



# **Algoritmos e Lógica de Programação**



**Algoritmo** = O passo a passo que você vai passar de instruções para o seu computador executar

**Algoritmo** = O passo a passo que você vai passar de instruções para o seu computador executar

**Algoritmo** = O passo a passo que você vai passar de instruções para o seu computador executar



# Como aprender Algoritmos e Lógica de Programação?

**Junto com a linguagem de programação**

Cuidado com a didática

**Fazendo um curso de Algoritmo/Lógica de Programação e depois aprendendo a linguagem**

Como você está fazendo nesse vídeo

```
function getGAData() {  
  3   var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Analytics');  
  4  
  5   // clean last content  
  6   sheet.getRange(1, 3, 4, 1).clearContent();  
  7  
  8   var dateRange = sheet.getRange(1, 3, 2, 1).getValues();  
  9  
 10   var gaData = [];  
 11  
 12   // get data from resultadosdigitais.com.br  
 13   var ids = "1234567890";  
 14  
 15   var options = {  
    'filters': 'ga:sourceMedium=@cpc,ga:sourceMedium=@postad,ga:sourceMedium=@paid',  
    'metrics': 'ga:pageviews,ga:uniquePageviews,ga:conversionRate',  
    'dimensions': 'ga:sourceMedium',  
    'maxResults': 1000  
  };  
 16  
 17   var result = Analytics.Data.Ga.get(ids, startDate, endDate, metrics, options);  
 18   if (result.rows) {  
    for (var i = 0; i < result.rows.length; i++){  
      if (result.rows[i][2] != 0) gaData.push(result.rows[i]);  
    }  
  }  
}
```

## E como usamos isso na prática?

Imagina que você está desenvolvendo um código, digamos uma automação de um relatório de vendas para uma loja.

**Você pode criar essa lógica de 3 grandes formas:**

- Fazer a descrição (em português mesmo)
- Fluxogramas
- Pseudocódigos (Portugol)



## E como usamos isso na prática?

Imagina que você está desenvolvendo um código, digamos uma automação de um relatório de vendas para uma loja.

**Você pode criar essa lógica de 3 grandes formas:**

- **Fazer a descrição (em português mesmo)**
- Fluxogramas
- Pseudocódigos (Portugol)

Passo 1: Importar base de vendas

Passo 2: Agrupar vendas por dia

Passo 3: Agrupar vendas por loja

Passo 4: Somar vendas agrupadas

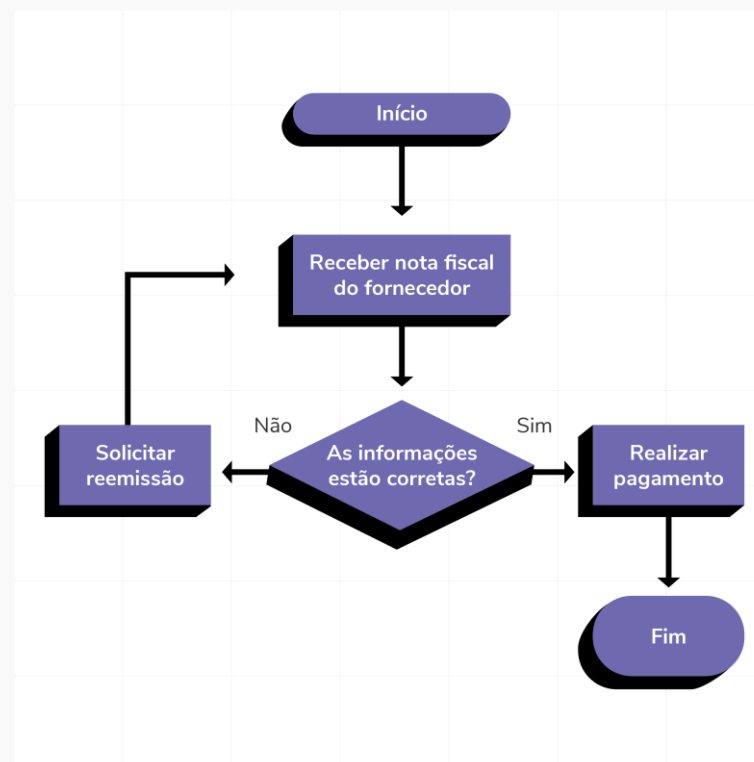
Passo 5: ...

## E como usamos isso na prática?

Imagina que você está desenvolvendo um código, digamos uma automação de um relatório de vendas para uma loja.

**Você pode criar essa lógica de 3 grandes formas:**

- Fazer a descrição (em português mesmo)
- **Fluxogramas**
- Pseudocódigos (Portugol)



## E como usamos isso na prática?

Imagina que você está desenvolvendo um código, digamos uma automação de um relatório de vendas para uma loja.

**Você pode criar essa lógica de 3 grandes formas:**

- Fazer a descrição (em português mesmo)
- Fluxogramas
- **Pseudocódigos (Portugol)**

```
5 // Autor(a)      : Nome do(a) aluno(a)
6 // Data atual    : 25/10/2017
7 Var
8   aluno: caractere
9   nota1, nota2, nota3, nota4: real
10  media: real
11
12 Inicio
13 escreval ("Qual é o seu nome mesmo?")
14 leia (aluno)
15 escreval ("digite a primeira nota")
16 leia (nota1)
17 escreval ("digite sua segunda nota")
18 leia (nota2)
19 escreval ("digite sua terceira nota")
20 leia (nota3)
21 escreval ("digite sua quarta nota")
22 leia (nota4)
23 media <- (nota1 + nota2 + nota3 + nota4) / 4
24 escreval (aluno.", sua média é ".media)
25
26
27 Fimalgoritmo
```



## Cuidados de lógica de programação

O programa é sempre executado exatamente do jeito que você escreveu, de cima para baixo.

