**Conteúdo da aula 009 - Python 3**

-Usa-se o print'...' para fazer o programa escrever algo

-Usa-se ... = X para associar algo a algo

-Usa-se # antes de uma linha de comandos para fazer o python entender que aquilo é somente uma observação e não executá-los

-Usa-se ...=input('...') para fazer uma pergunta e associar a resposta ao ...

-Usa-se + para somar duas coisas e , para juntar (O + também pode juntar se houver aspas)

\* print('...' , algo) junta uma frase pronta com algo variável já programado

\* ... =print 'Exemplo' faz com que a frase dita após uma pergunta escreva algo

-Operações: + mais , - menos , \* vezes , / dividido , \*\* potência , // divisão inteira , \*\*(1/2) raiz quadrada.

\*Para somar dois valores você tem que colocar int (de inteiro) antes dos input dos números

\* O símbolo % não representa a operação porcentagem e sim o módulo de algo (O resto) Ordem de precedência: 1°() ; 2°\* ; 3°\*,/,//,% ; 4° + e -

Na biblioteca math:

\*A função sqrt da biblioteca math serve para tirar a raiz quadrada

\*A função math.cos mostra o cosseno, math.sin o seno e math.tan a tangente

\*A função radians transforma o número em radianos

-Você pode fazer algo str se repetir multiplicando ele por um número (Ex: 'Oi' \* 5)

-Pode-se escrever em alguma quantidade de caracteres usando {:.20} com as {} significando algo no forma de .format{} \* < > ^ servem para informar onde a palavra ficará

-Pode-se definir a quantidade de casas depois da vírgula com {:.2f} por exemplo

-Usa-se /n para quebrar as linhas e end=' ' para não quebrar a linha (juntar com a linha de baixo)

-Tipos primitivos: int() números inteiros, float() números quebrados(reais), bool() true ou false, str() string

-Para importar alguma biblioteca usa-se import ... já para importar algo específico da biblioteca usa-se from ... import ...

\*Quando se importa com o from não precisa colocar o nome da biblioteca

-Uma lista é criada com [ ]

-Na biblioteca random:

\*random.shuffle(...) é colocado só em alguma linha e embaralha o ...

\*random.choice(...) escolhe algum da lista já feita anteriormente com [ ]

-Para executar uma música podemos utilizar o programa mixer da biblioteca pygame. Utilizamos mixer.init() para iniciar o programa, mixer.load.music('Nome da música com .mp3') e mixer.play().

\*É necessário manter o programa funcionando para a música tocar.Ex: executar um input('Qual é a música?'), por exemplo.

-Manipulando textos:

Os textos ocupam um espaço de micro-memoria no PC sendo contados do 0, uma frase normal está na mesma ''família'' de micro-memórias

\*Para fatiar uma frase e escolher uma letra se usa frase[3] por exemplo.

\*Para mostrar a frase do início ao 3 caractere por exemplo, usa-se print(frase[:3]), lembrando que o primeiro caractere sempre é o 0

\*Para mostrar do 3 caractere ao final usa-se print(frase[3:])

\*O comando [::2] por exemplo, vai mostrar do início ao final pulando de 2 em 2

\*O comando [3::2] por exemplo, serve para o python escrever do 3 ao final pulando de 2 em 2, mostrando o 2 (Tipo: ABCDEFGHI ficaria DFH)

\*O comando len(frase) serve para indicar a quantidade de caracteres na frase já atribuída por um input, por exemplo

\*O comando frase.count('o') por exemplo, serve para o python contar quantas vezes o ''o'' apareceu na frase

\*No fatiamento o último valor sempre é ignorado pelo python.

\*O comando frase.find('deo') serve para indicar onde começou o “deo” na frase ''curso em vídeo'' por exemplo, sendo igual a 11

\*Quando mandamos o python procurar algo que não existe na frase ele te dá como resultado -1, indicando que aquilo não está na frase. Exemplo: print(frase.find('cagão') da frase cheira peido não existe, logo, ele vai ter como resultado -1

\*O operador in serve para indicar se existe aquela parte já associada na frase também já associada. Ex: print(parte in frase)

\*O comando frase.replace('python , 'android') por exemplo, serve para substituir o python da frase já associada por android. Ex: print(frase.replace('python, 'android') da frase já associada: curso em vídeo python

\*O comando frase.upper() serve para deixar todas as letras da frase, já associada, mas, letras já em maiúsculas continuam em maiúsculas

\*O comando frase.lower() deixa as letras maiúsculas em minúsculas, minúsculas permanecem em minúsculas

\*A função frase.capitalize() serve para deixar todos os caracteres da frase já associada para minúsculo e deixar somente a primeira letra em maiúsculo

\*O comando frase.title() serve para deixar as letras após espaços em maiúsculas

\*A função frase.strip() remove todos os espaços inúteis, os entre palavras e letras continuam. \*A função frase.rstrip() retira os espaços da direita, final do texto

\*A função frase.lstrip() retira os espaços da esquerda, início do texto

\*A função frase.split() serve para separar as palavras em diferentes famílias de micro-memórias

\*A função '-'.join(frase) por exemplo, serve para juntar famílias de micro-memórias com o que está entre as aspas

\*A função print com """ serve para o python escrever do mesmo modo que foi escrito, com quebras de linhas, bom para textos grandes.

\*Quando usamos o comando frase.split() associado a dividido por exemplo, em uma frase já associada, o comando print(dividido[0]) mostrará a primeira família de micro-memórias, 1 a segunda, 2 a terceira e assim em diante