|  |  |  |
| --- | --- | --- |
|  | **PYTHON PARA CIÊNCIA DE DADOS**  **ESTRUTURA CONDICIONAL** | Imagem relacionada |

**Conceito:** serve para testar variáveis e fazer comparações, criando fluxos de execução independentes dentro do programa.

|  |  |  |  |
| --- | --- | --- | --- |
| **SINTAXE GERAL** | | | |
| **Estrutura Simples** | **Estrutura Composta** | **Estrutura Encadeada/Aninhada** | **Estrutura Aninhada/Encadeada** |
| if condição:  comandos | if condição:  comandos  else:  comandos | if condição1:  comandos  elif condição2:  comandos  else:  comandos | if condição 1:  comandos  if condição 2:  comandos  else:  comandos  else:  comandos |

**Exemplo resolvido**: ler do usuário a temperatura do dia de hoje (em °C). Se o valor estiver abaixo de 15°C, deve-se mostrar na tela a mensagem: “*está frio!*”. Se estiver acima de 25ºC, mostrar a mensagem: “*Está calor!*”. Caso contrário, mostrar a mensagem: “*Temperatura agradável*”.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *# lê valor do usuário*  t = float(input("Digite a temperatura (em ºC) de hoje:"))  *# testa o valor lido*  if (t < 15):  print("Está frio!")  elif (t < 25):  print("Temperatura agradável")  else:  print("Está calor!") | |  | Também é possível testar mais de uma condição ao mesmo tempo.   |  |  | | --- | --- | | if (15<=t<=25):  print('Temp. agradável') | if (t>=15 **and** t<=25):  print('Temp. agradável') | | Essas duas formas são equivalentes. Aqui, o *if* só executa se **as duas** condições forem verdadeiras **ao mesmo tempo**. | |  |  | | --- | | if (t<15 **or** t>25):  print('Temp. desagradável: quente ou frio') | | Aqui, o *if* executa se pelo menos uma das condições for verdadeira | | | |
|  | **PYTHON PARA CIÊNCIA DE DADOS**  **FUNÇÕES (MODULARIZAÇÃO)** | | | Imagem relacionada |

Na programação, é possível **modularizar** trechos do código por meio de **funções**.

**Função:** conjunto de comandos que executam uma **tarefa particular** agrupados em um **bloco**, que recebe um **nome** e através deste pode ser **chamado**.

Abaixo uma comparação entre a solução tradicional e a solução usando função para um programa que calcula a soma entre dois valores lidos do usuário.

|  |  |
| --- | --- |
| **PROGRAMA – SOLUÇÃO TRADICIONAL** | **PROGRAMA – SOLUÇÃO COM FUNÇÃO** |
|  | *# função de soma* |
|  | def **soma**(x,y): |
|  | result = x+y  return result  Isso é uma **função**! |
|  |  |
| *# 1ª etapa: lê dois valores do usuário* | *# 1ª etapa: lê dois valores do usuário* |
| a = int(input("Digite o 1º valor:")) | a = int(input("Digite o 1º valor:")) |
| b = int(input("Digite o 2º valor:")) | b = int(input("Digite o 2º valor:")) |
|  |  |
| *# 2ª etapa: soma os valores de forma simples* | *# 2ª etapa: soma os valores por meio da função* |
| c = a+b; | c = **soma**(a,b) |
|  | Aqui a função está sendo **chamada** |
| *# 3ª etapa: mostra o resultado na tela* | *# 3ª etapa: mostra o resultado na tela* |
| print("A soma vale:",c) | print("A soma vale:",c) |
|  |  |
| Nesta solução, os comandos são executados sequencialmente, de cima para baixo, sem nenhum desvio. | Nesta solução, ao chamar a função, a **execução** é desviada para que os comandos da função possam ser executados. Ao término, o programa retorna na linha em que ocorreu a chamada da função (c = soma(a,b)). |

**EXERCÍCIOS**

Em todos os exercícios abaixo:

* Utilize função no desenvolvimento.
* Garanta que as entradas lidas do usuário sejam válidas, de maneira que o programa não prossiga enquanto forem informadas entradas inválidas.

1. Criar um programa em Python que, a partir dos valores de peso (em kg) e de altura (em m) de uma pessoa digitados pelo usuário, imprime o IMC daquela pessoa e também sua classificação conforme a diretriz de saúde abaixo. Utilizar duas funções: uma para calcular e retornar o IMC, e outra para obter e retornar a classificação de peso do indivíduo.

|  |  |
| --- | --- |
| **IMC** | **Classificação** |
| (0,0 - 18,5) | Abaixo do peso |
| [18,5 - 25,0) | Peso saudável |
| [25,0 - ∞) | Sobrepeso |

parênteses representam *intervalo aberto* e colchetes representam *intervalo fechado*.