Isso significa que a ordem faz diferença.

E isso também significa que podem surgir alguns problemas que a gente não está acostumado

### **Exemplo:**

Modo de usar – Shampoo

1. Aplicar sobre os cabelos úmidos
2. Massagear suavemente o couro cabeludo
3. Enxaguar
4. Repetir





# O que você precisa aprender?

Vamos aprender nesse curso toda a base de lógica de programação e algoritmo que você precisa, independente da linguagem de programação

## **Ementa:**

1. Variáveis e Constantes
2. Tipos de Dados
  1. inteiro -> números inteiros
  2. float -> números “ponto flutuante” (ou seja, com casa decimal)
  3. string-> textos
  4. boolean -> Verdadeiro ou Falso
3. Estruturas de Dados / Coleções
  1. Listas -> [1, 2, 3, 4, 5]
  2. Dicionários -> {“Iphone”: 5000, “Ipad”: 7000, “Airpod”: 2000}
4. Operadores
  1. Matemáticos
  2. Lógicos (e, ou)
  3. Comparativos (maior que, menor que)



## O que você precisa aprender?

Vamos aprender nesse curso toda a base de lógica de programação e algoritmo que você precisa, independente da linguagem de programação

### **Ementa:**

1. Variáveis e Constantes
2. Tipos de Dados
3. Estruturas de Dados / Coleções
4. Operadores
5. Estruturas Condicionais
  1. If, else -> se chover, então não sai de casa. Caso contrário, sai de casa
6. Estruturas de repetição
  1. For -> Faça isso X vezes
  2. While -> Faça isso enquanto...

## Exemplo 1

Construa um programa que calcule: Quanto devemos cobrar em um projeto de programação se trabalhamos 8h por dia, demoramos 15 dias para fazer o projeto e cobramos R\$100/h?



## Como resolver problemas de programação?

Sempre tenha 3 perguntas em mente:

1. Qual o resultado que você quer?
2. O que eu preciso de informação para chegar nesse resultado?
3. Qual o passo a passo para fazer o que você quer?



## Aplicando no exemplo

Construa um programa que calcule: Quanto devemos cobrar em um projeto de programação se trabalhamos 8h por dia, demoramos 15 dias para fazer o projeto e cobramos R\$100/h?

1. Qual o resultado que você quer?  
Saber o preço que devo cobrar pelo projeto
2. O que eu preciso de informação para chegar nesse resultado?
  - Quanto tempo o projeto vai demorar (quantas horas de trabalho vou dedicar ao projeto)
  - Quanto custa a minha hora de trabalho
3. Qual o passo a passo para fazer o que você quer?
  - Passo 1: Descobrir quantas horas de trabalho vamos usar no projeto
  - Passo 2: Pegar o custo de uma hora de trabalho e multiplicar pelo resultado do passo 1
  - Passo 3: Exibir o resultado final do custo do projeto



## Vamos agora criar o que seria um código disso

Construa um programa que calcule: Quanto devemos cobrar em um projeto de programação se trabalhamos 8h por dia, demoramos 15 dias para fazer o projeto e cobramos R\$100/h?

Qual o passo a passo para fazer o que você quer?

**Passo 1:** Descobrir quantas horas de trabalho vamos usar no projeto

`horas_por_dia = 8`

`dias_totais = 15`

`horas_trabalho = horas_por_dia * dias_totais`

**Passo 2:** Pegar o custo de uma hora de trabalho e multiplicar pelo resultado do passo 1

`custo_hora = 100`

`custo_total = horas_trabalho * custo_hora`

**Passo 3:** Exibir o resultado final do custo do projeto

`exibir(custo_total)`



## Vamos agora criar o que seria um código disso

Seu pseudocódigo:

```
horas_por_dia = 8
dias_totais = 15
horas_trabalho = horas_por_dia * dias_totais
custo_hora = 100
custo_total = horas_trabalho * custo_hora
exibir(custo_total)
```

Código real:

```
horas_por_dia = 8
dias_totais = 15
horas_trabalho = horas_por_dia * dias_totais
custo_hora = 100
custo_total = horas_trabalho * custo_hora
print(custo_total)
```

Resultado:

12000

## Exemplo 2

Construa um programa que calcule: Quanto custa encher o tanque de um carro que tem 50 litros de capacidade, está com 20 litros de combustível atualmente e o custo do combustível é de R\$5,80/litro?





## Aplicando no exemplo

Construa um programa que calcule: Quanto custa encher o tanque de um carro que tem 50 litros de capacidade, está com 20 litros de combustível atualmente e o custo do combustível é de R\$5,80/litro?

