

Sumário

1 INTRODUÇÃO	2
2 INSTALAÇÃO	3
2.1 Instalando pacotes	3
3 PRIMEIROS PASSOS EM PYTHON	6
3.1 Indentação	10
3.2 Variáveis	10
3.2.1 Tipos de Variáveis	10
3.2.2 Verificando o tipo de variável – Comando type	11
3.2.3 Operadores	11
3.2.4 Casting de Variáveis	13
3.2.5 Escopo de uma variável	14
3.2.6 Padronização da nomenclatura	14
3.3 Leitura de dados do teclado - Comando input	15
3.4 Exibição de dados na tela - Comando print	15
3.5 Comando Condicional if eles	15

1 INTRODUÇÃO

O Python é uma linguagem de programação interpretada de alto nível, orientada a objetos, que é amplamente utilizada no desenvolvimento de aplicações para internet, softwares, acesso a banco de dados e em machine learning.

Esta linguagem foi criada em 1990 por Guido Van Rossum, no Centro de Matemática Stichting, na Holanda, tendo como principal foco ser uma linguagem de fácil uso, tendo uma escrita simples e bem legível, utilizando comando com uma escrita mais simples, quando comparados a outras linguagens, gerando um código claro e curto.

Devido a isso a linguagem Python é muito difundida tendo uma comunidade com milhões de desenvolvedores, bem como incontáveis fóruns particulares de suporte a linguagem e com uma vasta gama de bibliotecas que podem ser anexadas aos projetos, permitindo que o Python seja muito versátil, sendo capaz de ser configurado para trabalhar de forma simples e fácil em diferentes áreas do conhecimento.

Apesar de ser uma linguagem interpretada, o Python apresenta a possibilidade de que seus programas sejam inteiramente compilados, isso torna a execução mais rápida e facilita a criação e instalação de aplicações.

2 INSTALAÇÃO

O Python pode ser baixado através do seu site: <https://www.python.org/downloads/>.

Execute o instalador, leia atentamente as instruções e proceda com a instalação. Note que o Python possui dependências associadas ao Visual Studio .Net, sendo necessária a instalação prévia deste último para que certas funcionalidades sejam instaladas.

2.1 Instalando pacotes

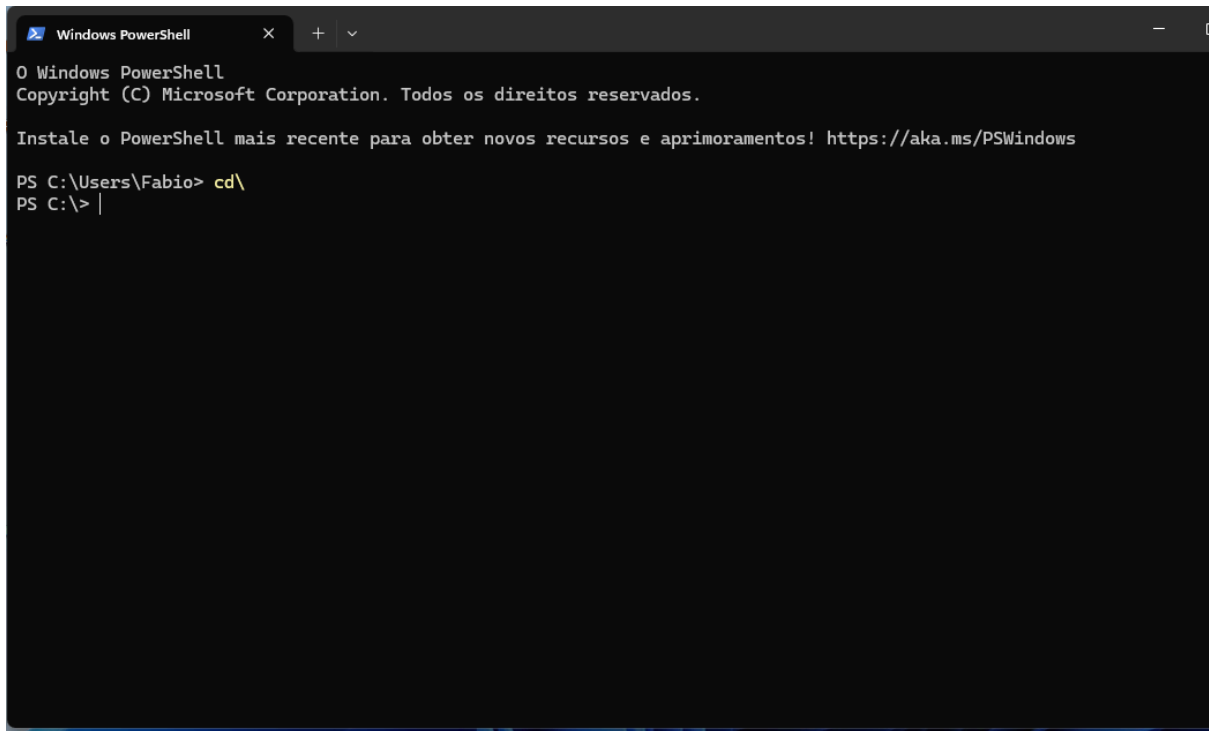
Pacotes contêm conjuntos de rotinas que adicionam funcionalidades a um programa. Como um dos principais pacotes, temos o numpy, utilizado para lidar com matrizes. Sua instalação é feita através do aplicativo pip, contido no Python. Caso sua versão não o possua, deve-se proceder a sua instalação manual. O pip encontra-se dentro da pasta onde o Python foi instalado em um diretório denominado de scripts. Normalmente está no drive do sistema Windows, dentro do diretório Users, nome do perfil, AppData, Local Programs, Python, PythonXXX (versão), Scripts, como o exemplo a seguir do Python 3.1.2

