

Secretaria
de Educação e
Esportes



GOVERNO DE
**PER
NAM
BU**CO
ESTADO DE MUDANÇA

Unidade Curricular

Programação com Python

Material de apoio à ação docente

PERNAMBUCO



**SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO**

Secretária de Educação e Esportes

Ivaneide Dantas

Secretária Executiva Planejamento e Coordenação

Mônica Maria Andrade

Secretária Executiva de Desenvolvimento da Educação

Tárcia Regina da Silva

Secretário Executivo de Ensino Médio e Profissional

Gilson Alves do Nascimento Filho

Secretário Executivo de Administração e Finanças

Gilson Monteiro Filho

Secretário Executivo de Gestão da Rede

Igor Fontes Cadena

Secretário Executivo de Esportes

Luciano Leonídio



SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

Equipe de elaboração

Gabriel Pimenta Carneiro Campelo

Equipe de coordenação

Gerente de Políticas Educacionais do Ensino Médio (GGPEM/SEDE)

Janine Furtunato Queiroga Maciel

Gestor Pedagógico (GGPEM/SEDE)

Rômulo Guedes e Silva

Chefe da Unidade do Ensino Médio (GGPEM/SEDE)

Andreza Shirlene Figueiredo de Souza

Revisão

Ana Caroline B. F. Pacheco

Ana Karine Pereira de Holanda Bastos

Andreza Shirlene Figueiredo de Souza

Sumário

1. Introdução	5
2. Princípios Básicos da Linguagem de Programação Python.	8
Orientações para a realização de atividades	19
3. Concatenação e Interpolação de Strings	21
4. Estrutura de Decisão	23
Orientações para a realização de atividades	25
Outras sugestões de atividades	27
5. Coleções do tipo Listas	29
Orientações para a realização de atividades	31
6. Coleções do tipo Tuplas	33
Orientações para a realização de atividades	34
Outras sugestões de atividades	36
7. Sub rotina de repetição do tipo Funções	37
Sugestões para a realização de atividades	41
Orientações para a realização de atividades	55
8. Referências bibliográficas	56

I. Introdução

Prezado/a professor/a,

Seguindo a reformulação proposta pela promulgação da lei nº13.415/2017 que regulamenta a implementação da Base Nacional Comum Curricular e, portanto, passa a seguir este documento de referência para estabelecer as diretrizes educacionais das escolas de ensino regular e integral. Seguindo esta mudança, inclusive dos saberes discutidos em sala de aula, estamos apresentando este material de apoio baseado neste novo referencial que serviu para elaboração dos Currículos Escolares nas diferentes redes educacionais. Ainda a este respeito, utilizando o que preconiza o Novo Currículo de Pernambuco, vigente desde 2021 na educação oficial de Pernambuco, desenvolvemos um texto que parte dos conceitos mais básicos ligados à Unidade Curricular de Linguagem com Python, suas aplicações e, principalmente, sugestões de atividades que podem fazer parte da jornada de aulas ou servir como inspiração para a elaboração de outras. Já no último capítulo, dedicamos a demonstrar como operar com a Linguagem de programação Python para construir jogos para computador.

Muito além de ser um ambiente formal de educação, as escolas precisam contemplar o acolhimento da juventude em suas diversas necessidades e potencialidades. Neste sentido, a BNCC aborda a necessidade de cultivar no ambiente escolar *“viabilizar o acesso dos estudantes às bases científicas e tecnológicas dos processos de produção do mundo contemporâneo, relacionando teoria e prática – ou o conhecimento teórico à resolução de problemas da realidade social, cultural ou natural?”* (BNCC, p. 466). A incorporação da tecnologia nas ações pedagógicas deve ser feita indo além do consumo de aparelhos e *softwares*.

Perceber que a atual geração pode passar do patamar de usuário para ser desenvolvedor de tecnologias, adquirindo a capacidade de compreender como pode ocorrer

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

o pensamento computacional e seus pormenores. Sempre que possível, estimular que o conhecimento discutido possa contribuir para *“revelar os contextos nos quais as diferentes formas de produção e de trabalho ocorrem, sua constante modificação e atualização nas sociedades contemporâneas e, em especial, no Brasil”* (BNCC, P. 466) já que é evidente a necessidade de profissionais na área de Tecnologia da Informação em suas mais diversas aplicações na sociedade. No intuito de abarcar a evolução que a sociedade sofreu com a popularização da internet e de aparelhos conectados é preciso *“prever o suporte aos jovens para que reconheçam suas potencialidades e vocações, identifiquem perspectivas e possibilidades, construam aspirações e metas de formação e inserção profissional presentes e/ou futuras, e desenvolvam uma postura empreendedora, ética e responsável para transitar no mundo do trabalho e na sociedade em geral”* (BNCC, P. 466), tendo na escola um ponto de partida seguro para projetar no mundo formas de interagir com ele e contribuir para o desenvolvimento pessoal e também social.

A sistematização da aprendizagem passa pela tomada de consciências daquilo que é o alicerce da Unidade Curricular, ainda segundo a BNCC, o *“pensamento computacional: envolve as capacidades de compreender, analisar, definir, modelar, resolver, comparar e automatizar problemas e suas soluções, de forma metódica e sistemática, por meio do desenvolvimento de algoritmos”* (BNCC, p. 474) para isso é preciso fortalecer a compreensão evitando copiar um código já pronto e utilizar sem antes analisar e adaptar ao propósito da aprendizagem. Sempre que possível construir seus próprios programas no intuito de ter a percepção de como opera um algoritmo e sensibilizar para a importância de compreender do que se trata o pensamento computacional.

Apesar da grande maioria dos jovens que, estão em idade escolar, já serem nativos digitais é preciso explorar o sentido do termo *“mundo digital”* que é explicitado pela BNCC, visto que *“envolve as aprendizagens relativas às formas de processar, transmitir e distribuir a informação de maneira segura e confiável em diferentes artefatos digitais – tanto físicos (computadores, celulares, tablets etc.) quanto virtuais (internet, redes sociais e nuvens de dados, entre outros) –, compreendendo a importância contemporânea de codificar, armazenar e proteger a informação”* (BNCC, p 473), prevendo um comportamento ético e solidários das ferramentas tecnológicas à serviço da promoção da igualdade social e de gênero, estimulado e exercitado dentro do ecossistema escolar a

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

riedade e segurança do armazenamento de dados e, principalmente, integridade e sigilo da informação coletada.

A Unidade Curricular de *Linguagem com Python* faz parte da Trilha de **Tecnologias Digitais** que faz parte dos Itinerários Formativos, compondo a etapa de escolha do estudante durante a sua jornada ao longo dos três anos do Ensino Médio. Neste sentido, há vantagem de contar com participantes que têm predisposição para enveredar no tema de Tecnologias.

O Novo Currículo de Pernambuco preconiza a respeito das Habilidades dessa Unidade Curricular de acordo com dois eixos:

Processos Criativos (EMIFMAT06PE): Propor e testar soluções éticas, estéticas, criativas e inovadoras para problemas reais, considerando a lógica de programação Python aplicada a partir dos conhecimentos matemáticos associados ao domínio de operações e relações matemáticas simbólicas e formais. (Novo Currículo de Pernambuco, 2021)

Apesar deste material de apoio explicitar atividades que partem de motivações ou situações-problema hipotéticas é desejado que seja estimulado nos estudantes a capacidade de analisar a sua própria realidade e propor soluções para casos reais do cotidiano que envolvam a comunidade escolar e a vida dos jovens. Assim como outras linguagens de programação, *Python* se utiliza de estruturação de algoritmos para executar sua rotina e de princípios da Matemática para estruturar suas decisões. Uma ótima oportunidade de entrelaçar interesses e conhecimentos numa ação pedagógica.

O segundo eixo fala sobre:

Mediação e Intervenção Sociocultural (EMIFMAT09PE): Propor e testar estratégias de mediação e intervenção para resolver problemas de natureza sociocultural e de natureza ambiental relacionados à lógica de programação Python aplicada a partir de conhecimentos matemáticos. (Novo Currículo de Pernambuco, 2021)

Com base nesses pressupostos, esta **Unidade Curricular** propõe, a seguinte **ementa**:

Proposição, testagem e resolução de problemas matemáticos utilizando algoritmo e a linguagem de programação Python. Seleção e mobilização dos conhecimentos da lógica de programação aplicada a sintaxes, variáveis, estruturas de repetição (loops), estruturas condicionais, noções de armazenamento e tratamento de dados. (Pernambuco, 2021)

**SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO**

A necessidade de projetar, para além do ambiente escolar as aprendizagens e descobertas dos estudantes, seguindo as atividades propostas nos momentos pedagógicos é precedida pela necessidade de se trabalhar a capacidade socioemocional levando em consideração que o uso do conhecimento adquirido deve ser utilizado de acordo com as leis vigentes e o comportamento ético da sociedade. Utilizar o armazenamento de dados pessoais em pesquisas ou bancos de dados com a linguagem *Python*, implica em se responsabilizar com a utilização sigilosa das informações recolhidas para fins exclusivamente da atividade a ser desenvolvida.

Por fim, informamos que este material de apoio pode ser disponibilizado aos/às estudantes para fins didáticos sem maiores problemas.

2. Princípios Básicos da Linguagem de Programação Python.

O que é um algoritmo?

Numa pesquisa rápida no dicionário Oxford Languages descobre-se que a palavra “algoritmo” está associada a duas áreas do conhecimento:

1. MATEMÁTICA

sequência finita de regras, raciocínios ou operações que, aplicada a um número finito de dados, permite solucionar classes semelhantes de problemas.

2. INFORMÁTICA

conjunto das regras e procedimentos lógicos perfeitamente definidos que levam à solução de um problema em um número finito de etapas.

