

Victor Mesquita Lima de Santana

**Mundo 1 de Python:**  
**Curso Em Vídeo**



São Paulo

2021

## História da Linguagem Python

Amsterdan, Holanda – 1982

Ali ficava o CWI (centro de matemática e computação, onde criavam-se algumas linguagens de programação)

Autor da linguagem ABC – Guido Van Rossum (a grosso modo, ABC foi precursor do Python – com dificuldades na área, ele resolveu criar uma linguagem simples e eficaz para a resolução de seus problemas).

2001 – Python Software Foundation (ONG mantenedora da linguagem)

S.O. que utilizam Python:

- MacOS
- Linux
- NetBSD
- Amiga

Raspberry Pi – Mini Computador que utiliza a linguagem como base para seu funcionamento (quando se instala Minecraft Pi, é possível aprender a linguagem ao mesmo tempo que joga).



*Guido Van Rossum – autor da linguagem Python.*

## Propósito Geral da Linguagem

- Pode-se usar para qualquer coisa, diferente de outras linguagens
- Fácil e intuitiva, quando comparada com outras
- Multiplataformas (Mobile, Web, Desktop, ...)
- “ *Batteries included*” (vem com a maioria das coisas instaladas)
- Totalmente gratuita e livre
- Linhas de códigos organizadas
- Orientada a objetos

## Principais Áreas de Atuação

Como ela comporta números gigantescos, ela acaba sendo boa para a realização de diversos cálculos complexos, podendo se destacar, dessa forma, em áreas como:

- Inteligência Artificial
- Biotecnologia
- Computação 3D
- Engenharia
- Matemática
- Ciência de Dados



## Quem utiliza Python?

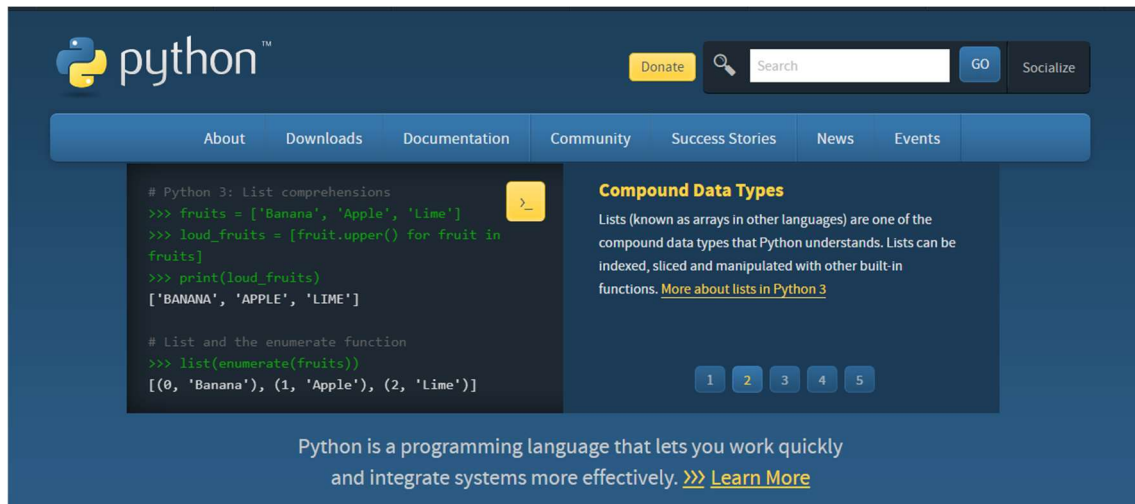
- Zope
- Air Canada
- BitTorrent
- Globo
- Google
- Youtube
- NASA
- Industrial Light & Magic (fazem a renderização de Star Wars, por exemplo)
- Maya, Blender, Gimp, Raspberry, Arduino também utilizam
- Games como Civilization IV, Battlefield 2, Frets on Fire, Jewel Quest, EVE Online



## Instalações e Funcionalidades

Para baixar basta acessar o site oficial e baixar a última versão disponível:

<https://www.python.org/>



Após isso, é recomendável baixar um ambiente de desenvolvimento integrado (IDE) para facilitar o contato com a linguagem. Uma IDE amplamente utilizada para Python é o PyCharm, que pode ser localizado no site da JetBrains: <https://www.jetbrains.com/pt-br/pycharm/>



## Variáveis

Variáveis são espaços de memória utilizados para guardar uma informação, como um texto ou um número. São sempre acompanhados com o nome da variável, um sinal de igual e então a informação a ser guardada.

