ALGORITMOS E LÓGICA DE PROGRAMAÇÃO AULA 03



Estrutura de Decisão e Repetição

Como vimos anteriormente em "Operações Lógicas", verificamos que na maioria das vezes necessitamos tomar decisões no andamento do algoritmo. Essas decisões interferem diretamente no andamento do programa. Trabalharemos com dois tipos de estrutura. A estrutura de Decisão e a estrutura de Repetição.

Comandos de Decisão

Os comando de decisão ou desvio fazem parte das técnicas de programação que conduzem a estruturas que não são totalmente seqüenciais. Com as instruções de **SALTO** ou **DESVIO** pode-se fazer com que o programa proceda de uma ou outra maneira, de acordo com as decisões lógicas tomadas em função dos dados ou resultados anteriores. As principais estruturas de decisão são: "**Se Então**", "**Se então Senão**" e "**Caso Selecione**".

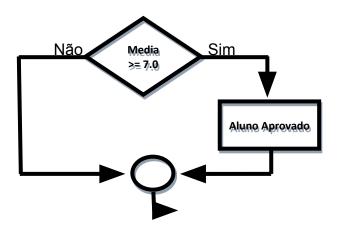
SE ENTÃO / IF ... THEN

A estrutura de decisão "SE/IF" normalmente vem acompanhada de um comando, ou seja, se determinada condição for satisfeita pelo comando SE/IF então execute determinado comando.

Imagine um algoritmo que determinado aluno somente estará Aprovado se sua média for maior ou igual a 7.0, veja no exemplo de algoritmo como ficaria.

SE MEDIA >= 7.0 ENTÃO ALUNO APROVADO

Em diagrama de blocos ficaria assim:



SE ENTÃO SENÃO / IF ... THEN ... ELSE

A estrutura de decisão "SE/ENTÃO/SENÃO", funciona exatamente como a estrutura "SE", com apenas uma diferença, então "SE" somente podemos executar comandos caso a condição seja verdadeira, diferente de "SE/SENÃO" pois sempre um comando será executado independente da condição, ou seja, caso a condição seja "Verdadeira" o comando da condição será executado, caso contrario o comando da condição "Falsa" será executado.

Em algoritmo ficaria assim:

SE MEDIA >= 7.0 ENTÃO
ALUNO APROVADO
SENÃO
ALUNO REPROVADO

```
exercicio01.py ×
```

```
1 a = 10
```

```
if a > b:
```

```
print("A é <u>maior</u> que B")
```

```
media = 8
if media >= 7:
    print("Aluno aprovado")
else:
    print("Aluno reprovado!")
```

Programação (PYTHON)

Receba 2 números do usuário e mostre eles em ordem crescente

```
"""Exercicio 01 da aula 03
iremos aprender sobre decisão IF """
#código para verificar qual o maior número e ordená-los
print("vamos Ordenar os números? \n")
a= int(input("Digite o Primeiro número: "))
b= int(input("Digite o segundo número: "))
if a>b:
  print("esta é a ordem crescente dos números", b , a)
else:
  print("esta é a ordem crescente dos números", a, b)
```

Faça um código que receba as 3 notas de um aluno e verifique se ele está aprovado ou reprovado.

Considere que a média para aprovação é 7,0

```
nota1=float(input("digite a primeira nota "))
nota2=float(input("digite a segunda nota "))
nota3=float(input("digite a terceira nota "))
media = (nota1 + nota2 + nota3)/3
if media>=7:
  print("Parabens! aluno aprovado, sua media foi: ", media)
else:
  print("OPS! aluno reprovado, sua media foi: ", media)
print("Calculo finalizado!")
```

Desafio: Altere o código anterior e acrescente a opção de aluno em recuperação, caso sua média seja menor que 7,0 e maior que 4,00

```
nota1=float(input("digite a primeira nota "))
nota2=float(input("digite a segunda nota "))
nota3=float(input("digite a terceira nota "))
media = (nota1 + nota2 + nota3)/3
if media>=7:
  print("Parabens! aluno aprovado, sua media foi: ", media)
elif media>=4:
  print("UFA! Foi por pouco, aluno em recuperação, sua media foi: ",
  media)
else:
  print("OPS! aluno reprovado, sua media foi: ", media)
print("Calculo finalizado!")
```

Ler o nome de 2 times e o número de gols marcados na partida (para cada time). Escrever o nome do vencedor. Caso não haja vencedor deverá ser impressa a palavra EMPATE.

Escreva um algoritmo que leia o número de litros vendidos e o tipo de combustível (codificado da seguinte forma: E-etanol, G-gasolina), calcule e imprima o valor a ser pago pelo cliente sabendo-se que o preço do litro da gasolina é R\$ 5,80 e o preço do litro do etanol é R\$ 4,90.