C:\Users\Nome_do_usuario\AppData\Local\Programs\Python\Python312\Scripts>

A instalação deve ser feita via terminal, que pode ser acessado via atalho Windows + X ou Windows + R, cmd, executar.

Para entrar em um diretório devemos utilizar o comando cd, onde:

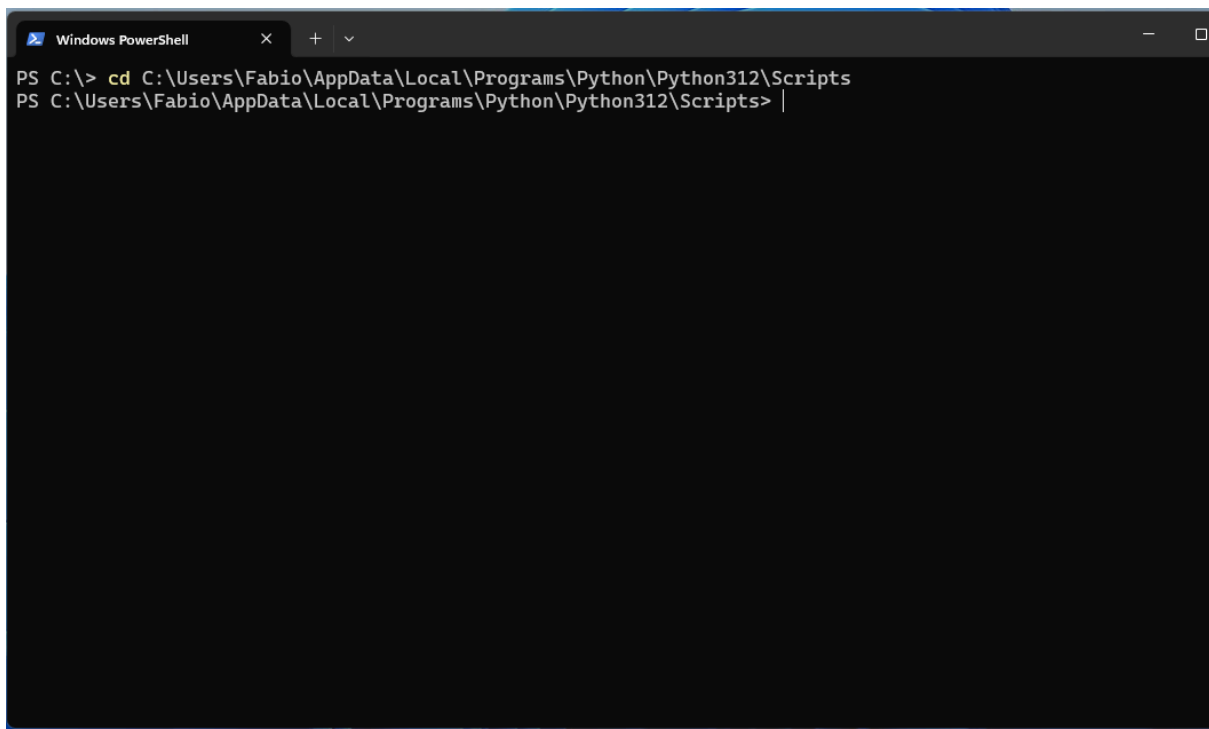
Comando	Ação
cd\	vai para o diretório raiz (o drive)
cd..	desce um nível
cd nome_do_diretório	Acessa o diretório nome_do_diretório dentro da pasta atual. Note que podemos colocar o caminho completo e acessá-lo diretamente.



```
Windows PowerShell
O Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Instale o PowerShell mais recente para obter novos recursos e aprimoramentos! https://aka.ms/PSWindows

PS C:\Users\Fabio> cd\
PS C:\> |
```



```
Windows PowerShell
PS C:\> cd C:\Users\Fabio\AppData\Local\Programs\Python\Python312\Scripts
PS C:\Users\Fabio\AppData\Local\Programs\Python\Python312\Scripts> |
```

