

## Zen do Python

- Bonito é melhor que feio.
- Explícito é melhor que implícito.
- Simples é melhor que complexo.
- Complexo é melhor que complicado.
- Linear é melhor do que aninhado.
- Esparso é melhor que denso.
- Legibilidade conta.
- Casos especiais não são especiais o bastante para quebrar as regras.
- Ainda que praticidade vença a pureza.
- Erros nunca devem passar silenciosamente.
- A menos que sejam explicitamente silenciados.
- Diante da ambiguidade, recuse a tentação de adivinhar.
- Dever haver um — e preferencialmente apenas um — modo óbvio para fazer algo.
- Embora esse modo possa não ser óbvio a princípio a menos que você seja holandês.
- Agora é melhor que nunca.
- Apesar de que nunca normalmente é melhor do que exatamente agora.
- Se a implementação é difícil de explicar, é uma má ideia.
- Se a implementação é fácil de explicar, pode ser uma boa ideia.
- Namespaces são uma grande ideia — vamos ter mais dessas!

---

---

## FUNDAMENTOS

---

---

## ▼ Comandos

### ▼ Texto e Número

#### Estruturas condicionais simples

**if** teste:

comando se verdadeiro

**else:**

comando se falso

```
Comando
n1 = 10
n2 = 2
if n1 > n2:
    print(f'{n1} é maior que {n2}')
else:
    print(f'{n1} não é maior que {n2}')
```

```
Saída
10 é maior que 2
```

---

#### Estruturas condicionais compostas

**if** teste:

comando se verdadeiro

**elif** teste:

comando se verdadeiro

**else:**

comando se falso

```
Comando
n1 = 7
n2 = 10
if n1 > n2:
    print(f'{n1} é maior que {n2}')
elif n1 == n2:
    print(f'{n1} é igual a {n2}')
else:
    print(f'{n1} é menor que {n2}')
```

```
Saída
7 é menor que 10
```

---

## MANIPULANDO TEXTO

uma string é uma cadeia de caracteres ou uma cadeia de texto.

### Fatiamento

utilizamos colchetes para fazer o fatiamento, é possível usar de diversas formas o fatiamento.

variável[início:fim]

retorna o texto no intervalo.

Comando

```
frase = 'Curso em Vídeo Python'  
print(frase[9:14])
```

Saída

Vídeo

`variável[início:fim:salto]`

retorna o texto no intervalo declarado dando saltos.

`variável[início]`

retorna somente o texto específico do "início" declarado.

`variável[início:]`

retorna todo o texto, à partir do "início" declarado.

`variável[:fim]`

retorna todo o texto até o "fim" declarado.

`variável[início::salto]`

retorna todo o texto, à partir do "início" declarado, dando saltos.

---

## **len**

`len(variável)`

Retorna o número de caracteres de uma variável.

Comando

```
frase = 'Curso em Vídeo Python'  
print(len(frase))
```

Saída

21

---

### **.count**

`variável.count('texto procurado')`

retorna a quantidade de vezes que se repete o "texto procurado" na "variável".

Comando

```
frase = 'Curso em Vídeo Python'  
print(frase.count('o'))
```

Saída

3

---

`variável.count('texto procurado', início, fim)`

retorna a quantidade de vezes que se repete o "texto procurado" com o "intervalo declarado" na "variável".

---

### **.find**

`variável.find('texto procurado')`

retorna a localização em que inicia o "texto procurado" na "variável".

Comando

```
frase = 'Curso em Vídeo Python'  
print(frase.find('em'))
```

Saída

6

`variável.rfind('texto procurado')`

retorna a localização em que inicia o "texto procurado" na "variável" da direita para esquerda.

---

**in**

`'texto procurado' in variável`

retorna true(verdadeiro) se encontrar o "texto procurado" na "variável".

