



Introdução ao Processamento de Dados

Turma 3 (2020.1)



Introdução ao Python

Gilson. A. O. P. Costa (IME/UERJ)

`gilson.costa@ime.uerj.br`

Histórico



- Foi concebida no final de 1989 pelo holandês Guido Van Rossum
 - O código só foi publicado em 1991
 - A versão 1.0 foi lançada em 1994



Histórico



- Foi concebida no final de 1989 pelo holandês Guido Van Rossum
 - O código só foi publicado em 1991
 - A versão 1.0 foi lançada em 1994



Histórico

- De acordo com a revista [Spectrum da IEEE](#) (Institute of Electrical and Electronics Engineers) o Python é em 2020 a **principal linguagem de programação** no mundo.
- É a linguagem escolhida como a **mais adequada para o ensino** introdutório de computação na maior parte das principais universidades dos EUA.
- Algumas organizações que usam Python: Google, Yahoo!, Wikipedia, CERN, NASA, Facebook, Amazon, Instagram, Spotify, ...

Histórico

- Muito utilizada para computação científica, na área financeira, em programação web, etc.
- É a linguagem mais utilizada em *machine learning* e *deep learning*.
- Se você for trabalhar com **inteligência artificial**, não vai escapar do Python.

Um exemplo...

- Estas faces foram produzidas por um programa escrito em Python (e com a biblioteca TensorFlow).
- Elas não são de pessoas vivas/reais.
- O programa (StyleGAN) aprende a gerar faces através de exemplos.
- Baseado em um conjunto de técnicas chamadas de *deep learning*.



Karras et al., "Analyzing and Improving the Image Quality of StyleGAN", 2020

Um exemplo...



Karras et al., "A Style-Based Generator Architecture for Generative Adversarial Networks" 2019

Características

- Python é uma linguagem interpretada (não compilada).
- Linguagem de altíssimo nível (*Very High Level Language*),
- Moderna, de sintaxe fácil e concisa.
- Tipagem dinâmica (não há declaração de variáveis).
- Modular.
- Multiplataforma.
- Código livre.

Características

- Python é uma linguagem interpretada (não compilada).
- Linguagem de altíssimo nível (*Very High Level Language*),
- Permite usar seguintes metodologias de programação:
 - Estruturada
 - Funcional
 - Orientada a objetos

Documentação

- Existem diversos livros e sites sobre programação com Python.
- Documentação oficial: <http://wiki.python.org.br/DocumentacaoPython>
- Livro de professores do IME UERJ (Alexandre Rojas e Sérgio Kostin).
- Introdução à Programação com Python.
- Editora Ciência Moderna.



Versões do Python

As principais versões são (<https://www.python.org/downloads/>)

- 2.7.x (última versão 2.7.13 – dezembro de 2016)
- 3.8.x (última versão 3.8.5 – julho de 2020)

Python 2

- Foi o padrão da linguagem por muito tempo (primeira versão lançada em 2010). **Descontinuada** em janeiro de 2020.

Python 3

- Algumas mudanças a tornaram **incompatível com a versão 2**.
- Constantemente evoluindo e recebendo novas funcionalidades, que não estarão presentes na versão anterior.

Versões do Python

- Existem inúmeras ferramentas de desenvolvimento para Python.
- ***Integrated Development Environments*** (IDEs) são pacotes de software integram várias ferramentas de desenvolvimento, com o objetivo de aumentar a produtividade do desenvolvedor.
- Geralmente as IDEs incluem recursos como ***syntax highlight*** (código fonte colorizado conforme a sintaxe da linguagem), ***code completion*** (o editor apresenta durante a digitação formas possíveis de completar o texto), ferramentas de ***depuração*** (*debug*), ***Shell*** integrado, etc.

Integrated Development Environments (IDEs) Python

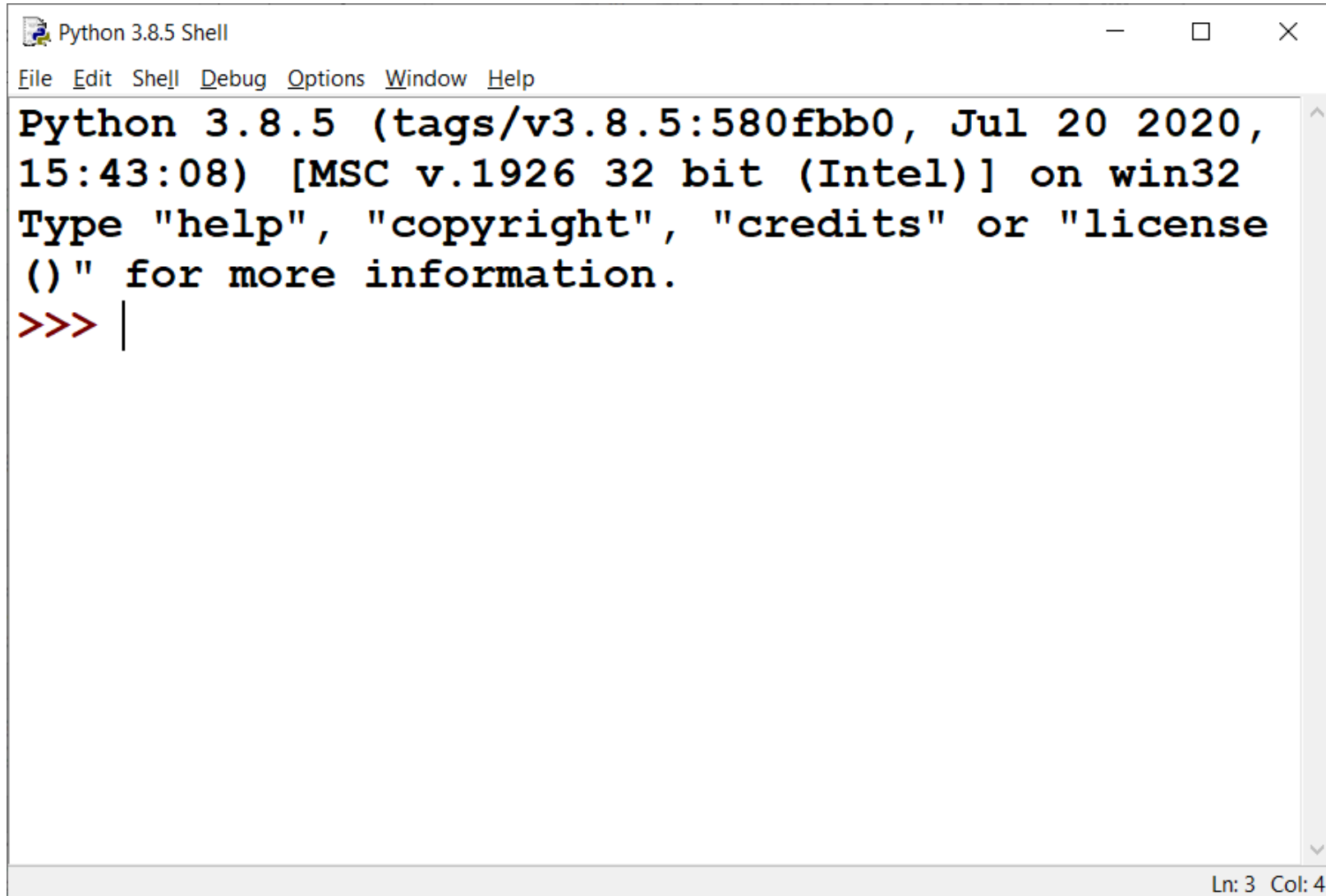
Python.org (IDLE)

