

Tópico 05

Introdução à programação de computadores

Estruturas de controle – condicional

1. Introdução

Aluna(o), é importante que entenda a necessidade das estruturas de controle condicionais para seu código Python e também para outras linguagens de programação!

Algumas estruturas de programação existem desde linguagens conhecidas como BAIXO NÍVEL (como C, Assembly etc.), pois são essenciais para que o código tenha utilidade e para que fique compreensível para que o lê. Este é o caso das condicionais.

O IF, ELSE e ELIF são estruturas que variam muito pouco de linguagem para linguagem. Na maioria das vezes, muda-se na forma, mas não no funcionamento. Se você ainda não conhece, vai ficar fascinada(o).

Se você já programou em outras linguagens, aqui terá pouca novidade. No entanto, recomendamos que siga o passo a passo, pois, muitas vezes, não sabemos de certas características quanto ao funcionamento do código. Também, a linguagem Python, apesar de ser muito fácil, é muito exigente e qualquer erro na sintaxe pode impedir o funcionamento de seu código.



2. Estrutura de seleção simples (IF)

A estrutura de seleção **if** é um dos conceitos básicos mais importantes e significativos da programação, não só da linguagem Python. Sem ela, nada de verdadeiramente útil pode ser feito em um código.



William Henry Gates III, mais conhecido como Bill Gates, foi um dos fundadores da Microsoft e, neste vídeo, ele explica a importância e o funcionamento do if. Hour of Code – Bill Gates explains If statements.

Hour of Code - Bill Gates explains If stateme...



No vídeo acima, podemos ver, de forma didática e prática, o funcionamento do if em uma linguagem de blocos de programação. Aqui, vamos aprender o mesmo, porém vamos compreender mais profundamente seu funcionamento e, especialmente, na linguagem de programação Python.

Por vezes, nós nos deparamos com situações em que decisões precisam ser tomadas durante a execução do programa. Se uma administradora, por exemplo, loga no sistema, ela pode ter acesso aos dados dos funcionários, porém, se um funcionário logar, ele terá acesso apenas às próprias informações.

No caso acima, trechos do código poderão ser executados, ou não. E, para tratar tal possibilidade, necessitaremos de expressões lógicas, para representar escolhas.

O problema acima descrito se chama **CONTROLE DE FLUXO**.

No dia a dia, na linguagem natural, utilizamos a estrutura if constantemente, que, traduzindo, para o português, significa SE. Por exemplo, se a água estiver morna, tomarei banho.

Em Python, a estrutura usada para decisões é o **if**.

Formato geral do if.



Caso a condição seja verdadeira, o bloco de código identificado dentro do if será executado.

Utilizando nosso exemplo de linguagem natural, ficaria assim:

if água morna:

 tomar banho

Exemplo 1:

- No código a seguir, há um exemplo de uso do if no qual verificamos se a variável idade é menor que 25;
- Em caso positivo, imprimimos uma mensagem na tela e, em caso negativo, o código seguirá normalmente, desconsiderando a terceira linha.

Código do exemplo 1.

Observe que é necessário indentar (usar espaços em branco e múltiplos de 4), a fim de que o código entenda que o BLOCO DE CÓDIGO está dentro da estrutura if. Assim, o bloco é tudo que está com o mesmo espaçamento.

“Dois pontos denota o início de um bloco de código indentado após o qual todo o código deve ser recuado no mesmo valor até o final do bloco.”

(MCKINNEY, 2019, p. 31).

Segue um exemplo do que acontece, quando falta indentação:



Erro de indentação.

Parece estranho esse erro, caso você venha de outras linguagens, certo?



Você sabia que esta estrutura foi usada, para forçar quem desenvolve o código a deixá-lo organizado? Isto aumenta a qualidade do código, pois facilita enxergar os

erros, aumenta a reusabilidade, porque facilita a compreensão dele, mesmo após muito tempo sem usá-lo, e, dentre outras coisas, facilita também o trabalho em equipe.

Exemplo 2:

- No exemplo a seguir, vamos verificar a possibilidade de separar o objeto observado em faixas de qualificação;
- Podemos observar que, de tal maneira, todos os **if** são verificados, podendo haver mais de um verdadeiro.



Código do exemplo 2.

Perceba que, desta forma, é impossível decretar exatamente a faixa de idade que estamos avaliando, forçando-nos a usar mais de uma verificação por **if**.