Comando

```
frase = 'Curso em Vídeo Python'  
print('Curso' in frase)
```

Saída

True

**.replace**

`variável.replace('texto procurado', 'novo texto')`

substitui no lugar do "texto procurado" o "novo texto" na "variável".

Comando

```
frase = 'Curso em Vídeo Python'  
print(frase.replace('Python', 'Android'))
```

Saída

Curso em Vídeo Android

---

### **.upper**

`variável.upper()`

transforma todas as letras da "variável" em maiúsculas.

---

### **.lower**

`variável.lower()`

transforma todas as letras da "variável" em minúsculas.

---

### **.capitalize**

`variável.capitalize()`

transforma todas as letras da "variável" em minúsculas, exceto a primeira letra que transforma em maiúscula.

---

### **.title**

`variável.title()`

transforma todas as palavras da "variável", deixa a primeira letra em maiúscula e as demais em minúsculas.

---

### **.strip**

`variável.strip()`

remove todos espaços da esquerda e da direita na "variável".

`variável.rstrip()`

remove todos espaços da direita na "variável".

`variável.lstrip()`

remove todos espaços da esquerda na "variável".

---

### **.split**

`variável.split()`

gera uma "lista" com cada palavra da "variável", separando pelo espaço, cada palavra é colocada em uma lista.

Comando

```
texto = "Teste do Python"
x = texto.split()
print(x)
```

Saída

```
['Teste', 'do', 'Python']
```

---

### **.join**



'texto'.join(variável)

retorna valores de uma lista colocando entre cada palavra o ""texto"".

Comando

```
lista = ['John', 'Peter', 'Vicky']  
x = '/'.join(lista)  
print(x)
```

Saída

John/Peter/Vicky

## ▼ Tipos Primitivos e Operadores

### TIPOS PRIMITIVOS

#### Número Inteiro (int)

O tipo inteiro é um tipo composto por caracteres numéricos (algarismos) inteiros.

#### Ponto Flutuante ou Número Decimal (float)

É um tipo composto por caracteres numéricos (algarismo) decimais.

#### String (str)

É um conjunto de caracteres dispostos numa determinada ordem, geralmente utilizada para representar palavras, frases ou textos.

#### Boolean (bool)

Tipo de dado lógico que pode assumir apenas dois valores: falso ou verdadeiro.

#### Tipo Complexo (complex)

Tipo de dado usado para representar números complexos. Esse tipo normalmente é usado em cálculos geométricos e científicos.

### **List (list)**

Listas agrupam um conjunto de elementos variados, podendo conter: inteiros, floats, strings, outras listas e outros tipos.

### **Tuple.**

Assim como Lista, Tupla é um tipo que agrupa um conjunto de elementos.

### **Dictionary (dic)**

Eles são utilizados para agrupar elementos através da estrutura de chave e valor, onde a chave é o primeiro elemento seguido por dois pontos e pelo valor.

---

## **OPERADORES ARITMÉTICOS**

### **Adição (+)**

Realiza a soma entre operandos - Adiciona o sinal de positivo ao número.

### **Subtração (-)**

Realiza a subtração entre operandos - Adiciona o sinal de negativo ao número.

### **Multiplicação (\*)**

Realiza a multiplicação entre operandos.

### **Divisão (/)**

Realiza a divisão entre operandos.

### **Divisão inteira (//)**

Realiza a divisão entre operandos e a parte decimal do resultado.

### **Módulo (%)**

Retorna o resto da divisão entre operandos.

### **Exponenciação ( \*\* )**

Retorna um número elevado a potência de outro.

---

## **OPERADORES DE ATRIBUIÇÃO**

### **Operador "="**

Exemplo:  $x = 1$

Equivalente a:  $x = 1$

### **Operador "+="**

Exemplo:  $x += 1$

Equivalente a:  $x = x + 1$

### **Operador "-="**

Exemplo:  $x -= 1$

Equivalente a:  $x = x - 1$

### **Operador "\*="**

Exemplo:  $x *= 1$

Equivalente a:  $x = x * 1$

### **Operador "/="**

Exemplo:  $x /= 1$

Equivalente a:  $x = x / 1$

### **Operador "%="**

Exemplo:  $x \% = 1$

Equivalente a:  $x = x \% 1$

---

## OPERADORES DE COMPARAÇÃO

### Maior que (>)

Verifica se um valor é maior que outro.