- Ambiente Python padrão.
- Vocês podem baixar a versão apropriada para o seu sistema operacional (Windows, MacOS, Linux) através do link:

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

- O IDE propriamente dito chama-se **IDLE**: contém interfaces para a criação, execução e depuração de programas Python.
- Permite a execução de comandos do Python de forma interativa (Python *Shell*).

IDLE (Python.org)



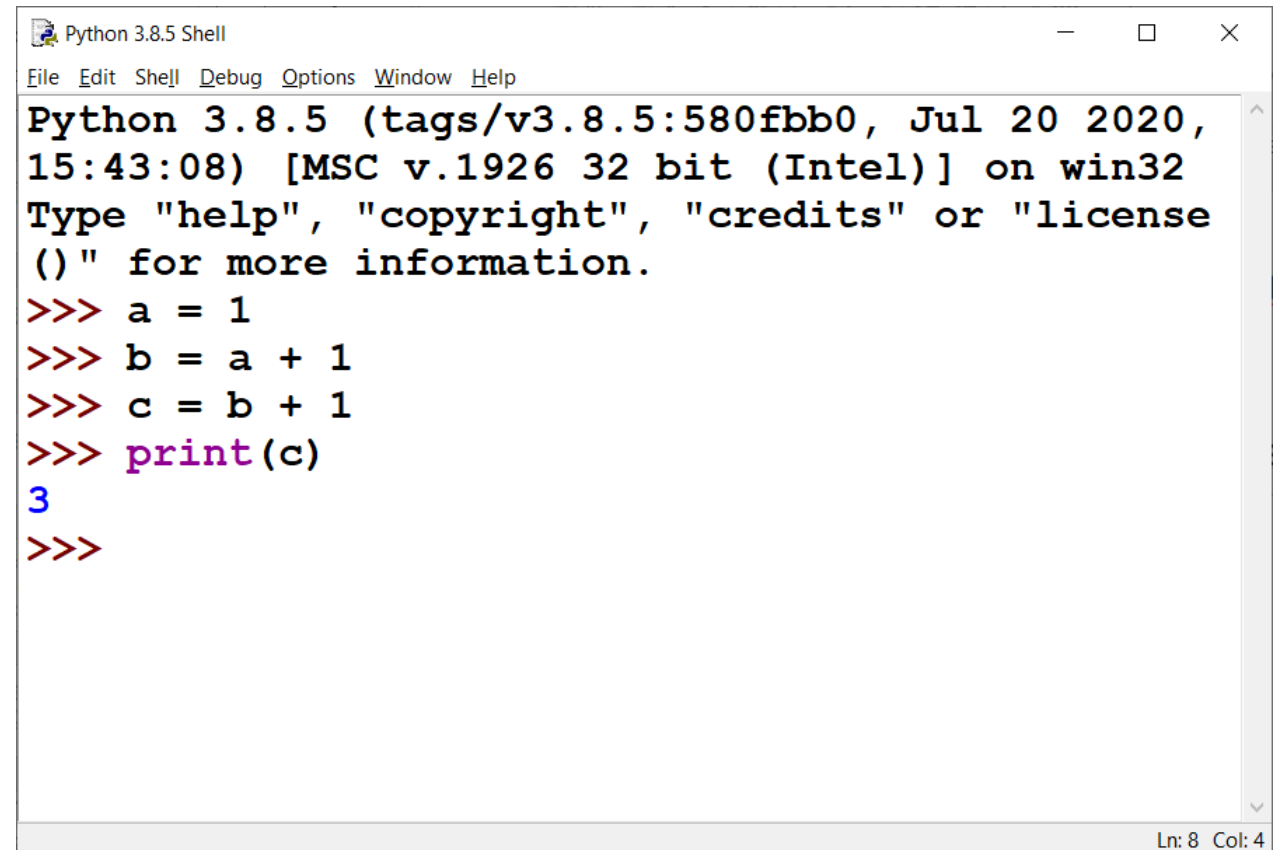
The image shows a screenshot of the Python 3.8.5 Shell window within the IDLE environment. The window has a title bar that reads "Python 3.8.5 Shell" and standard Windows window controls (minimize, maximize, close). Below the title bar is a menu bar with the following options: File, Edit, Shell, Debug, Options, Window, and Help. The main text area contains the following text: "Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32", followed by "Type 'help', 'copyright', 'credits' or 'license ()' for more information.", and then a red prompt ">>>" with a vertical cursor line. A vertical scrollbar is visible on the right side of the text area. At the bottom right of the window, a status bar displays "Ln: 3 Col: 4".