1. Alterar o programa em Python do exercício anterior de forma a considerar mais categorias de peso, conforme a relação abaixo:

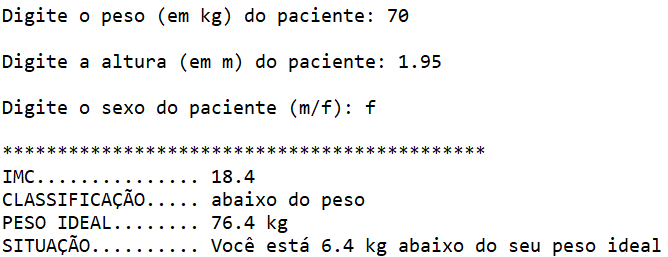
|  |  |
| --- | --- |
| **IMC** | **Classificação** |
| (0,0 - 18,5) | Abaixo do peso |
| [18,5 - 25,0) | Peso saudável |
| [25,0 - 30,0) | Sobrepeso |
| [30,0 - 35,0) | Obesidade grau I |
| [35,0 - 40,0) | Obesidade grau II (severa) |
| [40,0 - ∞) | Obesidade grau III (mórbida) |

parênteses representam *intervalo aberto* e colchetes representam *intervalo fechado*.

1. O programa em Python fornecido abaixo lê do usuário o peso (em kg), a altura (em m) e o sexo (m/f) de um paciente qualquer. Depois, por meio de funções, o programa calcula/obtém: o IMC do paciente, sua categoria de peso, seu peso ideal e sua situação, ou seja, quantos kg está acima ou abaixo do peso ideal. Por fim, o programa apresenta na tela essas quatro informações.

|  |  |
| --- | --- |
| **Programa em Python** |  |
| # -- definição das funções –  # \*\* os estudantes devem fazer \*\*  # -- programa principal --  # entrada de dados (validar as entradas)  # \*\* os estudantes devem fazer \*\*  # obtenção dos resultados  imc = calcula\_imc(p,h)  cat = categoria\_peso(imc)  pi = peso\_ideal(h,s)  sit = situacao(p,pi)  # impressão dos resultados  print('\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('IMC...............', round(imc,1))  print('CLASSIFICAÇÃO.....', cat)  print('PESO IDEAL........', round(pi,1), 'kg')  print('SITUAÇÃO..........', sit) | Faça o que se pede:   1. Copie o código fornecido e cole em um editor de sua preferência (Spyder, Pycharm, Idle, etc.). 2. Crie a função calcula\_imc(p,h), que calcula e retorna o IMC do paciente em função de seu peso e altura, sabendo-se que . 3. Crie a função categoria\_peso(imc), que retorna a classificação de peso do paciente em função de seu IMC, sabendo-se que: 4. Crie a função peso\_ideal(h,s), que calcula a retorna o peso ideal (em kg) do paciente em função de sua altura e sexo, sabendo-se que:  |  |  | | --- | --- | | **Sexo** | **Peso ideal** | | Masculino | (72,7 × altura) - 58,0 | | Feminino | (62,1× altura) - 44,7 |  1. Crie a função situacao(p,pi), que calcula e retorna quantos kg o paciente está acima ou abaixo do seu peso ideal. |

Veja abaixo um exemplo de execução do programa:



1. Criar um programa em Python que, a partir de dois números quaisquer e também de uma operação aritmética (+, -, \*, /) digitados pelo usuário, apresenta na tela o resultado do cálculo solicitado. Utilize quatro funções, uma para cada operação matemática elementar. Exemplos de execução:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

1. Criar um programa em Python que, a partir dos coeficientes ‘*a’*, ‘*b’* e ‘*c’* de uma equação do 2º grau digitados pelo usuário, apresenta na tela quantas raízes reais existem e quais são seus valores. As saídas possíveis são: “Não existem raízes reais”, “Há uma única raiz real que vale:\_\_\_” e “Há duas raízes reais que valem \_\_\_ e \_\_\_”. Utilizar uma única função para resolver o problema. Obs.: se necessário, utilize a função **str( )** para converter um número em texto.
2. Criar um programa em Python que imprime se um triângulo é “equilátero”, “isósceles” ou “escaleno” a partir das medidas dos três lados informadas pelo usuário.
3. Criar um programa em Python que imprime se um valor inteiro qualquer informado pelo usuário é “par” ou “ímpar”.
4. Criar um programa em Python que imprime se um visitante do Hopi Hari está “liberado” ou “proibido” de andar na montanha russa do parque com base na idade (em anos) e no peso (em kg) informados pelo usuário. Para estar liberado, é preciso ser maior de 15 anos e não pesar mais que 120 kg.
5. Crie um programa em Python que, a partir do tempo (em minutos) de permanência de um veículo no estacionamento de um shopping informado pelo usuário, imprime o valor a ser pago para retirar o veículo. A cobrança é feita da seguinte forma:

|  |  |  |  |
| --- | --- | --- | --- |
| Até 15 minutos, R$ 3,00 | De 15 a 30 minutos, R$ 6,00 | De 30 a 60 minutos, R$ 10,00 | Acima de 60 minutos, R$10,00 + R$ 1,00 por minuto excedente aos 60 iniciais. |