### Menor que (<)

Verifica se um valor é menor que outro.

### Igual a (==)

Verifica se um valor é igual a outro.

### Diferente de (!=)

Verifica se um valor é diferente de outro.

### Maior ou igual a (>=)

Verifica se um valor é maior ou igual a outro.

### Menor ou igual a (<=)

Verifica se um valor é menor ou igual a outro.

---

## OPERADORES LÓGICOS

### and

Retorna True se todas as condições forem verdadeiras, caso contrário retorna False

### or

Retorna True se uma das condições for verdadeiras, caso contrário retorna False

**not**

Inverte o resultado: se o resultado da expressão for True, o operador retorna false

---

## OPERADORES DE IDENTIDADE

**is**

Retorna True se as variáveis comparadas forem o mesmo objeto

**is not**

Retorna True se as variáveis comparadas não forem o mesmo objeto

---

## OPERADORES DE ASSOCIAÇÃO

**in**

Retorna True caso o valor seja encontrado na sequência

**not in**

Retorna True caso o valor não seja encontrado na sequência

## ▼ Módulos e Bibliotecas

### IMPORTAR MÓDULOS

**import** "grupo módulos"

Importa todos os módulos do grupo

**from** "grupo módulos" **import** "módulo específico"

Importa apenas o módulo específico do grupo de módulos

---

## MATH

**ceil** - arredonda para cima um número

**floor** - arredonda para baixo um número

**trunc** - retorna a parte inteira de um número sem modifica-lo

**sqrt** - retorna a raiz quadrada de um número.

**pow** - retona o resultado de uma exponenciação.

**pi** - retorna a constante pi, com várias casas decimais de precisão.

## ▼ Cores no Terminal

\033[estilo;texto;fundom

---



A terminal window with a dark background. On the left, there's a list of ANSI escape codes and their effects: 0 (None), 1 (Bold), 4 (Underline), and 7 (Negative). To the right of this list are two columns of numbers (30-37 and 40-47) each paired with a colored box. On the far right, a video call inset shows a man with glasses. In the center of the terminal, the text '\033[033[0:33:44m' is displayed in a colorful, pixelated font.

Code	Effect	30	40
0	None	30	40
1	Bold	31	41
4	Underline	32	42
7	Negative	33	43
		34	44
		35	45
		36	46
		37	47

\033[033[0:33:44m

```
nome = 'Wictor'

print(f'Olá! Muito prazer em te conhecer, \033[32m{nome} \033[33;41m{nome}')
```

Olá! Muito prazer em te conhecer, Wictor Wictor

```
nome = 'Wictor'
cores = {'limpa': '\033[37m',
        'azul': '\033[34m',
        'amarelo': '\033[33m',
        'pretoebranco': '\033[7;30m'}

print(f"Olá! Muito prazer em te conhecer, {cores['azul']}{nome} {cores['pretoebranco']}!!!")

Olá! Muito prazer em te conhecer, Wictor !!!
```

## ▼ Exercícios

### ▼ Tratando dados e fazendo contas

#### **exercício 1** - deixando tudo pronto

Crie um programa que escreva "Olá, Mundo!" na tela

```
msg = 'Olá, Mundo!'
print(msg)
print('Olá, Mundo!')
```

---

#### **exercício 2** - respondendo ao usuário

Faça um programa que leia o nome de uma pessoa e mostre a mensagem de boas-vindas.



```
nome = input('Digite seu nome: ')
print(f'É um prazer te conhecer {nome}!')
```

