**Programa**: Uma sequência de instruções que especifica como executar um cálculo.

**Programação**: Pegar uma tarefa e escreve-la em uma linguagem de programação que o computador possa compreender e executar.

(A atividade de dizer ao computador para fazer algo por nós.)

**Strings**: Valores de texto compostos por uma sequência de caracteres.

**Operadores Aritméticos**

**Adição (+)**

**Subtração (-)**

**Divisão (/)**

**Multiplicação (\*)**

**Porcentagem (%)**

**Elevado (Ao quadrado) (\*\*)**

**Duplo Sinal de Igualdade (==):** tal coisa é igual a tal coisa

**Reatribuindo Valores:** O último valor atribuído é o que valera.

**Continuação de Linha:** Quando tiver necessidade de quebra linha, pode-se usar a contrabarra (\).

**Indexação de Elementos:** A contagem começa do 0.

**Operadores Lógicos**

**Operadores Booleanos: NOT, AND, OR**

True AND True = True

True AND False = False

False AND False = False

False OR False = False

True OR False = True

NOT True = False

NOT False = True

4>6 AND 10<=15 = False

**Operadores de Identidade: IS, IS NOT**

10 IS 8 = False

5 == 6 = False

6 IS NOT 8 = True

6 != 8 = True

**Declarações Condicionais**

**Comando IF**

IF = SE (CONDIÇÃO)

Para colocar a condição. Sempre tem que colocar a condição.

**Comando ELSE**

**Comando ELIF**

O computador sempre lê seus comandos de cima para baixo! (ELIF é uma abreviação de (ELSE IF)

O comando ELIF é usado quando temos mais de duas alternativas

def compare\_com\_oito(y):

    if y > 8:

        return 'Maior'

    elif y < 8:

        'Menor'

    elif y < 0:

        return 'Negativo'

    else:

        return 'Igual'

**COMO DEFINIR UMA FUNÇÃO**

**DEF – É uma palavra reservada**

**Definindo uma função**

def aprendi():

print(“minha primeira função”)

print(aprendir())

**Criando uma função com um Parâmetro**

def mais\_nove(a):

return (a + 9)

**Criando funções contendo alguns argumentos**

Def conta(a,b,c):

Resultado = a – b \* c

Print (‘paramentro a é igual’, a)

Print (‘paramentro a é igual’, b)

Print (‘paramentro a é igual’, c)

Return resultado

**Funções Nativas Importantes no Python**

**Type**

**Str** – String

**Max** – Ela vai pegar e mostrar o maior valor do que estiver nas funções

Max(8,18,4)

18

**Min** - Ela vai pegar e mostrar o menor valor do que estiver nas funções

**Abs** – número absoluto

Z = -20

Abs(z)

**SUM** – Soma todos os números da lista

Lista = (1,2,3,4)

Sum(lista)

10

**ROUND –** Arredondar número quebrado (Float)

Round(3.5564, 2) Quantas casas decimais eu quero

**POW - potência**

Pow(2,10)

1024

**LEN –** Quantas letras tem uma palavra

Len(‘cachorro’)

**USANDO UMA FUNÇÃO EM OUTRA FUNÇÃO**

Def salario(hrs\_trab):

Return hrs\_trab \* 40

Def com\_bonus(hrs\_trab):

Return Def salario(hrs\_trab) + 50

Salario(8), com\_bonus(8)

(320, 370)

**Listas no Python**

Uma lista é um tipo de Dados como: números inteiros, frações ou texto

Quando for contar de frente para trás, colocasse o sinal de negativo (-) e contasse a partir de -1.

Trocar nome na lista

Nome da lista[posição que quer trocar] = ‘nome que quer trocar’

Deletar nome na lista

Del Nome da lista[posição que quer trocar

**Métodos Nativos do Python**

Nome da lista.append(‘add alguma coisa’) – Adicionar alguma coisa na lista

convidados.extend(['Vitor', 'Lopes']) – Adiciona mais de uma coisa

**Slicing de Listas**

É fatiamento, no nosso caso fatiamento da nossa lista.

print(convidados[1:3])

Quando for selecionar o 0 não coloca ele

print(convidados[:2])

Selecionar ao contrário, sem a posição 0

print(convidados[1:])

print(convidados[-3:])

Saber a posição do item

print(convidados.index('Lopes'))

Unir duas listas

vida = [convidados, novos]

Ordem alfabética

novos.sort()

Trocar ordem das listas

vida.sort(reverse=True)

print(vida)

**Tupla** – é uma função automática do python, Ex: x = (30,50,14) o que está dentro do parêntese é uma tupla

OBS: Tuplas não podem ser alteradas ou modificadas (não dá para acrescentar ou excluir elementos)

Tupla vem com parêntese

Lista vem com colchetes

.split(‘) – serve para separar

**Dicionários** representam outra maneira de armazenar dados

Dicionário = {} com chaves

dicionario = {'c1': 'Caio', 'v1': 'Vitor', 'd1': 'Deus','j1' : 'Jesus'}

print(dicionario['d1'])

Adicionando algo no dicionário

dicionario['c1'] = 'Cristo'

Substituindo

dicionario['c2'] = 'caio'

.get – procura algo

print(time.get('atacante'))

Usa-se parêntese