Ex.:

```
nome = 'Mesquitta'
idade = 19
peso = 75.0
```

Contudo, nesse exemplo, as variáveis nome, idade e peso não “variam” realmente ao longo do código. Elas estão pré-determinadas e para alterarem precisam ser trocadas por outra informação no código-fonte. Porém, há uma maneira de possibilitar que o usuário escolha que tipo de informação será anexada a cada variável com a função input.

## Input

A função input permite que uma entrada de dados vinda do teclado seja guardada dentro de uma variável:

Ex.:

Código

```
nome = input("Qual é o seu nome? ")
idade = input("Qual é a sua idade? ")
peso = input("Qual é o seu peso? ")
print(nome, idade, peso)
```

Run:

```
Qual é o seu nome? Victor
Qual é a sua idade? 20
Qual é o seu peso? 75
Victor 20 75
```

Em verde escuro no “run” estão os dados que foram introduzidos.

## Tipos Primitivos Fundamentais

### Integer (Inteiros)

Servem para representar que há números inteiros na função.

Exemplo de inteiros: 7, -4, 0, 9875.

### Float (Decimais)

"Float" vem de ponto flutuante, os que dividem a parte inteira da decimal. Lembrando que o Python é gringo, então o ponto deles simboliza a nossa vírgula. serve para representar que há números reais na função.

Exemplo de Floats: -4.5, 0.076, 15.223, 7.0

### Boolean (Verdade ou Falso)

Só aceita dois valores - True ou False. São escritos sempre com a primeira letra maiúscula.

### Strings (Caracteres)

Serve para representar um conjunto de caracteres, sejam palavras, números, símbolos ou strings vazias (sem nenhum caractere entre elas). São colocados sempre entre aspas, simples ou duplas.

Exemplo de Strings: 'Olá', '7.5', ''

## Type()

Para saber qual é tipo primitivo de algo, usa-se o comando `type()` e dentro dos parênteses, a variável que deseja consultar o tipo primitivo,

Código:

```
a = 0
print(type(a))
```

Run:

```
<class 'int'>
```

## Função Print

A função “`print()`”, que tem o significado “impressão”, permite que o computador possa imprimir um texto na tela do usuário. O texto que queremos receber para exemplificar a função será: Olá, Mundo. Para imprimirmos essa mensagem, escrevemos o código:

```
print("Olá, Mundo!")
```

## Método Format

O “`.format()`” serve para formatar strings, isto é, encaixar no meio das strings valores que não são fixos (variáveis). Por exemplo, se queremos saber o resultado de duas variáveis, podemos executar o código com duas formas diferentes de formatação:

1.

Código:

```
n1 = int(input("Digite o primeiro número: "))
n2 = int(input('Digite o segundo número: '))
soma = n1+n2
print('A soma vale {}'.format(soma))
```

Run:

```
Digite o primeiro número: 5
Digite o segundo número: 10
A soma vale 15.
```

Nesse caso, o `.format` vem após o fechamento das aspas e a variável deve ser colocada dentro dos parênteses. Dentro da string, é necessário abrir e fechar chaves aonde será anexada variável.

2.

Código:

```
n1 = int(input('Digite o primeiro número:'))
n2 = int(input('Digite o segundo número:'))
soma = n1+n2
print(f'A soma vale {soma}')
```

Run:

```
Digite o primeiro número: 10
Digite o segundo número: 20
A soma vale 30.
```

Dessa outra forma, um “f” é colocado atrás String (chamamos essas strings formatadas desse jeito de f-strings). Dentro da string, basta escrever entre chaves o nome da variável a ser utilizada.

## Exercício 001 – Deixando Tudo Pronto

**Crie um programa que escreva “Olá, Mundo!” na tela.**

Código:

```
print("Olá, Mundo!")
```

Run:

```
Olá, Mundo!
```

## Exercício 002 – Respondendo ao Usuário

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

Possíveis Códigos:

```
nome = input('Qual é o seu nome? ')
print('É um prazer te conhecer, {}!'.format(nome))
```

ou

```
nome = input('Qual é o seu nome? ')
print('É um prazer te conhecer, {}'.format(nome))
```

Run:



```
Qual é o seu nome? Victor  
É um prazer te conhecer Victor!
```

## Exercício 003 – Somando Dois Números

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

Código:

```
n1 = int(input('Digite um número: '))  
n2 = int(input('Digite outro: '))  
s = n1 + n2  
print('A soma entre {} e {} é {}'.format(n1, n2, s))
```