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020,
15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license
()" for more information.
>>> |
```

Ln: 3 Col: 4

IDLE (Python.org)

- Quando se abre o IDLE, ele mostra um **Python *Shell***.
- No Python *Shell* você pode **executar comandos Python** interativamente.



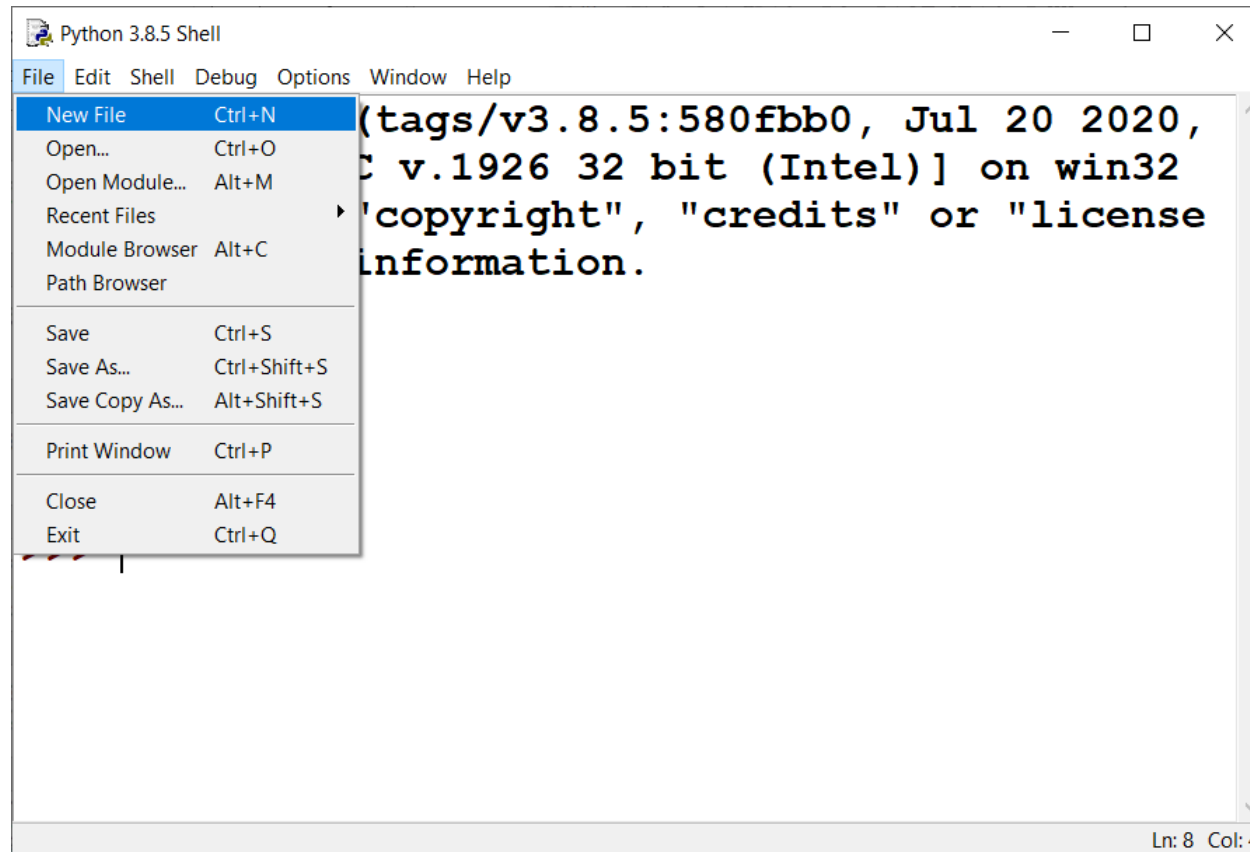
The screenshot shows a window titled "Python 3.8.5 Shell" with a menu bar (File, Edit, Shell, Debug, Options, Window, Help). The main text area displays the following content:

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> a = 1
>>> b = a + 1
>>> c = b + 1
>>> print(c)
3
>>>
```

The status bar at the bottom right indicates "Ln: 8 Col: 4".