(Oxford Languages, 2023)

Em ambas as explicações observamos que se trata de um ordenamento de instruções (de forma mais clara possível) com uma quantidade limitada de processos que, uma vez seguidos, podem ocasionar a solução de um problema proposto.

Para exemplificar, podemos fazer uma analogia com uma receita de bolo. Após a separação dos ingredientes, é preciso seguir cada etapa e seus parâmetros (temperatura, quantidade e tempo) para fazer a sobremesa. Se alguma das etapas for invertida ou ignorada, pode comprometer todo resultado. Assim é um algoritmo, uma sequência rígida de argumentos, com parâmetros explícitos que devem ser seguidas em busca do resultado desejado. Estas mesmas explicações podem ser aplicadas à compreensão de operações matemáticas. Pegue o exemplo de como resolver uma divisão com números inteiros. Há um processo que seguido corretamente ocasiona o quociente esperado. É importante trazer exemplos que sejam ressonantes com a vivência dos estudantes para motivar a aprendizagem e atenção dos discentes com relação a este conceito, uma vez que ele é essencial na tomada de consciência do conhecimento dos discentes.

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

O algoritmo deve fazer parte do planejamento do programa que se deseja elaborar. Pode ser feito no formato de pseudocódigo, que seria uma sucessão de frases que explicam o que acontece em cada etapa do código que ajuda a depurar o planejamento e ter mais controle das decisões tomadas para fazer o código.

O que é uma Linguagem de Programação?

Por incrível que pareça, um computador é burro. A verdade dos fatos é que toda a informação que está contida num computador está limitada a dois valores: Zero e Um. Então, tudo que digitamos, lemos, imprimimos e demais tarefas são executadas com conjuntos de bilhões de zeros e uns escritos na ordem correta. Esta forma de expressão é denominada *código binário*.

Então, para qualquer instrução que precisamos atribuir a um computador deveríamos sequenciar um conjunto quase infinito de zeros e uns numa ordem perfeita. Isso seria humanamente impossível. É preciso traduzir a linguagem dos seres humanos (alto nível) para a linguagem de máquina (baixo nível) de forma mais prática possível. A transcrição, em ambos os sentidos, é feita através de uma **Linguagem de Programação**. Cada uma delas tem seu conjunto de diretivas que guardam em si um conjunto de significados entendidos pelo processador do computador e passa a executar ações mediante às instruções. É importante destacar que a sintaxe dos termos que compõem a Linguagem de Programação seja escrita de maneira correta e, principalmente, na ordem coerente para surtir o resultado desejado pelo desenvolvedor.

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

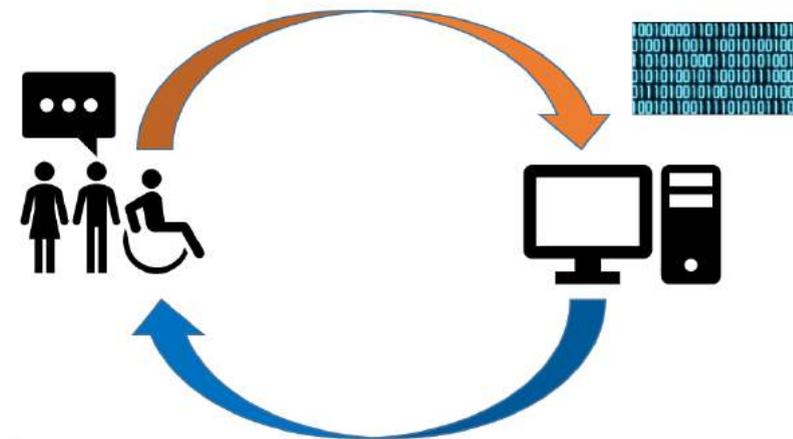


Fig.1: Representação de como uma linguagem de programação opera.
Fonte: elaborada pelo autor.

As quatro principais linguagens de programação mais utilizadas por desenvolvedores de softwares são: *Java*, *Python*, *C#* e *C/C++*. Cada uma delas tem suas aplicações, facilidades e dificuldades. O que têm em comum é que todas elas têm a capacidade de interagir entre o desenvolvedor e as rotinas mais profundas que ocorrem num computador. Cada linguagem de programação tem um gama de softwares que apresentam um ambiente de desenvolvimento chamado de *Compilador*. Para *Python* existem vários compiladores, muitos deles gratuitos. Toda aprendizagem que está sendo exercitada pode ser usada em qualquer um dos compiladores disponíveis. Fica a critério do discente a escolha do *software* que melhor lhe servir.

Sintaxe em diferentes linguagens para a instrução: **Escreva na tela: Hello World**	
Java: System.out.println("Hello World");	Python: print("Hello World")
C#: int main() { cout << "Hello World"; return 0; }	C/C++: Console.WriteLine("Hello World");

Tab.1: Comparação entre as diferentes sintaxes para a mesma instrução das principais linguagens de Programação.
Fonte: elaborada pelo autor.

A escolha pela Linguagem de Programação *Python* se dá pela busca que ela tem pela simplicidade. O seu próprio inventor, Guido van Rossum (matemático e programador de origem dos Países Baixos), relata que o que mais importa é o esforço do programador sobre o esforço do computador. Ela faz uso de uma sintaxe concisa e clara junto com bibliotecas muito úteis. A primeira vez que ela foi apresentada foi no ano de 1991. Hoje em dia a fundação *Python Softwares* cuida da distribuição (gratuita) do compilador que pode ser baixado e utilizado livremente.

Ambiente de desenvolvimento do Python (IDLE)

Existem vários compiladores para desenvolver programas com Python. Cabe ao/à usuário/a escolher qual o que lhe convém. No site oficial do *Python* (python.org) é possível baixar gratuitamente o console oficial para escrita de códigos em *Python*. Acessando o site, selecione a aba *Download*, escolha o sistema operacional, aguarde o arquivo executável ser baixado e prossiga com a instalação.

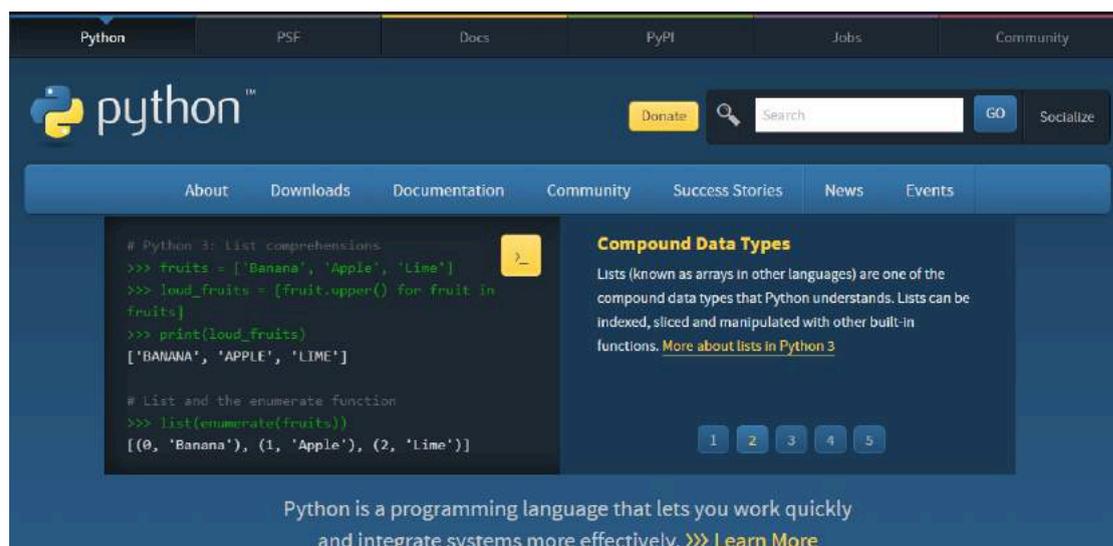
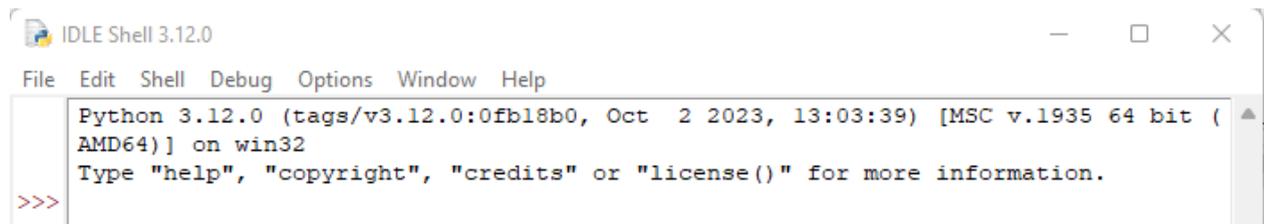


Fig.2 - Python
Fonte: Site oficial do Python (2024).