---

### **exercício 3** - somando dois números

Crie um programa que leia dois números e mostre a soma entre eles.

```
n1 = int(input('Digite um valor: '))
n2 = int(input('Digite outro valor: '))
s = n1 + n2
print(f'A soma entre {n1} e {n2} é igual a {s}!')
```

---

### **exercício 4** - dessecando uma variável

Faça um programa que leia algo pelo teclado e mostre na tela o seu tipo primitivo e todas as informações possíveis sobre ele.

```
a = input('Digite algo: ')
print('O tipo primitivo desse valor é ', type(a))
print('Só tem espaços? ', a.isspace())
print('É um número? ', a.isnumeric())
print('É alfabético? ', a.isalpha())
print('É alfanumérico? ', a.isalnum())
print('Está em maiúscula? ', a.isupper())
print('Está em minúscula? ', a.islower())
print('Está capitalizada? ', a.istitle())
```

---

### **exercício 5** - antecessor e sucessor

Faça um programa que leia um número inteiro e mostre na tela o seu sucessor e seu antecessor.

```
n = int(input('Digite um número: '))
a = n - 1
```

```
s = n + 1
print(f'Analisando o valor {n}, seu antecessor é {a} e o sucessor é {s}')
```

---

### **exercício 6** - dobro, triplo, raiz quadrada

Crie um algoritmo que leia um número e mostre o seu dobro, triplo e raiz quadrada.

```
n = int(input('Digite um número: '))
d = n * 2
t = n * 3
r = n ** (1/2)
print(f'O dobro de {n} vale {d}.')
print(f'O triplo de {n} vale {t}.')
print(f'A raiz quadrada de {n} vale {r:.2f}.')
# ou
print(f'O dobro de {n} vale {d}. \nO triplo de {n} vale {t}. \nA raiz quadrada de {n} vale {r:.2f}.')
```

---

### **exercício 7** - média aritmética

Desenvolva um programa que leia as duas notas de um aluno, calcule e mostre a sua média.

```
n1 = float(input('Primeira nota do aluno: '))
n2 = float(input('Segunda nota do aluno: '))
m = (n1 + n2) / 2
print(f'A média entre {n1} e {n2} é igual a {m:.2f}')
```

---

### **exercício 8** - conversor de medidas

Escreva um programa que leia um valor em metros e o exiba convertido em centímetros e milímetros.

```
medida = float(input('Uma distância em metros: '))
cm = medida * 100
```

```
mm = medida * 1000
print(f'A medida de {medida}m corresponde a {cm:.2f}cm e {mm:.2f}mm')
```

---

### **exercício 9** - tabuada

Faça um programa que leia um número inteiro qualquer e mostre na tela a sua tabuada.

```
num = int(input('Digite um número para ver sua tabuada: '))
print(f'{num} x 1 = {num*1:.2f}')
print(f'{num} x 2 = {num*2:.2f}')
print(f'{num} x 3 = {num*3:.2f}')
print(f'{num} x 4 = {num*4:.2f}')
print(f'{num} x 5 = {num*5:.2f}')
print(f'{num} x 6 = {num*6:.2f}')
print(f'{num} x 7 = {num*7:.2f}')
print(f'{num} x 8 = {num*8:.2f}')
print(f'{num} x 9 = {num*9:.2f}')
print(f'{num} x 10 = {num*10:.2f}')
```

---

### **exercício 10** - conversor de moedas

Crie um programa que leia quanto dinheiro uma pessoa tem na carteira e mostre quantos dólares ela pode comprar.

```
real = float(input('Quanto dinheiro você tem na carteira? '))
dolar = real / 3.27
print(f'Com R${real} você pode comprar US${dolar:.2f}')
```

