



SISTEMAS DE
INFORMAÇÃO®



ENGENHARIA
DE SOFTWARE®

APOSTILA DE ESTUDOS

ALGORITMOS E PROGRAMAÇÃO



CIÊNCIA DA
COMPUTAÇÃO®



JOGOS
DIGITAIS®

MADIANITA BOGO MARIOTI
DOUGLAS AQUINO MORENO

SUMÁRIO

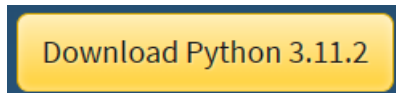
1. INSTRUÇÕES BÁSICAS DO PYTHON.....	3
1.1 Download do Python (para instalar no computador).....	3
1.2 Python on-line.....	3
1.3 Definição de Algoritmos e Programação.....	3
1.4 Definição de Programa, Linguagem de programação e Sintaxe.....	3
1.5 Impressão/Saída.....	4
1.6 Comentários.....	4
1.7 Variáveis.....	4
1.8 Operadores Matemáticos.....	5
1.9 Precedência matemática.....	6
1.10 Leitura/Entrada de dados pelo teclado.....	7
1.11 Exercícios de Fixação.....	8
1.12 Desafios.....	10
1.13 Exercícios de Teste de Mesa.....	11
2. ESTRUTURA DE DECISÃO OU ESTRUTURA CONDICIONAL: IF .. ELIF .. ELSE.....	12
2.1 Operadores Relacionais.....	12
2.2 Estrutura de decisão (condicional) simples - if.....	12
2.3 Estrutura de decisão (condicional) composta – if..else.....	13
2.4 Estrutura de decisão (condicional) encadeada - – if..elif..else.....	14
2.5 Estrutura de decisão (condicional) encadeada.....	16
2.6 Exercícios de Fixação.....	17
2.7 Desafios.....	19

1. INSTRUÇÕES BÁSICAS DO PYTHON

A programação é a arte de escrever instruções para um computador executar tarefas específicas. Essas instruções são escritas usando linguagens de programação, que são conjuntos de comandos compreensíveis tanto para humanos quanto para máquinas.

1.1 Download do Python (para instalar no computador)

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



1.2 Python on-line

<https://repl.it/signup>

1.3 Definição de Algoritmos e Programação

❑ **Algoritmo** - sequência finita de passos para a execução de uma tarefa e resolução de um problema. Eles formam a base do processo de programação.

❑ Um mesmo problema pode ser resolvido, corretamente, de maneiras diferentes.



Por exemplo, existem muitas formas de expressar esse algoritmo ou receita de sanduíche. Uma possível solução poderia ser a seguinte:

1. Pegue duas fatias de pão e coloque-as separadamente sobre o prato.
2. Coloque duas fatias de queijo sobre uma das fatias de pão.
3. Coloque uma fatia de presunto sobre as fatias de queijo.
4. Coloque duas folhas de alface sobre a fatia de presunto.
5. Acrescente uma pitada de sal e um fio de azeite.
6. Coloque a fatia de pão vazia sobre as alfaces.

“Seu sanduíche está pronto!”

1.4 Definição de Programa, Linguagem de programação e Sintaxe

- **Programa:** Grupo de instruções que determinam tarefas a serem executadas por um computador.
 - Os algoritmos e os programas de computadores apresentam a seguinte estrutura:

Entrada □ Processamento □ Saída

- **Linguagem de programação** – É uma linguagem formal que define comandos, símbolos, palavras-chave e regras para a criação de programas. Assim, seguindo as definições das

linguagens, o programador consegue criar programas com instruções que serão “compreendidas” e executadas pelo computador.

- Nós usaremos programas escritos na linguagem Python.
- **Sintaxe** – É o conjunto de regras que define como os comandos devem ser escritos em uma linguagem de programação. Erros de sintaxe podem impedir a execução do programa.

* **Execute os códigos a seguir**

1.5 Impressão/Saída

A impressão ou saída é uma parte fundamental da programação, pois permite que o programa mostre informações para o usuário ou para o desenvolvedor, facilitando a comunicação entre o computador e as pessoas. Em muitas linguagens de programação, incluindo Python, você usa funções específicas para imprimir informações na saída padrão (geralmente a tela).

```
print ('Hello World!!')
```

Neste exemplo, a função “print()” é usada para mostrar uma mensagem na tela.

1.6 Comentários

Comentários são trechos de texto que você pode adicionar em seu código para fornecer explicações, documentação ou anotações. Eles não são executados pelo interpretador ou compilador da linguagem, ou seja, não afetam o funcionamento do programa. Comentários são úteis para explicar o propósito de partes do código, documentar funcionalidades ou fazer anotações para você mesmo ou para outros desenvolvedores que possam ler o código.