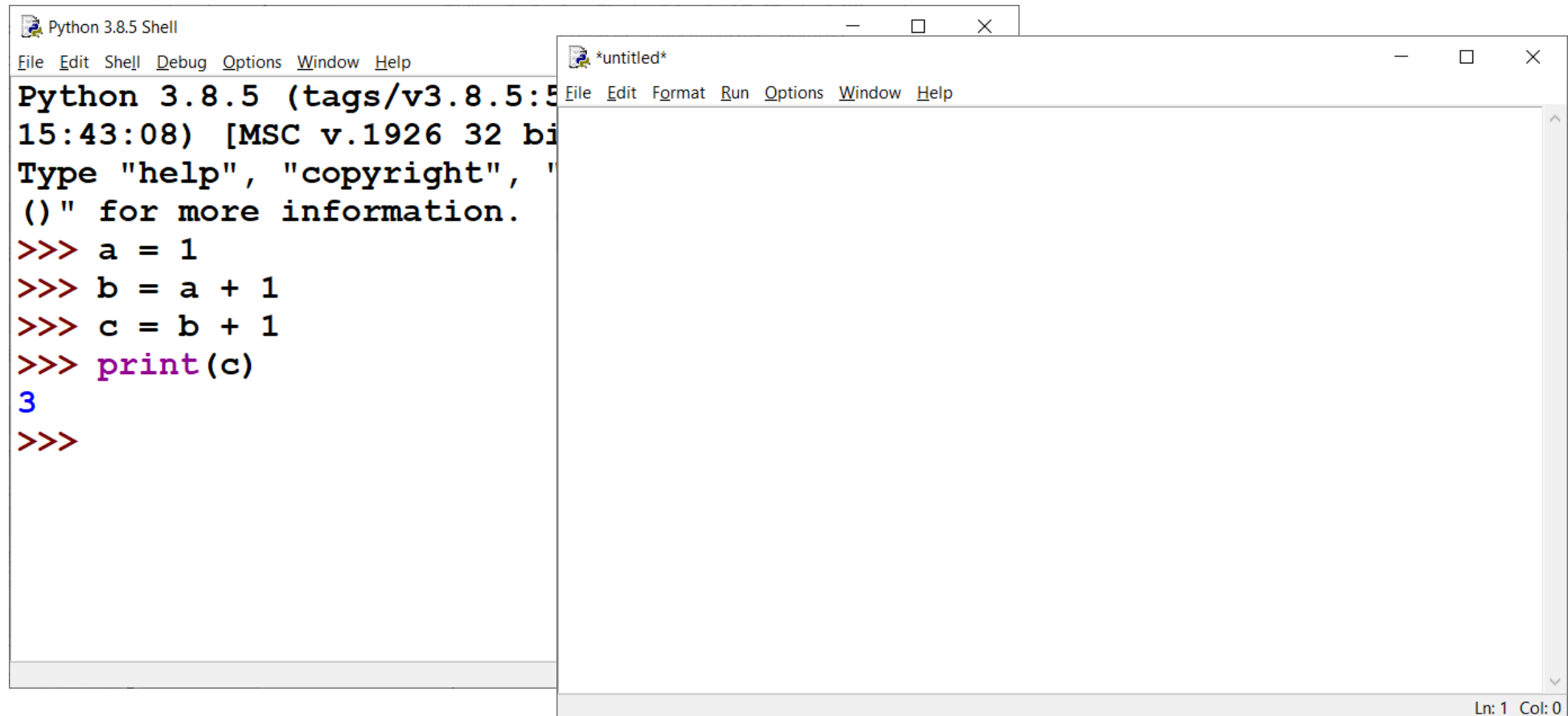
IDLE (Python.org)

Para criar um programa Python, você deve **abrir uma nova janela**, escrever seu programa e salvá-lo num arquivo com extensão `.py`



IDLE (Python.org)

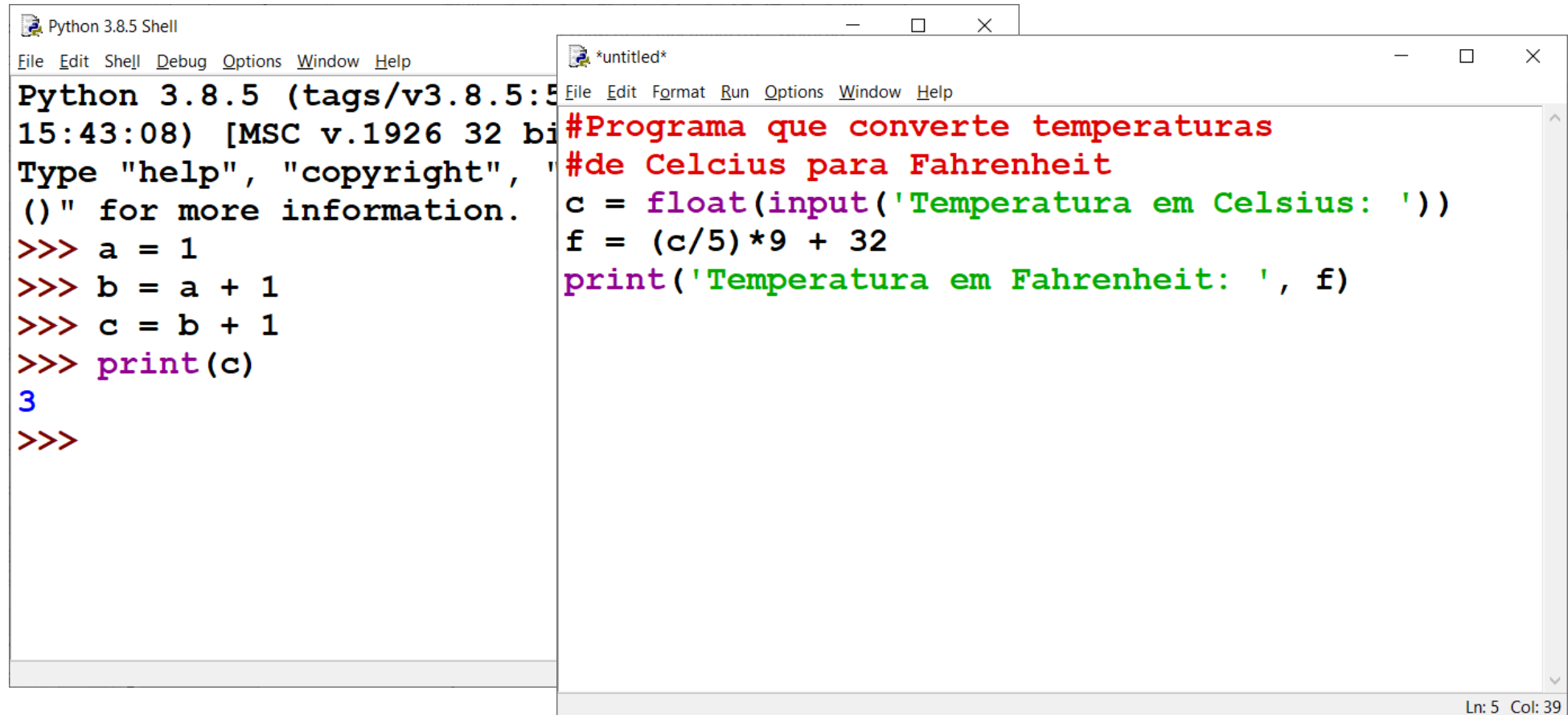
Para criar um programa Python, você deve **abrir uma nova janela**, escrever seu programa e salvá-lo num arquivo com extensão `.py`



The screenshot displays the Python IDLE interface. On the left, the 'Python 3.8.5 Shell' window is active, showing the Python prompt and a series of commands: `>>> a = 1`, `>>> b = a + 1`, `>>> c = b + 1`, and `>>> print(c)`. The output `3` is displayed below the last command. On the right, an 'untitled*' editor window is open, showing a blank text area for writing code. The status bar at the bottom right of the editor window indicates 'Ln: 1 Col: 0'.