---

### **exercício 11** - pintando parede

Faça um programa que leia a largura e a altura de uma parede em metros, calcule a sua área e a quantidade de tinta necessária para pintá-la, sabendo que cada litro de tinta, pinta uma área de 2 metros quadrados.

```
larg = float(input('Largura da parede: '))
alt = float(input('Altura da parede: '))
area = larg * alt
print(f'Sua parede tem a dimensão de {larg}x{alt} e sua área é de {area:.2f} metros quadrados')
tinta = area / 2
print(f'Para pintar essa parede, você precisará de {tinta:.2f}l de tinta.')
```

---

### **exercício 12** - calculando descontos

Faça um algoritmo que leia o preço de um produto e mostre seu novo preço, com 5% de desconto.

```
preco = float(input('Qual é o preço do produto? R$'))
novo = preco - (preco * 5 / 100)
print(f'O produto que custava R${preco}, na promoção com desconto de 5% vai custar R${novo:.2f}')
```

---

### **exercício 13** - reajuste salarial

Faça um algoritmo que leia o salário de um funcionário e mostre seu novo salário, com 15% de aumento.

```
salário = float(input('Qual é o salário do funcionário? R$'))
novo = salário + (salário * 15 / 100)
print(f'Um funcionário que ganhava R${salário:.2f}, com 15% de aumento, passa a receber R${novo:.2f}')
```

---

### **exercício 14** - conversor de temperaturas

Escreva um programa que converta uma temperatura digitada em °C e converta em °F.

```
c = float(input('Informe a temperatura em °C: '))
f = ((9 * c) / 5) + 32
print(f'A temperatura em {c}°C corresponde a {f}°F!')
```

---

### exercício 15 - aluguel de carros

Escreva um programa que pergunte a quantidade de km percorridos por um carro alugado e a quantidade de dias pelos quais ele foi alugado. Calcule o preço a pagar, sabendo que o carro custa R\$60 por dia e R\$0,15 por km rodado

```
dias = int(input('Quantos dias alugados? '))
km = float(input('Quantos km rodados? '))
pago = (dias * 60) + (km * 0.15)
print(f'O total a pagar é de R${pago:.2f}')
```

## ▼ Usando módulos

### exercício 16 - quebrando um número

Crie um programa que leia um número Real qualquer pelo teclado e mostre na tela a sua porção inteira.

```
from math import trunc
num = float(input('Digite um valor: '))
print(f'O valor digitado foi {num} e a sua porção inteira é {trunc(num)}')
```

---

### exercício 17 - catetos e hipotenusa

Faça um programa que leia o comprimento do cateto oposto e do cateto adjacente de um triângulo retângulo, calcule e mostre o comprimento da hipotenusa.

```
from math import hypot
co = float(input('Comprimento do cateto oposto: '))
ca = float(input('Comprimento do cateto adjacente: '))
hi = hypot(co, ca)
print(f'A hipotenusa vai medir {hi:.2f}')
```

---

### **exercício 18** - seno, cosseno e tangente

Faça um programa que leia um ângulo qualquer e mostre na tela o valor do seno, cosseno e tangente desse ângulo

```
from math import radians, sin, cos, tan
angulo = float(input('Digite o ângulo que você deseja: '))
seno = sin(radians(angulo))
cosseno = cos(radians(angulo))
tangente = tan(radians(angulo))
print(f'O ângulo de {angulo} tem o SENO de {seno:.2f}')
print(f'O ângulo de {angulo} tem o COSSENO de {cosseno:.2f}')
print(f'O ângulo de {angulo} tem a TANGENTE de {tangente:.2f}')
```

---

### **exercício 19** - sorteando um item na lista

Um professor quer sortear um dos seus quatro alunos para apagar o quadro. Faça um programa que ajude ele, lendo o nome deles e escrevendo o nome do escolhido

```
from random import choice
n1 = str(input('Primeiro aluno: '))
n2 = str(input('Segundo aluno: '))
n3 = str(input('Terceiro aluno: '))
n4 = str(input('Quarto aluno: '))
lista = [n1, n2, n3, n4]
escolhido = choice(lista)
print(f'O aluno escolhido foi {escolhido}')
```