Exemplo de Comentário (após o símbolo #):

Em muitas linguagens de programação, incluindo Python, você pode criar comentários colocando um símbolo “#” no início da linha. Tudo que vem depois desse símbolo é tratado como um comentário e é ignorado pelo compilador ou interpretador.

```
#não será considerado  
print ('Hello World!!') #não será considerado
```

Comentários de Múltiplas Linhas (“Comentário”):

Comentários de múltiplas linhas são usados quando você precisa adicionar explicações mais longas que abrangem várias linhas de código. Você pode usar três aspas duplas “""" ou três aspas simples “''' para criar um bloco de comentário.

```
"""  
Este é um comentário de múltiplas linhas.  
Pode se estender por várias linhas sem a necessidade de repetir  
o símbolo.  
"""
```

1.7 Variáveis

- **Variáveis:** armazenam um valor na memória, que pode ser consultado ou alterado em um momento posterior. Cada variável tem um nome único e um valor associado. Os tipos de dados especificam o tipo de informação que uma variável pode conter. Alguns tipos de dados comuns incluem:
 - o **Inteiro (int):** Armazena números inteiros, como -3, 0, 42.
 - o **Flutuante (float):** Armazena números com casas decimais, como 3.14, -0.5.
 - o **Texto (string):** Armazena sequências de caracteres, como "Olá, mundo!".
 - o **Booleano (bool):** Armazena valores verdadeiro (true) ou falso (false).

Código 1 – variável **nome** recebe o conteúdo **Ana**

```
nome = 'Ana'
print (nome)
```

Código 2 – variável **nome** recebe o conteúdo **Ana**

```
nome = 'Ana'
print('Olá', nome)
```

Obs.: altere a impressão para `print('Olá', 'nome')` e verifique que a saída não é a desejada.

Código 3 – variável **nome** recebe o conteúdo **Ana** e a variável **sobrenome** recebe o conteúdo **Silva**

```
nome = 'Ana'
sobrenome = 'Silva'
print('Olá', nome, sobrenome)
```

1.8 Operadores Matemáticos

Os operadores matemáticos são elementos fundamentais na programação, pois permitem que você realize cálculos e operações matemáticas em seus programas. Aqui estão alguns dos operadores matemáticos mais comuns:

Operador	Descrição	Sintaxe	Exemplo
+	Adição	$x+y$	$3+3 \square 6$
-	Subtração	$x-y$	$7-5 \square 2$
*	Multiplicação	$x*y$	$2*3 \square 6$
/	Divisão	x/y	$8/2 \square 4$
**	Exponenciação/Potenciação	$x**y$	$2**3 \square 8$
0.5	Radiciação/Raiz Quadrada	$x0.5$	$25**0.5 \square 5$
%	Resto da divisão (mod)	$x\%y$	$8\%2 \square 0$ $9\%2 \square 1$
//	Quociente da divisão \square apenas a parte inteira	$x//y$ Obs.: mesmo que $\text{int}(x/y)$	$25//3 \square 8$ $\text{int}(25/3) \square 8$

Código 4 – variável **numero** recebe o conteúdo **6**, que é um valor numérico (não precisa estar entre '')

```
numero = 6
print(numero)
```

Código 5

```
numero = 6
print('Número:', numero)
resultado = numero+1
print('Resultado:', resultado)
```

Código 6

```
x = 10
result = x+2
print('Resultado:', result)
```

Código 7

```
x = 10
result = x/3
print('Resultado:', result)
```

Obs.: para mostrar apenas **duas casas decimais**:

```
print('Resultado: %.2f' % result)
print(f'Resultado {result:.2f}')
```

Código 8

```
x = 10
result = x%4
print('Resto:', result)
```

Código 9

```
x = 25
result = x**0.5
print('Raiz Quadrada:', result)
```

Código 10

```
x = 25
result = x//3
print('Quociente:', result)
```

Código 11

```
x = 5
result = x**3
print('Potenciação (x elevado a 3):', result)
```

1.9 Precedência matemática

Ordem (prioridade) de execução das operações matemáticas nas expressões. A precedência matemática, também conhecida como ordem de operações, é uma convenção que define a ordem na qual as operações matemáticas devem ser realizadas quando uma expressão contém vários operadores. Isso evita ambiguidades e garante que os cálculos sejam executados de maneira consistente. Aqui está a ordem de precedência comum, da mais alta para a mais baixa:

1. Parênteses **()**
2. Radiação ****0.5**, Exponencial ******
3. Multiplicação *****, Divisão **/**, mod **%** e quociente **//** na ordem que aparecerem na expressão
4. Adição **+** e Subtração **-** na ordem que aparecerem na expressão

Parênteses **()**

- Os Parênteses têm a precedência mais alta e podem ser usados para forçar a avaliação de uma expressão na ordem que quisermos!!!
- Como as expressões em parênteses são avaliadas primeiro, quais seriam os resultados para:
 - o $2 * (3-1)$ é igual a: 4
 - o $(1+1)**(5-2)$ é igual a: 8

Exponencial ******

- A Exponenciação tem a próxima precedência mais alta, então quais seriam os resultados para:
 - o $1 + 2**3$ é igual a: 9
 - o $2 * 3**2$ é igual a: 18

Multiplicação *****, Divisão **/** e Resto(mod) **%**

- A Multiplicação, Divisão e o Cálculo do Resto têm precedência mais alta que a Adição **+** e Subtração **-**. Assim, quais seriam os resultados para:
 - o $2 * 3 - 1$ é igual a: 5
 - o $6 + 4 / 2$ é igual a: 8
 - o $8+7\%2$ é igual a: 9

Código 9

```
print(2 * (3-1))
print((1+1)**(5-2))
print(1 + 2**3)
print(2 * 3**2)
print(2 * 3 - 1)
print(6 + 4 / 2)
print(8+7%2)
```

1.10 Leitura/Entrada de dados pelo teclado

Ler dados de entrada pelo teclado é uma tarefa comum em programação, pois permite que os programas interajam com os usuários. A entrada de dados do teclado geralmente é usada para obter informações inseridas pelo usuário, como números, texto ou opções. Vou te mostrar exemplos em Python, mas o conceito se aplica a muitas linguagens de programação.

input **()** instrução que espera o usuário digitar um valor e retorna como uma string (valor alfanumérico – não numérico)

Código 10 - entrada de strings (Tipo de dado Alfanumérico – não é utilizado para operações matemáticas)

```
nome = input('Digite o nome: ')
print('Nome', nome)
```

Código 11 - entrada de valores inteiros (Tipo de dado numérico – sem casas decimais)

```
a = int(input('Digite o valor de a: '))
b = int(input('Digite o valor de b: '))
soma = a+b
print('Soma:', soma)
```

Código 12 – entrada de valores reais (Tipo de dado numérico – números flutuantes, que podem ter casas decimais, para uma melhor exatidão)

```
#programa que lê a base e a altura de um triângulo e apresenta a área do mesmo
base = float (input('Digite o valor da base: '))
alt = float (input('Digite o valor da altura: '))
area = base*alt/2
print('Área:', area)
```

Lembre-se de que a função “input()” retorna uma string, independentemente do tipo de dado que o usuário tenha digitado. Se você espera um tipo específico (como inteiro ou ponto flutuante), é necessário converter a string para esse tipo usando “int()” ou “float()”.

1.11 Exercícios de Fixação

Exemplos de apresentação de valores reais com duas casas decimais:

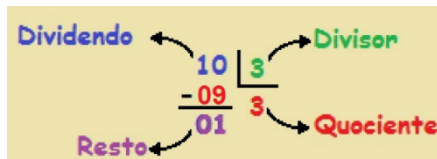
area = 25.3658974521

print('A área é de %.2f' % area, 'centímetros quadrados.')

print(f'A área é de {area:.2f} centímetros quadrados.')

1. Faça um programa que leia o nome e duas notas de um aluno, calcule e apresente seu nome e a sua média aritmética, no seguinte modelo:
 - a. (Fulano obteve a média XX).
2. Faça um programa que leia 2 valores reais, calcule e imprima:
 - a. a soma dos números;
 - b. o dobro da soma dos números.
3. O restaurante “Tudo Gostoso” cobra R\$ 30.00 pelo quilo de refeição. Escreva um programa que lê o peso do prato cheio de um cliente (em quilos) e imprima o valor a pagar.
 - a. Considere que o peso do prato é 0.1 Kg.
4. Antes do racionamento de energia ser decretado, quase ninguém falava em quilowatts, mas agora, todos a incorporaram a seu vocabulário. Sabendo que cada quilowatt de energia custa um centésimo do salário mínimo, escreva um algoritmo que receba o valor do salário mínimo e a quantidade de quilowatts gasta em uma residência, calcule e imprima o valor pago pela residência em questão.
5. Uma fábrica produz dois tipos de clips, plástico ou metal, sendo que a caixa de cada tipo é vendida por 5 e 10 reais, respectivamente. Faça um algoritmo em que o usuário forneça a quantidade de caixas vendidas de clips de plástico e de metal e informe qual será o valor arrecadado com cada um dos tipos e o valor total arrecadado.