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

Uma vez instalada a IDLE *Python* temos uma tela inicial que pode servir como ensaio para os programas que vamos fazer. O símbolo de três maiores que (“>>>”) indicam a linha do código que está sendo desenvolvido. Ao apertar enter, o código passa para a linha seguinte que pode ser uma nova instrução ou pode ser a execução da linha anterior.



```
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct 2 2023, 13:03:39) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

Fig.3: Tela de desenvolvimento da IDLE Python.
Fonte: adaptada pelo autor.

Exemplo:

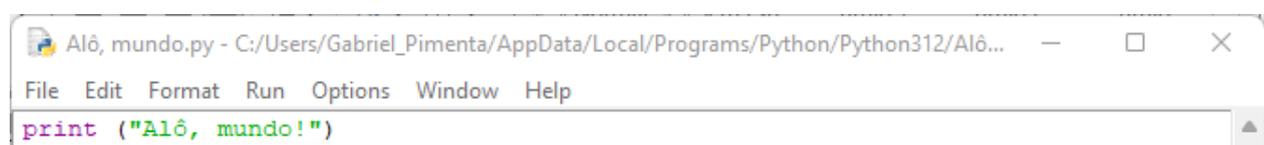
Faça um código que exibe a frase “Alô, mundo!” na tela do IDLE

```
print ("Alô, mundo!")
```

Saída na IDLE do Python

```
print ("Alô, mundo!")
'Alô, mundo!'
```

Outra forma de editar um programa em desenvolvimento é usar um “novo arquivo”. Para isso clique em **File > New File** ou aperte **Ctrl+N**. Nesta nova página que se abriu, temos um ambiente que comporta o código em desenvolvimento que deve ser escrito, e só ao final será executado por um comando dado pelo/a usuário/a. Para visualizar o resultado dos comandos é preciso clicar em **Run > Run Module** ou apertar **F5**. Com isso, vai ser solicitado que o programa seja salvo antes de ser executado. Escolha um nome sugestivo e boa sorte! Em seguida, surgirá uma nova tela que segue as instruções escritas anteriormente. Veja como fica o código do exemplo anterior:

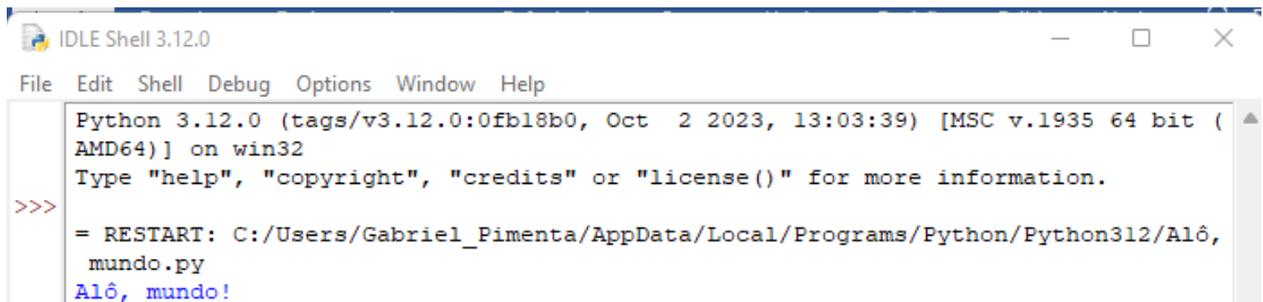


```
Alô, mundo.py - C:/Users/Gabriel_Pimenta/AppData/Local/Programs/Python/Python312/Alô...
File Edit Format Run Options Window Help
print ("Alô, mundo!")
```

Fig.4: Tela de desenvolvimento da IDLE Python.

Fonte: adaptada pelo autor.

Após selecionar “Run”, a tela do console vai surgir da seguinte forma:



```
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct 2 2023, 13:03:39) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Gabriel_Pimenta/AppData/Local/Programs/Python/Python312/Alô,
mundo.py
Alô, mundo!
```

Fig.5: Tela do console Python.

Fonte: adaptada pelo autor.

Sintaxe

Um programa feito em Python segue o ordenamento da linha de cima para baixo. É possível dividir uma linha usando a barra invertida (“\”) que faz o compilador pular para a próxima.

O caractere hashtag (“#”) possibilita incluir comentários que não são executados pelo código, mas permitem ao desenvolvedor que ele explique quais os objetivos ou significados daquela parte do programa.

Para fazer o texto de um código mudar para a linha seguinte (durante a interface com a tela do usuário) basta acrescentar “\n” no ponto de mudança.

O uso de dois pontos (“:”) indica o final de uma instrução. Sendo uma figura de controle, aponta para o início de uma nova estrutura que deve ficar com a tabulação a frente da linha de código inicial. Esta é a forma correta de usar a indentação que tem o papel de fazer com que o código execute sub rotinas antes de seguir com o programa principal. Para isso, utiliza-se como padrão quatro espaços ou um parágrafo.

Veja a figura ilustrativa:

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
 GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
 GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

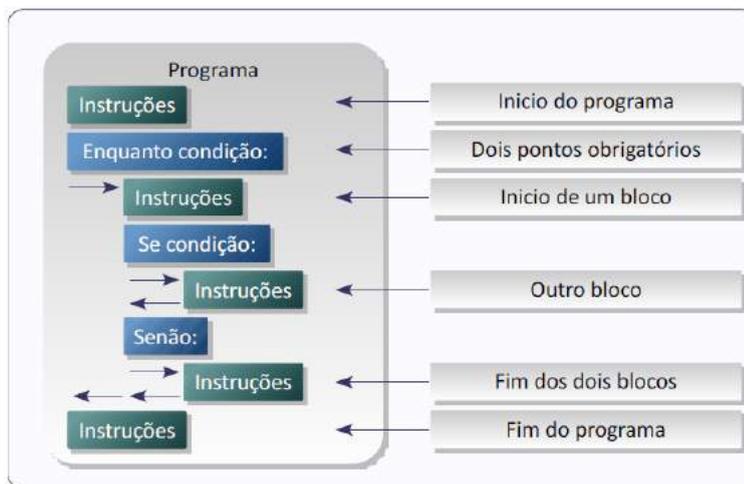


Fig.5: Estrutura de indentação.
 Fonte: FRANCO, João (2010, p.06)

A linguagem de programação *Python* utiliza Objeto, Método e Módulo para estruturar sua sintaxe e ordenar as instruções.

- **Objeto:** alocação de memória que armazena letras, palavras, frases, listas, valores e tudo aquilo que se deseja referenciar no código desenvolvido.
- **Método:** comando que acumula um conjunto de atividades pré-programadas que utiliza o objeto como parâmetro/atributo.
- **Módulo:** são algumas rotinas mais comuns que já estão organizadas na linguagem Python. Um bom exemplo é o módulo “math” que implicitamente tem operações mais complexas como raiz quadrada, fatorial dentre outras.

Exemplo:

Instrução	Sintaxe
Atribuir a palavra “Einstein” à variável “nome”.	<pre>nome = "Einstein"</pre> <p>Objeto Atributo</p>
Incluir ao final do objeto “lista” o valor 25. Obs.: dentro dos parênteses pode ser um objeto também.	<pre>lista.append(25)</pre> <p>Objeto Método</p>
Utilizar o módulo math para extrair a raiz quadrada do valor 16. Obs.: dentro dos parênteses pode ser um objeto também.	<pre>math.sqrt(16)</pre> <p>Módulo Função</p>

Tab. 2: Aplicações de Objeto, Método e Módulo
 Fonte: LabTIME (2017, p.09).

Declaração de variável

Variável é um termo ou palavra que adotamos no código que passa a alocar uma certa quantidade de memória do computador para armazenar a informação. Na linguagem *Python* temos 4 tipos de dados:

1. **Interge:** expressa quantidades exatas de valores não aceitando números decimais.
2. **Float:** armazena valores decimais.
3. **String:** usado para agrupamentos de letras, em geral para palavras.
4. **Booleanos:** armazena estado como falso ou verdadeiro.

OBS.: Existem outros formatos para declaração de variáveis.
 Para os próximos passos vamos nos concentrar nesses quatro.

O ambiente de desenvolvimento da linguagem de programação em *Python* permite ao usuário que a entrada de dados seja reconhecida automaticamente após a digitação. Desta forma, é possível alocar a quantidade de memória ideal para cada informação que armazenamos no código.

Os nomes das variáveis devem começar com letras ou sublinhados. Deve-se respeitar o uso de letras maiúsculas e minúsculas.

Nome	Válido	Comentários
a3	Sim	Embora contenha um número, o nome a3 inicia com letra.
velocidade	Sim	Nome formado com letras.
velocidade90	Sim	Nome formado por letras e números, mas inicia com letras.
salario_médio	Sim	O símbolo (“_”) é permitido e facilita a leitura de nomes grandes.
salario médio	Não	Nomes de variáveis não podem conter espaços em branco.
salário	Sim	O sublinha (“”) é aceito em nomes de variáveis, mesmo no início.
5A	Não	Nomes de variáveis não podem começar com números.

Tabela 3: Exemplos de nomes válidos e inválidos.

Fonte: FRANCO, João(2010, p.09).

Atribuição e exibição de Objetos

Para que uma variável receba do usuário uma informação, usamos a sintaxe `input()` atribuindo a variável, usando o sinal gráfico de igualdade (“=”). O compilador IDLE vai aguardar até que a informação seja inserida no console e só então, dará continuidade ao programa.

Sempre que for necessário exibir na tela a informação armazenada numa variável, basta escrever `print (NOME_DA_VARIÁVEL)` numa das linhas do código. Veja nossos exemplos abaixo.

Exemplos:

Código para solicitar e exibir o nome de uma pessoa:

```
nome = input ("Digite o seu nome: ")  
print (nome)
```

Saída na IDLE do *Python*

```
Digite seu nome: Albert Einstein  
Albert Einstein
```

Código para solicitar e exibir a idade de uma pessoa:

```
idade = input ("Digite a sua idade: ")  
print (idade)
```

Saída na IDLE do *Python*

```
Digite a sua idade: 78  
78
```

Código para solicitar e exibir a altura de uma pessoa:

```
altura = input ("Digite a sua altura: ")  
print (altura)
```

Saída na IDLE do *Python*

```
Digite a sua altura: 1.71  
1.71
```

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

OBS.: Seguindo o padrão ASCII, devemos utilizar ponto (.) e não vírgula (,) para indicar o início das casas decimais. **Atenção com isso!**

É possível direcionar o tipo de dado que o usuário pode inserir no código. Para isso, antes da declaração `input()`, devemos direcionar o tipo aguardado usando `int()`, para inteiros, `float()` para decimais. Por padrão, `input()` sempre devolve string, se não for direcionado. Vamos refazer os exemplos anteriores redirecionando os tipos de variáveis:

Código para solicitar e exibir a idade de uma pessoa:

```
idade = int (input ("Digite a sua idade: "))  
print (idade)
```

Saída na IDLE do Python

```
Digite sua idade: 78  
78
```

OBS.: Neste último código, teste lançar letras, símbolos ou valores decimais e veja o que acontece.

Código para solicitar a altura de uma pessoa:

```
altura = float (input ("Digite a sua altura: "))  
print (altura)
```

Saída na IDLE do Python

```
Digite sua altura: 1.71  
1.71
```

Operadores numéricos

Não é necessário escrever um código que explique ao IDLE como fazer operações matemáticas. A linguagem de programação Python já possui uma série de funções designadas a isso. Porém é preciso declarar de forma correta sua sintaxe para obter o resultado desejado.

Operadores Aritméticos

Operação	Símbolo	Exemplo	Saída
Soma	+	5 + 7	12
Subtração	-	5 - 7	-2
Multiplicação	*	5 * 7	35
Divisão	/	5 / 7	0.71428571428571
Potenciação/Radiciação	**	5**7	78125
Resto da divisão	%	7%5	2
Potenciação/Radiciação	**	5**0.5	2.23606797749979
Arredondamento	round (valor,casa decimais)	round(5**0.5,2)	2.24

Tabela 4: Operadores aritméticos.
Fonte: FRANCO, João (2010, p.09)

Operadores de Comparação

Descrição	Símbolo	Exemplo
Menor que	<	a < 7
Maior que	>	b > 7
Menor ou igual	<=	a <= 7
Maior ou igual	>=	b >= 7
Igual (comparação)	==	a == 7

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
 GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
 GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

Diferente	!=	B != 7
-----------	----	--------

Tabela 5: Operadores comparativos.
 Fonte: FRANCO, João(2010, 09)

Operadores Lógicos

Descrição	Sintaxe	Exemplo
Negação	not	not a
Aditivo	and	a < 10 and b > 5
Comutativo	or	a < 10 or b > 5

Tabela 6: Operadores lógicos.
 Fonte: BORGES, Luiz Eduardo (2010)

Orientações para a realização de atividades

Exercícios:

- 1.1. Faça um Programa que peça as 4 notas bimestrais e mostre a média.
- 1.2. Faça um programa que converta metros para centímetros.
- 1.3. Faça um Programa que peça o raio de um círculo, calcule e mostre a sua área e o comprimento.
- 1.4. Escreva um programa que recebe dois valores do tipo inteiro (x e y), e calcule o valor de z com três casas decimais de precisão:

$$z = \frac{(x^2+y^2)}{(x-y)^2}$$

- 1.5 Escreva um programa que recebe dois valores decimais, um da massa corpórea (m) e outro da altura (h) e informe a IMC da pessoa usando a fórmula:

$$IMC = \frac{m}{(h)^2}$$

- 1.6 Faça um Programa que pergunte quanto você ganha por hora e o número de horas trabalhadas no mês. Calcule e mostre o total do seu salário no referido mês, sabendo-se que são descontados 11% para o Imposto de Renda, 8% para o INSS e 5% para o sindicato, faça um programa que nos dê:

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

- a. salário bruto.
- b. quanto pagou ao INSS.
- c. quanto pagou ao sindicato.
- d. o salário líquido.
- e. calcule os descontos e o salário líquido, conforme a tabela abaixo:

+ Salário Bruto: R\$
- IR (11%): R\$
- INSS (8%): R\$
- Sindicato (5%): R\$
= Salário Líquido: R\$

Obs.: Salário Bruto - Descontos = Salário Líquido.

3. Concatenação e Interpolação de Strings

É uma variável utilizada para armazenar letras e palavras. Uma vez armazenadas, elas ficam imutáveis. Isto é, só podem ser reescritas, mas não podem ser editadas. As funções mais comuns de manipulação com Strings são as de Concatenação e Interpolação:

- Concatenação consiste na alternância entre texto e variáveis na tela de exibição do console da IDLE. Veja o exemplo:

Um código que solicita e exibe o nome e a data de nascimento daquela pessoa:

```
nome = input ("Digite o seu nome: ")
nascimento = input ("Digite sua data de nascimento: ")
print ("O sr. ou sra. " + nome + " nasceu no dia: " + nascimento)
```

Saída na IDLE do *Python*

```
Digite seu nome: Albert Einstein
Digite a data de nascimento: 14/03/1879
O sr. ou sra. Albert Einstein nasceu no dia 14/03/1879
```

- Interpolação usa o operador “%” para atribuir a exibição da informação contida numa variável que deve ser corretamente declarada de acordo com o seu tipo. Segue um exemplo de como aplicar:

Um código que solicita e exibe o nome e a altura daquela pessoa:

```
nome = input ("Digite o seu nome: ")
altura = float (input ("Digite a sua altura: "))
print ("O/A sr./sra. %s tem a altura de %.2f metros." %(nome,
altura))
```

Saída na IDLE do *Python*

```
Digite seu nome: Albert Einstein
Digite sua altura: 1.7
O sr. ou sra. Albert Einstein tem a altura de 1.70 metros
```

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

O uso de interpolação é mais indicado para a economia da alocação de memória pelo programa em *Python*.

Símbolos usados na interpolação:

- %s: *string*.
- %d: inteiro.
- %o: octal.
- %x: hexadecimal.
- %f: real.
- %e: real exponencial.
- %%: sinal de percentagem.

OBS. No código usado para fazer a interpolação de dados com escrita foi acrescentado ao operador "%f" o valor ".2" ficando "%.2f". Com isso limitamos a exibição para duas casas decimais após à vírgula.

4. Estrutura de Decisão

Para controlar o fluxo de um programa podemos utilizar as estruturas de decisão. A partir de parâmetros estabelecidos uma variável pode atingir o estado de verdadeira ou falsa. Se verdadeira, executa o código no interior da condicionante. Se falso, passa para a sessão seguinte.

- **Laço de repetição if**

O uso do if coordena se uma sequência do código deve ou não ser executada. Ela precisa de um parâmetro para comparar o estado da variável e com isso distribuir o fluxo do software. A sua estrutura tem o seguinte formato:

```
if "condição":  
    "código"
```

Exemplo:

Um programa que identifica se o usuário pode dirigir de acordo com a sua idade:

```
idade = int (input ("Qual a sua idade? "))  
if idade < 18:  
    print ("Não pode dirigir")
```

Saída na IDLE do *Python*

```
Qual a sua idade? 15  
Não pode dirigir
```

- **Laço de repetição if...else**

Semelhante ao if, essa estrutura executa um código quando parâmetro assume o estado de verdadeiro, porém há também um código caso o critério atinja o estado de falso. Devemos seguir o seguinte formato:

```
if "condição1":  
    "código1"  
else:  
    "código2"
```

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

Exemplo:

Um programa que identifica se o usuário pode ou não dirigir de acordo com a sua idade:

```
idade = int (input ("Qual a sua idade? "))  
if idade < 18:  
    print ("Não pode dirigir")  
else:  
    print ("Você é o cara")
```

Saída na IDLE do *Python*

```
Qual a sua idade? 20  
Você é o cara
```

- **Laço de repetição if...elif...else**

Caso tenha diversas condições de comparação é possível associar cada uma delas a uma estrutura do elif. Veja a sintaxe:

```
if "condição1":  
    "código1"  
  
elif "condição2":  
    "código2"  
elif "condiçãoN":  
    "códigoN"  
else:  
    "código de default"
```

Exemplo:

Um programa que identifica se o usuário pode ou dirigir e votar de acordo com a sua idade:

```
idade = int (input ("Qual a sua idade? "))  
if idade < 16:  
    print ("Não pode dirigir e não pode votar")  
elif idade >=16 and idade < 18:  
    print ("Pode votar mas não pode dirigir")  
elif idade >= 18  
    print ("Pode votar e pode dirigir")
```

Saída na IDLE do *Python*

Qual a sua idade? 17
Pode votar, mas não pode dirigir

Orientações para a realização de atividades

Exercícios:

2.1. Faça um programa que leia 2 notas de um aluno, calcule a média e imprima “aprovado” ou “reprovado” (para ser aprovado a média deve ser no mínimo 6).

2.2. Refaça o exercício 1, identificando o conceito “aprovado” (média superior a 6), “exame” (média entre 4 e 6) ou “reprovado” (média inferior a 4).

2.3. Faça um programa que pergunte o preço de três produtos e informe qual produto você deve comprar, sabendo que a decisão é sempre pelo mais barato.

2.4. Faça um Programa que peça os 3 lados de um triângulo. O programa deverá informar se os valores podem ser um triângulo. Indique, caso os lados formem um triângulo, se o mesmo é: equilátero, isósceles ou escaleno.

Dicas: Três lados formam um triângulo quando a soma de quaisquer dois lados for maior que o terceiro;

Triângulo Equilátero: três lados iguais;

Triângulo Isósceles: quaisquer dois lados iguais;

Triângulo Escaleno: três lados diferentes;

2.5. Faça um programa que calcule as raízes de uma equação do segundo grau, na forma $Ax^2 + Bx + C = 0$. O programa deverá pedir os índices de A, B e C e fazer as consistências, informando ao usuário nas seguintes situações:

- Se o valor de A igual a zero, a equação não é do segundo grau e o programa não deve pedir os demais valores, sendo encerrado;

- Se o discriminante (delta) calculado for negativo, a equação não possui raízes reais e iguais. Informe ao usuário e encerre o programa;

- Se o discriminante (delta) calculado for igual a zero a equação possui duas raízes reais e iguais; informe ao usuário;

- Se o discriminante (delta) for positivo, a equação possui duas raízes reais e distintas; informe-as ao usuário;

2.6. Um posto está vendendo combustíveis com a seguinte tabela de descontos:

Álcool:

- até 20 litros, desconto de 3% por litro

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

- acima de 20 litros, desconto de 5% por litro
- Gasolina:
- até 20 litros, desconto de 4% por litro
 - acima de 20 litros, desconto de 6% por litro.

Escreva um algoritmo que leia o número de litros vendidos, o tipo de combustível (codificado da seguinte forma: A-álcool, G-gasolina), calcule e imprima o valor a ser pago pelo cliente sabendo-se que o preço do litro da gasolina é R\$ 2,50 o preço do litro do álcool é R\$ 1,90.

2.7. O Hipermercado Tabajara está com uma promoção de carnes que é imperdível. Confira:

	Até 5 Kg	Acima de 5 Kg
o File Duplo	R\$ 4,90 por Kg	R\$ 5,80 por Kg
o Alcatra	R\$ 5,90 por Kg	R\$ 6,80 por Kg
o Picanha	R\$ 6,90 por Kg	R\$ 7,80 por Kg

Cada cliente poderá levar apenas um dos tipos de carne da promoção, porém não há limites para a quantidade. Caso a compra seja feita com o cartão Tabajara, o cliente receberá ainda um desconto de 5% sobre o total da compra. Escreva um programa que peça o tipo, a quantidade de carne comprada pelo usuário e gere um cupom fiscal, contendo as informações da compra: tipo e quantidade de carne, preço total, tipo de pagamento, valor do desconto e valor a pagar.

Desafio:

2.8. Faça um programa que leia dois números e em seguida um caracter que representa um operador aritmético (+, -, * ou /). Seu programa então deve imprimir o resultado do operador aplicado aos dois números dados.

- **Laço de repetição while**

A partir de uma condição, o laço while repete a rotina enquanto o parâmetro designado para ela não se torna verdadeiro. Quando isso acontece, passa ao trecho seguinte do código. É preferível usar esse laço de repetição quando não se conhece o número total de interações que o usuário vai ter. Veja a sintaxe:

```
while "condição":
    "código"
```

Exemplo:

Um código de acesso mediante a cadastro e reconhecimento de senha:

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

```
senha = input ("Cadastre a senha: ")
leitura = input ("Digite a senha: ")
while leitura != senha:
    leitura = input ("Digite a senha correta: ")
    if leitura == senha:
        print ("Acesso liberado")
    else:
        print ("Senha Incorreta")
```

Saída na IDLE do *Python*:

```
Cadastre a senha: 123456
Digite a senha correta: abcde
Senha Incorreta
Digite a senha correta: 654321
Senha Incorreta
Digite a senha correta: 123456
Acesso liberado
```

- **Laço while True**

Associado ao laço while podemos introduzir uma variável de controle que mantém a repetição até que uma interação com o/a usuário/a a torne falsa e saia do laço ou encerre o programa.

Veja o exemplo da sintaxe:

```
while True:
    "código"
    if "condição":
        break
```

A instrução break ordena que o while seja interrompido e volte ao programa raiz, o uso dela é opcional.

Outras sugestões de atividades

Exercícios:

3.1. Faça um programa que peça uma nota, entre zero e dez. Mostre uma mensagem caso o valor seja inválido e continue pedindo até que o usuário informe um valor válido.

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

3.2. Supondo que a população de um país A seja da ordem de 80.000 habitantes com uma taxa anual de crescimento de 3% e que a população de B seja 200.000 habitantes com uma taxa de crescimento de 1.5%. Faça um programa que calcule e escreva o número de anos necessários para que a população do país A ultrapasse ou iguale a população do país B, mantidas as taxas de crescimento.

3.3. Faça um programa que receba dois números inteiros e gere os números inteiros que estão no intervalo compreendido por eles.

3.4. Desenvolva um gerador de tabuada, capaz de gerar a tabuada de qualquer número inteiro entre 1 a 10. O usuário deve informar de qual número ele deseja ver a tabuada. A saída deve ser conforme o exemplo abaixo:

Tabuada de 5:

$$5 \times 1 = 5$$

$$5 \times 2 = 10$$

...

$$5 \times 10 = 50$$

3.5. Faça um programa que calcule o fatorial de um número inteiro fornecido pelo usuário. Ex.: $5! = 5.4.3.2.1=120$

3.6. Numa eleição existem três candidatos. Faça um programa que peça o número total de eleitores. Peça para cada eleitor votar e ao final mostrar o número de votos de cada candidato.

5. Coleções do tipo Listas

Um conjunto de objetos que podem ser heterogêneos ou até mesmo uma outra lista com atribuição de índices com localização. A principal característica de uma lista é que ela pode ser alterada a qualquer momento.

Para declarar uma lista é preciso seguir a escrita:

```
nome_da_lista = ['valor1', 'valor2', 'valor3', ..., 'valorN']
```

Exemplo:

Um código que cria uma lista com o nome das capitais do Nordeste do Brasil:

```
capitais = []  
i = 0  
while i < 9:  
    cidade = capitais.append(input("Digite a Capital: "))  
    i = i + 1  
print (capitais)
```

Saída na IDLE do *Python*:

```
Nome da Capital: Recife  
Nome da Capital: João Pessoa  
Nome da Capital: Maceió  
Nome da Capital: Salvador  
Nome da Capital: Natal  
Nome da Capital: Teresina  
Nome da Capital: São Luís  
Nome da Capital: Fortaleza  
Nome da Capital: Aracaju  
['Recife', 'João Pessoa', 'Maceió', 'Salvador', 'Natal',  
'Teresina', 'São Luís', 'Fortaleza', 'Aracaju']
```

A lista `capitais` inicialmente está vazia. Neste caso, o laço `while` foi utilizado para repetir a ação de preenchimento até que o parâmetro fosse concluído (mudar do estado

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

False para o True). Cada vez que o laço era percorrido, a variável *i* é incrementada em uma unidade. De zero até oito temos nove repetições, que é a quantidade total de capitais no Nordeste, isto é reflexo da escolha do menor que “<”. Se colocado menor igual que “<=”, ocorreriam 10 repetições que desvirtua o objetivo. No final, o programa exibe a lista capitais com os dados na ordem cronológica.

É possível manipular e exibir os valores de uma lista utilizando as sintaxes certas. Como foi feito dentro do while com a lista capitais. Usamos o método append para atribuir a entrada da informação armazenada na variável cidade ao final da lista. Veja outras aplicações:

Função	Descrição	Exemplo
len	retorna o tamanho da lista.	L = [1, 2, 3, 4] len(L) → 4
min	retorna o menor valor da lista.	L = [10, 40, 30, 20] min(L) → 10
max	retorna o maior valor da lista.	L = [10, 40, 30, 20] max(L) → 40
sum	retorna a soma dos elementos da lista.	L = [10, 20, 30] sum(L) → 60
append	adiciona um novo valor no final da lista.	L = [1, 2, 3] L.append(100) L → [1, 2, 3, 100]
extend	insere uma lista no final de outra lista.	L = [0, 1, 2] L.extend([3, 4, 5]) L → [0, 1, 2, 3, 4, 5]
del	remove um elemento da lista, dado seu índice.	L = [1,2,3,4] del L[1] L → [1, 3, 4]
in	verifica se um valor pertence à lista.	L = [1, 2, 3, 4] 3 in L → True
sort()	ordena em ordem crescente de valor.	L = [3, 5, 2, 4, 1, 0] L.sort() L → [0, 1, 2, 3, 4, 5]
reverse()	inverte os elementos de uma lista.	L = [0, 1, 2, 3, 4, 5] L.reverse() L → [5, 4, 3, 2, 1, 0]
list(i-1)	acessar o enésimo termo de uma lista. (O primeiro termo de uma lista ocupa a posição zero.)	L = [0, 1, 2, 3, 4, 5] L.list (3) 3

Tabela 7: Operadores para manipular listas.

Orientações para a realização de atividades

Exercícios:

4.1. Dada a lista $L = [5, 7, 2, 9, 4, 1, 3]$, escreva um programa que imprima as seguintes informações:

- tamanho da lista.
- maior valor da lista.
- menor valor da lista.
- soma de todos os elementos da lista.
- listar em ordem crescente.
- listar em ordem decrescente.

4.2. Gere uma lista contendo os múltiplos de 3 entre 1 e 50.

4.3. Faça um Programa que leia um vetor de 5 números inteiros e mostre-os.

4.4. Faça um Programa que leia um vetor de 10 números reais e mostre-os na ordem inversa.

4.5. Faça um Programa que leia 10 números inteiros e armazene-os num vetor. Separe os números pares no vetor PAR e os números ÍMPARES no vetor ímpar. Imprima os três vetores.

4.6. Faça um programa que receba a temperatura média de cada mês do ano e armazene-as em uma lista. Após isto, calcule a média anual das temperaturas e mostre todas as temperaturas acima da média anual, e em que mês elas ocorreram (mostrar o mês por extenso: 1 – Janeiro, 2 – Fevereiro,..).

4.7. Crie uma lista com os nomes dos super-heróis que devem participar da Iniciativa Vingadores seguindo a ordem:

- Homem de Ferro
- Capitão América
- Thor
- Hulk
- Viúva Negra
- Gavião Arqueiro

4.8. No exercício anterior, inclua o Homem-Aranha no final da lista e imprima em qual posição está o Thor.

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

4.9. No exercício anterior, infelizmente, a Viúva Negra e o Homem de Ferro não fazem mais parte da Iniciativa Vingadores, então retire-os da lista.

4.10. Em uma competição de ginástica, cada atleta recebe votos de sete jurados. A melhor e a pior nota são eliminadas. A sua nota fica sendo a média dos votos restantes. Você deve fazer um programa que receba o nome do ginasta e as notas dos sete jurados alcançadas pelo atleta em sua apresentação e depois informe a sua média, conforme a descrição acima informada (retirar o melhor e o pior salto e depois calcular a média com as notas restantes).

As notas não são informadas, ordenadas conforme o exemplo abaixo:

```
Atleta: Aparecido Parente  
Nota: 9.9  
Nota: 7.5  
Nota: 9.5  
Nota: 8.5  
Nota: 9.0  
Nota: 8.5  
Nota: 9.7
```

```
Resultado final:  
Atleta: Aparecido Parente  
Melhor nota: 9.9  
Pior nota: 7.5  
Média: 9.04
```

6. Coleções do tipo Tuplas

Semelhante a lista, tuplas armazenam dados que podem ser heterogêneos ou mesmo outras tuplas. A principal característica das tuplas é que elas são imutáveis, ou seja, seus elementos não podem ser alterados.

Declaração de uma tupla:

```
nome_da_tupla = ('valor1', 'valor2', ..., 'valorN')
```

Dicionários

Agrupa valores associados a uma chave de acesso. Declaração:

```
nome_do_dicionário = {chave1: valor1,
                      chave2: valor2,
                      ...,
                      chaveN: valorN}
```

Exemplo:

Uma lista de produtos e respectivos preços:

```
preco = {"arroz": 6.90, "feijão": 12.39, "farinha": 7.95,
         "macarrão": 5.86, "biscoito": 2.30}
print (preco)
print (preco["farinha"])
```

Saída na IDLE:

```
{'arroz': 6.9, 'feijão': 12.39, 'farinha': 7.95, 'biscoito': 2.3}
7.95
```

Alguns comandos podem ser usados para manipular o conteúdo dos dicionários.

Segue a tabela referente a isso:

Função	Descrição	Exemplo	
del	Exclui o item	del preco {"biscoito"}	
in	Verifica a existência da chave	"batata" in preco False	"feijão" in preco True

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
 GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
 GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

keys()	Exibe as chaves	preco.keys() dict_keys(['arroz', 'feijão', 'farinha', 'macarrão', 'biscoito'])
values()	Exibe os valores	preco.values() dict_values([6.9, 12.39, 7.95, 5.86, 2.3])
items()	Retornar uma lista contendo pares de tuplas (o primeiro elemento será a chave do e o segundo elemento o valor).	preco.items() dict_items([('arroz', 6.9), ('feijão', 12.39), ('farinha', 7.95), ('macarrão', 5.86), ('biscoito', 2.3)])

Tabela 8: Operadores para manipular dicionários.
 Fonte: FRANCO, João(2010, p. 14).

Para inserir valores no dicionário procedemos da seguinte forma de escrita:

```
nome_do_dicionário ["chave"] = valor
```

Exemplo:

Para reintroduzir a chave “biscoito” no dicionário:

```
precos ["biscoito"] = 2.35
```

Orientações para a realização de atividades

Exercícios:

5.1. Dada a tabela a seguir, crie um dicionário que a represente:

Lanchonete	
Produtos	Preços R\$
Salgado	R\$ 4.50
Lanche	R\$ 6.50
Suco	R\$ 3.00
Refrigerante	R\$ 3.50
Doce	R\$ 1.00

Fonte: FRANCO, João (2010, p.14).

5.2. Considere um dicionário com 5 nomes de alunos e suas notas. Escreva um programa que exiba a lista de alunos e notas e calcule a média dessa turma.

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

5.3. Crie um dicionário “d” e coloque nele seus dados: nome, idade e telefone. Usando o dicionário “d”, imprima seus dados.

5.4. Faça um programa, utilizando Dicionários, que peça ao usuário para inserir os nomes de três produtos do mercado, seus respectivos preços e os mostre na tela.

5.5. Faça um Programa que peça o nome, a idade e a altura de 5 pessoas, armazene cada informação no seu respectivo vetor. Imprima o nome, a idade e a altura desse grupo.

- **Laço de repetição for**

Essa estrutura de repetição possui um controle de ciclos controlado pela declaração de uma variável. Pode ser parametrizada por sequência numérica (range) ou mesmo por uma lista. Forma de argumentação:

```
for "variável" in "sequência":  
    "código"
```

- **Função range**

A função range(m, n, p) é muito útil em laços pois retorna uma lista de inteiros começando em m, menores que n e em passos de comprimento p, que podem ser usados como sequência.

Exemplo:

Programa que exhibe múltiplos de 3 entre 6 e 25:

```
i = 0  
for i in range (6, 25, 3):  
    print (i)
```

Saída na IDLE:

```
6  
9  
12  
15  
18  
21  
24
```

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

Um código que determina a soma dos valores inteiros entre 0 e 100:

```
s = 0
for x in range (1,100):    #na falta do "passo" valor é 1
    s = s + x
print ("Soma ", s)
```

Saída na IDLE Python:

```
Soma 4950
```

Outras sugestões de atividades

Exercícios:

- 6.1. Escreva um programa para encontrar a soma $S = 3 + 6 + 9 + \dots + 333$.
- 6.2. Escreva um programa que leia 10 notas e informe a média do aluno.
- 6.3. Escreva um programa que leia um número de 1 a 10, e mostre a tabuada desse número.
- 6.4. Faça um programa que peça as quatro notas de 5 alunos, calcule e armazene num vetor a média de cada aluno.
- 6.5. Usando o código anterior, imprima o número de alunos com média maior ou igual a 7.0.
- 6.5. Elaborar um programa para ler 10 números e imprimir a soma, o maior e o menor desses números.
- 6.6. Elaborar um programa para calcular a soma de 1 até 50.

7. Sub rotina de repetição do tipo Funções

Um conjunto de códigos é executado quando a função é chamada dentro do programa principal. Uma função admite parâmetros ou argumentos que podem ser utilizados dentro dela inclusive gerando resultados a ser utilizado de volta ao escopo principal usando a sintaxe return.

Como declarar uma função:

```
def "nome_da_função" ("parâmetros/argumentos"):  
    "Bloco de códigos"  
    return "variável"
```

O uso da diretiva return é opcional.

Um código que recebe dois valores e devolve a soma deles (usando função):

```
def soma (x,y):  
    total = x + y  
  
x = float(input("Digite um número: "))  
y = float(input("Digite outro número: "))  
s = soma (x,y)  
print ("Soma = %.2f" %(s))
```

Saída na IDLE Python:

```
Digite um número: 8  
Digite outro número: 4  
Soma = 12.00
```

Pré-código chamado de Biblioteca

Uma biblioteca é um pré-código com uma coleção de rotinas e funções que encurta o tempo de desenvolvimento de um programa. Ao instalar a IDLE *Python*, algumas

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

bibliotecas já vêm instaladas e podem ser acrescentadas ao código desde que sejam convocadas. Há duas maneiras de fazer isso.

Sintaxe para acrescentar bibliotecas:

```
import "nome_da_biblioteca"
```

É possível acrescentar apenas uma função específica da biblioteca:

```
from "nome_da_biblioteca" import "função"
```

Exemplos de Bibliotecas Nativas:

Matemática: o módulo `math` define funções logarítmicas, exponenciais, trigonométricas, hiperbólicas e conversões angulares, entre outras.

Exemplo do uso da biblioteca Matemática para sortear um número entre 1 e 10. Em seguida escolhe aleatoriamente um valor decimal entre 0 e 1.

```
import random
#escolha um inteiro entre 1 e 10
print(random.randrange(1,11))
#escolha um valor decimal entre 0 e 1
print(random.random())
```

Saída na IDLE Python:

```
9
0.74182277131113139
```

Tempo: o Python possui dois módulos para lidar com tempo:

- *time*: implementa funções que permitem utilizar o tempo gerado pelo sistema.
- *datetime*: implementa tipos de alto nível para realizar operações de data e hora.

Uso da biblioteca *time*:

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

```
import time
#retorna uma coleção com ANO, MÊS, DIA, HORA, MINUTO, SEGUNDO,
#DIA DA SEMANA, DIA DO ANO e HORÁRIO DE VERÃO.
print(time.localtime())
#mostra a DATA e HORA atual como string
print(time.asctime())
#faz uma contagem de 5 segundos
for i in range(5):
    time.sleep(1)
    print(i+1, 's')
```

Saída na IDLE Python:

```
time.struct_time(tm_year=2024, tm_mon=4, tm_mday=23, tm_hour=14,
tm_min=45, tm_sec=28, tm_wday=1, tm_yday=114, tm_isdst=0)
Tue Apr 23 14:45:28 2024
1 s
2 s
3 s
4 s
5 s
```

Em *datetime*, estão definidos quatro tipos para representar o tempo:

- *datetime*: data e hora.
- *date*: apenas data.
- *time*: apenas hora.
- *timedelta*: diferença entre tempos.

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

Aplicações da biblioteca *datetime*:

```
import datetime
#guarda no objeto ANO, MÊS, DIA, HORA, MINUTO, SEGUNDO
dt = datetime.datetime(2030,4,23,14,56,50)
#Manipula o objeto date e time
data = dt.date()
hora = dt.time()
#Quanto tempo falta para 23/04/2030
dd = dt - dt.today()
print('Data:',dt)
print('Hora:',hora)
print('Faltam ',dd, ' para', dt)
```

Saída IDLE Python:

```
Data: 2030-04-23 14:56:50
Hora: 14:56:50
Faltam 2190 days, 23:35:17.468993 para 2030-04-23 14:56:50
```

Exemplos de bibliotecas para Python que estão disponíveis na internet

NumPy:

Considerada uma biblioteca fundamental para todos que trabalham com computação numérica em Python, já que essa ferramenta permite que se trabalhe com tratamento de dados. Oferece suporte a arrays multidimensionais e funções matemáticas essenciais para operações numéricas eficientes.

Matplotlib:

Biblioteca de gráficos que oferece visualizações poderosas e atraentes. É amplamente empregada em Python para a visualização de dados. Além disso, pode servir como substituta do MATLAB, com as vantagens de ser gratuita e de código aberto. Matplotlib é utilizada para criar visualizações estáticas, animadas e interativas em Python, gerar figuras com qualidade de publicação, disponibilizar uma ampla gama de gráficos e diagramas.

Tkinter:

É uma biblioteca padrão em Python com foco na criação de interfaces gráficas sem tantos recursos. Simples de usar para o desenvolvimento de aplicações básicas e intermediárias. Conta com widgets pré definidos e fácil integração.

Pygame:

Usada na criação de aplicações multimídia como jogos. Ela é uma camada de abstração construída sobre a biblioteca SDL, e assim como a SDL ela é altamente portátil e roda em praticamente todas plataformas e sistemas operacionais.

Sugestões para a realização de atividades

Elaborar um programa Python para calcular a distância entre dois pontos no eixo cartesiano.

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

Exemplo de respostas:

Questão 1.1.