Run:

```
Digite um número: 5  
Digite outro: 7  
A soma entre 5 e 7 é 12.
```

## Exercício 004 – Dissecando uma Variável

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

Código:

```
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 maiúsculas? ', a.isupper())  
print('Está em minúsculas? ', a.islower())  
print('Está capitalizada? ', a.istitle())
```

Run:

```
Digite algo: Victor
O tipo primitivo desse valor é <class 'str'>
Só tem espaços? False
É um número? False
É alfabetico? True
É alfanumérico? True
Está em maiúsculas? False
Está em minúsculas? False
Está capitalizada? True
```

```
Digite algo: 12345
O tipo primitivo desse valor é <class 'str'>
Só tem espaços? False
É um número? True
É alfabetico? False
É alfanumérico? True
Está em maiúsculas? False
Está em minúsculas? False
Está capitalizada? False
```

```
Digite algo: ABc123
O tipo primitivo desse valor é <class 'str'>
Só tem espaços? False
É um número? False
É alfabetico? False
É alfanumérico? True
Está em maiúsculas? False
Está em minúsculas? False
Está capitalizada? False
```

## Operadores Aritméticos

### Operadores

Para realizar as operações numéricas, se faz, primeiramente, necessário saber quais são seus símbolos:

- Adição (+):  $5+2==7$
- Subtração (-):  $5-2==3$
- Multiplicação (\*):  $5*2==10$
- Divisão (/):  $5/2==2.5$
- Potência (\*\*):  $5**2==25$
- Radiciação ( $**(1/x)$ ):  $5**(1/2)==2.2360679774...$
- Divisão Inteira (//):  $5//2==2$

- Resto da Divisão ( % ):  $5\%2==1$

**Percebe-se que foi utilizado dois símbolos de igualdade ( == ). Isso porque um símbolo de igualdade só ( = ) na verdade é um sinal de atribuição, que se lê como “recebe”.**

## Ordem de Precedência

Ordem para resolução de operações matemáticas:

1. ( ) : Parênteses;
2. \*\* : Exponenciação;
3. \* / // % : Multiplicação, divisão, divisão inteira e resto de divisão;
4. + - : Adição e Subtração

## Para exemplificar:

Código:

```
n1 = int(input('Digite um valor: '))
n2 = int(input('Digite outro: '))
so = n1 + n2
sub = n1 - n2
m = n1 * n2
d = n1 / n2
di = n1 // n2
e = n1 ** n2
print('A soma é {}, a subtração é {} e a multiplicação é {}.'.format(so, sub, m))
print('A divisão é {:.3f}, a divisão inteira é {} e a exponenciação é {}.'.format(d, di, e))
```

Run:

```
Digite um valor: 3
Digite outro: 2
A soma é 7, a subtração é 3 e a multiplicação é 10.
A divisão é 2.500, a divisão inteira é 2 e a exponenciação é 25.
```

## Quebra e Manutenimento de Linha

### Quebra de Linha

Para realizar uma quebra de linha, basta acrescentar “\n” em qualquer parte do código, como segue o exemplo:

```
print("Testando a \n quebra de linha")
```

```
Testando a
quebra de linha
```

### End=' '

Para realizar um manutenção de linha, basta acrescentar “, end=' ’” em qualquer parte do código, como segue o exemplo:

```
print("Testando o argumento", end=' ')
print("que mantém o código na mesma linha")
```

```
Testando o argumento que mantém o código na mesma linha
```

## Exercício 005 – Antecessor e Sucessor

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

Código:

```
n = int(input('Digite um número: '))
n1 = n + 1
n2 = n - 1
print("O sucessor de {} é {} e o seu antecessor é {}." .format(n, n1, n2))
```

Run:

```
Digite um número: 12
O sucessor de 12 é 13 e o seu antecessor é 11.
```

## Exercício 006 – Dobro, Triplo e Raiz Quadrada

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

Código:

```
n = int(input("Digite um número: "))
d = n * 2
t = n * 3
r = n ** (1/2)
print('O dobro de {} é {}. Já o triplo é {}. E por fim, a raiz quadrada é {:.2f}'.format(n, d, t, r))
```

Run:

```
Digite um número: 4
O dobro de 4 é 8. Já o triplo é 12. E por fim, a raiz quadrada é 2.00.
```

## Exercício 007 – Média Aritmética

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

Código:

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

Run:

```
Primeira nota do aluno: 9.5
Segunda nota do aluno: 7.5
A média entre 9.50 e 7.50 é igual a 8.50.
```