6. Ler dois números inteiros, diferentes de zero, representando o dividendo e o divisor. Calcular e imprimir o quociente e o resto da divisão.



7. Escreva um algoritmo que leia três números inteiros e positivos (A, B, C) e calcule a seguinte expressão:

$$D = \frac{R+S}{2}, \text{ onde } R = (A + B)^2 \text{ e } S = (B + C)^2.$$

8. Uma professora precisa calcular a média de um aluno. O aluno realizou 3 provas, sendo que a primeira prova tem peso 1, a segunda tem peso 2 e a terceira prova tem peso 3. Escreva um programa que leia o nome do aluno e as notas, calcule e apresente a média ponderada do aluno.
9. Uma doceira levou pães de mel para vender em uma feira de gastronomia. Escreva um algoritmo que calcule o lucro obtido pela doceira na feira, baseando-se na quantidade vendida e nos preços de custo e de venda da unidade, considerando que o valor gasto pela doceira com o aluguel do *stand* foi de R\$ 500.00.
10. Escreva um programa que leia o número de um funcionário, seu número de horas trabalhadas em um mês e o valor que ele recebeu por hora. Imprima o número e o salário que o funcionário receberá no final do mês.

Exemplos de Entrada	Exemplos de Saída
25 100 5.50	Número = 25 Salário = R\$ 550.00
6 145 15.55	Número = 6 Salário = R\$ 2254.75

11. João recebeu seu salário e precisa pagar duas contas que estão atrasadas. Como as contas estão atrasadas, João terá de pagar multa de 2% sobre cada conta. Faça um programa que leia o valor do salário, o valor de cada uma das contas, calcule e mostre o valor que João pagou em cada conta e quanto restará do salário do João.
12. Clara é uma estudante de matemática e precisa calcular a área e o perímetro de um retângulo, dados os valores de sua largura e altura. Para ajudá-la, crie um programa em Python que solicite esses valores como entrada e calcule a área e o perímetro.
13. Imagine que você é responsável por criar um programa para gerenciar as informações de um cadastro de clientes de uma loja virtual. Escreva um programa que leia os dados dos clientes (nome, e-mail e telefone) e os armazene em variáveis.
- O programa deve solicitar ao usuário que insira as informações do cliente e, em seguida, exibir os dados do cliente cadastrado na tela.
14. A loja de eletrônicos "TecnoPlus" está realizando uma promoção em que o cliente que se

cadastrar no site da loja ganha um cupom de desconto de 10% para a primeira compra. Para se cadastrar, o cliente precisa informar seu nome completo e seu e-mail.

Escreva um código em Python que:

- Solicite ao usuário que informe seu nome completo e seu e-mail.
- Imprima na tela a mensagem: "Parabéns, <nome do usuário>! Seu cadastro foi realizado com sucesso e você ganhou um cupom de desconto de 10% para a sua primeira compra na TecnoPlus."
- Solicite ao usuário que informe o valor total de sua compra.
- Calcule o valor do desconto que o usuário terá direito.
- Imprima na tela a mensagem: "O valor total da sua compra é R\$ <valor da compra>. Com o cupom de desconto, você terá um desconto de R\$ <valor do desconto>, e o valor final a ser pago será de R\$ <valor final>."

Considere que o desconto é de 10% sobre o valor total da compra e que o valor final a ser pago é o valor total da compra menos o desconto.

- Faça um algoritmo que leia a idade de uma pessoa expressa em anos, meses e dias e mostre-a expressa apenas em dias.
 - Considere que o ano sempre tem 365 dias e que o mês sempre tem 30 dias.
- Faça um programa que obtenha o valor da hora trabalhada, a quantidade de horas trabalhadas na semana e o percentual de descontos com impostos, para realizar o cálculo do salário do funcionário de uma empresa. Considerando que o mês comercial, para a realização do cálculo, é composto por 4.5 semanas. Calcule e apresente:
 - o salário bruto (salário sem desconto);
 - o valor do desconto;
 - o salário líquido (salário bruto menos o desconto).

1.12 Desafios

- Faça um programa que leia um valor em segundos e informe quanto aquele valor representa em horas, minutos e segundos.