Expressões Aritméticas - MOD

Problema:

Receba um número e diga se ele é par ou ímpar

```
numero = int(input("Digite um numero inteiro: "))
if numero%2 == 0:
    print("Numero é par", numero)
else:
    print("Numero é impar", numero)
```

Desafio

Problema:

Dados os valores de horários abaixo, decifre a lógica e faça um código para executar.

entrada01 3:45

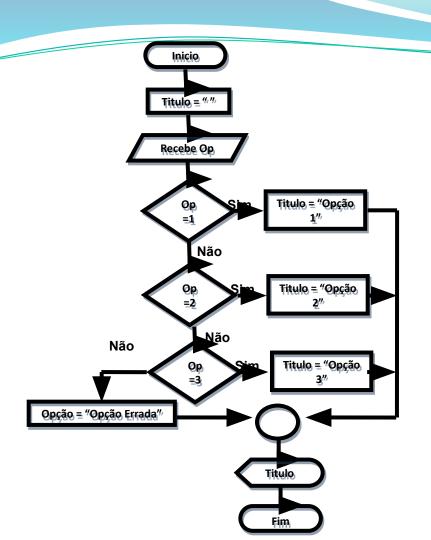
entrada02 14:20

saída 6:05

CASO SELECIONE / SELECT... CASE

A estrutura de decisão **CASO/SELECIONE** é utilizada para testar, na condição, uma única expressão, que produz um resultado, ou, então, o valor de uma variável, em que está armazenado um determinado conteúdo. Compara-se, então, o resultado obtido no teste com os valores fornecidos em cada cláusula "**CASO**".

No exemplo do diagrama abaixo, é recebido uma variável "Op" e testado seu conteúdo, caso uma das condições seja satisfeita, é atribuído para a variável Titulo a String "Opção X", caso contrario é atribuído a String "Opção Errada".



Programação (Python)

receba, do usuário, um número de 1 a 12 e mostre o nome do mês correspondente. Caso o mês não exista, mostrar "Valor inválido".

Obs: usando IF/elif/else

O que é o comando switch case?

Descrição. Da mesma forma que o comando **if**, o comando switch case controla o fluxo do programa permitindo ao programador especificar código diferente para ser executado em várias condições. Em particular, um comando switch compara o valor de uma variável aos valores especificados nos comandos case

```
mes = int(input("Digite um
  numerò inteiro: "))
match mes:
  case 1:
    print("Janeiro")
  case 2:
    print("Fevereiro")
  case 3:
    print("Março")
  case 4:
    print("Abril")
  case 5:
    print("Maio")
  case 6:
    print("Junho")
```

```
case 7:
  print("Julho")
case 8:
  print("Agosto")
case 9:
  print("Setembro")
case 10:
  print("Outubro")
case 11:
  print("Novembro")
case 12:
  print("Dezembro")
case _ :
  print("Numero
Inválido")
```

Comandos de Repetição ou Laços

- **Laços de Repetição** são usados para repetir comandos até que uma condição ocorra. O Laço evita redundância de código e é uma das práticas mais usadas em programação.
- O Laço de Repetição é usados para:
 - Criação de contadores;
 - Obtenção e recuperação de dados, etc.

Utilizamos os comandos de repetição quando desejamos que um determinado conjunto de instruções ou comandos sejam executados, um numero definido ou indefinido de vezes, ou enquanto um determinado estado de coisas prevalecer ou até que seja alcançado.

Imagine que queremos imprimir todos as letras de um nome. Então, faça um programa que receba um nome de uma pessoas e mostre letra por letra

nome=input("Digite o seu nome: ")

for c in nome: print(c)

A função range()

A função range() retorna uma série numérica no intervalo enviado como argumento. A série retornada é um objeto iterável tipo range e os elementos contidos serão gerados sob demanda.

É comum o uso da função range() com a estrutura for loop. Desta forma temos que a cada ciclo o próximo elemento da sequência será utilizado de tal forma que é possível partirmos de um ponto e ir incrementando, decrementando x unidades.

range([start], stop[, step])

Faça um programa que mostre na tela, a contagem de 1 até 10.

for x in range(10): print(x)

Escreva um algoritmo para ler 10 números e ao final da leitura escrever a soma total dos 10 números lidos.

Receber um número inteiro e mostrar a tabuada de multiplicação desse número (1-10).

Formato da saída:

 $1 \times 5 = 5$

 $2 \times 5 = 10$

numero = int(input("Digite um número inteiro"))

```
for x in range(1, 11):
    print("A tabuada de", numero, "Vezes ", x, "é:",
    numero*x," \n ")
```

receber um número do usuário e mostrar todos os números ímpares do intervalo de 1 até o número digitado

Receber 10 números do usuário e mostre a soma de todos os números ímpares.

Ler o número de alunos existentes em uma turma e, após isto, ler as notas destes alunos, calcular e escrever a média aritmética dessas notas lidas

Laço de Repetição While

O while é a estrutura de repetição mais simples. Ele repete a execução de um bloco de sentenças enquanto uma condição permanecer verdadeira. Na primeira vez que a condição se tornar falsa, o while não repetirá a execução do bloco, e a execução continuará com a sentença ou comando que vem logo após o bloco do while, na seqüência do programa.

```
n = 10

soma = 0
i = 1

while i <= n:
    soma = soma + i
    i = i+1

print("A Soma é : ", soma)</pre>
```