

EAD - Algoritmos e Lógica de Programação

Itens de reforço para prova

- variaveis basicas
- estruturas condicionais
- estruturas de repetição
- input utilizando dicionarios
- material extra sobre funções

#Variaveis são espaços na memoria volátil que armazenam determinado dado

```
numero = 1
letras = 'abc'
fracao = 1.2
```

```
print(type(numero))
print(type(letras))
print(type(fracao))
```



```
numero = 'b'
```

```
print(numero)
```



```
#Alem dos tipos básicos de variaveis temos as listas, que armazenam vários dados em um
lista = [1,2,3,4]
```

```
print(lista)
```



```
lista[0]
```



```
lista[1]
```



```
lista[-1]
```



```
lista[:3]
```



```
lista=[1,2,3]
```

```
#copiando dados de uma variavel para outra  
list1=lista
```

```
list1
```



```
lista
```



```
#montando uma lista  
lista=list(range(0,30,1))  
lista
```



```
#montando listas incrementais  
impares=list(range(1,20,2))  
print(impares)
```



```
l1 = [5,6,7,[8,9,10]]
```

```
lista1=[1,2,3]
```

```
lista1
```



```
len(lista1)
```



```
len(l1[3])
```



```
dicionario = {"melhor amigo": "Jailson", "inimigo": "estevão"}  
print(dicionario)  
dicionario["melhor amigo"]
```



```
dicionario["inimigo"]
```



```
clientes={}
```

```
clientes
```



```
clientes["8887777888"]=["carinha que mora logo ali", "mora logo ali"]
```

```
clientes
```



```
clientes['99998888777']
```



A variavel clientes recebe o conjunto de clientes cadastrados. Ela é modificada toda vez que o código é executado

```
#Criando um dicionario com os dados dos clientes
clientes = { "987899":['Seu carlos',"1787887","Rua sem beco, numero 4"]}
print(clientes)
```



```
#Operações matemáticas com python
#Soma
n1=10
n2=5
print(n1+n2)
print(n1-n2)
print(n1/n2)
print(n2**2)
print(n1*n2)
```



```
#Mostrando valores na tela
#Utiliza a função print para mostrar os valores de variaveis ou apresentar mensagens r
print("Vem vindo ao curso de Lógica")
```



```
#Lendo valores do teclado e atribuindo a variavel
var1 = input("Informe o seu nome:")
print("Bem vindo",var1)
```



```
texto = "Quantos caracteres possui nesta frase?"
len(texto)
```



```
#Posso pegar parte da string, utilizando operações de split
texto[0]
```



```
#Posso pegar um conjunto de caracteres  
texto[0:7]
```

```
#Funciona tambem de tras pra frente  
texto[-1:]
```



```
#As operações mostradas para fatiar a string tambem valem para listas ok  
listas=[1,2,3,4]  
listas[-1]
```



```
#Separando as palavras de uma frase  
texto.split()
```



```
#Deixando todo o texto em maiusculo  
texto.upper()
```



```
#Tudo em minusculo  
texto.lower()
```



▼ Trabalhando com operadores Lógicos e expressões relacionais

```
#Verificando se uma variavel é igual a outra  
n1=2  
n2=4  
n1 > n2
```



```
#Verificando se n1 esta em lista  
lista = [1,2,3,4]  
31 in lista
```



```
#Operador And  
(n1 and 2) > 0
```



```
#Operador Or  
(n1 or -1) > 0
```



▼ Estruturas Condicionais

```
#Estruturas condicionais  
a=11  
if(a<10):  
    print("Menor que 10")  
else:  
    print("Maior que 10")
```



```
9%2
```



```
b=4  
if (b%2==0):  
    print("numero par")  
    print("eh mesmo par")  
else:  
    print("impar")
```



```
var1 = eval(input("digite o primeiro valor:"))  
var2 = eval (input("digite o segundo valor:"))  
if(var2):
```

```
divisao = var1/var2
print("o valor da divisao eh:",divisao)
else:
    print("Numeros não podem ser divididos por zero")
```



```
numero = eval(input("digite um numero:"))
if (numero%2 !=0):
    print("numero impar")
else:
    print("numero par")
```



