... letra E

frase[9:13] =>>> Pega os caracteres das posições indicadas ........................... ENDE

frase[9:18:2] =>>> Pega os caracteres das posições indicadas pulando 2........ EDNOA

Análise com:

len() =>>> Mostra quantas letras tem a frase............................... len(frase) = 38 letras

count() =>>> Conta quantas vezes aparece a letra escolhida ...... frase.count('s')

count('o',0,13) quantos caracteres 'o' tem dentro de 0 a 13 caracteres, lembrando sempre o último caractere é ignorado 13 e sim o 12.

find() =>>> Procura os caracteres escolhido ................................ frase.find('aprendendo')

Transformações com:

replace() =>>> Troca uma palavra por outra na frase.. frase.replace('python','JavaScript')

upper() =>>> Colocar todas as outras letras em maiúsculo....... frase.upper()

lower() =>>> Colocar todas as outras letras em minúsculo........ frase.lower()

capilalize() =>>> Coloca todas a frase em minusculo menos a 1 letra.. frase.capitalize()

title() =>>> Todas as palavras começa com letra maiúscula....... frase.title()

strip() =>>> Tira o espaço do começo e no fim da frase.............. frase.strip()

lstrip() =>>> Tira o espaço do começo e no fim da frase lado esquerdo.. frase.lstrip()

rstrip() =>>> Tira o espaço do começo e no fim da frase lado direito...... frase.rstrip()

Divisão com:

split() => Vai ocorrer uma divisão entre os espaços da frase, gera uma lista... frase.split()

Junção com:

.join() =>>> Juntar uma coisa com algum caracteres...... '-'.join.frase Estou-aprendendo-a-programar-em-python

<<<>>><<<>>><<<>>><<<>>><<<>>><<<>>><<<>>><<<>>><<<>>>

**LISTA MÉTODOS**

**LISTAS ADICIONAR**

lista\_nomes = [‘João’, ‘Maria’, ‘Guilherme’, ‘Diego’]

para adicionar um nome na lista usar se **.append**

lista\_nomes.append (‘Geralda’)

**LISTAS REMOVER**

lista\_nomes = [‘João’, ‘Maria’, ‘Guilherme’, ‘Diego’]

para remover um nome na lista usar se **.remove**

lista\_nomes.remove (‘João’)

lista\_nomes = [‘João’, ‘Maria’, ‘Guilherme’, ‘Diego’]

para remover um nome da última lista usar se **.pop**

lista\_nomes.pop (‘João’)

print (lista\_nomes)

#Run

[‘João’, ‘Maria’, ‘Guilherme’]

**LISTAS INSERIR**

lista\_nomes = [‘João’, ‘Maria’, ‘Guilherme’, ‘Diego’]

para inserir um nome na lista em um determinado local usar se **.insert**

lista\_nomes.insert (1, ‘José’)

print (lista\_nomes)

#Run

[‘João’, ‘José’, ‘Maria’, ‘Guilherme’, ‘Diego’]

**DICIONÁRIO INSERIR**

dicionario\_ordenado = {('color1', 'Red'), ('color2', 'Green'), ('color3', 'Blue')}

para inserir um elemento no dicionário em um determinado local usar se

**.update** e também **.append**

dicionario\_ordenado.update({('color4', 'Orange')})

dicionario\_ordenado.append({('color4', 'Orange')})

print(dicionario\_ordenado)

#Run

{('color2', 'Green'), ('color3', 'Blue'), ('color1', 'Red'), **('color4', 'Orange')**}

**LISTAS EM ORDEM CRESCENTE**

lista\_valores = [7, 5, 3, 9, 4, 6, 8, 0]

para colocar a lista em uma ordem crescente usar se **.sort()**

lista\_valores.sort()

*lista\_valores.sort(key=int) # Crescente*

*lista\_valores.sort(key=int, reverse=True) # Decrescente*

print (lista\_valores)

#Run

[0, 3, 4, 5, 6, 7, 8, 9]

