

Nome: _____ Data: ____/____/____

Revisão Geral de Condicional, Laço de Repetição, Vetor e Função

Nesta etapa vamos apenas lembrar a sintaxe para condicionais, laço de repetição e manipulação de vetores utilizando a linguagem Python. Mas o objetivo principal será trabalhar com vetores e os diferentes algoritmos de ordenação.

Para trabalharmos com condicional, utilizaremos o famoso 'if' existente em praticamente todas as linguagens. O ponto importante a ser considerado é o cuidado com a endentação dos blocos de códigos para atingirmos o resultado esperado em nossos códigos. Abaixo seguem o 2 exemplos de condicionais.

Exemplo 01

```
num = int(input("Digite um número inteiro qualquer: "))
if (num % 2 == 0) :
    print("O número %d é par!" %num)
else :
    print("O número %d é ímpar!" %num)
```

Exemplo 02

```
anoNasc = int(input("Digite o ano de nascimento de uma pessoa: "))
anoAtual = int(input("Digite o ano atual: "))
idade = anoAtual - anoNasc
if (idade <= 12) :
    print("Idade = %d ano(s). É uma criança!" %idade)
elif (idade <= 17) :
    print("Idade = %d ano(s). É um adolescente!" %idade)
elif (idade <= 64) :
    print("Idade = %d ano(s). É um adulto!" %idade)
else :
    print("Idade = %d ano(s). É um idoso!" %idade)
```

Agora iremos lembrar um pouco a utilização de laço de repetição. Segue abaixo exemplos com "while" e "for".

Exemplo 03 – for

```
#Lista pré-definida
for cont in (1,2,3,4,5) :
    print(cont)
```

Exemplo 04 – for

```
#Lista com intervalo
for cont in range(5, 10, 1) :
    print(cont)
```

Nome: _____ Data: ____/____/____

Exemplo 05 – while

```
print("Programa para ler e imprimir 5 números \n")
cont = 1
while (cont <= 5) :
    num = int(input("Digite o %d número: " %cont))
    print("O %d número digitado foi %d " %(cont, num))
    cont = cont + 1
print("Fim do Programa")
```

Em qualquer linguagem quando necessitamos armazenar várias informações utilizamos vetores ou arrays para isso. Só como reforço, segue 2 exemplos de utilização de vetores em Python.

Exemplo 06 - Vetor

```
#definido o tamanho do vetor
lista = list(range(5))
#laço para preenchimento dos valores no vetor
for i in range(5):
    lista[i] = i * 3
#laço para impressão dos valores
for i in range(5):
    print("O valor da posição %d do vetor é: %d" %(i, lista[i]))
```

Exemplo 07 – Vetor

```
vetor = []
for i in range(10):
    vetor.append(i)
for val in vetor:
    print(val)
```

Exemplo 08 – Vetor

```
lista = ['foo', 'bar', 'baz']

# removendo elementos do vetor através do índice
print(lista)
del lista[0]
print(lista)

# removendo elementos do vetor através do valor
lista2 = ['aaa', 'bbb', 'ccc', 'ddd']
print(lista2)
lista2.remove('ccc');
print(lista2)
```

Nome: _____ Data: ____/____/____

Mais Exemplos de Manipulação com Vetor

<https://pythonhelp.wordpress.com/2013/06/26/brincando-com-listas/>

Para finalizarmos está rápida revisão sobre a programação estruturada com a linguagem Python, podemos também modularizar nossos programas em Python tendo como objetivo facilitar a manutenção, a leitura e o reaproveitamento de código. Ao dividir um grande programa em pequenos programas ou módulos, estes podem retornar valores ou não. Em caso de retorno de valor denominamos de Função e quando não se retorna valor denominamos de Procedimento. Abaixo um pequeno exemplo de declaração de procedimento e função, além da chamada dos mesmos.

Exemplo 09 – Procedimento e Função

```
# Procedimento = não retorna valor
def imprimirMens(mens):
    print(mens)

# Função = retorna valor
def calcularSoma(n1, n2):
    soma = n1 + n2
    return soma

imprimirMens("Programa - Testando Procedimentos e Funções")
x = calcularSoma(5, 10)
imprimirMens("Soma = %d" %x)
```

Atividade 01

Criar um programa em Python que manipule um vetor com N números inteiros. Os números deverão ser digitados pelo usuário, podendo este escolher: adicionar um novo número no vetor, excluir um número ou imprimir a lista de números digitados. O programa deverá ter o seguinte menu:

Menu

0 - Sair
1 - Inserir um número na Lista
2 - Excluir um número da Lista
3 - Imprimir a Lista

Atividade 02

Pesquisar e Implementar o seguinte método simples de ordenação: Bubble Sort. Adicionar ao programa da atividade anterior com a opção = 4.