1. Qual o resultado que você quer?  
Saber o preço para encher o tanque, em R\$
2. O que eu preciso de informação para chegar nesse resultado?
  - Qual a capacidade do tanque
  - Quanto custa 1 litro de combustível
  - Quantos litros faltam para encher o tanque
3. Qual o passo a passo para fazer o que você quer?
  - Passo 1: Descobrir quantos litros faltam para encher o tanque
  - Passo 2: Pegar o custo de 1 litro e multiplicar pelo resultado do passo 1
  - Passo 3: Exibir o resultado final do custo para encher o tanque



## Aplicando no exemplo

Construa um programa que calcule: Quanto custa encher o tanque de um carro que tem 50 litros de capacidade, está com 20 litros de combustível atualmente e o custo do combustível é de R\$5,80/litro?

Qual o passo a passo para fazer o que você quer?

**Passo 1: Descobrir quantos litros faltam para encher o tanque**

`capacidade_tanque = 50`

`volume_atual = 20`

`litros_encher = capacidade_tanque - volume_atual`

**Passo 2: Pegar o custo de 1 litro e multiplicar pelo resultado do passo 1**

`custo_litro = 5.80`

`custo_total = litros_encher * custo_litro`

**Passo 3: Exibir o resultado final do custo para encher o tanque**

`exibir(custo_total)`



## Vamos agora criar o que seria um código disso

Seu pseudocódigo:

```
capacidade_tanque = 50  
volume_atual = 20  
litros_encher = capacidade_tanque - volume_atual  
custo_litro = 5.80  
custo_total = litros_encher * custo_litro  
exibir(custo_total)
```

Código real:

```
capacidade_tanque = 50  
volume_atual = 20  
litros_encher = capacidade_tanque - volume_atual  
custo_litro = 5.80  
custo_total = litros_encher * custo_litro  
print(custo_total)
```

Resultado:

174.0



## E se fôssemos encher com diesel, a R\$5,03 o litro?

Seu pseudocódigo:

```
capacidade_tanque = 50
volume_atual = 20
litros_encher = capacidade_tanque - volume_atual
custo_litro = 5.03
custo_total = litros_encher * custo_litro
exibir(custo_total)
```

Código real:

```
capacidade_tanque = 50
volume_atual = 20
litros_encher = capacidade_tanque - volume_atual
custo_litro = 5.03
custo_total = litros_encher * custo_litro
print(custo_total)
```

Resultado:

**150.9**

## Exemplo 3

Construa um programa que calcule: Qual deve ser o bônus de um funcionário? Se ele vendeu mais de 1.000 unidades, o bônus tem que ser de R\$250, caso contrário, o bônus tem que ser R\$50



## Aplicando no exemplo

Construa um programa que calcule: Qual deve ser o bônus de um funcionário? Se ele vendeu mais de 1.000 unidades, o bônus tem que ser de R\$250, caso contrário, o bônus tem que ser R\$50

1. Qual o resultado que você quer?  
O valor do bônus do funcionário
2. O que eu preciso de informação para chegar nesse resultado?
  - Quanto ele vendeu
  - Qual a meta de vendas
  - Quanto ele ganha se bateu a meta
  - Quanto ele ganha se não bateu a meta

3. Qual o passo a passo para fazer o que você quer?

Passo 1: Definir quanto o funcionário vendeu e a meta de vendas

Passo 2: Analisar, se ele vendeu mais do que a meta

Passo 3: Exibir definir bônus quando bate a meta

Passo 4: Caso contrário, definir bônus quando não bate a meta

Passo 5: Exibir bônus

## Aplicando no exemplo

Construa um programa que calcule: Qual deve ser o bônus de um funcionário? Se ele vendeu mais de 1.000 unidades, o bônus tem que ser de R\$250, caso contrário, o bônus tem que ser R\$50

Passo 1: Definir quanto o funcionário vendeu e a meta de vendas  
vendas = 1200  
meta = 1000

Passo 2: Analisar, se ele vendeu mais do que a meta  
Se vendas > meta então

Passo 3: Definir bônus quando bate a meta  
bonus = 250

Passo 4: Caso contrário, definir bônus quando não bate a meta  
Caso contrário  
bonus = 50

Passo 5: Exibir bônus  
exibir(bonus)



## Aplicando no exemplo

Construa um programa que calcule: Qual deve ser o bônus de um funcionário? Se ele vendeu mais de 1.000 unidades, o bônus tem que ser de R\$250, caso contrário, o bônus tem que ser R\$50

Passo 1: Definir quanto o funcionário vendeu e a meta de vendas  
vendas = 1200  
meta = 1000

Passo 2: Analisar, se ele vendeu mais do que a meta  
Se vendas > meta então

Passo 3: Definir bônus quando bate a meta  
bonus = 250

Passo 4: Caso contrário, definir bônus quando não bate a meta  
Caso contrário **Caso contrário do que?**  
bonus = 50

Passo 5: Exibir bônus  
exibir(bonus) **Isso aqui deve rodar nos 2 casos? Como o computador sabe disso?**





## Aplicando no exemplo

Construa um programa que calcule: Qual deve ser o bônus de um funcionário? Se ele vendeu mais de 1.000 unidades, o bônus tem que ser de R\$250, caso contrário, o bônus tem que ser R\$50

### Opção 1:

vendas = 1200  
meta = 1000

```
Se vendas > meta então {  
  bonus = 250  
}  
Caso contrário {  
  bonus = 50  
}  
exibir(bonus)
```

### Opção 2:

vendas = 1200  
meta = 1000

```
Se vendas > meta então  
  bonus = 250  
Caso contrário  
  bonus = 50  
  
exibir(bonus)
```



