Fundamentos de Programação

Estruturas CONDICIONAIS

O Comando *If*

Aula 09

Post It da Aula Passada

Fundamentação em *Strings*

- Cadeia de caracteres
- Índices
  - Percorrendo os caracteres
  - Iniciando em zero
- Concatenação
- Notação Head/Tail
  - [n:m]

Funções de *Strings*

- Funções de *string*
  - Notação ponto
  - “abcdef”.função()
- Exemplos
  - startswith
  - endswith
  - replace

Tomada de Decisão

Enquanto seres humanos, passamos por vários momentos de decisão durante as nossas vidas... sejam elas das mais simples:

Eu devo comer uma maçã agora?

Até as mais complexas:

Qual graduação eu devo cursar?

Tomada de Decisão

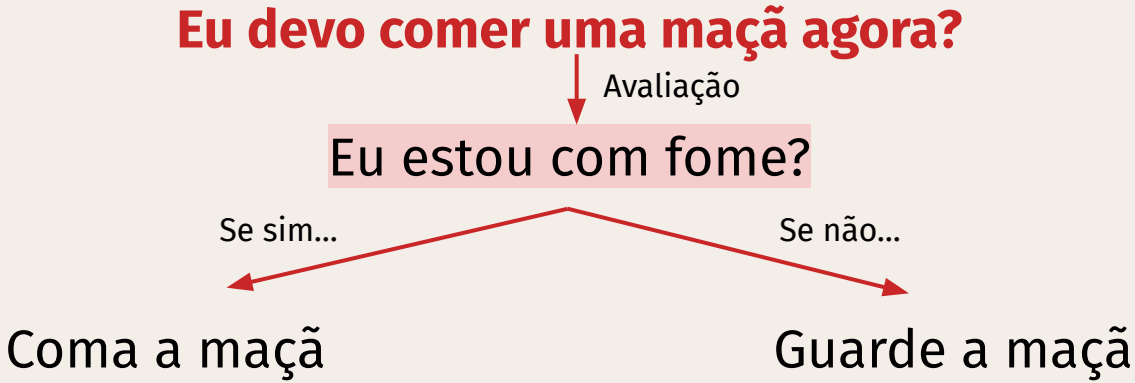
Naturalmente, para decidir, nós projetamos diversos cenários e realizamos uma série de perguntas respondidas conforme o nosso contexto e situação:

Eu devo comer uma maçã agora?: eu estou com fome?

Qual graduação eu devo cursar?: as matérias de um curso X me agradam? Eu posso seguir o curso X? Eu quero seguir o curso X?