Dica:

%	Resto da divisão (mod)	x%y	8%2 à 0 e 9%2 à 1
//	Quociente da divisão à apenas a parte inteira	x//y Obs.: mesmo que int(x/y)	25//3 à 8 int(25/3) à 8

EXEMPLOS	
Entrada	Saída
Segundos: 3680	1 hora(s) – 1 minuto(s) – 20 segundo(s)
Segundos: 7408	2 hora(s) – 3 minuto(s) – 28 segundo(s)
Segundos: 352	0 hora(s) – 5 minuto(s) – 52 segundo(s)

1.13 Exercícios de Teste de Mesa

# Definição das variáveis	# Expressões matemáticas	# Exibição dos resultados
a = 10 b = 3 c = 7.2 d = 2.1 e = 4 f = 2.7 g = "Olá " h = "mundo" i = "!" j = 8	exp1 = a + b + e + j exp2 = c - d exp3 = a * b * e exp4 = c / d exp5 = b ** e exp6 = j // b exp7 = j % b exp8 = a * (b + e) exp9 = a * b + e exp10 = a + b / e + j exp11 = g + h + i	print("Expressões matemáticas:") print("exp1:", exp1) print("exp2:", exp2) print("exp3:", exp3) print("exp4:", exp4) print("exp5:", exp5) print("exp6:", exp6) print("exp7:", exp7) print("exp8:", exp8) print("exp9:", exp9) print("exp11:", exp11) print("expressão 3:", exp3)

2. ESTRUTURA DE DECISÃO OU ESTRUTURA CONDICIONAL: IF .. ELIF .. ELSE

A estrutura de decisão, também conhecida como estrutura condicional, é usada quando você quer que o programa tome decisões com base em condições específicas. A estrutura mais comum nesse contexto é a combinação de “if”, “elif” (abreviação de “else if”) e “else”. Essa estrutura é utilizada quando existem duas ou mais possibilidades de execução, de acordo com alguma condição.

2.1 Operadores Relacionais

Os operadores relacionais são usados para comparar valores e expressões, retornando um valor booleano (Verdadeiro ou Falso) com base na relação entre eles. Eles são frequentemente usados em estruturas condicionais para tomar decisões com base em comparações. Aqui estão os operadores relacionais mais comuns:

Operador	Função
==	Verifica se dois valores são iguais.
>	Verifica se o valor à esquerda é maior que o valor à direita
<	Verifica se o valor à esquerda é menor que o valor à direita.
>=	Verifica se o valor à esquerda é maior ou igual ao valor à direita.
<=	Verifica se o valor à esquerda é menor ou igual ao valor à direita.
!=	Verifica se dois valores são diferentes

Exemplo:

```
x = 5
y = 8

igual = x == y # False
diferente = x != y # True
maior_que = x > y # False
menor_que = x < y # True
maior_ou_igual = x >= y # False
menor_ou_igual = x <= y # True
```

2.2 Estrutura de decisão (condicional) simples - if

Na programação, muitas vezes tem-se a necessidade de verificar condições e alterar o comportamento do programa de acordo com os resultados das condições. Isso pode ser feito usando a estrutura de decisão **if**, que é uma estrutura condicional em que as instruções serão executadas quando determinada condição for verdadeira.

❖ **Condição** é uma expressão em que o resultado é verdadeiro (true) ou falso (false),

```
if condição:
    se a condição1 for verdadeira
    as instruções deste bloco serão executadas

    essa instrução é executada sempre
```

Na estrutura a seguir:

- as instruções 1 e 2 serão executadas apenas **se a condição for verdadeira**;
- já a instrução 3 será sempre executada.

```
if condição:
    instrução 1
    instrução 2
instrução 3
```

Programa Exemplo 1:

Programa que lê duas notas de um aluno, apresenta a média e parabeniza-o se ele for aprovado (média mínima para aprovação é 6.0).

```
nota1 = float(input("Nota 1:"))
nota2 = float(input("Nota 2:"))

media = (nota1+nota2)/2
print("A média é ", media)

if media >= 6.0:
    print("Parabéns você foi aprovado!")
```

2.3 Estrutura de decisão (condicional) composta – if..else

Na estrutura **if..else**, se a condição for verdadeira, as instruções (ou comandos) dentro do **if** (recuadas/tabuladas dentro do if) serão executadas. Se for falsa, são executadas as instruções dentro de **else** (recuadas/tabuladas dentro do else).