## Exercício 008 – Conversor de Medidas

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

Código:

```
n = float(input('Digite uma distância em metros: '))
print('{}km'.format(n/1000))
print('{}hm'.format(n/100))
print('{}dam'.format(n/10))
print('{}dm'.format(n*10))
print('{}cm'.format(n*100))
print('{}mm'.format(n*1000))
```

Run:

```
Digite uma distância em metros: 800
0.8km
8.0hm
80.0dam
8000.0dm
80000.0cm
800000.0mm
```

## Exercício 009 – Tabuada

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

Código:

```
n = int(input('Digite um número para ver sua tabuada: '))
print('-'*12)
print('{} x 1 = {}'.format(n, n*1))
print('{} x 2 = {}'.format(n, n*2))
print('{} x 3 = {}'.format(n, n*3))
print('{} x 4 = {}'.format(n, n*4))
print('{} x 5 = {}'.format(n, n*5))
print('{} x 6 = {}'.format(n, n*6))
```

```
print('{} x 7 = {}'.format(n, n*7))
print('{} x 8 = {}'.format(n, n*8))
print('{} x 9 = {}'.format(n, n*9))
print('{} x 10 = {}'.format(n, n*10))
print('-'*12)
```

Run:

```
Digite um número para ver sua tabuada: 10
-----
10 x 1 = 10
10 x 2 = 20
10 x 3 = 30
10 x 4 = 40
10 x 5 = 50
10 x 6 = 60
10 x 7 = 70
10 x 8 = 80
10 x 9 = 90
10 x 10 = 100
-----
```

## Exercício 010 – Conversor de Moedas

**Crie um programa que leia quanto dinheiro uma pessoa tem na carteira e mostre quantos dólares, euros e libras ela pode comprar (considerando [US\$1,00 = R\$5.20] // [1,00 € = R\$6,13] // [£1,00 = R\$7,23] – 11/08/2021).**

Código:

```
reais = float(input('Quanto reais você tem na carteira? R$ '))
dolar = reais/5.22
euro = reais/6.13
libra = reais/7.23
print('Com R${:.2f} você pode comprar:\nUS${:.2f}.'.format(reais,
dolar))
print('€{:.2f}.'.format(euro))
print('£{:.2f}.'.format(libra))
```

Run:

```
Quanto reais você tem na carteira? R$ 50
Com R$50.00 você pode comprar:
US$9.58.
€8.16.
£6.92.
```

## Exercício 011 – 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.**

Código:

```
larg = float(input('Largura da parede em metros: '))
alt = float(input('Altura da parede em metros: '))
area = larg * alt
print('Sua parede tem a dimensão de {}mx{}m.'.format(larg, alt))
print('Portanto, sua área é de {}m².'.format(area))
tinta = area/2
print('Para pintar essa parede, são necessários {:.2f}l de tinta.'.format(tinta))
```

Run:

```
Largura da parede em metros: 20
Altura da parede em metros: 30
Sua parede tem a dimensão de 20.0mx30.0m.
Portanto, sua área é de 600.0m².
Para pintar essa parede, são necessários 300.00l de tinta.
```

## Exercício 012 – Calculando Descontos

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

Código:

```
preco = float(input('Qual é o preço do produto? R$'))
desconto = preco * 0.95
print('O produto que custava R${}, na promoção, com desconto de 5%, vai custar R${}.'.format(preco, desconto))
```

Run:

```
Qual é o preço do produto? R$100
```



```
O produto que custava R$80.0, na promoção, com desconto de 5%, vai custar R$76.0.
```

## Exercício 013 – Reajuste Salarial

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

Código:

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

Run:

```
Qual é o salário do funcionário? R$1500
Um funcionário que ganhava R$1500.00, com 15% de aumento, passará a
receber R$1725.00.
```

## Exercício 014 – Conversor de Temperatura

**Escreva um programa que converta uma temperatura digitando em graus Celsius e converta para graus Fahrenheit.**

Código:

```
celsius = float(input('Informe a temperatura em °C: '))
fahrenheit = celsius * 1.8 + 32
print('A temperatura de {:.2f}°C corresponde a
{:.2f}°F!'.format(celsius, fahrenheit))
```

Run:

```
Informe a temperatura em °C: 100
A temperatura de 100.00°C corresponde a 212.00°F!
```

## Exercício 015 – 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.**

Código:

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

Run:

```
Quantos dias alugados? 5
Quantos Km rodados? 10
O total a pagar é de R$301.50.
```

## Módulos

Módulos são conteúdos adicionais de uma determinada biblioteca, dentro de uma linguagem de programação.