IDLE (Python.org)

Para criar um programa Python, você deve abrir uma nova janela, **escrever seu programa** e salvá-lo num arquivo com extensão **.py**



The screenshot displays the Python IDLE interface. On the left, the 'Python 3.8.5 Shell' window shows the interactive prompt with the following code and output:

```
Python 3.8.5 (tags/v3.8.5:5b3443cd, Sep 11 2019, 15:43:08) [MSC v.1926 32 bit (AMD64)]
Type "help", "copyright", "credits()" or "quit()" for more information.
>>> a = 1
>>> b = a + 1
>>> c = b + 1
>>> print(c)
3
>>>
```

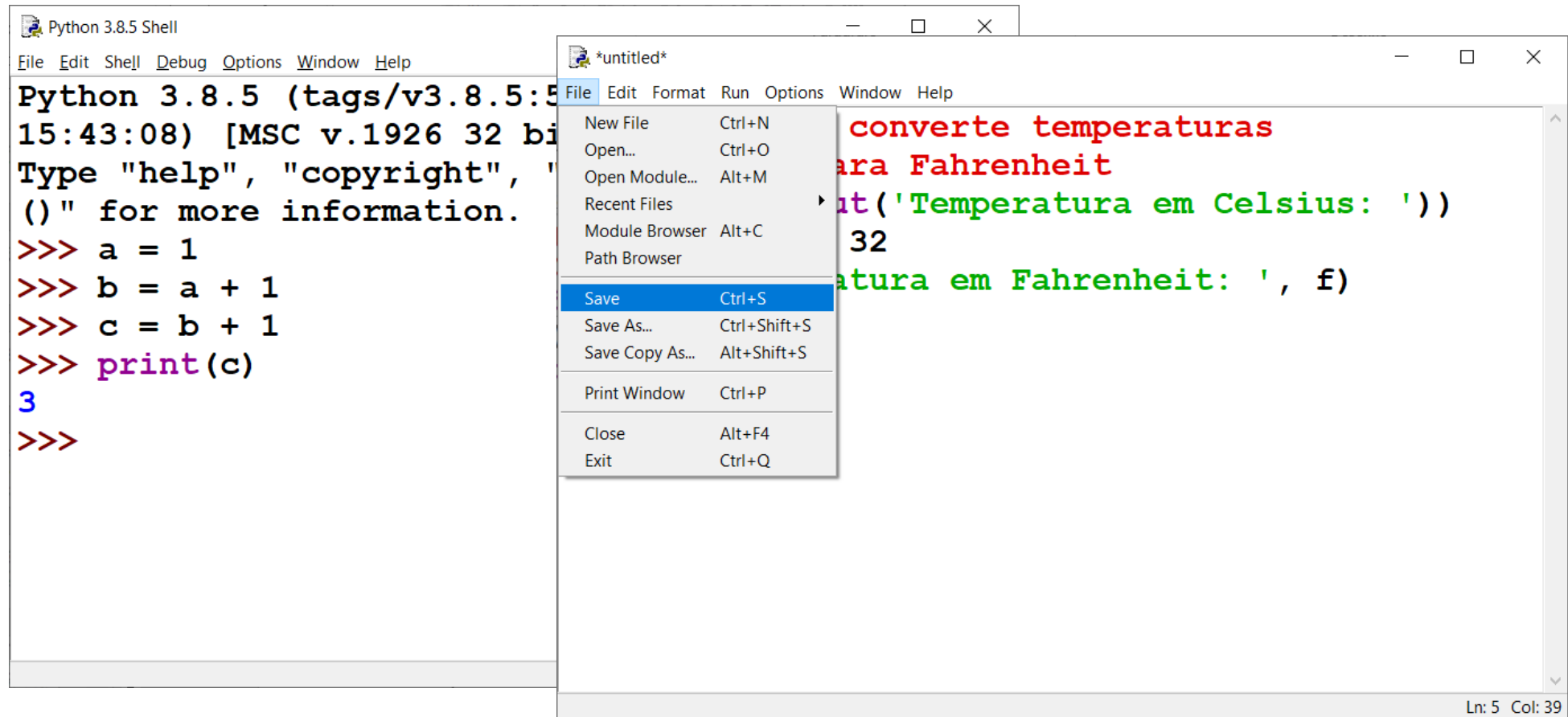
On the right, the '*untitled*' editor window shows a Python script for temperature conversion:

```
#Programa que converte temperaturas
#de Celcius para Fahrenheit
c = float(input('Temperatura em Celsius: '))
f = (c/5)*9 + 32
print('Temperatura em Fahrenheit: ', f)
```

The status bar at the bottom right of the editor window indicates 'Ln: 5 Col: 39'.