- ❖ O conjunto de instruções recuadas/tabuladas que seguem o **if** são chamadas de bloco. O primeiro comando com tabulação diferente marca o fim do bloco.
- ❖ A primeira instrução com tabulação/recuo diferente marca o fim do bloco.

```
if condição:
    se a condição1 for verdadeira
    as instruções deste bloco serão executadas
else:
    caso a condição seja falsa
    as instruções deste bloco de código serão executadas

essa instrução é executada sempre
```

Na estrutura a seguir:

- as instruções 1 e 2 serão executadas apenas **se a condição1 for verdadeira**;
- as instruções 3 a 4 serão executadas apenas **se a condição 1 for falsa**;
- já a instrução 5 será sempre executada.

```
if condição1:
    instrução 1
    instrução 2
else:
    instrução 3
    instrução 4
instrução 5
```

Programa Exemplo 2

Programa que lê duas notas de um aluno, apresenta a média e informa se ele foi aprovado ou reprovado (média mínima para aprovação é 6.0).

```
nota1 = float(input("Nota 1:"))
nota2 = float(input("Nota 2:"))

media = (nota1+nota2)/2
print("A média é ", media)

if media >= 6.0:
    print("Aprovado!")
else:
    print("Reprovado!")
```

2.4 Estrutura de decisão (condicional) encadeada - – if..elif..else

O comando **elif** é uma abreviação de **else if**. Cada condição é verificada em ordem, da seguinte forma: se a primeira é falsa, a segunda é verificada; se a segunda é falsa, a terceira é verificada, e assim por diante. Se uma das condições é verdadeira, o bloco de instruções correspondente é executado e a execução da estrutura encadeada (**if..elif..else**) é finalizada.

- ❖ Mesmo que mais do que uma condição seja verdadeira, apenas um bloco de instruções é executado.
- ❖ Não existe limite para a quantidade de **elif**, mas apenas um **else** final (opcional) é permitido e precisa ser o último da estrutura encadeada.

```
if condição1:
    se a condição1 for verdadeira
    as instruções deste bloco serão executadas
elif condicao2:
    caso a condição1 seja falsa e a condição2 seja verdadeira
    as instruções deste bloco de código serão executadas
elif condicao3:
    se condições 1 e 2 forem falsas e a condição3 seja verdadeira
    as instruções deste bloco de código serão executadas
else:
    caso todas as condições anteriores sejam falsas
    as instruções deste bloco de código serão executadas

essa instrução é executada sempre
```

Na estrutura a seguir:

- as instruções 1 e 2 serão executadas apenas **se a condição1 for verdadeira**;
- as instruções 3 e 4 serão executadas apenas **se a condição1 for falsa e a condição2 for verdadeira**;
- as instruções 5 a 6 serão executadas apenas **se as condições 1 e 2 forem falsas**;
- já a instrução 7 será sempre executada, pois está fora da estrutura do condicional.

```

if condição1:
    instrução 1
    instrução 2
elif condição2:
    instrução 3
    instrução 4
else:
    instrução 5
    instrução 6
instrução 7

```

Programa Exemplo 3

Programa que lê duas notas de um aluno, apresenta a média e informa o resultado:

- Aprovado: media maior ou igual a 6;
- Exame: media maior ou igual a 4 e menor que 6;
- Reprovado: nota menor que 3.

```

nota1 = float(input("Nota 1:"))
nota2 = float(input("Nota 2:"))

media = (nota1+nota2)/2
print("A média é ", media)

if media >= 6.0:
    print("Aprovado!")
elif media >= 3.0:
    print("Exame!")
else:
    print("Reprovado!")

```

Programa Exemplo 4

Programa que lê a idade de uma pessoa e apresenta a categoria da mesma, de acordo com a tabela a seguir.

Idade	Categoria
Até 10 anos	Categoria A
11 até 20 anos	Categoria B
21 até 30 anos	Categoria C
31 até 40 anos	Categoria D
Acima de 40 anos	Categoria E

```

idade = int (input('Digite a idade: '))

if idade <= 10:
    print('Categoria A')
elif idade <= 20:
    print ('Categoria B')
elif idade <= 30:
    print ('Categoria C')
elif idade <= 40:
    print ('Categoria D')

```

```
else:
    print('Categoria E')
```

2.5 Estrutura de decisão (condicional) encadeada

A estrutura de decisão (condicional) encadeada é uma extensão da estrutura de decisão básica (com “if”, “elif” e “else”). Ela permite lidar com várias condições de maneira sequencial, tomando decisões com base nas condições em uma ordem específica. Essa estrutura é útil quando você tem várias condições exclusivas e quer executar um bloco de código correspondente à primeira condição verdadeira encontrada. É possível colocar uma sentença condicional dentro de outra.

```
if condição1:
    se a condição1 for verdadeira
    as instruções deste bloco serão executadas
elif condição2:
    caso a condição1 seja falsa e a condição2 seja verdadeira
    as instruções deste bloco, inclusive o if condição3
    serão executadas

    if condição3:
        se a condição3 for verdadeira
        as instruções deste bloco serão executadas
    else:
        caso a condição3 seja falsa
        as instruções deste bloco serão executadas
else:
    caso as condições 1 e 2 sejam falsas
    as instruções deste bloco de código serão executadas

essa instrução é executada sempre
```