**LISTA MENOR VALOR**

lista\_valores = [7, 5, 3, 9, 4, 6, 8, 0]

para colocar indicar a lista um menor valor usa se **min(xxx)**

menor = min(lista\_valores)

print (menor)

#Run

0

**LISTA MAIOR VALOR**

lista\_valores = [7, 5, 3, 9, 4, 6, 8, 0]

para colocar indicar a lista um menor valor usa se **max(xxx)**

maior = max(lista\_valores)

print (maior)

#Run

9

**LISTA SOMA DENTRO DA LISTA**

lista\_valores = [7, 5, 3, 9, 4, 6, 8, 0]

sum(lista\_valores)

print (sum(lista\_valores))

#Run

42

**LISTA FAZENDO UMA MÉDIA DE SOMA DENTRO DA LISTA**

lista\_valores = [7, 5, 3, 9, 4, 6, 8, 0]

sum(lista\_valores)

len(lista\_valores)

media = sum(lista\_valores) / len(lista\_valores)

print (sum(lista\_valores))

print (len(lista\_valores))

print (media)

#Run

42

8

5.25

<<<>>><<<>>> **Cálculos Aritméticos** <<<>>><<<>>>

Potência se usa \*\*

ou também comando pow (base,potência), exemplo qual a potência de 4 elevado a 3

pow(4,3)

Raiz quadrada se usa \*\*(1/2)

exemplo: 81\*\*(1/2)

Raiz cubica se usa \*\*(1/3)

exemplo: 127\*\*(1/3)

#alinhando a direita, esquerda e centralizando, em X caracteres.

# em X caracteres

nome=input('Qual seu nome ? ')

print('Prazer te conhecer, {:20}!'.format(nome))

#Run

Qual seu nome ? Jacques

Prazer te conhecer Jacques !

# alinhamento a direita

nome=input('Qual seu nome ? ')

print('Prazer te conhecer, {:>20}!'.format(nome))

#Run

Qual seu nome ? Jacques

Prazer te conhecer, Jacques!

# alinhamento a esquerda

nome=input('Qual seu nome ? ')

print('Prazer te conhecer, {:<20}!'.format(nome))

#Run

Qual seu nome ? Jacques

Prazer te conhecer, Jacques !

# alinhamento centralizado

nome=input('Qual seu nome ? ')

print('Prazer te conhecer, {:^20}!'.format(nome))

#Run

Qual seu nome ? Jacques

Prazer te conhecer, Jacques !

# alinhamento centralizado com um caracteres '='

nome=input('Qual seu nome ? ')

print('Prazer te conhecer, {:=^20}!'.format(nome))

#Run

Qual seu nome ? Jacques

Prazer te conhecer, ======Jacques=======!

Abreviações:

s para soma ( + )

m para mutiplicação ( \* )

d para divisão ( / )

di para divisão inteira ( // )

e para exponênciação ( \*\* )

#Prática

n1=int(input('Um valor: '))

n2=int(input('Outro valor: '))

s = n1 + n2

m = n1 \* n2

d = n1 / n2

di = n1 // n2

e = n1 \*\* n2

print('A soma é {}, a multiplicação é {}, a divisão é {}'.format(s, m, d))

print('A divisão inteira é {}, e a exponênciação é {}'.format(di, e))

#Run

Um valor: 4

Outro valor: 3

A soma é 7, a multiplicação é 12, a divisão é 1.3333333333333333

A divisão inteira é 1, e a exponênciação é 64

#Para diminuir a dizima nesta divisão ou seja formatando as casas decimas.

{:.3f} nesta opção 3 casas após o ponto

print('A soma é {}, a multiplicação é {}, a divisão é {:.3f}'.format(s, m, d))

#Run

A soma é 7, a multiplicação é 12, a divisão é 1.333

{:.0f} nesta opção nenhuma casa após o ponto

print('A soma é {}, a multiplicação é {}, a divisão é {:.0f}'.format(s, m, d))

#Run

A soma é 7, a multiplicação é 12, a divisão é 1

# utilizando o comando end= ' ' você deixa os carateres ou frase na mesma linha.

print('A soma é {}, a multiplicação é {}, a divisão é {:.3f}'.format(s, m, d), end=' ')

print('A divisão inteira é {}, e a exponênciação é {}'.format(di, e))