## Aplicando no exemplo

### Seu pseudocódigo:

```
vendas = 1200  
meta = 1000
```

```
Se vendas > meta então {  
    bonus = 250  
}  
Caso contrário {  
    bonus = 50  
}  
exibir(bonus)
```

### Código real:

```
vendas = 1200  
meta = 1000  
if vendas > meta:  
    bonus = 250  
else:  
    bonus = 50  
print(bonus)
```

250

```
vendas = 700  
meta = 1000  
if vendas > meta:  
    bonus = 250  
else:  
    bonus = 50  
print(bonus)
```

50



## Exemplo 4

Construa um programa que calcule: Qual deve ser o bônus de um funcionário? Se a empresa bateu a meta de 10.000 vendas E se ele vendeu mais de 1.000 unidades, o bônus tem que ser de R\$250, caso contrário, o bônus tem que ser R\$50



## Aplicando no exemplo

Construa um programa que calcule: Qual deve ser o bônus de um funcionário? Se a empresa bateu a meta de 10.000 vendas E se ele vendeu mais de 1.000 unidades, o bônus tem que ser de R\$250, caso contrário, o bônus tem que ser R\$50

1. Qual o resultado que você quer?  
O valor do bônus do funcionário
2. O que eu preciso de informação para chegar nesse resultado?
  - Quanto ele vendeu e quanto a empresa vendeu
  - Qual a meta de vendas dele e da empresa
  - Quanto ele ganha se bateu a meta
  - Quanto ele ganha se não bateu a meta

3. Qual o passo a passo para fazer o que você quer?

Passo 1: Definir quanto o funcionário vendeu e a meta de vendas

Passo 2: Definir quanto a empresa vendeu e a meta de vendas

Passo 3: Analisar, se ele vendeu mais do que a meta e AO MESMO TEMPO a empresa bateu a meta

Passo 4: Definir bônus quando bate a meta

Passo 5: Caso contrário, definir bônus quando não bate a meta

Passo 6: Exibir bônus



## Aplicando no exemplo

Passo 1: Definir quanto o funcionário vendeu e a meta de vendas

vendas = 1200

meta = 1000

Passo 2: Definir quanto a empresa vendeu e a meta de vendas

vendas\_empresa = 11000

meta\_empresa = 10000

Passo 3: Analisar, se ele vendeu mais do que a meta e AO MESMO TEMPO a empresa bateu a meta

Passo 4: Definir bônus quando bate a meta

Se vendas > meta E vendas\_empresa > meta\_empresa então {

bonus = 250

}

Passo 5: Caso contrário, definir bônus quando não bate a meta

Caso contrario{

bonus = 50

}

Passo 6: Exibir bônus

exibir(bonus)



## Aplicando no exemplo

```
vendas = 1200
meta = 1000
vendas_empresa = 11000
meta_empresa = 10000
Se vendas > meta E vendas_empresa > meta_empresa então {
    bonus = 250
}
Caso contrario{
    bonus = 50
}
exibir(bonus)
```

```
vendas = 1200
meta = 1000
vendas_empresa = 11000
meta_empresa = 10000
if vendas > meta and vendas_empresa > meta_empresa:
    bonus = 250
else:
    bonus = 50
print(bonus)
```

250



## Aplicando no exemplo

```
vendas = 800
meta = 1000
vendas_empresa = 11000
meta_empresa = 10000
if vendas > meta and vendas_empresa > meta_empresa:
    bonus = 250
else:
    bonus = 50
print(bonus)
```

50

```
vendas = 1200
meta = 1000
vendas_empresa = 9000
meta_empresa = 10000
if vendas > meta and vendas_empresa > meta_empresa:
    bonus = 250
else:
    bonus = 50
print(bonus)
```

50



## Exemplo 4 – Só que mudando 1 regra

Construa um programa que calcule: Qual deve ser o bônus de um funcionário? Se a empresa bateu a meta de 10.000 vendas E se ele vendeu mais de 1.000 unidades, o bônus tem que ser de R\$250, caso contrário, o bônus tem que ser R\$50. Se a empresa não bateu a meta de 10.000 vendas, o bônus dele tem que ser 0.





## Exemplo 4 – Só que mudando 1 regra

Construa um programa que calcule: Qual deve ser o bônus de um funcionário? Se a empresa bateu a meta de 10.000 vendas E se ele vendeu mais de 1.000 unidades, o bônus tem que ser de R\$250, caso contrário, o bônus tem que ser R\$50. Se a empresa não bateu a meta de 10.000 vendas, o bônus dele tem que ser 0.

Qual o passo a passo para fazer o que você quer?

Opção 1:

Passo 1: Definir quanto o funcionário vendeu e a meta de vendas

Passo 2: Definir quanto a empresa vendeu e a meta de vendas

Passo 3: Analisar, se ele vendeu mais do que a meta e AO MESMO TEMPO a empresa bateu a meta

Passo 4: Definir bônus quando bate a meta

Passo 5: Caso contrário, definir bônus quando não bate a meta

Passo 6: Caso a empresa tenha vendido menos do que a meta, edita o bônus para 0

Passo 7: Exibir bônus



## Aplicando no exemplo

Passo 1: Definir quanto o funcionário vendeu e a meta de vendas

vendas = 1200

meta = 1000

Passo 2: Definir quanto a empresa vendeu e a meta de vendas

vendas\_empresa = 11000

meta\_empresa = 10000

Passo 3: Analisar, se ele vendeu mais do que a meta e AO MESMO TEMPO a empresa bateu a meta