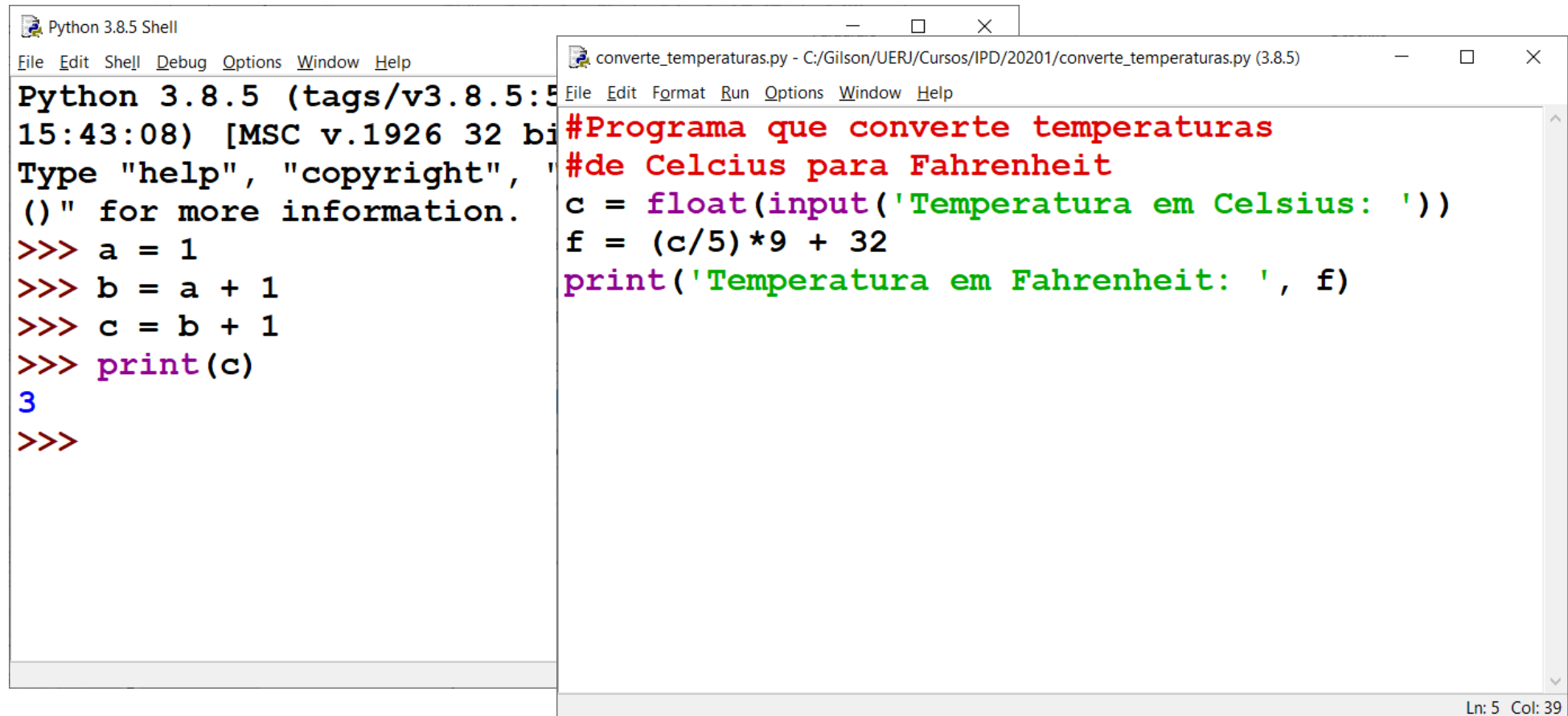
IDLE (Python.org)

Para criar um programa Python, você deve abrir uma nova janela, escrever seu programa e **salvá-lo num arquivo** com extensão **.py**



IDLE (Python.org)

Para criar um programa Python, você deve abrir uma nova janela, escrever seu programa e **salvá-lo num arquivo** com extensão **.py**



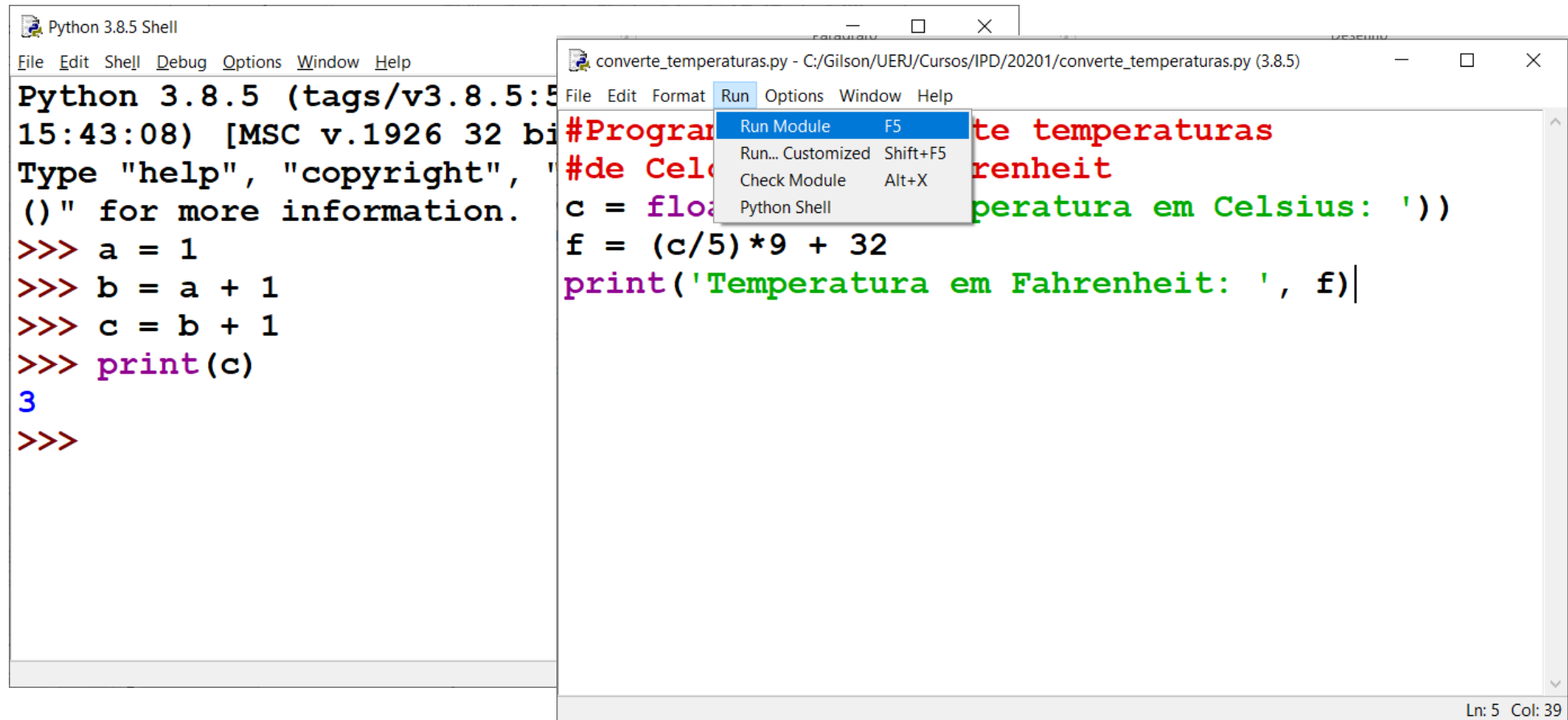
```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:5b3443d, Sep 15:43:08) [MSC v.1926 32 bit Intel x86_64]
Type "help", "copyright", "credits()" or "help()" for more information.
>>> a = 1
>>> b = a + 1
>>> c = b + 1
>>> print(c)
3
>>>
```

```
converte_temperaturas.py - C:/Gilson/UERJ/Cursos/IPD/20201/converte_temperaturas.py (3.8.5)
File Edit Format Run Options Window Help
#Programa que converte temperaturas
#de Celcius para Fahrenheit
c = float(input('Temperatura em Celsius: '))
f = (c/5)*9 + 32
print('Temperatura em Fahrenheit: ', f)
```