Com o comando `import` (nome da biblioteca), você importa uma biblioteca inteira de funcionalidades.

```
import math
```

Já com o `from` (nome da biblioteca) `import`, você importa uma determinada funcionalidade dentre tantas outras.

```
from math import sqrt
```

A biblioteca math é a mais conhecida e usada dentre todas elas. Partes delas já vêm instaladas com a linguagem – por isso já usamos a parte dos operadores. Mas coisas mais complexas dela não, como por exemplo o fatorial.

Exemplos:

- Ceil – arredonda um determinado número pra cima
- Floor – arredonda um determinado número pra baixo
- Trunc – elimina o que tiver depois da vírgula sem arredondar nem nada
- Pow – potencializa um número
- Sqrt – “Square root”: calcular raiz quadrada
- Factorial – calcula o fatorial de um número

Exemplo:

```
from math import sqrt
num = int(input('Digite um número: '))
raiz = sqrt(num)
print('A raiz de um {} é igual a {}'.format(num, raiz))
```

```
Digite um número: 25
A raiz de um 25 é igual a 5.0.
```

Para saber quais são todas as bibliotecas que o próprio Python deixa importar, basta entrar no site: <https://docs.python.org/3/library/index.html>

Agora para saber bibliotecas públicas (criadas pela própria comunidade), esse é o site:

<https://pypi.org>

## Exercício 016 - Utilizando Módulos

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

Código:

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

Run:

```
Digite um valor: 12.5871
O valor digitado foi 12.5871 e a sua porção inteira é 12.
```

## Exercício 017 – 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.**

Código:

```
import math
oposto = float(input('Comprimento do cateto oposto: '))
adjacente = float(input('Comprimento do cateto adjacente: '))
hipotenusa = math.hypot(oposto, adjacente)
print('A hipotenusa irá medir {:.2f}'.format(hipotenusa))
```

Run:

```
Comprimento do cateto oposto: 3
Comprimento do cateto adjacente: 4
A hipotenusa irá medir 5.00.
```

## Exercício 018 – 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.**

Código:

```
import math
angulo = float(input('Digite o ângulo que você deseja: '))
sen = math.sin(math.radians(angulo))
cos = math.cos(math.radians(angulo))
tan = math.tan(math.radians(angulo))
print('Para o ângulo de {:.2f}°:'.format(angulo))
print('O seno é de {:.2f}'.format(sen))
print('O cosseno é de {:.2f}'.format(cos))
print('A tangente é de {:.2f}'.format(tan))
```

Run:

```
Digite o ângulo que você deseja: 90
Para o ângulo de 90.00°:
O seno é de 1.00.
O cosseno é de 0.00.
A tangente é de 16331239353195370.00.
```

## Exercício 019 – 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 dos alunos e escrevendo na tela o nome do escolhido.**

Código:

```
import random
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]
sorteado = random.choice(lista)
print('O aluno escolhido foi {}'.format(sorteado))
```

Run:

```
Primeiro aluno: Victor
Segundo aluno: Paula
Terceiro aluno: Marcelo
Quarto aluno: Roberta
O aluno escolhido foi Roberta.
```

## Exercício 020 – Sorteando uma Ordem na Lista

**O mesmo professor do desafio 19 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.**

Código:

```
import random
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]
ordem = random.sample(lista, 4)
print('A ordem de apresentação será:\n{}'.format(ordem))
```

Run

```
Primeiro aluno: Victor
Segundo aluno: Pedro
Terceiro aluno: Cristiane
Quarto aluno: Flávia
A ordem de apresentação será:
['Pedro', 'Cristiane', 'Flávia', 'Victor'].
```

## Exercício 021 – Tocando um MP3

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

Código:

```
from pygame import mixer # é necessário antes instalar o package
pygame
mixer.init()
mixer.music.load('ex021music.mp3')
mixer.music.play()
input('Agora você escuta? ')
```

Run:

```
Hello from the pygame community.
https://www.pygame.org/contribute.html
Agora você escuta?
```

## Manipulando Textos

Toda string, ou cadeia de texto, é transcrita para a memória do computador ocupando mini pedaços. Cada letra, número, espaço e afins são inumerados de 0 até o número que for necessário para caber o texto. Assim, cada mini pedaço a gente denomina de caractere.