▼ Estruturas de repetição

```
l1 = list(range(1,100))
```

```
l1[0]+1
```



```
l1[2]+1
```



```
l1
```



1

Estrutura de repetição permitem executar um comando até que determinada condição seja atendida

```
i=0  
lista = [1,2,3,4,5,6,7,8]
```

```
for i in lista:  
    print(i)
```



```
# o for permite que seja analisado item a item de uma lista  
#Script para encontra o maior elemento da lista  
maior=0  
for i in lista:  
    print("O maior valor ate agora eh:",maior)  
    if i > maior:  
        maior = i  
  
print(maior)
```



```
#Script para verificar o menor elemento da lista  
menor=1000  
for i in lista:  
    if i < menor:  
        menor = i  
print(menor)
```



Estruturas de repetição permitem iterar entre os diversos itens de uma lista, aplicando a estes operações de soma, subtração, divisão ou qualquer outra ação que o python permitir como remoção de dados

```
#Para iterar entre os objetos pode ser utilizado os comandos while e for
```

```
a = 0
while a<10:
    print(a)
    a=a+1
```



```
#soma dos numeros de 1 a 10
```

```
a=10
soma=0
while a>1:
    soma=soma+a
    a=a-1
print(soma)
```



```
#Muito cuidado ao utilizar o while, pois é necessário gerar uma condição de parada
```

```
#soma dos numeros de 1 a 10
```

```
a=10
soma=0
while a>1:
    soma=soma+a
    a=a-1
print(soma)
```

```
soma=0
impares = [1,3,5,9,11]
for i in impares:
    soma=soma+i
print(soma)
```



```
#tem um jeito fácil  
soma=sum(impares)  
print(soma)
```



Vamos trabalhar agora? Resolva a seguinte lista de exercicios utilizando python

- Crie um programa que realiza a soma dos 10 primeiros numeros inteiros
- Crie um programa que realiza a soma dos 20 primeiros numeros pares

[] ↪ 2 cells hidden

Funções

As funções permitem o reaproveitamento de código, evitando a necessidade de repetir linhas toda vez que é necessário executar determinados comandos

```
var1 = input("Informe um valor:")  
print("O valor informado é:",var1)  
var2 = input("Informe um valor:")  
print("O valor informado é:",var2)  
var3 = input("Informe um valor:")  
print("O valor informado é:",var3)
```



```
#Refazendo a tarefa acima com função  
def leitura():  
    var1 = input("Informe um valor:")  
    print("O valor informado é:",var1)
```

```
leitura()  
leitura()
```



```
#Funções podem receber variaveis e executar determinadas ações
a = eval(input("Informe um numero inteiro:"))
b = eval(input("Informe outro numero inteiro:"))
def soma(a,b):
    soma = a+b
    print("A soma dos dois numeros eh:",soma)
soma(a,b)
```



```
soma('a','p')
```



```
'eu estou '+'aqui'
```



```
#Faça uma função que recebe dois numeros, faz a divisão e apresenta o resultado
a = eval(input("Informe um numero inteiro:"))
b = eval(input("Informe outro numero inteiro:"))
def divisao(a,b):
    resultado = a/b
    print("O resultado da divisão dos dois numeros eh:",resultado)
divisao(a,b)
```



```
#Uma função pode retornar um determinado valor
#Funções podem receber variaveis e executar determinadas ações
a = eval(input("Informe um numero inteiro:"))
b = eval(input("Informe outro numero inteiro:"))
def soma(a,b):
    soma = a+b
    return soma
soma(a,b)
```



```
conta = soma(1,8)
```

```
print(conta)
```



```
#recebendo o resultado de uma função  
resultado = soma(1,2)  
print("O valor da soma eh:",resultado)
```



▼ Dever de casa

- Gere uma lista com os numeros de um a 30 e faça a soma apenas dos numeros impares. Utilize o if dentro do for para verificar se o numero é par ou impar
- Desenvolva uma função que recebe uma lista de numeros e faz a soma dos mesmos.
- Crie uma função que recebe dois valores e faz a divisão, verificando se o divisor é diferente de zero

```
nota_1= float (input ("digite nota_1:"))
```

```
nota_2= float (input ("digite nota_2:"))
```

```
média= ((nota_1)+(nota_2))/2
```

```
print ("a média é:", média)
```