---

### **exercício 20** - sorteando uma ordem na lista

O mesmo professor do desafio anterior quer sortear a ordem de apresentação de trabalhos dos alunos. Faça um programa que leia o nome dos quatro alunos e mostre a ordem sorteada.

```
from random import shuffle
n1 = str(input('Primeiro aluno: '))
n2 = str(input('Segundo aluno: '))
n3 = str(input('Terceiro aluno: '))
n4 = str(input('Quarto aluno: '))
lista = [n1, n2, n3, n4]
shuffle(lista)
print('A ordem de apresentação será ')
print(lista)
```

---

### **exercício 21** - tocando um MP3 - inútil no momento

Faça um programa em python que abra e reproduza o áudio de um arquivo MP3.

```
import pygame
pygame.init()
pygame.mixer.music.load('ex021.mp3')
pygame.mixer.music.play()
pygame.event.wait()
```

---

### **exercício 22** - analisador de textos

Crie um programa que leia o nome completo de uma pessoa e mostre:

o nome com todas as letras maiúsculas.

o nome com todas as letras minúsculas.

quantas letras ao todo(sem considerar espaços).

quantas letras tem o primeiro nome.

```
nome = str(input('Digite o seu nome: '))
print(nome.upper())
```

```
print(nome.lower())
print(len(nome)-nome.count(' '))
nome1 = nome.split()
print(f'Seu primeiro nome é {nome1[0]} e ele tem {len(nome1[0])} letras')
```

---

### **exercício 23** - separando dígitos de um número

Faça um programa que leia um número de 0 a 9999 e mostre na tela cada um dos dígitos separados.

Ex: Digite um número: 1834

unidade: 4

dezena: 3

centena: 8

milhar: 8

```
num = int(input('Digite um número: '))
num2 = str(num)
print(f'Unidade: {num2[-1]}')
print(f'Dezena: {num2[-2]}')
print(f'Centena: {num2[-3]}')
print(f'Milhar: {num2[-4]}')
```

---

### **exercício 24** - verificando as primeiras letras de um texto

Crie um programa que leia o nome de uma cidade e diga se ela começa ou não com o nome "Santo".

```
cidade = str(input('Digite uma cidade: '))
print(cidade[:5].upper() == 'SANTO')
```

---



### **exercício 25** - procurando uma string dentro de outra

Crie um programa que leia o nome de uma pessoa e diga se ela tem "Silva" no nome.

```
nome = str(input('Digite o seu nome: '))
print('silva' in nome.lower())
```

---

### **exercício 26** - primeira e última ocorrência de uma string

Faça um programa que leia uma frase pelo teclado e mostre:

quantas vezes aparece a letra "a".

em que posição ela aparece a primeira vez.

em que posição ela aparece a última vez.

```
frase = str(input('Digite uma frase:')).upper()
print(f'A letra A aparece {frase.count("A")} vezes')
print(f'A letra A aparece pela primeira vez na posição {frase.find("A")+1}')
print(f'A letra A aparece pela última vez na posição {frase.rfind("A")+1}')
```

---

### **exercício 27** - primeiro e último nome de uma pessoa

Fala um programa que leia o nome completo de uma pessoa, mostrando em seguida o primeiro e o último nome separadamente.

Ex: Ana Maria de Souza

Primeiro = Ana

Último = Souza

```
nome = str(input('Digite o seu nome: '))
nome1 = nome.split()
```

```
print(f'Primeiro = {nome1[0]}')
print(f'Último = {nome1[-1]}')
```