String referência:



```
frase = "Curso em Video Python"
```

Operações mais utilizadas na manipulação de textos:

- **Fatiamento**

```
print(frase[9]) # ele vai retornar o caractere 9, que e a letra 'V'.
print(frase[9:14]) # pega um a menos sempre - 'Vídeo'
print(frase[9:21:2]) # faz o fatiamento e pula de dois em dois -
'VdoPto'
print(frase[:5]) # vai do início da string até o 5º caractere -
'Curso'
print(frase[15:]) # vai do 15º até o final da string - 'Python'
print(frase[9::3]) # começa no 9º caractere e vai até o final da
string e pula de três em três - 'VePh'
```

```
V
Vídeo
VdoPto
Curso
Python
VePh
```

- **Análise**

```
print(len(frase)) # len: comprimento - retorna o total de caracteres
print(frase.count("o")) # conta quantos desses caracteres tem na
string
```

```
print(frase.count("o", 0, 13)) # conta quantos desses caracteres tem
da posição 0 à 13
print(frase.find("deo")) # retorna a posição de início de um
determinado conjunto de caracteres
print(frase.find("android")) # se não encontra, retorna -1
print("Curso" in frase) # retorna True se o conjunto existe na frase
e false caso não
```

- **Transformação**

String referência:



```
print(frase.replace("Python", "Android")) # Substitui a string Python
por Android
print(frase.upper()) # Deixa todas as letras em maiúsculas
print(frase.lower()) # Deixa todas as letras em minúsculas
print(frase.capitalize()) # Deixa tudo minúsculo e a primeira letra
em maiúscula
print(frase.title()) # Cada palavra fica com a primeira letra
maiúsculas.
print(frase.strip()) # Tira os espaços inúteis da string
print(frase.rstrip()) # retira os espaços inúteis à direita da string
print(frase.lstrip()) # retira os espaços inúteis à esquerda da
string
```

```
Curso em Vídeo Android
CURSO EM VÍDEO PYTHON
curso em vídeo python
Curso em vídeo python
Curso Em Vídeo Python
Curso em Vídeo Python
Curso em Vídeo Python
Curso em Vídeo Python
```

- **Divisão**





```
print(frase.split()) # Divisões das strings aonde tem espaços
print('-'.join(frase)) # Agrupa conjuntos de strings com o caractere
dentro dos parênteses
```

```
['Curso', 'em', 'Video', 'Python']
C-u-r-s-o- -e-m- -V-í-d-e-o- -P-y-t-h-o-n
```

## Exercício 022 – Analisador de Textos

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

- O nome com todas as letras maiúsculas e minúsculas.**
- Quantas letras ao todo (sem considerar espaços).**
- Quantas letras tem o primeiro nome.**

Código:

```
nome = str(input('Digite seu nome: ')).strip()
print('Analisando seu nome...')
print('Seu nome em letras maiúsculas é {}'.format(nome.upper()))
print('Seu nome em letras minúsculas é {}'.format(nome.lower()))
print('Seu nome tem ao todo {} letras'.format(len(nome)-nome.count(' ')))
separa = nome.split()
print('Seu primeiro nome é {} e ele tem {} letras'.format(separa[0], len(separa[0])))
```

Run:

```
Digite seu nome: Victor Mesquita
Analisando seu nome...
Seu nome em letras maiúsculas é VICTOR MESQUITA
Seu nome em letras minúsculas é victor mesquita
Seu nome tem ao todo 14 letras
Seu primeiro nome é Victor e ele tem 6 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.**

Código:

```
num = int(input('Informe um número: '))
print('Analisando o número {}'.format(num))
u = num // 1 % 10
d = num // 10 % 10
c = num // 100 % 10
m = num // 1000 % 10
print('Unidade: {}'.format(u))
print('Dezena: {}'.format(d))
print('Centena: {}'.format(c))
print('Milhar: {}'.format(m))
```

Respostas:

```
Informe um número: 12340
Analisando o número 12340
Unidade: 0
Dezena: 4
Centena: 3
Milhar: 2
```

## Exercício 024 – Verificando as Primeiras Letras de um Texto

**Crie um programa que leia o nome de uma cidade diga se ela começa ou não com o nome “SANTO”.**

Código:

```
cid = str(input('Em que cidade você? ').strip())
print(cid[:3].upper() == 'RIO')
```

Run:

```
Em que cidade você? Rio de Janeiro
True
```

## 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.**

Código:

```
nome = str(input('Qual é o seu nome completo? ')).strip()
print('Seu nome tem Silva? {}'.format('SILVA' in nome.upper()))
```

Run:

```
Qual é o seu nome completo? Fulano Beltrano da Silva
Seu nome tem Silva? True
```

,

## 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 e em que posição ela aparece a última vez.**

Código:

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

Run:

```
Digite uma frase: Eu amo tecnologia!
A letra A aparece 2 vezes na frase.
A primeira letra A apareceu na posição 4.
A última letra A apareceu na posição 17.
```

## Exercício 27 – Primeiro e Último Nome de uma Pessoa

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

Código:

```
n = str(input('Digite seu nome completo: ')).strip()
nome = n.split()
print('Seu primeiro nome é {}'.format(nome[0]))
print('Seu último nome é {}'.format(nome[len(nome)-1]))
```

Run:

```
Digite seu nome completo: Fulano Beltrano da Silva
Seu primeiro nome é Fulano
Seu último nome é Silva
```

## Condições em Python

Ifs são comando que possibilitam ao programa seguir fluxos de códigos diferentes a depender das condições estabelecidas.

Para fazer os blocos ficarem dentro da condição, basta apertar a tecla tab, como se tivesse aplicando um parágrafo. Assim, o código ficará dessa forma:

```
carro.siga()
se carro.esquerda()
    carro.siga()
    carro.direita()
    carro.siga()
    carro.direita()
    carro.esquerda()
    carro.siga()
    carro.direita()
    carro.siga()
senão
    carro.siga()
    carro.esquerda()
    carro.siga()
    carro.esquerda()
    carro.siga()
carro.para()
```

```
if carro.esquerda():
    bloco True
else:
    bloco False
```

Exemplo:

Determinando se um carro é velho ou não pela sua idade.

```
tempo = int(input('Quantos anos tem seu carro? '))
if tempo <= 3:
    print('carro novo')
else:
    print('carro velho')
print('--FIM--')
```

Ou dessa forma, de uma maneira simplificada:

```
tempo = int(input('Quantos anos tem seu carro? '))
print('carro novo' if tempo <= 3 else 'carro velho')
print('--FIM--')
```

## Exercício 028 – Jogo da Adivinhação v1.0

**Escreva um programa que faça o computador “pensar” em um número inteiro entre 1 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.**

Código:

```
import random
import time
print('--'*20)
print('Vou pensar em um número entre 0 e 5. Tenta adivinhar...')
print('--'*20)
numero = int(input('Em que número pensei? ')) # Jogador tenta adivinhar.
computador = random.randint(1, 5) # Randomiza o numero que o computador irá pensar.
print('PROCESSANDO...')
time.sleep(3) #Sleep faz o resto do código aparecer depois de tantos segundos.
if random != numero: # != é o símbolo de diferente na matemática.
    print('PERDEU! Eu pensei no número {} e não no número {}!'.format(computador, numero))
else:
    print('GANHOU! Você conseguiu me vencer')
```

Run:

```
-----  
Vou pensar em um número entre 0 e 5. Tenta adivinhar...  
-----  
Em que número pensei? 4  
PROCESSANDO...  
PERDEU! Eu pensei no número 1 e não no número 4!
```

## Exercício 029 – 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.**

Código:

```
vel = int(input('Qual é a sua velocidade? [em km/h] '))  
if vel > 80:  
    print('MULTADO! Você excedeu o limite permitido de 80km/h')  
    print('Você deve pagar uma multa de R${}.00!'.format(7 * (vel -  
80)))  
print('Tenha um bom dia! Dirija com segurança.')
```

Run:

```
Qual é a sua velocidade? [em km/h] 80  
Tenha um bom dia! Dirija com segurança.
```

```
Qual é a sua velocidade? [em km/h] 120  
MULTADO! Você excedeu o limite permitido de 80km/h  
Você deve pagar uma multa de R$280.00!  
Tenha um bom dia! Dirija com segurança.
```

## Exercício 030 – Par ou Ímpar ?

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

Código:

```
num = int(input('Me diga um número qualquer: '))
if (num % 2) > 0:
    print('O número {} é ÍMPAR.'.format(num))
else:
    print('O número {} é PAR.'.format(num))
```

Resultado:

```
Me diga um número qualquer: 10
O número 10 é PAR.
```

```
Me diga um número qualquer: 3
O número 10 é ÍMPAR.
```

## Exercício 031 – Custo da Viagem

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

Código:

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

Run:

```
Qual é a distância da sua viagem? 100
Você está prestes a começar a viagem de 100.00Km.
E o preço da sua passagem será de R$50.00.
```

```
Qual é a distância da sua viagem? 300
Você está prestes a começar a viagem de 300.00Km.
E o preço da sua passagem será de R$135.00.
```

## Exercício 032 – Ano Bissexto

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

Código:

```
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('O ano de {} É BISSEXTO.'.format(ano))
else:
    print('O ano de {} NÃO É BISSEXTO.'.format(ano))
```

Run:

```
Que ano quer analisar? Coloque 0 para analisar o ano atual: 2022
O ano de 2022 NÃO É BISSEXTO
```

```
Que ano quer analisar? Coloque 0 para analisar o ano atual: 2012
O ano de 2012 É BISSEXTO.
```

## Exercício 033 – Maior e Menor Valores



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

Código:

```
a = int(input('Primeiro valor: '))
b = int(input('Segundo valor: '))
c = int(input('Terceiro valor: '))
# Verificando quem é o menor
menor = a
if b < a and b < c:
    menor = b
if c < a and c < b:
    menor = c
# Verificando quem é o maior
maior = a
if b > a and b > c:
    maior = b
if c > a and c > b:
    maior = c
print('O menor valor digitado foi {}'.format(menor))
print('O maior valor digitado foi {}'.format(maior))
```

Run:

```
Primeiro valor: 5
Segundo valor: 10
Terceiro valor: 14
O menor valor digitado foi 5
O maior valor digitado foi 14
```

## Exercício 034 – Aumentos Múltiplos

**Escreva um programa que pergunte o salário de um funcionário e calcule o valor do seu aumento. Para salários superiores a R\$1250,00, calcule um aumento de 10%. Para os inferiores ou iguais, o aumento é de 15%.**

Código:

```
salario = float(input('Qual é o salário do funcionário? R$'))
if salario <= 1250.00:
    aumento = salario * 1.15
else:
    aumento = salario * 1.10
print('Quem ganhava R${:.2f} passa a ganhar R${:.2f}.'.format(salario, aumento))
```

Run:

```
Qual é o salário do funcionário? R$1200
Quem ganhava R$1200.00 passa a ganhar R$1380.00.
```

## Exercício 35 – Analisando Triângulo v1.0

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

Código:

```
print('-=' * 20)
print('Analisador de Triângulos')
print('-=' * 20)
s1 = float(input('Primeiro segmento: '))
s2 = float(input('Segundo segmento: '))
s3 = float(input('Terceiro segmento: '))
if s1 < s2 + s3 and s2 < s1 + s3 and s3 < s1 + s2:
    triangulo = 'PODEM FORMAR'
else:
    triangulo = 'NÃO PODEM FORMAR'
print('Os segmentos acima {} um triângulo.'.format(triangulo))
```

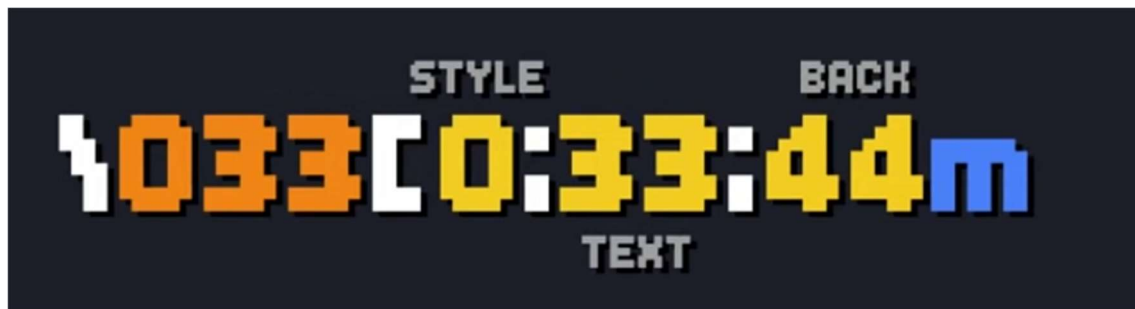
Run:

```
-----
Analisador de Triângulos
-----
Primeiro segmento: 5
Segundo segmento: 6
Terceiro segmento: 8
Os segmentos acima PODEM FORMAR um triângulo.
```

## Cores no Terminal

Esse número limitado de cores é disponibilizado pela funcionalidade ANSI.

E esse é o código para adicionar cores e estilos aos códigos:



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

Exemplo:

```
print('\033[7;30;42mOlá, Mundo!\033[m')
print('\033[1;31;43mOlá, Mundo!\033[m')
print('\033[4;32;42mOlá, Mundo!\033[m')
print('\033[0;33;41mOlá, Mundo!\033[m')
print('\033[4;34;45mOlá, Mundo!\033[m')
print('\033[1;35;47mOlá, Mundo!\033[m')
print('\033[0;36;44mOlá, Mundo!\033[m')
print('\033[7;37;44mOlá, Mundo!\033[m')
print('\033[7;30;42mOlá, Mundo!\033[m')
```