O próximo exemplo nos mostrará como fazer.

Exemplo 3:

- No código a seguir, temos um exemplo de uso de vários **if**;
- Neste caso, cada bloco de código identificado dentro o **if** será executado, se a CONDIÇÃO for verdadeira;
- Usamos aqui mais de uma verificação em alguns dos **if**, para consertar o problema do código do exemplo 2.

Código do exemplo 3.

No exemplo acima, existe um problema que não pode ser percebido facilmente. Mesmo o código encontrando sua chave de resposta no terceiro **if**, ele ainda assim tenta resolver o quarto **if**. Além disso, ao todo, são seis diferentes testes feitos, para encontrar a faixa de idade.

Imaginemos se este código fosse acessado centenas de vezes por segundo, ou um caso que analise centenas, ou milhares de **if**. Seria muito mais rápido, se o código saísse da estrutura condicional, assim que encontrasse sua chave de resposta, certo?



Exemplos de utilização da Estrutura de Seleção Simples (IF) – Parte 01



Exemplos de utilização da Estrutura de Seleção Simples (IF) – Parte 02

3. Estrutura de seleção composta (IF-ELSE)



Vimos anteriormente como utilizar a sentença **if**, para executar uma ação, caso a condição seja atendida. No entanto, nenhum comportamento específico foi definido para o caso de a condição não ser satisfeita. Para tal, usamos a sentença **else**.

Formato geral do if-else.

Assim, caso a CONDIÇÃO seja satisfeita, o BLOCO DE CÓDIGO A será executado. Caso contrário, o BLOCO DE CÓDIGO B será executado.



Lembra-se do uso do SE e do SE NÃO no pensamento lógico? Se não se lembra, vale a pena visitar o livro Iniciação à Lógica Matemática para lembrar! ALENCAR FILHO, Edgard de. Iniciação à lógica matemática. São Paulo: Nobel, 2006 e edições anteriores.

Um exemplo de linguagem natural seria, SE A ÁGUA ESTIVER MORNA, EU TOMO BANHO, SE NÃO ESTIVER MORNA, EU A ESQUENTO.

```
if água morna:  
    tomar banho  
else:  
    esquentar água
```



Quer aprender a partir de um exemplo prático de temperatura? O livro Python para Desenvolvedores tem um exemplo bacana na página 26. BORGES, Luiz Eduardo de. Python para desenvolvedores. Rio de Janeiro: Edição do autor, 2010.

Exemplo 4

- No exemplo a seguir, consideramos uma análise simples de idade, para observar se uma pessoa é jovem, ou não;
- É utilizada a estrutura **if-else**, para evitar que o **if** seja processado duas vezes.

Código do exemplo 4.

Repare que, no exemplo acima, não precisamos colocar dois **if**, tampouco mais de uma verificação no mesmo **if**.

No entanto, tudo fica mais complexo, quando temos várias condições.

Exemplo 5:

- O código a seguir tem a mesma proposta do exemplo 3, encontrar a faixa de idade que a pessoa se encontra;
- No entanto, desta vez, usaremos a estrutura **if-else**.



Código do exemplo 5

Note que, no exemplo 5, o código faz apenas três testes para encontrar o resultado, diferente do exemplo 3 em que o código fazia seis testes.

Quando o código encontra o primeiro **if** correto (idade < 60), ele processa seu bloco de código (print (“adulto”)) e sai da estrutura

condicional.

Vamos buscar um exemplo ainda mais rápido:

Código do exemplo 6.

No exemplo acima, apenas o primeiro **if** é processado. No Código do exemplo 3, mesmo se tivéssemos a mesma entrada, teríamos seis **if** sendo processados. Isso significa dizer que nosso código melhor escrito executa apenas um sexto das tarefas, aumentando consideravelmente a velocidade.



Tudo bem, resolvemos o problema da velocidade do código, uma vez que, agora, ele faz menos testes, mas parece que ficou mais complexo programar, não acha?

Pode parecer que essas várias condições aninhadas não sejam um grande problema, mas isso é porque o nosso exemplo é simples. No caso de grandes projetos em que existem várias condições, esse código ficaria mais complicado de ser compreendido.

Vamos ver um modo de ter uma alta velocidade e uma fácil programação!