Para instalar o numpy digitamos:
pip install numpy

```
C:\Windows\system32\cmd.exe X + v
--client-cert <path>      Path to SSL client certificate, a single file containing the private key and the
                           certificate in PEM format.
--cache-dir <dir>         Store the cache data in <dir>.
--no-cache-dir            Disable the cache.
--disable-pip-version-check
                           Don't periodically check PyPI to determine whether a new version of pip is available for
                           download. Implied with --no-index.
--no-color                Suppress colored output.
--no-python-version-warning
                           Silence deprecation warnings for upcoming unsupported Pythons.
--use-feature <feature>   Enable new functionality, that may be backward incompatible.
--use-deprecated <feature> Enable deprecated functionality, that will be removed in the future.

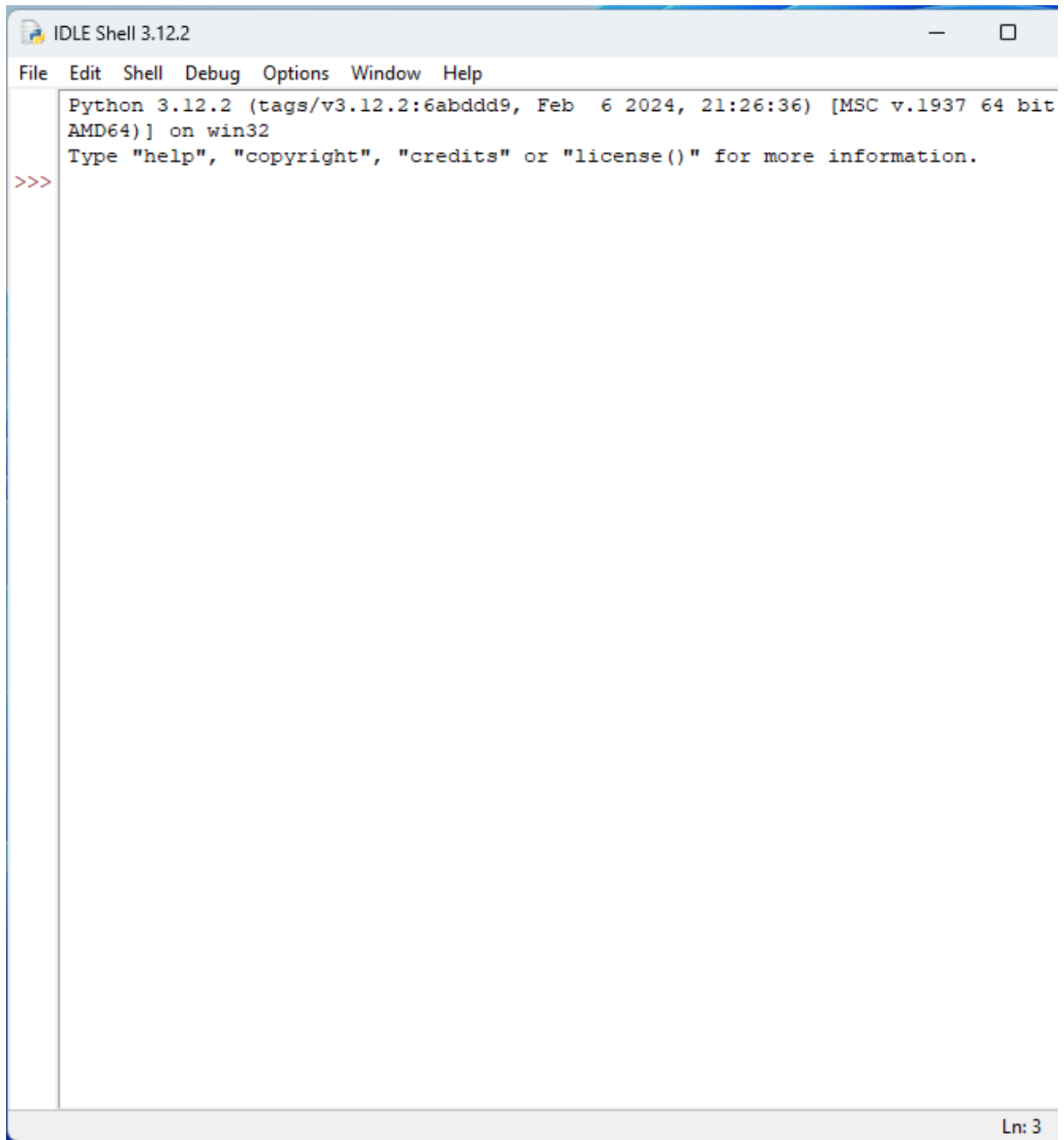
C:\Users\Fabio\AppData\Local\Programs\Python\Python312\Scripts>pip install numpy
Collecting numpy
  Downloading numpy-1.26.4-cp312-cp312-win_amd64.whl.metadata (61 kB)
    61.0/61.0 kB 325.6 kB/s eta 0:00:00
  Downloading numpy-1.26.4-cp312-cp312-win_amd64.whl (15.5 MB)
    15.5/15.5 MB 32.8 MB/s eta 0:00:00
Installing collected packages: numpy
  WARNING: The script f2py.exe is installed in 'C:\Users\Fabio\AppData\Local\Programs\Python\Python312\Scripts' which
  is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed numpy-1.26.4

C:\Users\Fabio\AppData\Local\Programs\Python\Python312\Scripts>pip install numpy
Requirement already satisfied: numpy in c:\users\fabio\AppData\Local\Programs\Python\Python312\lib\site-packages (1.
26.4)

C:\Users\Fabio\AppData\Local\Programs\Python\Python312\Scripts>
```