Na estrutura a seguir:

- as instruções 1 e 2 serão executadas apenas **se a condição1 for verdadeira**;
- a instrução 3 e a instrução “if condição3:” serão executadas se a **condição1 for falsa e a condição2 for verdadeira**;
- as instruções 4 e 5 serão executadas apenas **se a condição1 for falsa e as condições 2 e 3 forem verdadeiras**;
- a instrução 6 será executada apenas **se a condição1 for falsa, a condição2 for verdadeira e a condição 3 for falsa**;
- a instrução 7 será executada **se as condições 1 e 2 forem falsas**;
- já a instrução 8 será sempre executada, pois está fora da estrutura do condicional.

```
if condição1:
    instrução 1
    instrução 2
elif condição2:
    instrução 3
    if condição3:
        instrução 4
        instrução 5
    else:
        instrução 6
else:
    instrução 7
```


Programa Exemplo 5

Programa que lê duas notas de um aluno, apresenta a média e informa o resultado:

- ❖ Aprovado: media maior ou igual a 6;
- ❖ Exame: media maior ou igual a 3 e menor que 6.
 - Caso o aluno esteja em exame, solicita a notado exame e informe se ele foi “Aprovado no exame!” ou “Reprovado no exame!”. Nota mínima para aprovação no exame é 6.
- ❖ Reprovado: nota menor que 3.

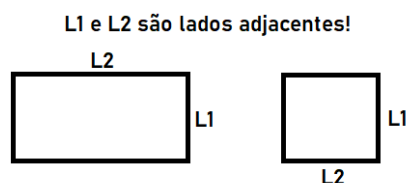
```
nota1 = float(input("Nota 1:"))
nota2 = float(input("Nota 2:"))

media = (nota1+nota2)/2
print("A média é ", media)

if media >= 6.0:
    print("Aprovado!")
elif media >= 3.0:
    print("Exame!")
    notaExame = float(input("Nota do exame:"))
    if notaExame >= 6.0:
        print("Aprovado no exame!")
    else:
        print("Reprovado no exame!")
else:
    print("Reprovado!")
```

2.6 Exercícios de Fixação

1. Faça um programa que leia dois números, calcule e imprima a soma. Caso a soma seja par, calcula e imprime a metade da soma.
2. Escreva um algoritmo que **leia os valores de dois lados adjacentes** de uma figura geométrica e informe se eles representam um quadrado perfeito ou um retângulo. Caso represente um retângulo, informe o tamanho do maior lado.
*Quadrado perfeito é aquele que possui todos os lados iguais.



3. Faça um programa que leia **dois** números e calcule a divisão do primeiro pelo segundo. Porém, se o usuário digitar zero para o segundo número, não realize o cálculo e apresente uma mensagem de erro “Não pode ser feita divisão por zero!”.

4. Faça um programa que solicite ao usuário que digite um valor para a variável num. Verifica o número digitado e informa se o é positivo, negativo ou zero. Caso o número seja positivo, apresente o seu dobro. Caso o número seja negativo, informe se ele é par ou ímpar.
5. Ler dois valores e um código de condição. Se o código for “c” os valores devem ser escritos em ordem crescente. Se o código for “d”, deve-se escrevê-los em ordem decrescente.
6. Um posto está com uma promoção, de forma que oferece os seguintes descontos, de acordo com a quantidade abastecida:
 - de 20 a 30 litros o desconto é de 5%; e
 - acima de 30 litros o desconto é de 10%.

Elabore um algoritmo que leia a quantidade de litros e o tipo de combustível (A - Álcool ou G - Gasolina), calcule e imprima o valor a ser pago, sabendo-se que o preço do litro da gasolina é de R\$ 7.00 e o preço do litro do álcool é de R\$ 5.00.

7. Uma companhia de saneamento calcula o valor da conta de água de acordo com o tipo de consumidor:
 - Residencial: R\$ 5.00 de taxa mais R\$ 0.05 por m³ gasto;
 - Comercial: R\$ 500.00 para os primeiros 80 m³, acrescido de R\$ 0.25 por m³ gastos acima dos 80 m³;
 - Industrial: R\$ 800.00 para os primeiros 100 m³, acrescido de R\$ 0.04 por m³ gastos acima dos 100 m³.

Baseando-se nessas informações, escreva um algoritmo que leia o tipo do cliente e o seu consumo de água em metros cúbicos. Depois, calcule e apresente a conta de água a ser paga pelo cliente.