Passo 4: Definir bônus quando bate a meta

Se vendas > meta E vendas\_empresa > meta\_empresa então {  
bonus = 250 }

Passo 5: Caso contrário, definir bônus quando não bate a meta

Caso contrario{  
bonus = 50}

Passo 6: Caso a empresa tenha vendido menos do que a meta, edita o bônus para 0

Se vendas\_empresa < meta\_empresa então {  
bonus=0}

Passo 7: Exibir bônus

exibir(bonus)



## Aplicando no exemplo

```
vendas = 1200
meta = 1000
vendas_empresa = 11000
meta_empresa = 10000
Se vendas > meta E vendas_empresa > meta_empresa então {
    bonus = 250 }
Caso contrario{
    bonus = 50}
Se vendas_empresa < meta_empresa então {
    bonus=0}
exibir(bonus)
```

```
vendas = 1200
meta = 1000
vendas_empresa = 11000
meta_empresa = 10000
if vendas > meta and vendas_empresa > meta_empresa:
    bonus = 250
else:
    bonus = 50
if vendas_empresa < meta_empresa:
    bonus = 0
print(bonus)
```



## Exemplo 4 – Só que mudando 1 regra

Construa um programa que calcule: Qual deve ser o bônus de um funcionário? Se a empresa bateu a meta de 10.000 vendas E se ele vendeu mais de 1.000 unidades, o bônus tem que ser de R\$250, caso contrário, o bônus tem que ser R\$50. Se a empresa não bateu a meta de 10.000 vendas, o bônus dele tem que ser 0.

Qual o passo a passo para fazer o que você quer?

Opção 2:

Passo 1: Definir quanto o funcionário vendeu e a meta de vendas

Passo 2: Definir quanto a empresa vendeu e a meta de vendas

Passo 3: Analisar, se a empresa bateu a meta

Passo 4: Se bateu a meta, analisa se ele vendeu mais do que a meta

Passo 5: Definir bônus quando bate a meta (da empresa e do funcionário)

Passo 6: Caso contrário do passo 4, definir bônus quando não bate a meta do funcionário

Passo 7: Caso contrário do Passo 3, definir bônus como 0

Passo 8: Exibir bônus



## Aplicando no exemplo

Passo 1: Definir quanto o funcionário vendeu e a meta de vendas

vendas = 1200

meta = 1000

Passo 2: Definir quanto a empresa vendeu e a meta de vendas

vendas\_empresa = 11000

meta\_empresa = 10000

Passo 3: Analisar, se a empresa bateu a meta

Se vendas\_empresa > meta\_empresa então {

Passo 4: Se bateu a meta, analisa se ele vendeu mais do que a meta

Se vendas > meta então {

Passo 5: Definir bônus quando bate a meta (da empresa e do funcionário)

bonus = 250}

Passo 6: Caso contrário do passo 4, definir bônus quando não bate a meta do funcionário

Caso contrario{

bonus = 50}

}

Passo 7: Caso contrário do Passo 3, definir bônus como 0

Caso contrario{

bonus = 0}

Passo 8: Exibir bônus

exibir(bonus)



## Aplicando no exemplo – Com indentação

Passo 1: Definir quanto o funcionário vendeu e a meta de vendas

vendas = 1200

meta = 1000

Passo 2: Definir quanto a empresa vendeu e a meta de vendas

vendas\_empresa = 11000

meta\_empresa = 10000

Passo 3: Analisar, se a empresa bateu a meta

Se vendas\_empresa > meta\_empresa então {

    Passo 4: Se bateu a meta, analisa se ele vendeu mais do que a meta

    Se vendas > meta então {

        Passo 5: Definir bônus quando bate a meta (da empresa e do funcionário)

        bonus = 250}

    Passo 6: Caso contrário do passo 4, definir bônus quando não bate a meta do funcionário

    Caso contrario{

        bonus = 50}

}

Passo 7: Caso contrário do Passo 3, definir bônus como 0

Caso contrario{

    bonus = 0}

Passo 8: Exibir bônus

exibir(bonus)



## Aplicando no exemplo – Com indentação

```
vendas = 1200
meta = 1000
vendas_empresa = 11000
meta_empresa = 10000
Se vendas_empresa > meta_empresa então {
    Se vendas > meta então {
        bonus = 250}
    Caso contrario{
        bonus = 50}
}
Caso contrario{
    bonus = 0}
exibir(bonus)
```

```
vendas = 1200
meta = 1000
vendas_empresa = 11000
meta_empresa = 10000
if vendas_empresa > meta_empresa:
    if vendas > meta:
        bonus = 250
    else:
        bonus = 50
else:
    bonus = 0
print(bonus)
```



## Exemplo 4 – Só que mudando 1 regra

Construa um programa que calcule: Qual deve ser o bônus de um funcionário? Se a empresa bateu a meta de 10.000 vendas E se ele vendeu mais de 1.000 unidades, o bônus tem que ser de R\$250, caso contrário, o bônus tem que ser R\$50. Se a empresa não bateu a meta de 10.000 vendas, o bônus dele tem que ser 0.

Qual o passo a passo para fazer o que você quer?