```
nota1 = float(input("primeira nota: "))
nota2 = float(input("segunda nota: "))
nota3 = float(input("terceira nota: "))
nota4 = float(input("quarta nota: "))
media = (nota1+nota2+nota3+nota4)/4
print ("A média é: ")
print (media)
```

Questão 1.2.

```
distancia = float(input("Digite a distância em metros: "))
resposta = distancia*100
print ("São ")
print (resposta)
print ("cm")
```

Questão 1.3.

```
raio = float(input("Digite o raio do círculo em metros: "))
area = 3.14*raio**2
comprimento = 2*3.14*raio
print ("A área é de: ")
print (round (area,2))
print ("O comprimento é de: ")
print (round(comprimento,2))
```

Questão 1.4.

```
x = int (input ("valor de x: "))
y = int (input ("Valor de y: "))
z = (x**2+y**2)/(x-y)**2
print ("Valor de z: ")
print (round(z,3))
```

Questão 1.5

```
m = float (input ("Massa corporal: "))
h = float (input ("Altura: "))
IMC = m/h**2
print ("Valor do IMC: ")
print (round(IMC,2))
```

Questão 1.6

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

```
salario_bruto = float (input ("Valor do salário: "))
IR = salario_bruto*0.11
INSS = salario_bruto*0.08
sindicato = salario_bruto*0.05
salario_liquido = salario_bruto - IR - INSS - sindicato
print ("Imposto de renda: R$ ")
print (round(IR,2))
print ("Contribuição do INSS: R$ ")
print (round(INSS,2))
print ("Sindicalização: R$ ")
print (round(sindicato,2))
print ("Valor do salário líquido: R$")
print (round(salario_liquido,2))
```

Questão 2.1.

```
notal = float (input("Primeira nota: "))
nota2 = float (input("Segunda nota: "))
media = (notal+nota2)/2
if media > 6:
    print("Aprovado")
else:
    print("Reprovado")
```

Questão 2.2

```
notal = float (input("Primeira nota: "))
nota2 = float (input("Segunda nota: "))
media = (notal+nota2)/2
if media > 6:
    print("Aprovado")
elif media > 4 and media < 5.9:
    print("Exame")
else:
    print("Reprovado")
```

Questão 2.3.

```
produto1 = float(input("Primeiro valor (R$): "))
produto2 = float(input("Segundo valor (R$): "))
produto3 = float(input("Terceiro valor (R$): "))

if produto1 >= produto2:
    menor_preco = produto2

elif produto2 >= produto3:
    menor_preco = produto3

else:
    menor_preco = produto1

print ("Menor preço: ")
print (menor_preco)
```

Questão 2.4.

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

```
ladoA = float (input("Lado A: "))
ladoB = float (input("Lado B: "))
ladoC = float (input("Lado C: "))

if ladoA + ladoB < ladoC:
    print("Não é um triângulo")
elif ladoA + ladoC < ladoB:
    print("Não é um triângulo")
elif ladoC + ladoB < ladoA:
    print("Não é um triângulo")
else:
    print("É um triângulo... ")
    if ladoA == ladoB and ladoA == ladoC:
        print("equilátero")
    elif ladoA == ladoB or ladoB == ladoC:
        print("isóceles")
    else:
        print("escaleno")
```

Questão 2.5.

```
A = float(input("Índice A: "))
B = float(input("Índice B: "))
C = float(input("Índice C: "))

if A == 0:
    print ("Não é uma equação do 2 grau")
else:
    delta = B**2-4*A*C
    if delta < 0:
        print ("Não possui raízes reais")
    elif delta == 0:
        X = -B/2*A
    else:
        X1 = (-B + delta**0.5)/2*A
        X2 = (-B - delta**0.5)/2*A
        print ("Raízes: ")
        print (X1)
        print (" ")
        print (X2)
```

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

Questão 2.6.

```
tipo = input ("Tipo de combustivel: 'A'- Álcool ou 'G'- Gasolina ")
volume = float (input("Quantidade (Litros): "))
precoA = 1.9
precoG = 2.5

if tipo == 'A':
    if volume < 20:
        preco = volume*precoA - volume*precoA*0.03
    else:
        preco = volume*precoA - volume*precoA*0.05

elif tipo == 'G':
    if volume < 20:
        preco = volume*precoG - volume*precoG*0.04
    else:
        preco = volume*precoG - volume*precoG*0.06

print ("Preço: ")
print (round (preco,2))
```

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

Questão 2.7.

```
file_duplo1 = 4.9
file_duplo2 = 5.8
alcatra1 = 5.9
alcatra2 = 6.8
picanha1 = 6.9
picanha2 = 7.8

escolha = input("Tipo de carne: 'F' - Filé, 'A' - Alcatra e 'P' - Picanha ")
quantidade = float(input("Qual a quantidade (Kg)? "))

if escolha == 'F':
    escolha = "Filé_duplo"
    if quantidade < 5:
        preco = file_duplo1*quantidade
    else:
        preco = file_duplo2*quantidade

if escolha == 'A':
    escolha = "Alcatra"
    if quantidade < 5:
        preco = alcatra1*quantidade
    else:
        preco = alcatra2*quantidade

if escolha == 'P':
    escolha = "Picanha"
    if quantidade < 5:
        preco = picanha1*quantidade
    else:
        preco = picanha2*quantidade

pagamento = input("Avista '1', Cartão '2': ")
if pagamento == '1':
    pagamento = "A vista"
    desconto = 0
else:
    pagamento = "Cartão"
    desconto = preco*0.05

valor_total = preco - desconto

print(" ")
print("-*-*-*- FATURA *-*-*-*-")
print(" ")
print(escolha)
print("%.2f Kg" %(quantidade))
print("Preço total: R$ %.2f" %(preco))
print("Forma de pagamento: " + pagamento)
print("Desconto: R$ %.2f" %(desconto))
print("Valor total: R$ %.2f" %(valor_total))
```

Questão 3.1

```
nota = float(input("Qual a nota (de 0.0 à 10.0)? "))

while nota < 0 or nota > 10:
    nota = float(input("Insira uma nota válida (de 0.0 à 10.0)"))

if nota > 0 and nota < 10:
    print("Obrigado por informar a nota: %.1f" %(nota))
```

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

Questão 3.2

```
paisA = 80000
paisB = 200000
anos = 0
while paisA <= paisB:
    paisA = paisA + paisA*0.03
    paisB = paisB + paisB*0.015
    anos = anos + 1

if paisA > paisB:
    print("%i anos"%(anos))
```

Questão 3.3.

```
valor1 = int(input("Digite um inteiro: "))
valor2 = int(input("Digite outro inteiro: "))

while valor2 > valor1:
    valor1 = valor1 + 1
    print (valor1)
```

Questão 3.4.

```
tabuada = int (input("Número: "))
n = 0
produto = 0
while n < 11:
    produto = tabuada*n
    print ("%i x %i = %i" %(tabuada,n,produto))
    n = n + 1
```

Questão 3.5.

```
n = int(input("Entre com um número inteiro positivo: "))
fat = 1
i = 2

while i <= n:
    fat = fat * i
    i = i + 1

print("Resultado:", fat)
```

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

Questão 3.6.

```
n = int (input ("Número de votantes: "))
candidatoA = 0
candidatoB = 0
candidatoC = 0

while n != 0:

    voto = int (input ("Número do candidato: "))

    if voto == 1:
        candidatoA = candidatoA + 1
        print ("voto computado")
    elif voto == 2:
        candidatoB = candidatoB + 1
        print ("voto computado")
    elif voto == 3:
        candidatoC = candidatoC + 1
        print ("voto computado")
    else:
        print ("voto anulado")
    n = n - 1

print (candidatoA, candidatoB, candidatoC)
```

Questão 4.1.

```
L = [5,7,2,9,4,1,3]

print ("Tamanho: %i" %(len(L)))
print ("Maior valor: %i" %(max(L)))
print ("Menor valor: %i" %(min(L)))
print ("Soma dos termos: %i" %(sum(L)))
print ("Ordenamento crescente")
L.sort()
print (L)
print ("Ordenamento decrescente")
L.reverse()
print (L)
```

Questão 4.2.

```
L = []
i = 3

while i <= 50:
    L.append (i)
    i = i + 3

print (L)
```

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

Questão 4.3.

```
L = []
i = 0

while i < 5:
    L.append (input("Digite o valor: "))
    i = i + 1

print (L)
```

Questão 4.4.

```
L = []
i = 0

while i < 10:
    L.append (input("Digite o valor: "))
    i = i + 1
L.reverse()
print (L)
```

Questão 4.5.

```
L = []
PAR = []
IMPAR = []
i = 0

while i < 10:
    valor = int(input("Digite um valor inteiro: "))
    L.append(valor)

    if valor % 2 == 0:
        PAR.append(valor)
    else:
        IMPAR.append(valor)
    i = i + 1

print (" ")
print ("Sua lista: ")
print (L)
print ("Lista dos pares: ")
print (PAR)
print ("Lista dos impares: ")
print (IMPAR)
```

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

Questão 4.6.