Exemplos de utilização da Estrutura de Seleção
Composta (IF – ELSE) – Parte 01



Exemplos de utilização da Estrutura de Seleção
Composta (IF – ELSE) – Parte 02



4. Estrutura de seleção encadeada (IF – ELIF – ELSE)

PYTHON tem uma sentença especial, a fim de ajudar os encadeamentos de **if-else**. Tal sentença se chama **elif**.

*“Uma instrução **if** pode ser opcionalmente seguida por um ou mais blocos **elif** e um bloco **else** pega-tudo caso todas as condições sejam falsas”.*

(MCKINNEY, 2019, p. 46, tradução nossa).

Assim, sempre que houver mais de uma condição alternativa que precisa ser verificada, podemos utilizar a sentença **elif**, para avaliar as expressões intermediárias antes de usar o **else**.

Formato geral do if-elif-else.



Em linguagem natural, poderíamos dizer **SE A ÁGUA ESTIVER MUITO QUENTE, DEIXO ESFRIAR. SE NÃO ESTIVER MUITO QUENTE, MAS SE ESTIVER MORNA, EU TOMO BANHO. SE NÃO ESTIVER MUITO QUENTE NEM MORNA, EU ESQUENTO.**

if água muito quente:

deixar a água esfriar

elif água morna:

tomar banho

else:

esquentar a água

Exemplo 7:

- No código a seguir, temos um exemplo de uso de sentenças **if-elif-else**;
- Desenvolveremos o mesmo objetivo dos exemplos 3 e 5, usando a nova estrutura.

Código do exemplo 7.

Apesar deste último exemplo ser tão rápido em relação ao exemplo anterior, temos de levar em consideração sua legibilidade.



Exemplo 8:

- No código a seguir, temos outro exemplo de uso de sentenças **if-elif-else**;
- Aqui, vamos mostrar o que acontece, quando o código encontra uma condição verdadeira logo no primeiro **if**.

Código do exemplo 8.



Vamos fazer um teste? Digite o código abaixo no seu interpretador Python (pode ser Jupyter Notebook, ou Google Colab).

```
idade = int (input('Digite sua idade:'))
if idade <= 12:
    print("Criança")
elif idade <= 25:
    print("Jovem")
elif idade < 60:
    print("Adulto")
else:
    print("Idoso")
```

Então, coloque sua idade e veja se o código a classifica corretamente.

Neste código, temos apenas uma verificação de condicional e o código já sai da estrutura de controle. Portanto, em termos de velocidade de código, não se nota diferença para um código que utilize apenas **if-else**. No entanto, podemos perceber uma diferença considerável, a fim de compreender a lógica da estrutura de controle condicional, quando utilizamos a estrutura **if-elif-else**.



Quando programamos em equipes, é necessário que uma pessoa entenda o código da outra. Nesse contexto, a linguagem Python usa diversas táticas, para tornar o código obrigatoriamente legível, sendo que indentação, **elif** etc. são exemplos de como fazê-lo.

Agora, temos um código rápido e legível, como se espera de um código PYTHON.



Exemplos de utilização da Estrutura de Seleção
Encadeada (IF – ELIF – ELSE) – Parte 01



Exemplos de utilização da Estrutura de Seleção
Encadeada (IF – ELIF – ELSE) – Parte 02



5. Conclusão

Este tópico procurou mostrar alguns exemplos básicos de como programar usando estruturas de controle condicionais na

linguagem Python e a sua importância.

Buscamos aprender conceitos e regras sobre a linguagem e a sua estrutura. A partir destes conceitos, vamos poder construir códigos mais complexos e até mesmo levar isto para outras linguagens de programação, quando necessário.

Assim, procuramos fundamentar as bases da compreensão de um tópico importante que será utilizado frequentemente daqui para frente em todas linguagens de programação.

6. Referências

MCKINNEY, Wes. Python para análise de dados: tratamento de dados com pandas, Numpy e IPython. São Paulo, SP: Novatec, 2019 e edições anteriores. 615 p. ISBN 9788575226476.

ALENCAR FILHO, Edgard de. Iniciação à lógica matemática. São Paulo: Nobel, 2006 e edições anteriores.

EDUARDO BORGES, Luiz de. Python para desenvolvedores. Rio de Janeiro: Edição do autor, 2010.



Parabéns, esta aula foi
concluída!

O que achou do conteúdo estudado?

Péssimo

Ruim

Normal

Bom

Excelente

Deixe aqui seu comentário

Mínimo de caracteres: 0/150

Enviar