Opção 3:

Passo 1: Definir quanto o funcionário vendeu e a meta de vendas

Passo 2: Definir quanto a empresa vendeu e a meta de vendas

Passo 3: Analisar, se a empresa NÃO bateu a meta

Passo 4: Se sim (realmente não bateu a meta), então o bônus é 0

Passo 5: Caso contrário do passo 3, analisa se ele vendeu mais do que a meta

Passo 6: Definir bônus quando bate a meta (da empresa e do funcionário)

Passo 7: Caso contrário do passo 6, definir bônus quando não bate a meta do funcionário, mas bate a da empresa

Passo 8: Exibir bônus





## Aplicando no exemplo – Com indentação

Passo 1: Definir quanto o funcionário vendeu e a meta de vendas

vendas = 1200

meta = 1000

Passo 2: Definir quanto a empresa vendeu e a meta de vendas

vendas\_empresa = 11000

meta\_empresa = 10000

Passo 3: Analisar, se a empresa NÃO bateu a meta

Se NÃO vendas\_empresa > meta\_empresa então {

    Passo 4: Se sim (realmente não bateu a meta), então o bônus é 0

    bonus=0}

Passo 5: Caso contrário do passo 3, analisa se ele vendeu mais do que a meta

Caso contrário{

    Se vendas > meta então {

        Passo 6: Definir bônus quando bate a meta (da empresa e do funcionário)

        bonus = 250}

        Passo 7: Caso contrário do passo 6, definir bônus quando não bate a meta do funcionário, mas bate a da empresa

        Caso contrário {

            bonus=50}

    }

Passo 8: Exibir bônus

exibir(bonus)



## Aplicando no exemplo – Com indentação

```
vendas = 1200
meta = 1000
vendas_empresa = 11000
meta_empresa = 10000
Se NÃO vendas_empresa > meta_empresa então {
    bonus=0}
Caso contrário{
    Se vendas > meta então {
        bonus = 250}
    Caso contrário {
        bonus=50}
}
exibir(bonus)
```

```
vendas = 1200
meta = 1000
vendas_empresa = 11000
meta_empresa = 10000
if not vendas_empresa > meta_empresa:
    bonus = 0
else:
    if vendas > meta:
        bonus = 250
    else:
        bonus = 50
print(bonus)
```

## Exemplo 5

Construa um programa que calcule: Qual seria o salário de um funcionário após 10 anos se todo ano ele ganhasse 10% de aumento, salário inicial de R\$2.000

## Aplicando no exemplo

Construa um programa que calcule: Qual seria o salário de um funcionário após 10 anos se todo ano ele ganhasse 10% de aumento, salário inicial de R\$2.000

1. Qual o resultado que você quer?  
Salário final do funcionário
2. O que eu preciso de informação para chegar nesse resultado?
  - Qual o aumento que ele recebe
  - Quantos anos ele vai receber aumento
  - Qual o salário inicial
3. Qual o passo a passo para fazer o que você quer?
  - Passo 1: Definir todas as informações que temos
  - Passo 2: Calcular 1 ano de aumento
  - Passo 3: Adaptar o programa para rodar para 10 anos
  - Passo 4: Exibir o resultado final do salário

## Aplicando no exemplo

Construa um programa que calcule: Qual seria o salário de um funcionário após 10 anos se todo ano ele ganhasse 10% de aumento, salário inicial de R\$2.000

Qual o passo a passo para fazer o que você quer?

**Passo 1: Definir todas as informações que temos**

salario = 2000

aumento = 0.1

tempo = 10

**Passo 2: Calcular 1 ano de aumento**

salario = salario \* (1 + aumento)

**Passo 3: Adaptar o programa para rodar para 10 anos**

Para cada ano, até tempo anos {

salario = salario \* (1 + aumento)

}

**Passo 4: Exibir o resultado final do salário**

exibir(salario)



## Aplicando no exemplo

Construa um programa que calcule: Qual seria o salário de um funcionário após 10 anos se todo ano ele ganhasse 10% de aumento, salário inicial de R\$2.000

```
salario = 2000  
aumento = 0.1  
tempo = 10
```

```
Para cada ano, até tempo anos{  
    salario = salario * (1 + aumento)  
}  
exibir(salario)
```

```
salario = 2000  
aumento = 0.1  
tempo = 10  
  
for i in range(tempo):  
    salario = salario * (1 + aumento)  
print(salario)
```

**5187.484**



## Exemplo 6

Construa um programa que calcule: Calcule o contrário agora, quanto tempo demora para esse funcionário chegar em um salário de 10.000 reais?

## Aplicando no exemplo

Construa um programa que calcule: Calcule o contrário agora, quanto tempo demora para esse funcionário chegar em um salário de 10.000 reais?

1. Qual o resultado que você quer?  
Quanto tempo demora para o salário > 10000
2. O que eu preciso de informação para chegar nesse resultado?
  - Qual o aumento que ele recebe
  - Qual o salário inicial
  - Qual a meta que ele quer bater
  - Algo para rastrear quanto tempo passou
3. Qual o passo a passo para fazer o que você quer?
  - Passo 1: Definir todas as informações que temos
  - Passo 2: Calcular 1 ano de aumento
  - Passo 3: Adaptar o programa para rodar enquanto o salário for menor do que a meta
  - Passo 4: Exibir o resultado final de quantos anos foram necessarios



## Aplicando no exemplo

Construa um programa que calcule: Calcule o contrário agora, quanto tempo demora para esse funcionário chegar em um salário de 10.000 reais?

Qual o passo a passo para fazer o que você quer?