## ▼ Condições if e else

### **exercício 28** - jogo da adivinhação v.1.0

Escreva um programa que faça o computador "pensar" em um número inteiro entre 0 e 5 e peça para o usuário tentar descobrir qual foi o número escolhido pelo computador. O programa deverá escrever na tela se o usuário venceu ou perdeu.

```
from random import randint
from time import sleep
computador = randint(0, 5)
print('-' * 20)
print('Vou pensar em um número entre 0 e 5. Tente adivinhar...')
print('-' * 20)
jogador = int(input('Em que número eu pensei? '))
print('PROCESSANDO...')
sleep(3)
if jogador == computador:
    print('PARABÉNS! Você conseguiu me vencer!')
else:
    print(f'GANHEI! Eu pensei no número {computador} e não no {jogador}')
```

---

### **exercício 29** - radar eletrônico

Escreva um programa que leia a velocidade de um carro. Se ele ultrapassar 80km/h, mostre uma mensagem dizendo que ele foi multado.

A multa vai custar R\$7,00 por cada km acima do limite.

```
velocidade = float(input('Qual é a velocidade atual do carro? '))
if velocidade > 80:
```

```
print('MULTADO! Você excedeu o limite permitido que é de 80km/h')
multa = (velocidade-80) * 7
print(f'Você deve pagar uma multa de R${multa:.2f}')
print('Tenha um bom dia! Dirija com segurança!')
```

---

### **exercício 30** - par ou impar?

Crie um programa que leia um número inteiro e mostre na tela se ele é PAR ou IMPAR

```
numero = int(input('Me diga um número qualquer! '))
resultado = numero % 2
if resultado == 0:
    print(f'O número {numero} é PAR')
else:
    print(f'O número {numero} é IMPAR')
```

---

### **exercício 31** - custo da viagem

Desenvolva um programa que pergunte a distância de uma viagem em km. Calcule o preço da passagem, cobrando 0,50 reais por km para viagens de até 200km e 0,45 reais para viagens mais longas.

```
distancia = float(input('Qual é a distância da sua viagem? '))
print(f'Você está prestes a começar uma viagem de {distancia}km.')
preco = distancia * 0.5 if distancia <= 200 else distancia * 0.45
print(f'E o preço da sua passagem será de R${preco:.2f}')
```

---

### **exercício 32** - ano bissexto

Faça um programa que leia um ano qualquer e mostre se ele é BISSEXTO.

```
from datetime import date
ano = int(input('Que ano quer analisar? Coloque 0 para analisar o ano atual: '))
if ano == 0:
    ano = date.today().year
if ano % 4 == 0 and ano % 100 != 0 or ano % 400 == 0:
    print(f'O ano {ano} é BISSEXTO')
else:
    print(f'O ano {ano} não é BISSEXTO')
```

---

### **exercício 33** - maior ou menor valores

Faça um programa que leia três números e mostre qual é o maior e qual é o menor.

```
a = int(input('Primeiro valor: '))
b = int(input('Segundo valor: '))
c = int(input('Terceiro valor: '))
menor = a
if b < a and b < c:
    menor = b
if c < a and c < b:
    menor = c
maior = a
if b > a and b > c:
    maior = b
if c > a and c > b:
    maior = c
print(f'O menor valor digitado foi {menor}')
print(f'O maior valor digitado foi {maior}')
```

---

```
salario = float(input('Qual é o salário do funcionário? R$'))
if salario <= 1250:
    novo = salario + (salario * 15 / 100)
else:
    novo = salario + (salario * 10 / 100)
print(f'Quem ganhava R${salario:.2f} passa a ganhar R${novo:.2f}')
```

---

### **exercício 35** - analisando triângulo v.1.0

Desenvolva um programa que leia o comprimento de três retas e diga ao usuário se elas podem ou não formar um triângulo.

```
r1 = float(input('Primeiro segmento: '))
r2 = float(input('Segundo segmento: '))
r3 = float(input('Terceiro segmento: '))
if r1 < r2 + r3 and r2 < r1 + r3 and r3 < r1 + r2:
    print('Os segmentos acima PODEM FORMAR triângulo!')
else:
    print('Os segmentos acima NÃO PODEM FORMAR triângulo!')
```

•  
•  
•  
•  
•