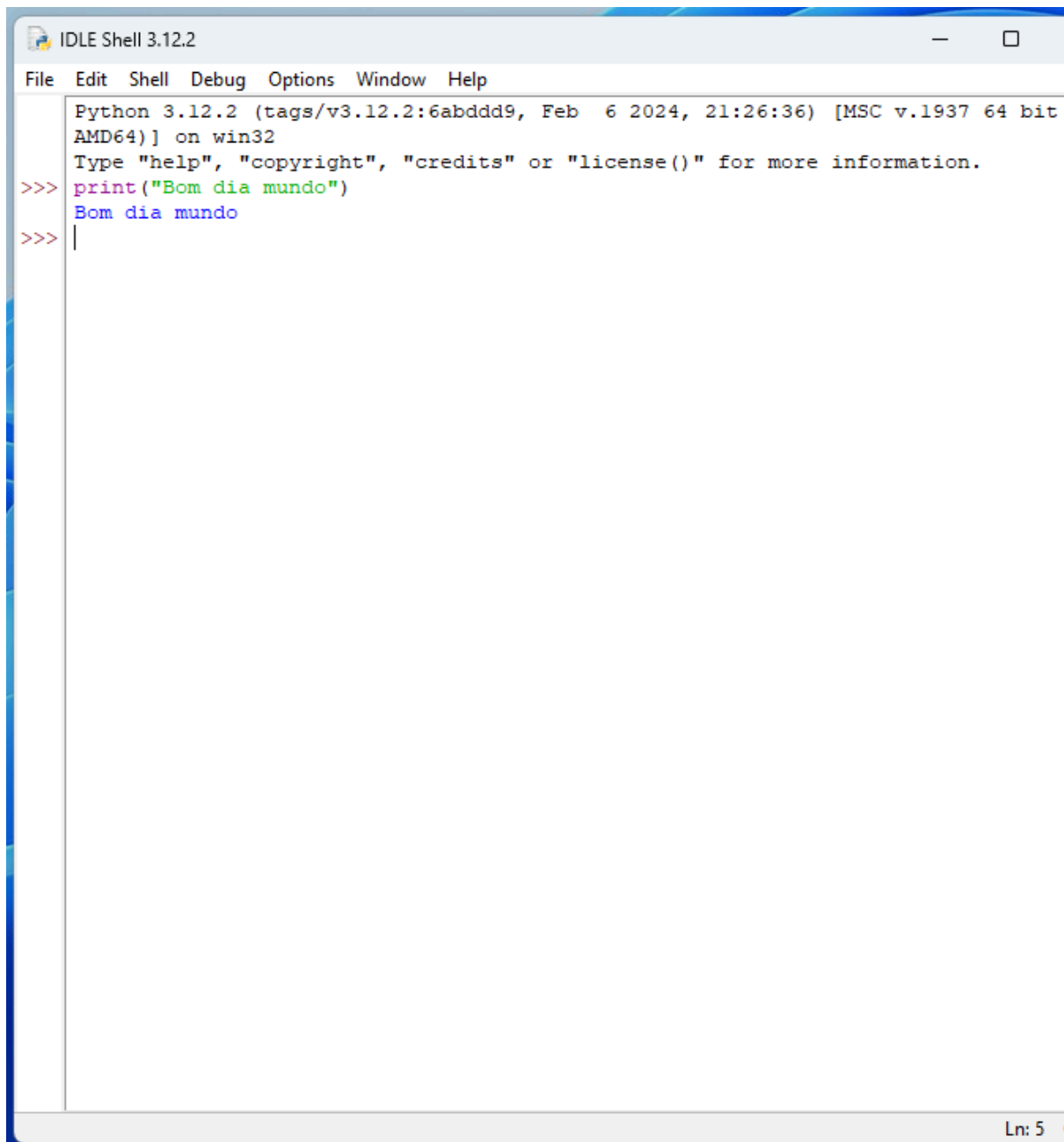
3 PRIMEIROS PASSOS EM PYTHON

Primeiramente devemos iniciar a IDE do Python.

The image shows a screenshot of the Python IDLE Shell 3.12.2 window. The window has a title bar with the text "IDLE Shell 3.12.2" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with the following options: File, Edit, Shell, Debug, Options, Window, and Help. The main area of the window is a text editor with a light blue background. It contains the following text: "Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit AMD64)] on win32" followed by "Type 'help', 'copyright', 'credits' or 'license()' for more information." and a prompt ">>>". The status bar at the bottom right of the window shows "Ln: 3".