**Passo 1: Definir todas as informações que temos**

salario = 2000

aumento = 0.1

tempo = 0

**Passo 2: Calcular 1 ano de aumento**

salario = salario \* (1 + aumento)

**Passo 3: Adaptar o programa para rodar enquanto o salário for menor do que a meta**

Enquanto salario < 10000 faça {

salario = salario \* (1 + aumento)

tempo = tempo + 1

}

**Passo 4: Exibir o resultado final do salário**

exibir(tempo)



## Aplicando no exemplo

```
salario = 2000
aumento = 0.1
tempo = 0
Enquanto salario < 10000 faça {
    salario = salario * (1 + aumento)
    tempo = tempo + 1
}
exibir(tempo)
```

```
salario = 2000
aumento = 0.1
tempo = 0
while salario < 10000:
    salario = salario * (1 + aumento)
    tempo = tempo + 1
print(tempo)
```

17



## Exemplo 7

Construa um programa que calcule: Tenho uma lista de preços de produtos, se fizermos um reajuste de 5% a mais em todos os produtos, como ficam os preços dos produtos?



## Aplicando no exemplo

Construa um programa que calcule: Tenho uma lista de preços de produtos, se fizermos um reajuste de 5% a mais em todos os produtos, como ficam os preços dos produtos?

1. Qual o resultado que você quer?

Lista de preços reajustados

2. O que eu preciso de informação para chegar nesse resultado?

- Lista de preços iniciais

- Valor do reajuste

3. Qual o passo a passo para fazer o que você quer?

Passo 1: Definir a lista de preços iniciais e a nova lista reajustada

Passo 2: Para cada produto na lista

Passo 3: Calcular o preço reajustado e adicionar na lista reajustada

Passo 4: Exibir o resultado final da lista reajustada



## Aplicando no exemplo

Construa um programa que calcule: Tenho uma lista de preços de produtos, se fizermos um reajuste de 5% a mais em todos os produtos, como ficam os preços dos produtos?

Qual o passo a passo para fazer o que você quer?

**Passo 1: Definir a lista de preços iniciais e a nova lista reajustada**

```
lista_precos = [100, 500, 1000, 1500]
```

```
reajuste = 0.05
```

```
lista_reajustada = []
```

**Passo 2: Para cada produto na lista**

Para cada preco na lista\_precos faça{

**Passo 3: Calcular o preço reajustado e adicionar na lista reajustada**

```
novo_preco = preco * (1 + reajuste)
```

```
Adicione novo_preco na lista_reajustada
```

```
}
```

**Passo 4: Exibir o resultado final da lista reajustada**

```
exibir(lista_reajustada)
```

## Aplicando no exemplo

```
lista_precos = [100, 500, 1000, 1500]
reajuste = 0.05
lista_reajustada = []
Para cada preço na lista_preços faça{
    novo_preco = preco * (1 + reajuste)
    Adicione novo_preco na lista_reajustada
}
exibir(lista_reajustada)
```

```
lista_precos = [100, 500, 1000, 1500]
reajuste = 0.05
lista_reajustada = [ ]
for preco in lista_precos:
    novo_preco = preco * (1 + reajuste)
    lista_reajustada.append(novo_preco)
print(lista_reajustada)
```

```
[105.0, 525.0, 1050.0, 1575.0]
```



## Exemplo 8

Construa um programa que calcule: Você tem uma lista de preços de produto, todos os produtos acima de R\$5.000 vão ser reajustados em 5% e todos abaixo de R\$5.000 vão ser reajustados em 10%, como ficam os preços dos produtos?



## Aplicando no exemplo

Construa um programa que calcule: Você tem uma lista de preços de produto, todos os produtos acima de R\$5.000 vão ser reajustados em 5% e todos abaixo de R\$5.000 vão ser reajustados em 10%, como ficam os preços dos produtos?

1. Qual o resultado que você quer?

Lista de preços reajustados

2. O que eu preciso de informação para chegar nesse resultado?

- Lista de preços iniciais
- Valor do reajuste faixa 1 e faixa 2
- Qual o corte entre faixa 1 e faixa 2

3. Qual o passo a passo para fazer o que você quer?

Passo 1: Definir a lista de preços iniciais e a nova lista reajustada

Passo 2: Para cada produto na lista

Passo 3: Calcular o preço reajustado e adicionar na lista reajustada

Passo 4: Se o preço do produto for maior do que o corte de faixa, aplicar reajuste 1

Passo 5: Caso contrário, aplicar reajuste 2

Passo 6: Exibir o resultado final da lista reajustada





## Aplicando no exemplo

Qual o passo a passo para fazer o que você quer?

**Passo 1:** Definir a lista de preços iniciais e a nova lista reajustada

```
lista_precos = [100, 6000, 1000, 1500]
```

```
reajuste_faixa1 = 0.05
```

```
reajuste_faixa2 = 0.1
```

```
corte_faixa = 5000
```

```
lista_reajustada = []
```

**Passo 2:** Para cada produto na lista

Para cada preco na lista\_precos faça{

**Passo 3:** Calcular o preço reajustado e adicionar na lista reajustada

**Passo 4:** Se o preço do produto for maior do que o corte de faixa, aplicar reajuste 2

Se preço > corte\_faixa{

```
    novo_preco = preco * (1 + reajuste_faixa1)}
```

**Passo 5:** Caso contrário, aplicar reajuste 1

Caso contrario{

```
    novo_preco = preco * (1 + reajuste_faixa2)
```

```
}
```

