**Introdução a Programação**

**Tipos de dados constantes e variáveis**

**Exemplo com pseudocódigo**

Para dar continuidade ao assunto, vamos analisar um exemplo mais técnico do mesmo código utilizado na aula de tipos de representação de algoritmos.

Início

1. Pegar primeira nota do aluno;
2. Pegar segunda nota do aluno;
3. Somar as duas notas;
4. Dividir a nota por dois;
5. Se o resultado for maior que sete então;
   1. O aluno foi aprovado;
6. Se não;
   1. O aluno foi reprovado;

Fim.

Após a leitura do [pseudocódigo](https://kenzie.com.br/blog/pseudocodigo/), é possível perceber que o resultado citado na quinta ação se refere ao resultado da média das duas notas. Como o computador não consegue entender esse comando,  é necessário informar passo a passo. Isso também acontece com o resultado da soma das duas notas.

Além disso, as notas dos alunos devem estar armazenadas em [variáveis](http://leg.ufpr.br/~silvia/CE055/node8.html) para que seja possível acessá-las durante o código.

A seguir, as variáveis e os tipos para identificação serão inseridos no exemplo.

Início

float **nota1** = valor da primeira nota do aluno;

float **nota2** = valor da segunda nota do aluno;

float **res\_soma** = **nota1** + **nota2;**

float **res\_media** = **res\_soma** / 2;

Se **res\_media** for maior que sete então:

O aluno foi aprovado;

Se não:

O aluno foi reprovado;

Fim

É possível perceber que o pseudocódigo passa a ser mais complexo, parecendo com um código de programação.

O tipo [float](https://docs.microsoft.com/pt-br/cpp/c-language/type-float?view=msvc-170) foi utilizado no exemplo, pois os valores das notas, das somas das notas e da média podem ser compostos por um número decimal.

**Linguagem Python**

Na maioria das linguagens, é preciso declarar o tipo do valor que será inserido na variável, porém, isso não acontece na linguagem Python, pois ela tem um nível de adaptabilidade maior. Então, a variável assumirá o tipo correspondente ao valor inserido nela.

Esse recurso torna essa linguagem mais simples, o que não prejudica o estudo e a compreensão das variáveis, pois você só consegue desenvolver o código se entender esses conceitos. Exemplo:

* [JavaScript](https://www.zup.com.br/blog/java) -> var numero = 5
* Python -> numero = 5

**Exemplo com código em Python**

Para compreender melhor os tipos de variáveis, você deve praticar alguns comandos em Python. No seu espaço de codificação desejado, escreva o código a seguir.

numero\_inteiro = 5

numero\_float = 5.7

texto = “este é um texto”

booleano = False

print(“Este é um tipo”)

print(type(numero\_inteiro))

print(“Este é um tipo”)

print(type(numero\_float))

print(“Este é um tipo”)

print(type(texto))

print(“Este é um tipo”)

print(type(booleano))

A função [type()](https://www.w3schools.com/python/ref_func_type.asp) em Python informa o tipo da variável que está sendo inserida. Então, esse código mostra na tela, através do comando [print()](https://docs.python.org/pt-br/3/library/functions.html#print), os tipos das variáveis criadas, sendo apresentados em sequência [int](http://www.univasf.edu.br/~marcelo.linder/arquivos_pc/aulas/aula13.pdf), float, [string](https://homepages.dcc.ufmg.br/~rodolfo/aedsi-2-06/cadeiadecaractere/cadeiadecaractere.html) e [boolean](https://docs.microsoft.com/pt-br/office/vba/language/reference/user-interface-help/boolean-data-type).

É possível alterar os tipos das variáveis, pois eles não são definidos na Python. Por exemplo:

**variavel = 5**

**print(“primeiro tipo”)**

**print(type(variavel))**

**variavel = “texto”**

**print(“segundo tipo”)**

**print(type(variavel))**

Nesse código, é possível observar que a mesma variável, inicialmente do tipoint, se tornou do tipo string. Isso ocorre devido à tipagem automática da Python.

**Introdução a Programação**

**Estruturas e comandos**

**lógicos de decisão**

**Estrutura de decisão**

Durante o desenvolvimento de um [algoritmo](https://www.significados.com.br/algoritmo/#:~:text=Um%20algoritmo%20%C3%A9%20uma%20sequ%C3%AAncia,descreveu%20regras%20para%20equa%C3%A7%C3%B5es%20matem%C3%A1ticas.), podem haver momentos em que o caminho do código mudará, dependendo do resultado adquirido ou da situação.

Para escolher qual caminho seguir, são utilizadas estruturas de decisão, as quais são as situações “se…então”, ou seja, as escolhas que podem ser feitas durante a execução do código.

Utilizando o exemplo básico da receita de cobertura de bolo, essas estruturas de condição seriam:

* coloque 100 g de granulado;
* **SE** não houver granulado **ENTÃO**;
  + coloque 200 g de chocolate ralado.

Por mais simples que possa parecer, essa simples adição já torna a receita mais genérica. Pois, é possível utilizar dois ingredientes diferentes para a finalização da cobertura.