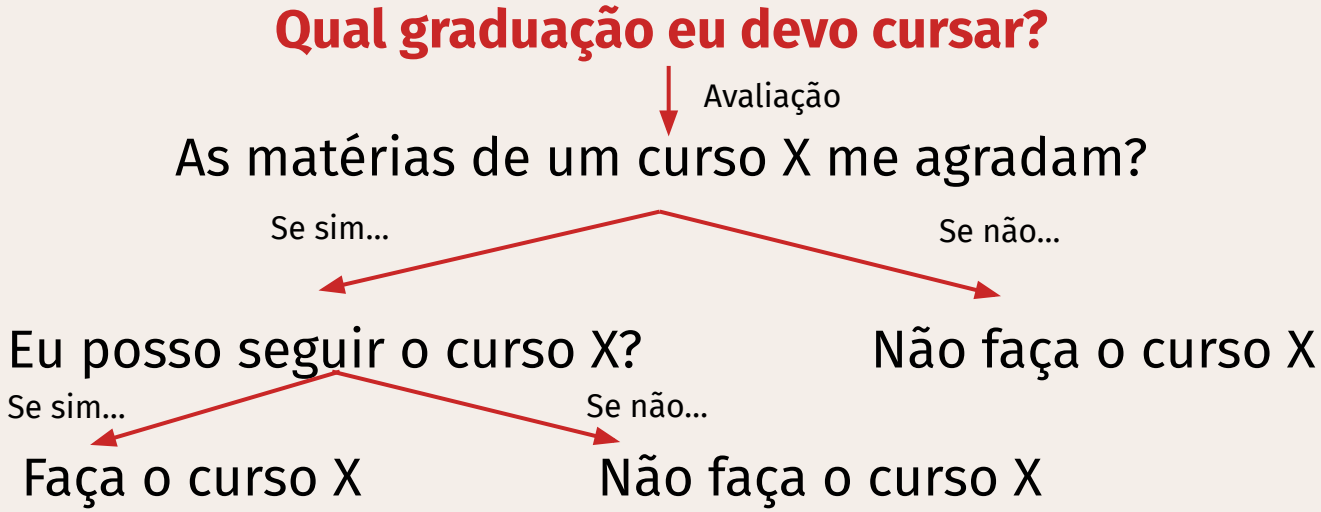
Tomada de Decisão

De acordo com que respondemos certas perguntas, vamos adquirindo clareza do que podemos ou iremos fazer. Algumas questões são simples, sendo que com uma avaliação já é possível decidir:



Tomada de Decisão

Outras questões são complexas e exigem a análise de vários aspectos para serem respondidas:



Estruturas Condicionais

Em linguagens de programação, podemos criar estruturas condicionais muito semelhantes às que criamos mentalmente quando precisamos tomar uma decisão

Essas estruturas condicionais são criadas através do comando popularmente chamado (na maior parte das linguagens de programação, inclusive em Python):

if

(palavra em inglês para “se”)

O Comando *If*

O comando *if* nos permite examinar o estado de um programa em um determinado momento e modificar o seu **fluxo de execução** conforme o resultado de uma avaliação condicional

Essas avaliações podem ser simples, concluídas com apenas **uma resposta de "sim" (*True*) ou "não" (*False*)**, ou podem ser complexas, exigindo múltiplas respostas de sim ou não estruturadas

O Comando *If*

O comando *if*, de forma geral, funciona por testes condicionais. Tais testes devem retornar um **valor lógico final (*True* ou *False*)** que determinará qual fluxo de execução deve ser tomado pelo programa

Assim, testes condicionais tipicamente se organizam através de **expressões relacionais e/ou expressões lógicas**

Também podemos utilizar **funções que retornam valores lógicos** como testes condicionais

O Comando *If*

15 + math.sqrt(4) == 17

True

2 ** 3 > 3 ** 2

False

x = input(“Número real:”)
math.ceil(float(x)) > 1

?

x = input(“Número inteiro:”)
x.isnum()

?

x = input(“Número inteiro:”)
x.isnum() and int(x) > 1

?

O Comando *If*

Assim, tendo em mente os testes condicionais que serão realizados podemos estruturar o comando *if*. Em Python, tal comando apresenta a seguinte estrutura:

if teste:
    Instruções executadas caso o teste seja verdadeiro (***True***)
Instruções executadas independentemente do condicional

Atenção para a INDENTAÇÃO

Indentação

A indentação define o **escopo de um determinado trecho de código**. Por exemplo, no caso do *if*, o código submetido ao mesmo está **indentado com uma tabulação**. Para sair do escopo operacional do if, basta remover a tabulação:


```
if teste:
    Instruções no escopo do if
Instruções fora do escopo do if
```

Podemos indentar com espaços ou tabulações, mas a indentação deve ser sempre a mesma

O Comando *If*

```
resposta = input(“Qual é a resposta para tudo?:”)
if resposta == “42”:
    print(“Segundo o guia do mochileiro das galáxias, você está certo!”)
if resposta == “nada”:
    print(“Você foi bastante filosófico!”)
if resposta != “42” and resposta != “nada”:
    print(“Hmm, interessante!”)
```

Exercício #09

Faça um programa que recebe um número inteiro pela entrada padrão (*input*), verifique se a *string* retornada da entrada padrão é de fato um valor numérico (*isnumeric*). Se não for, exiba o erro na tela e pare o programa (para parar o programa utilizamos a função *exit*). Se for um valor numérico, faça:

a) Faça o *casting* da *string* para o tipo de dados inteiro e armazene o resultado na variável x
b) Execute a operação x ** 2 + 5 e armazene o resultado em uma variável
c) Verifique se o valor é maior que 1000, se sim, exiba na tela “Resultado > 1000”
d) Verifique se o valor é menor ou igual que 1000, se sim, exiba na rela “Resultado <= 1000”

Fundamentos de Programação

Aula 09

Obrigado e ATÉ A PRÓXIMA!