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit  
AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>
```

Nesta interface digitamos os comandos da linguagem e após apertarmos enter, o comando é imediatamente executado.

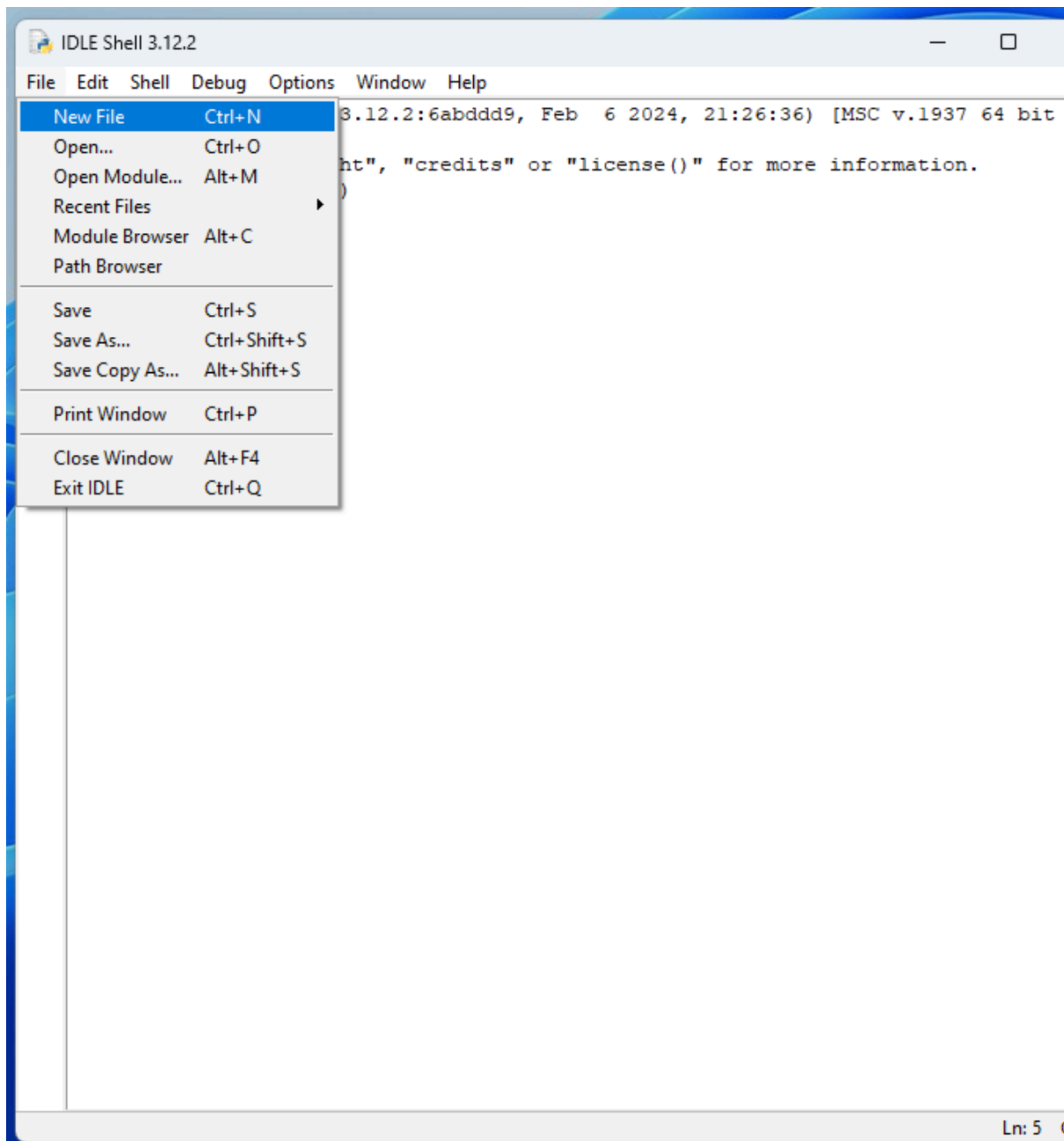


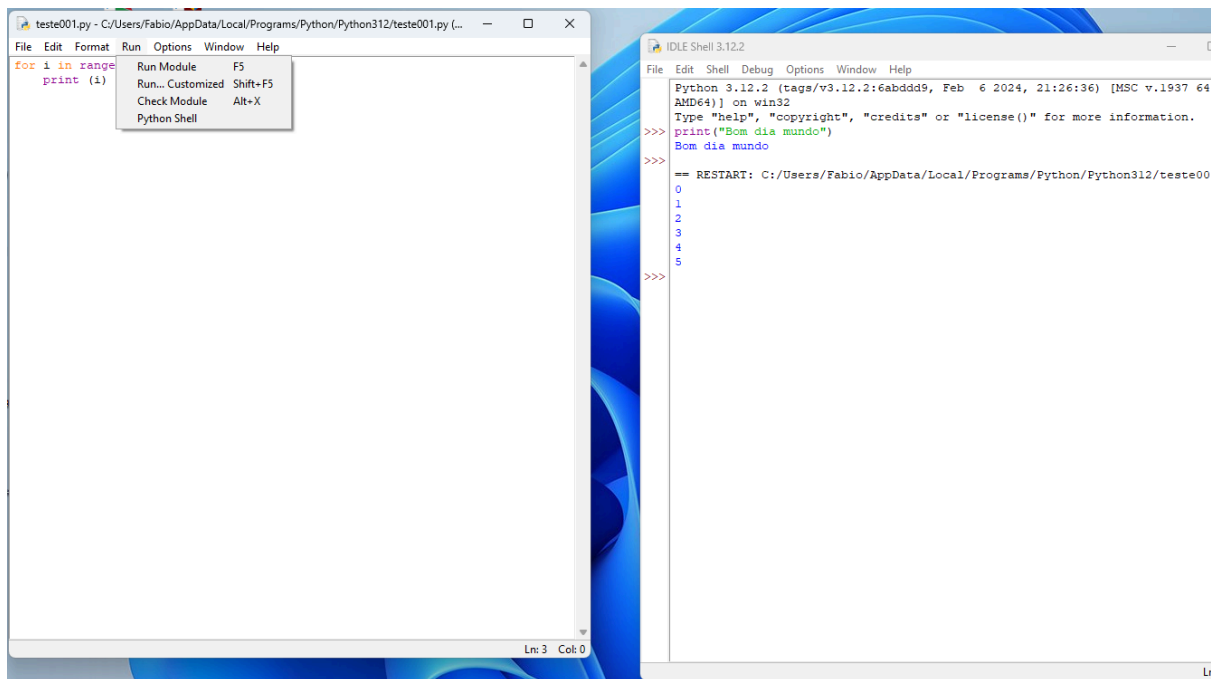
The image shows a screenshot of the IDLE Shell 3.12.2 window. The title bar at the top reads "IDLE Shell 3.12.2" and includes standard window controls (minimize, maximize, close). Below the title bar is a menu bar with the following options: File, Edit, Shell, Debug, Options, Window, and Help. The main text area contains the following text:

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit  
AMD64] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>> print("Bom dia mundo")  
Bom dia mundo  
>>> |
```

The text is color-coded: "print" is in green, "Bom dia mundo" is in blue, and the prompt ">>>" is in red. A vertical cursor is visible on the line following the last prompt. At the bottom right of the window, the status bar shows "Ln: 5".