8. Faça um programa que leia a idade de uma pessoa e informe se ela é maior ou menor de idade (atinge-se a maior idade com 18 anos).
9. O código a seguir gera um valor inteiro aleatório (entre 0 e 10) e guarda em uma variável chamada valorAleatorio:

```
import random

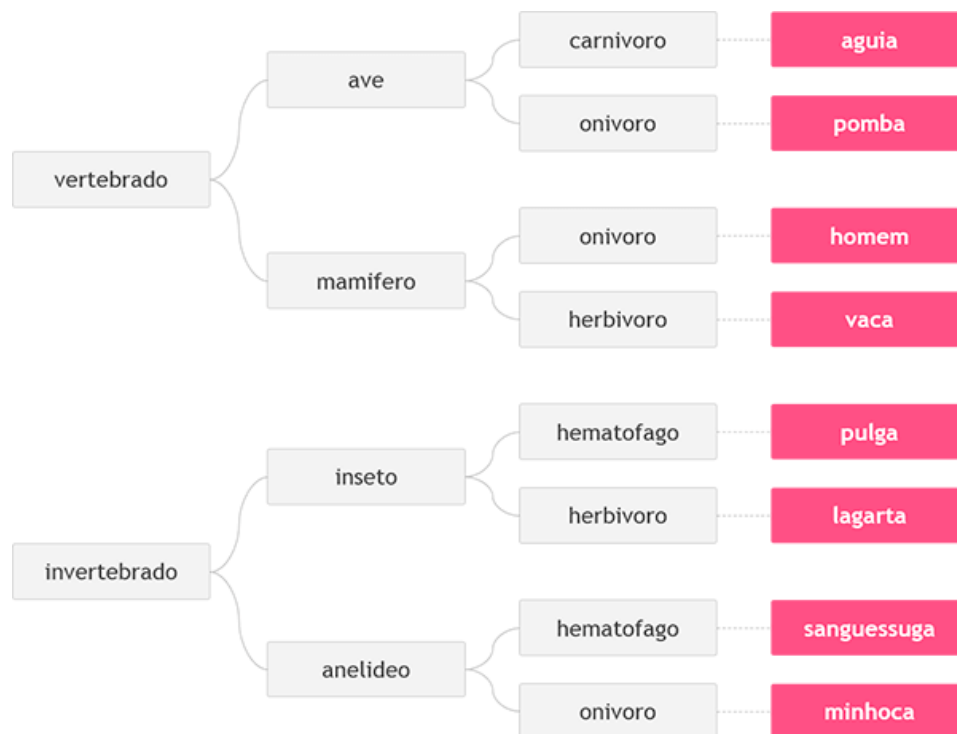
# Gerando um valor aleatório entre 1 e 10

valorAleatorio = random.randint(1, 10)
```

Continuando o código acima, após o programa gerar o valor aleatório, solicite ao usuário que digite um número e, na sequência, informe se o número digitado pelo usuário é maior, menor ou igual ao valor gerado aleatoriamente.

10. Faça um programa que leia três palavras em português. Essas palavras definem um animal de acordo com a imagem a seguir. Depois, imprima o animal escolhido pelo usuário, considerando as três palavras digitadas.

Obs.: considere que o usuário digitará as 3 palavras corretamente, ou seja, não precisa verificar se ele digitou um valor errado.



2.7 Desafios

1. Usando apenas as instruções vistas até o momento, escreva um algoritmo que receba a data de aniversário de duas pessoas (pessoa1 e pessoa2), sendo que a data consiste em dia, mês e ano (três variáveis separadas). Baseando nestas datas, informe se as duas pessoas têm a mesma idade ou, em caso negativo, informe qual é a mais velha.
2. Calcular e apresentar o total de crianças, a quantidade de litros de leite necessários e o custo para comprar o leite para alimentar as crianças uma determinada creche que contém de três salas. Para isso, o usuário deve informar a quantidade de crianças de cada sala. Para o cálculo considere que:
 - cada litro de leite alimenta 4 crianças;
 - o litro de leite custa R\$ 3.50;
 - pode ser que sobre leite em uma caixa, mas, todas as crianças devem ser alimentadas.

Exemplos	
Entrada	Saída
Quantidade de alunos da sala 1: 10 Quantidade de alunos da sala 2: 10 Quantidade de alunos da sala 3: 8	Total de crianças: 28 Qt. de litros de leite: 7 Custo: 24.5
Quantidade de alunos da sala 1: 9 Quantidade de alunos da sala 2: 9 Quantidade de alunos da sala 3: 3	Total de crianças: 21 Qt. de litros de leite: 6 Custo: 21.0