Ln: 5 Col: 39

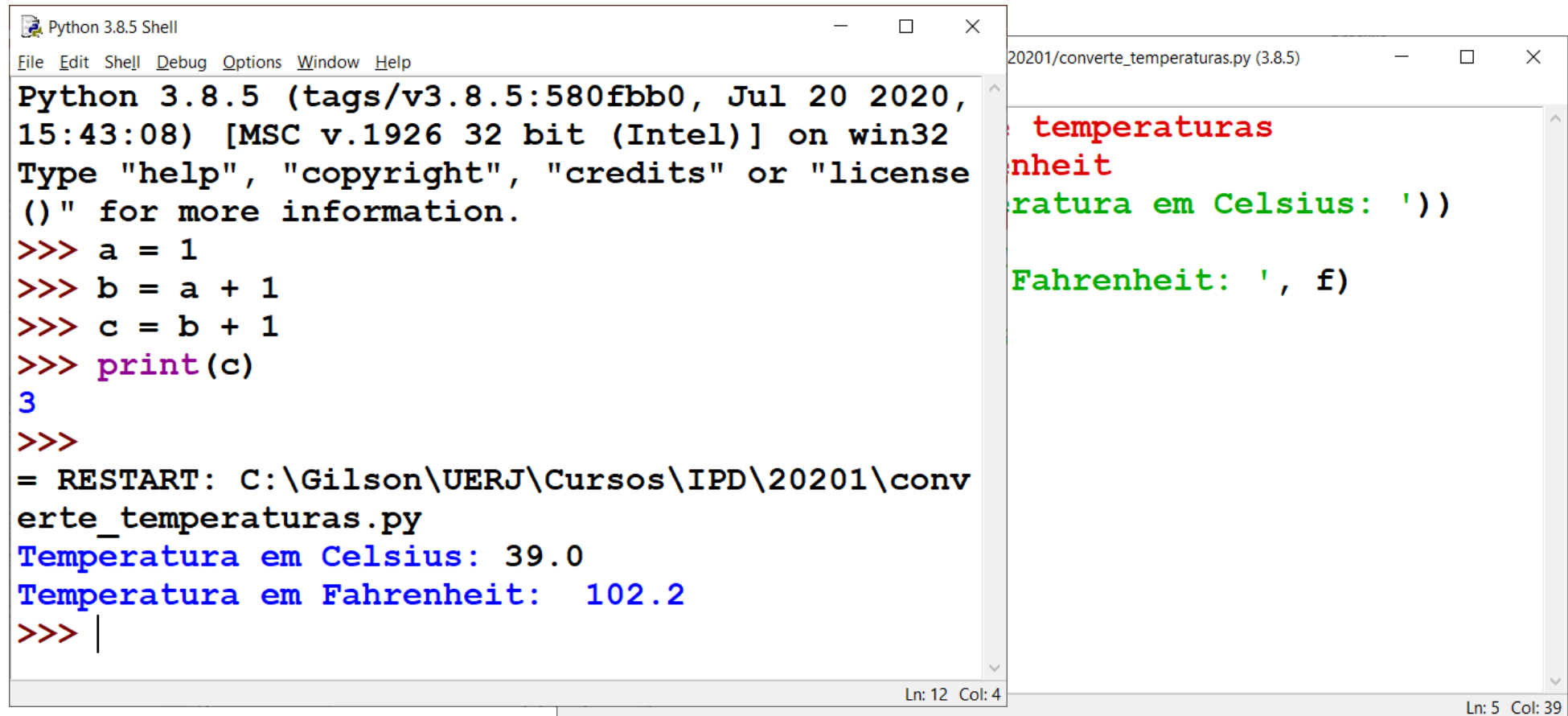
IDLE (Python.org)

Para **executar o programa** basta escolher a opção *Run/Run Module*, ou pressionar F5.



IDLE (Python.org)

As interações com o usuário e a apresentação do resultado ocorre no *Shell*.



The image shows two windows from the Python IDE (IDLE). The left window is titled 'Python 3.8.5 Shell' and displays the Python interpreter's startup message and a series of commands and their outputs. The right window is titled '20201/converte_temperaturas.py (3.8.5)' and shows a Python script for temperature conversion.