Para criar programas completos é mais conveniente utilizar o editor de programas do Python, indo em File, New File





3.1 Indentação

A indentação consiste no espaçamento dado entre o começo de uma linha e o primeiro caractere digitado. No Python o interior de determinados comandos como for, if e funções são delimitados pela indentação das linhas seguintes a esses comandos, não por separadores de blocos de comando como o begin e end e {} de outras linguagens. A própria indentação define o bloco de comando.

3.2 Variáveis

Variáveis são espaços de memória que armazenam dados utilizados em programas. Este espaço é representado normalmente por uma alias (apelido) apesar de que algumas linguagens, como o C++, são capazes de referenciar diretamente o endereço da variável.

Uma variável possui um atributo denominado tipo, que especifica qual espécie de valor será armazenado. Muitas linguagens necessitam que o tipo seja definido previamente, porém o Python é uma linguagem dinamicamente tipada, ou seja, não é preciso declarar previamente o tipo da variável, ele será definido quando um valor for inserido em uma variável. Também é possível alterar o tipo de uma variável, basta que seja inserido um tipo de valor diferente do tipo original. Para atribuir um valor a uma variável basta colocar o nome da variável = valor.

Exemplo:

```
a = 2      //a variável vira integer
a = 2.12   //a variável vira float
```

3.2.1 Tipos de Variáveis

- Inteiro (int): Variáveis do tipo inteiro representam números inteiros sem qualquer parte fracionária.

Exemplo:

```
ano = 1990
```

- Real (float): Variáveis do tipo real representam números que podem possuir casas decimais.

Exemplo

```
Pi = 3.1415
```

- Texto (str): Variáveis do tipo texto representam sequências de caracteres, são dados textuais, palavras e sentenças. Todo texto (**excetuando-se o nome das variáveis**) deve vir entre aspas (“ ”) ou apóstrofes (‘ ’).

Exemplo:

Nome = “Alex”

- Booleano (bool): Variáveis booleanas armazenam valores lógicos, True (verdadeiro, 1) ou False (falso, 0). Valores lógicos devem ser escritos diretamente, sem o uso de aspas ou apóstrofes e com apenas a primeira letra em maiúsculo.

Exemplo:

marca = False

- Listas (List): São coleções de objetos que podem ser alterados.

Exemplo:

Frutas = [‘maçã’ , ‘goiaba’ , ‘pera’]

- Tuplas (Tuples): São coleções de objetos que podem ser alterados.

Exemplo:

Ponto = (1, 2)

3.2.2 Verificando o tipo de variável – Comando type

O comando type retorna o tipo de uma variável em um determinado momento.

Exemplo:

```
type(nome)
<class 'str'>
```

3.2.3 Operadores

Aritméticos

Operador	Nome	Exemplo
+	Adição	c = a + b
-	Subtração	c = a - b
*	Multiplicação	c = a * b
/	Divisão	c = a / b (6//4 = 1.5)
//	Divisão inteira	c = a // b (6//4 = 1)
%	Módulo (resto da divisão inteira)	c = a % b (6%4 = 2)

Potenciação

$C = a^{**}b$ ($2^{**}3 = 8$)

Comparação

Operador	Nome	Função
=	Igual a	Verifica se um valor é igual ao outro
!=	Diferente de	Verifica se um valor é diferente ao outro
>	Maior que	Verifica se um valor é maior que outro
>=	Maior ou igual	Verifica se um valor é maior ou igual ao outro
<	Menor que	Verifica se um valor é menor que outro
<=	Menor ou igual	Verifica se um valor é menor ou igual ao outro

Atribuição

Operador	Equivalente a	Exemplo, com a = 4
=	$x = y$	$a = 1$ (1)
+=	$x = x + y$	$a += 2$ (a = 6)
-=	$x = x - y$	$a -= 2$ (a = 2)
*=	$x = x * y$	$a *= 2$ (a = 8)
/=	$x = x / y$	$a /= 2$ (a = 8)
%=	$x = x \% y$	$a \% = 2$ (a = 0)

Lógicos

Operador	Definição	Exemplo
and	Retorna True se todas as afirmações forem verdadeiras	if $a > b$ and $c < d$:
or	Retorna True se pelo menos uma afirmação for verdadeira	if $a > b$ or $c < d$:
not	Inverte o resultado (Falso se o resultado for verdadeiro)	if not ($a > b$):

Operadores de Identidade

Estes Operadores verificam se os objetos testados se referem ao mesmo objeto (is) ou não (is not).