Adicione novo\_preco na lista\_reajustada

```
}
```

**Passo 6:** Exibir o resultado final da lista reajustada

```
exibir(lista_reajustada)
```



## Aplicando no exemplo

```
lista_precos = [100, 6000, 1000, 1500]
reajuste_faixa1 = 0.05
reajuste_faixa2 = 0.1
corte_faixa = 5000
lista_reajustada = []
Para cada preco na lista_precos faça{
    Se preço > corte_faixa{
        novo_preco = preco * (1 + reajuste_faixa1)}
    Caso contrario{
        novo_preco = preco * (1 + reajuste_faixa2)
    }
    Adicione novo_preco na lista_reajustada
}
exibir(lista_reajustada)
```

```
lista_precos = [100, 6000, 1000, 1500]
reajuste_faixa1 = 0.05
reajuste_faixa2 = 0.1
corte_faixa = 5000
lista_reajustada = []
for preco in lista_precos:
    if preco > corte_faixa:
        novo_preco = preco * (1 + reajuste_faixa1)
    else:
        novo_preco = preco * (1 + reajuste_faixa2)
    lista_reajustada.append(novo_preco)
print(lista_reajustada)
```

```
[110.0, 6300.0, 1100.0, 1650.0]
```



## Exemplo 9

Construa um programa para:

- Calcular o salário dos funcionários com reajuste de 10% (a partir de uma lista de salários)
- Calcular o custo total de salários que o RH vai ter q pagar
- Calcular a diferença em R\$ entre o cenário anterior e o cenário atual



## Aplicando no exemplo

Construa um programa para:

- Calcular o salário dos funcionários com reajuste de 10% (a partir de uma lista de salários)
- Calcular o custo total de salários que o RH vai ter q pagar
- Calcular a diferença em R\$ entre o cenário anterior e o cenário atual

1. Qual o resultado que você quer?

Lista de salários reajustados

Custo total de salários a pagar

Diferença do novo custo total para o antigo custo total

2. O que eu preciso de informação para chegar nesse resultado?

- Lista de salários iniciais

- Valor de reajuste

3. Qual o passo a passo para fazer o que você quer?

Passo 1: Calcular custo total de salários

Passo 2: Calcular Reajustes dos Salários

Passo 3: Calcular novo custo total de Salários

Passo 4: Calcular diferença do novo custo total para o antigo custo total

Passo 5: Exibir as informações



## Aplicando no exemplo

Qual o passo a passo para fazer o que você quer?

**Passo 1: Calcular custo total de salários**

```
lista_salarios = [1500, 2500, 2000, 3000]
```

```
custo_total = somar(lista_salarios)
```

```
lista_salarios_reajustados = []
```

```
reajuste = 0.1
```

**Passo 2: Calcular Reajustes dos Salários**

```
Para cada salario na lista_salarios{
```

```
    novo_salario = salario * (1 + reajuste)
```

```
    Adicionar novo_salario na lista_salarios_reajustados
```

```
}
```

**Passo 3: Calcular novo custo total de Salários**

```
novo_custo_total = somar(lista_salarios_reajustados)
```

**Passo 4: Calcular diferença do novo custo total para o antigo custo total**

```
diferenca_custo = novo_custo_total - custo_total
```

**Passo 5: Exibir as informações**

```
exibir(lista_salarios_reajustados)
```

```
exibir(novo_custo_total)
```

```
exibir(diferenca_custo)
```



## Aplicando no exemplo

Qual o passo a passo para fazer o que você quer?

**Passo 1: Calcular custo total de salários**

```
lista_salarios = [1500, 2500, 2000, 3000]
```

```
reajuste = 0.1
```

```
calcular_custo_total(lista_salarios)
```

**Passo 2: Calcular Reajustes dos Salários**

```
calcular_lista_salarios_reajustados(reajuste, lista_salarios)
```

**Passo 3: Calcular novo custo total de Salários**

```
calcular_custo_total(lista_salarios_reajustados)
```

**Passo 4: Calcular diferença do novo custo total para o antigo custo total**

```
calcular_diferença_custo(novo_custo_total, custo_total)
```



## Aplicando no exemplo

```
Funcao calcular_lista_salarios_reajustados(lista_salarios, reajuste) {  
    lista_salarios_reajustada = []  
    Para cada salario na lista_salarios{  
        novo_salario = salario * (1 + reajuste)  
        Adicionar novo_salario na lista_salarios_reajustada  
    }  
    exibir(lista_salarios_reajustada)  
}
```

```
Funcao calcular_custo_total(lista_salarios){  
    total = somar todos os itens(lista_salarios)  
    exibir(total)  
}
```

```
Funcao calcular_diferença_custo(novo_custo_total, custo_total) {  
    diferenca = novo_custo_total - custo_total  
    exibir(diferenca)  
}
```

```
def calcular_lista_salarios_reajustados(lista_salarios, reajuste):  
    lista_salarios_reajustada = []  
    for salario in lista_salarios:  
        novo_salario = salario * (1 + reajuste)  
        lista_salarios_reajustada.append(novo_salario)  
    print(lista_salarios_reajustada)
```

```
def calcular_custo_total(lista_salarios):  
    total = sum(lista_salarios)  
    print(total)
```

```
def calcular_diferença_custo(novo_custo_total, custo_total):  
    diferenca = novo_custo_total - custo_total  
    print(diferenca)
```





**E agora, qual o  
próximo passo?**