```
meses = ['Jan', 'Fev', 'Mar', 'Abr', 'Mai', 'Jun', 'Jul', 'Ago', 'Set', 'Out', 'Nov',  
'Dez']  
temp = []  
maximaValor = []  
maximaMes = []  
  
i = 0  
while i < 12:  
    leitura = temp.append (float(input("Leitura: ")))  
    i = i + 1  
  
tempMax = max(temp)  
tempMedia = sum(temp)/12  
  
j = 0  
while j < 12:  
    if tempMedia < temp [j]:  
        maximaValor.append (temp [j])  
        maximaMes.append (meses[j])  
    j = j + 1  
  
n = len(maximaValor)  
  
print ("Temperatura Máxima: %.2f°C" %(tempMax))  
print ("Temperatura Média: %.2f°C" %(tempMedia))  
  
k = 0  
while k != n:  
    print ("%.2f°C - %s" %(maximaValor[k],maximaMes[k]))  
    k = k + 1
```

Questão 4.7.

```
vingadores = ["Homem de Ferro", "Capitão América", "Thor", "Hulk", "Viúva Negra", "Gavião  
Arqueiro"]
```

Questão 4.8.

```
vingadores = ["Homem de Ferro", "Capitão América", "Thor", "Hulk", "Viúva Negra", "Gavião  
Arqueiro"]  
vingadores.append("Homem -Aranha")  
if "Thor" in vingadores:  
    print (vingadores.index("Thor"))
```

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

Questão 4.9.

```
vingadores = ["Homem de Ferro", "Capitão América", "Thor", "Hulk", "Viúva Negra", "Gavião  
Arqueiro", "Homem -Aranha"]  
if "Viúva Negra" in vingadores:  
    vingadores.remove("Viúva Negra")  
if "Homem de Ferro" in vingadores:  
    vingadores.remove("Homem de Ferro")  
print(vingadores)
```

Questão 4.10.

```
nome = input ("Atleta: ")  
notas = []  
  
i = 0  
while i < 7:  
    valor = notas.append (float(input("Nota: ")))  
    i = i + 1  
  
print (" ")  
print ("Resultado final:")  
print ("Atleta: " + nome)  
print ("Melhor nota: %.2f" %(max(notas)))  
print ("Menor nota: %.2f" %(min(notas)))  
notas.sort ()  
del notas [0]  
del notas [5]  
print ("Média: %.2f" %(sum(notas)/5))
```

Questão 5.1.

```
cardapio = {'Salgado': 4.50,  
            'Lanche': 6.50,  
            'Suco': 3,  
            'Refrigerante': 3.50,  
            'Doce': 1}  
print (cardapio)
```

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

Questão 5.2

```
caderneta = {}  
guarda_nota = 0  
i = 0  
while i < 5:  
    nome = input ("Aluno/a: ")  
    nota = float (input("Nota: "))  
    guarda_nota = nota + guarda_nota  
    caderneta [nome] = nota  
    i = i + 1  
media = guarda_nota / 5  
print (caderneta)  
print (media)
```

Questão 5.3

```
d = {}  
d ["nome"] = input ("Digite seu nome: ")  
d ["idade"] = int (input("Qual a sua idade? "))  
d ["telefone"] = int (input("Telefone: "))  
print (d)
```

Questão 5.4

```
produtos = {}  
  
for i in range(3):  
    produto = input("Digite o nome do produto: ")  
    preco = float(input("Digite o preço do produto: R$"))  
    produtos[produto] = preco  
  
print("Produtos e preços inseridos:")  
print (produtos.items())
```

Questão 5.5

```
usuarios = {}  
  
i = 0  
for i in range(5):  
    nome = input("Nome: ")  
    idade = int (input("Idade: "))  
    altura = float (input("Altura: "))  
    pessoa = []  
    pessoa.append (nome)  
    pessoa.append (idade)  
    pessoa.append (altura)  
    usuarios [i] = pessoa  
  
print (usuarios)
```

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

Questão 6.1.

```
s = 0
for x in range (3,333,3):
    s = s + x
print (s)
```

Questão 6.2.

```
nota = 0
i = 0
for i in range (0, 10):
    guarda_nota = float (input ("Digite a nota: "))
    nota = nota + guarda_nota
    i = i + 1
media = nota/10
print (media)
```

Questão 6.3.

```
numero = int (input ("Insira o número: "))
i = 0
for i in range (1, 11):
    produto = i * numero
    print ("%i x %i = %i" %(numero, i, produto))
```

Questão 6.4.

```
caderneta = {}
nota = 0
i = 0
for i in range (0,5):
    nome = input ("Nome do/a aluno/a: ")
    j = 0
    nota = 0
    for j in range (0,4):
        guarda_nota = float (input ("Digite a nota: "))
        nota = nota + guarda_nota
        j = j + 1
    media = nota/4
    caderneta [nome] = media
    i = i + 1
print (caderneta)
```

SECRETARIA EXECUTIVA DE ENSINO MÉDIO E PROFISSIONAL
GERÊNCIA GERAL DE ENSINO MÉDIO E ANOS FINAIS DO ENSINO FUNDAMENTAL
GERÊNCIA GERAL DE POLÍTICAS EDUCACIONAIS DO ENSINO MÉDIO

Questão 6.5.

```
numeros = []
for i in range(10):
    num = int(input("Digite um número: "))
    numeros.append(num)
soma = sum(numeros)
maior = max(numeros)
menor = min(numeros)
print("Soma: ", soma)
print("Maior: ", maior)
print("Menor: ", menor)
```

Questão 6.6.

```
soma = 0
for i in range(1, 51):
    soma += i
print("Soma: ", soma)
```

Questão 7.1.

```
import math
def calcular_distancia(x1, y1, x2, y2):
    distancia = math.sqrt((x2 - x1)**2 + (y2 - y1)**2)
    return distancia
# Entrada dos dados
x1 = float(input("Digite a coordenada x do primeiro ponto: "))
y1 = float(input("Digite a coordenada y do primeiro ponto: "))
x2 = float(input("Digite a coordenada x do segundo ponto: "))
y2 = float(input("Digite a coordenada y do segundo ponto: "))
# Processamento dos dados
distancia = calcular_distancia(x1, y1, x2, y2)
# Resultado
print("Distância entre os pontos: ", distancia)
```

Orientações para a realização de atividades

Avaliar se os estudantes participam de forma ativa e/ou autônoma nas atividades propostas. Como também, se demonstram conhecer e respeitar opiniões dos colegas e professores. Além disso, se conseguem discernir entre fontes de pesquisa seguras e as não confiáveis. Ademais, se sistematizam e apresentam resultados coerentes com o tema sugerido nas atividades escolares. Outrossim, se comunicam suas descobertas e conclusões em eventos diversos que podem envolver a comunidade escolar ou públicos externos. Assim como, se selecionam ferramentas tecnológicas que proporcionem vivências propositivas e ligadas ao tema gerador.

Outra forma de avaliação pode ser rubrica, exposição e/ou feira acerca da temática estudada.

8. Referências bibliográficas

Python Software Foundation. Python. Disponível em: <https://www.python.org>. Acesso em: 30 nov. 2023.

Python para Desenvolvedores / Luiz Eduardo Borges. Rio de Janeiro, Edição do Autor, 2010.

Franco, João L. . INTRODUÇÃO À PROGRAMAÇÃO COM PYTHON, Programa de Educação Tutorial. Grupo PET - ADS IFSP - Câmpus São Carlos. 2010.

Botcity. "Bibliotecas Python: uma introdução para desenvolvedores". Blog Botcity, 15 de janeiro de 2024. Disponível em:

<https://blog.botcity.dev/pt-br/2024/01/15/bibliotecas-python/>.

Acesso em: 16/04/2024;

Python Brasil. "Games - Python Brasil". Python Brasil, [s.d.]. Disponível em:

<https://python.org.br/games/>. Acesso em: 16/04/2024.

Curso de Introdução à Programação de Jogos em Python, LabTIME, UFG, 2017.

Araújo, Stenio L. Exercícios resolvidos utilizando Python. Universidade Estadual do Sudoeste da Bahia. Série de Textos Didáticos. Edições UESB. Julho, 2023. Disponível em:

https://drive.google.com/file/d/1S-ecrRhDVO28mdAWJ7qe4aN_sluqrBMi/view?usp=sharing

Oxford Languages. Google Dictionary. Disponível em:

<https://languages.oup.com/google-dictionary-pt>. Acesso em: 19 dez. 2023.

Alura. Trabalhando com o dicionário no Python. Disponível em:

https://www.alura.com.br/artigos/trabalhando-com-o-dicionario-no-python?utm_term=&utm_campaign=%5BSearch%5D+%5BPerformance%5D+-+Dynamic+Search+Ads+-+Artigos+e+Conteúdos&utm_source=adwords&utm_medium=ppc&hsa_acc=7964138385&hsa_cam=11384329873&hsa_grp=111087461203&hsa_ad=662261158752&hsa_src=g&hsa_tgt=aud-574826424850:dsa-843358956400&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gad_source=1&gclid=CjwKCAiAu9yqBhBmEiwAHTx5pyuXphAw8VWVMA--mFIelhXntDR6ty5JW0ldiDt-IKUXc7TEa4Z8ihoCy0EQAvD_BwE. Acesso em: 17 nov. 2023.

Python Academy. A função range() do Python. Disponível em:

<https://pythonacademy.com.br/blog/a-funcao-range-do-python>. Acesso em: 20 nov. 2023.

Linux Console. Disponível em: <https://pt.linux-console.net/?p=26498>.