Operador	Definição
is	Retorna True se ambas as variáveis são o mesmo objeto
is not	Retorna True se ambas as variáveis não forem o mesmo objeto

Operadores de Associação

Verifica se um determinado objeto está associado ou pertence a uma determinada estrutura de dados.

```
L1= [1, 2, 3]
```

```
L2 = [1, 2, 3]
```

```
L1 is L2
```

```
L1 is not L2
```

Operador	Função
in	Retorna True caso o valor seja encontrado na sequência
not in	Retorna True caso o valor não seja encontrado na sequência

```
frutas = ['maçã', 'pera', 'uva', 'caqui']
```

```
# Verifica a existência de um dado valor dentro da lista
```

```
print('uva' in lista) # True
```

```
# Verifica a não existência de um dado valor dentro da lista
```

```
print('ovo' not in lista) # True
```

3.2.4 Casting de Variáveis

Consiste em transformar uma variável de um tipo em outro. Devemos respeitar o tipo de variável, pois podemos transformar um inteiro ou float em string, mas letras não podem ser transformadas de forma direta em números. Temos então:

`int()` - Constrói uma variável inteira a partir de um inteiro, real (remove a parte decimal) ou texto, desde que ele seja composto de algarismos numéricos.

`float()` - Constrói uma variável real a partir de um real, inteiro ou texto, desde que ele seja composto de algarismos numéricos.

`str()` - Constrói um texto a partir de outros tipos.

```
x = int(2) # 2
```

```
y = int(2.4) # 2
```

```
z = int("2") # 2
```

```
x = float(1) # 1.0
```

```
y = float(1.8) # 1.8
```

```
z = float("23") # 23.0
```

```
w = float("24.2") # 24.2
```

```
x = str("A1") # 'A1'
```

```
y = str(12) # '12'
```

```
z = str(31.0) # '31.0'
```

3.2.5 Escopo de uma variável

Quando criamos uma variável fora de qualquer função ou bloco de código, ela é considerada global e pode ser acessada em qualquer parte ou função do programa.

As variáveis que são inicializadas dentro de um bloco de comandos ou função, são locais, apenas são visíveis para o local onde foram criadas.

3.2.6 Padronização da nomenclatura

Primeiramente o Python é case sensitive, ou seja, é uma linguagem que faz distinção entre caracteres maiúsculos e minúsculos.

O nome das variáveis podem conter letras e dígitos, entretanto devem começar com uma letra ou sublinhado (`_`);

Variáveis devem ter seu nome completamente escrito com letras minúsculas;

Utilize nomes representativos para as variáveis e não genéricos e sem significado;
Não utilizar palavras reservadas, como comandos;
Nomes compostos devem ser unidos pelo underline (_).

3.3 Leitura de dados do teclado - Comando input

O comando input lê os dados digitados pelo usuário e o armazena em uma variável.

Sintaxe:

```
Variável = input("Texto a ser exibido para o usuário")
```

Exemplo:

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

Este comando pode ser utilizado em conjunto com o casting de valores, permitindo que valores inteiros e de ponto flutuante sejam lidos.

Exemplo:

```
num = int(input("Digite a posição desejada: "))
```

Digite a posição desejada: 12

num

12

3.4 Exibição de dados na tela - Comando print

É o comando utilizado para imprimir dados no monitor.

Exemplo:

```
print("ola")
```

Podemos escrever mais de uma sentença, desde que as separemos com vírgula (,)

Exemplo:

```
print("ola", "tudo bem?")
```

Podemos criar um texto que será impresso entre as sentenças com o comando sep = ' '

Exemplo:

```
print('ola', 'tudo bem?', " com você", sep = '====')
```

ola====tudo bem?===== com você

3.5 Comando Condicional if eles

Este comando testa uma condição lógica e responde de acordo com o resultado. Se for verdadeiro executa o código após a condição, se for falso, ou não executa nada ou uma outra

linha de execução, antecedida pelo comando else. A final da linha do if, bem como o final da linha do else devem ser sucedidos de `:`.

O bloco contendo os comandos a serem executados tanto após o if quanto após o else devem ser indentados.

Exemplo

```
if b > a:  
    print("b é o maior.")
```

Com uma ação caso a alternativa seja falsa:

```
if b > a:  
    print("b é o maior")  
  
else:  
    print("a é o maior")
```

Podemos colocar mais de uma condição utilizando o comando elif.

```
if b > a:  
    print("b é o maior")  
  
elif a == b:  
    print("são iguais")  
  
else:  
    print("a é o maior")
```

Este comando pode ser escrito de forma compacta, utilizando a seguinte sintaxe:

```
açãoV if condição else condiçãoF  
  
print("o maior é: ")  
print("A") if a > b else print("B")
```