Os [operadores](https://devschannel.com/logica-de-programacao/operadores), principalmente os relacionais e os lógicos, são muito utilizados na construção dessas condições. Por exemplo, no código da média do aluno:

1. media = (nota1 + nota2) / 2
2. **se** (media >= 7) **então**

Aprovado

1. **se** (media < 7) **então**

      Reprovado

Após obter a média do aluno, o programa analisa **se** (condição) a média dele foi maior que ou igual a sete. Caso essa operação seja verdadeira, o aluno é aprovado, caso seja falsa, ele é reprovado.

No exemplo anterior, foram aplicadas duas condições e só uma será executada, são elas: a média maior ou igual a sete; e a média menor que sete.

Ou seja, **se** a média do aluno for maior ou igual a sete, ele está aprovado, **senão**, está reprovado. O termo **senão** é utilizado na programação para facilitar a compreensão dos comandos.

1. media = (nota1 + nota2) / 2
2. **se** (media >= 7) **então**

Aprovado

1. **Senão**

Reprovado

E se a ação possuir mais de duas possibilidades? Nem sempre será possível simplesmente aplicar vários "**se**", pois isso pode acarretar alguns problemas.

O próximo exemplo apresenta um pseudocódigo que recebe a média do aluno como entrada e armazena a categoria de nota como letras.

Exemplo

1. media = 8
2. **se** (media == 10) **então**

mediaLetra = “A”

1. **se** (media >= 9) **então**

mediaLetra = “B”

1. **se** (media >= 8) **então**

mediaLetra = “C”

1. **se** (media >= 7) **então**

mediaLetra = “D”

1. **Senão**

mediaLetra = “F”

De início, “C” aparenta ser o valor armazenado na variável **mediaLetra**, mas, na quinta linha, a condição também é verdadeira, pois o valor da variável **media** é maior que sete. Então, o valor armazenado na variável seria “D” .

Isso acontece pois todos os “**se**” são independentes, sendo necessário inseri-los em uma condição para que apenas um deles seja executado. Para isso, é utilizado o “[**senão se**](https://panda.ime.usp.br/cc110/static/cc110/05-if.html)”.

Como mostra o próximo exemplo, quando o “**senão se**” é utilizado no lugar de cada “**se**”, se mantém o primeiro “**se**” e quando uma condição “**senão se**” for a correta, as outras não serão mais testadas. Observe o exemplo a seguir.

media = 8

**se** (media == 10) **então**

mediaLetra = “A”

**senão se** (media >= 9) **então**

mediaLetra = “B”

**senão se** (media >= 8) **então**

mediaLetra = “C”

**senão se** (media >= 7) **então**

mediaLetra = “D”

**senão**

mediaLetra = “F

**Condições maiores**

Os operadores lógicos também podem ser utilizados nas estruturas condicionais. Com eles, as condições ficam mais estruturadas e trabalham em situações complexas.

Por exemplo, uma outra forma de executar o pseudocódigo anterior seria adicionando outra condição nos “**se**”, definindo o limite superior da nota. Dessa forma, temos:

1. media = 8
2. **se** (media == 10) **então**

a. mediaLetra = “A”

1. **senão se** (media >= 9) **E** (media < 10) **então**

a. mediaLetra = “B”

1. **senão se** (media >= 8) **E** (media < 9) **então**

a. mediaLetra = “C”

1. **senão se** (media >= 7) **E** (media < 8) **então**

a. mediaLetra = “D”

1. **senão**

a. mediaLetra = “F”

Assim, o problema comentado anteriormente não ocorreria, mesmo se não fossem utilizados os “**senão se**”.

As condições que serão colocadas e como elas serão estruturadas dependerá do problema que está sendo solucionado.

**Na linguagem Python**

Todos os comandos que foram utilizados em pseudocódigo existem na [linguagem Python](https://www.hashtagtreinamentos.com/estruturas-condicionais-no-python?gclid=Cj0KCQjw29CRBhCUARIsAOboZbLKY8pyU34BkNnAZN5AwuznHa3MpvrqLEK7nsKXIGAy8fY6s1d17VAaAi06EALw_wcB). Se converter o pseudocódigo para Python, fica:

* **Se** CONDIÇÃO **então** -> **if** CONDIÇÃO :
* **Senão se** CONDIÇÃO então -> **Elif** CONDIÇÃO :
* **Senão** -> **Else** :

Os dois pontos significam o “**então**” e precisam ser inseridos. Todos os comandos dentro do “**if**” precisam ser indentados, pois a Python é uma linguagem indentada.

Praticando

**Problema**: Crie um programa que calcule o status de aprovação do aluno a partir da nota dele. O código deve seguir as instruções:

* nota menor que 4: ele está reprovado;
* nota menor que 7: ele está em recuperação;
* nota maior que 7: ele está aprovado

**Solução:**

nota = 4

if (nota < 4):

   print("Reprovado")

elif (nota < 7):

   print("Em recuperação")

elif(nota >=7):

print("Aprovado")