#Run

A soma é 7, a multiplicação é 12, a divisão é 1.333 A divisão inteira é 1, e a exponênciação é 64

# utilizando o comando \n você quebra a linha

print('A soma é {}, \n a multiplicação é {}, \n a divisão é {:.3f}'.format(s, m, d))

print('A divisão inteira é {}, \n e a exponênciação é {}'.format(di, e))

#Run

A soma é 7,

a multiplicação é 12,

a divisão é 1.333

A divisão inteira é 1,

e a exponênciação é 64

<<<>>><<<>>> **CORES NO TERMINAL** <<<>>><<<>>>

Nessa aula, vamos aprender como utilizar os códigos de escape sequence ANSI

para configurar cores para os seus programas em Python.

Veja como utilizar o código \033[m com todas as suas principais possibilidades.

Códigos para STYLE:

0 => sem style algum

1 => colocar o texto em negrito

4 => colocar o texto em sublinhado

7 => colocar o texto em fundo negativo

TEXT

30 => texto em preto

31 => texto em vermelho

32 => texto em verde

33 => texto em amarelo

34 => texto em azul

35 => texto em margenta

36 => texto em ciano

37 => texto em cinza

97 => texto em branco

BLACKGROUND

40 => fundo em preto

41 => fundo em vermelho

42 => fundo em verde

43 => fundo em amarelo

44 => fundo em azul

45 => fundo em margenta

46 => fundo em ciano

47 => fundo em cinza

107 => fundo em branco

STYLE BLACKGROUND

\033[ 0 : 33 : 44 m

TEXT

Exemplos:

\033[0:30:41m => sem formatação, texto branco e fundo vermelo

\033[4:33:44m => sublinado, texto amarelo e fundo azul

\033[1:35:43m => negrito, texto margenta e fundo amarelo

\033[30:42m => sem stylo, texto branco e fundo verde

\033[m => configuração padrão do terminal

\033[7:30m => sem formatação, texto negativo

NA PRÁTICA

cores = {'vermelho':'\033[1;31m','azul':'\033[1;34m','branco':'\033[0;30m','limpar':'\033[m','verde':'\033[32m'}

print("\033[1:29:45mOlá, Mundo!\033[m")

a = 3

b = 8

print(f'Os valores são \033[1:32m{a}\033[m e \033[1:31m{b} !!!')

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**#TEMPO**

from time import sleep

print()

print('-=<>='\*15)

print('\033[33mPROCESSANDO ........\033[m')

print()

sleep(3)

print('-=<>='\*15)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**#ANO ATUAL**

from datetime import date

atual = date.today().year

print(atual)

#Run

2022

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**#EMOJI**

import emoji

print(emoji.emojize('Olá, Mundo \U0001f985', use\_aliases=True))

Olá, Mundo 🦅

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**#TOCAR MÚSICA**

import pygame

pygame.mixer.init()

pygame.init()

pygame.mixer.music.load('acertouamizeravi.mp3rox.mp3')

pygame.mixer.music.play()

input()

pygame.event.wait()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**#TOCAR MÚSICA**

playsound.playsound('acertou.mp3')

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**COMPOSIÇÃO DE STRINGS**

%d Tipo: Números inteiros

%s Tipo: Strings

%f Tipo: Números decimais

**OPERADORES RELACIONAIS**

== Igual a “comparando”

> Maior que

>= Maior igual que

< Menor

<= Menor igual que

!= Diferente de

**OPERADORES LÓGICOS**

and operador “E”

or operador “OU”

not operador “NEGAÇÃO”

**CONVERTER NÚMERO INTERIO PARA BINÁRIO**

Utilize bin() Função para Converter Int para Binário em Python.

Em Python, pode utilizar uma função integrada, bin() para converter um número inteiro em binário.

**CONVERTER NÚMERO INTERIO PARA OCTAL**

oct()

**CONVERTER NÚMERO INTERIO PARA HEXADECIMAL**

hex()

**TIPOS DE CONJUNTO**

**len(s)**

Retorna o número de elementos no set s (cardinalidade de s).

**x in s**

Testa se x pertence a s.

**x not in s**

Testa se x não pertence a s.

**isdisjoint(other)**

Retorna True se o conjunto não tem elementos em comum com other.

Conjuntos são disjuntos se e somente se a sua interseção é o conjunto vazio.

**issubset(other)**

Testa se cada elemento do conjunto está contido em other.

**set < other**

Testa se o conjunto é um subconjunto próprio de other, isto é,

set <= other and set != other.

**issuperset(other)**

Testa se cada elemento em other está contido no conjunto.

**set > other**

Testa se o conjunto é um superconjunto próprio de other,

isto é, set >= other and set != other.

**union(\*others)**

Retorna um novo conjunto com elementos do conjunto e de todos os outros.

**intersection(\*others)**

Retorna um novo conjunto com elementos comuns do conjunto e de todos os outros.

**difference(\*others)**

Retorna um novo conjunto com elementos no conjunto que não estão nos outros.

**symmetric\_difference(other)**

Retorna um novo conjunto com elementos estejam ou no conjunto ou em other, mas não em ambos.

**copy()**

Retorna uma cópia rasa do conjunto.

*Observe que, as versões não-operador dos métodos union(), intersection(), difference(), symmetric\_difference(), issubset(), e issuperset() irão aceitar qualquer iterável como um argumento. Em contraste, suas contrapartes baseadas em operadores exigem que seus argumentos sejam conjuntos. Isso impede construções suscetíveis a erros como set('abc') & 'cbs' e favorece a forma mais legível set('abc').intersection('cbs').*

*Tanto set quanto frozenset suportam comparar um conjunto com outro. Dois conjuntos são iguais se, e somente se, cada elemento de cada conjunto está contido no outro conjunto (cada um é um subconjunto do outro). Um conjunto é menor que outro se, e somente se, o primeiro conjunto é um subconjunto adequado do segundo (é um subconjunto, mas não é igual). Um conjunto é maior que outro conjunto se, e somente se, o primeiro conjunto é um superconjunto próprio do segundo conjunto (é um superconjunto, mas não é igual).*

Instâncias de set são comparadas a instâncias de frozenset baseados nos seus membros. Por exemplo, set('abc') == frozenset('abc') retorna True e assim como set('abc') in set([frozenset('abc')]).

O subconjunto e comparações de igualdade não generalizam para a função de ordenamento total. Por exemplo, quaisquer dois conjuntos deslocados não vazios, não são iguais e não são subconjuntos um do outro, então todos os seguintes retornam False: a<b, a==b ou a>b.

Como conjuntos apenas definem ordenamento parcial (subconjunto de relacionamentos), a saída do método list.sort() é indefinida para listas e conjuntos.

Elementos de conjuntos, assim como chaves de dicionário, devem ser hasheáveis.

Operações binárias que misturam instâncias de set com frozenset retornam o tipo do primeiro operando. Por exemplo: frozenset('ab') | set('bc') retorna uma instância de frozenset.

A seguinte tabela lista operações disponíveis para set que não se aplicam para instâncias imutáveis de frozenset:

**update(\*others)**

Atualiza o conjunto, adicionando elementos dos outros.

**intersection\_update(\*others)**

Atualiza o conjunto, mantendo somente elementos encontrados nele e em outros.

**difference\_update(\*others)**

Atualiza o conjunto, removendo elementos encontrados em outros.

**symmetric\_difference\_update(other)**

Atualiza o conjunto, mantendo somente elementos encontrados em qualquer conjunto, mas não em ambos.

**add(elem)**

Adiciona o elemento elem ao conjunto.

**remove(elem)**

Remove o elemento elem do conjunto. Levanta KeyError se elem não estiver contido no conjunto.

**discard(elem)**

Remove o elemento elem do conjunto se ele estiver presente.

**pop()**

Remove e retorna um elemento arbitrário do conjunto. Levanta KeyError se o conjunto estiver vazio.

**clear()**

Remove todos os elementos do conjunto.