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> a = 1
>>> b = a + 1
>>> c = b + 1
>>> print(c)
3
>>>
= RESTART: C:\Gilson\UERJ\Cursos\IPD\20201\converte_temperaturas.py
Temperatura em Celsius: 39.0
Temperatura em Fahrenheit: 102.2
>>> |
```

```
def converte_temperaturas(temperatura, unidade):
    if unidade == 'C':
        return (temperatura * 9/5) + 32
    elif unidade == 'F':
        return (temperatura - 32) * 5/9
    else:
        raise ValueError('Unidade inválida')

# Exemplo de uso
temperatura = 39.0
unidade = 'C'
resultado = converte_temperaturas(temperatura, unidade)
print(f'Temperatura em Fahrenheit: {resultado}')

```

Ln: 12 Col: 4

Ln: 5 Col: 39

Integrated Development Environments (IDEs) Python

Programiz

- IDE de Python **on-line**.
- Permite a criação de programas Python e **execução diretamente num navegador**, independente do sistema operacional.
- Não há necessidade de instalar nada no computador.
- O Programiz pode ser acessado através do link:
<https://www.programiz.com/python-programming/online-compiler/>

Programiz

The screenshot shows the Programiz Online Python Compiler web application. The browser's address bar displays the URL `programiz.com/python-programming/online-compiler/`. The page features a navigation bar with various links and a sidebar with icons for different programming languages. The main content area is divided into two sections: a code editor on the left and a shell on the right. The code editor contains a file named `main.py` with the following Python code:

```
1 # Online Python compiler (interpreter) to run Python online.
2 # Write Python 3 code in this online editor and run it.
3 print("Hello world")
```

The shell on the right is currently empty, showing a prompt `>`. The interface also includes a "Run" button and a "Clear" button. An advertisement for Adobe Stock images is visible at the top of the page.

Programiz

The screenshot shows the Programiz Online Python Compiler web application. The browser's address bar displays the URL `programiz.com/python-programming/online-compiler/`. The page features a top navigation bar with various social media and utility links. A prominent advertisement for Adobe Stock images is visible, offering 10 free images for a limited time. The main workspace is divided into two sections: a code editor on the left and a shell on the right. The code editor contains a file named `main.py` with the following Python code:

```
1 # Online Python compiler (interpreter) to run Python online.
2 # Write Python 3 code in this online editor and run it.
3 print("Hello world")
```

A red arrow points to the **Run** button, which is located next to the code editor. The shell on the right displays the output of the code execution:

```
> a = 1
> b = a + 1
> c = b + 1
> print(c)
3
>
```

The interface also includes a **Clear** button for the shell output and a **Learn Python App** button in the top right corner.

Programiz

The screenshot displays the Programiz Online Python Compiler web application. The browser's address bar shows the URL `programiz.com/python-programming/online-compiler/`. The page features a header with the Programiz logo, a promotional banner for Adobe Stock images, and a 'Learn Python App' button. The main workspace is divided into two panels. The left panel, titled 'main.py', contains a Python script with three lines: a comment about the online compiler, another comment about writing code in the editor, and a `print("Hello world")` statement. Above the code editor are icons for full-screen, dark mode, and a blue 'Run' button. The right panel, titled 'Shell', shows the output of the program: 'Hello world' followed by a prompt character '>' and a cursor. A 'Clear' button is located in the top right corner of the Shell panel.

Online Python Compiler (Interpreter) × +

← → ↻ 🔒 programiz.com/python-programming/online-compiler/ ☆ 🔍 8 V PDF 🔄 ⚙️ 👤

Apps Google Netflix YouTube Globoplay globo.com WhatsApp Dictionaries News Bancos UERJ » Outros favoritos

Programiz
Python Online Compiler

GET 10 FREE ADOBE STOCK IMAGES.

Limited time offer: Get 10 free Adobe Stock images.

ADS VIA CARBON

[Learn Python App](#)

main.py

1 # Online Python compiler (interpreter) to run Python online.
2 # Write Python 3 code in this online editor and run it.
3 print("Hello world")

Run

Shell

Hello world
> |

Clear

Programiz

Online Python Compiler (Interpre

programiz.com/python-programming/online-compiler/

Apps Google Netflix YouTube Globoplay globo.com WhatsApp Dictionaries News Bancos UERJ Outros favoritos

Programiz
Python Online Compiler

[Learn Python App](#)

main.py

```
1 #Programa que converte temperaturas
2 #de Celcius para Fahrenheit
3 c = float(input('Temperatura em Celsius: '))
4 f = (c/5)*9 + 32
5 print('Temperatura em Fahrenheit: ', f)
```

Run

Shell

Clear

> |

Programiz

The screenshot displays the Programiz Online Python Compiler web application. The browser's address bar shows the URL `programiz.com/python-programming/online-compiler/`. The page header includes the Programiz logo, the text "Python Online Compiler", and a "Learn Python App" button. The main interface is divided into two sections: a code editor on the left and a shell on the right. The code editor shows a file named `main.py` with the following Python code:

```
1 #Programa que converte temperaturas
2 #de Celcius para Fahrenheit
3 c = float(input('Temperatura em Celsius: '))
4 f = (c/5)*9 + 32
5 print('Temperatura em Fahrenheit: ', f)
```

The shell on the right shows the output of the program:

```
Temperatura em Celsius: 39.0
Temperatura em Fahrenheit: 102.2
> |
```

Programiz

Para **salvar um programa** escrito no Programiz:

- Copiar o programa para memória (Ctrl+C).
- Abrir um editor de arquivos texto.
- Colar o programa no editor (Ctrl+V).
- Salvar o arquivo com a extensão .py

Integrated Development Environments (IDEs) Python

QPython 3L

- Ambiente de desenvolvimento Python para celulares com sistema operacional Android.
- É gratuito e pode ser obtido através do Google Play.

Pythonista e Pyto

- Ambientes de desenvolvimento Python para iPhone.
- Aplicativos pagos.
- Podem ser obtidos através do App Store.

Identificadores

- Um identificador é **qualquer nome aceito** pela linguagem:
 - Variáveis
 - Palavras reservadas
- Python é ***Case Sensitive***
 - Diferencia maiúscula de minúscula.
 - AB, Ab, aB e ab são variáveis diferentes!
- Podem ter qualquer tamanho.
- São formadas por letras, números e sublinhado.
- Iniciam sempre por uma letra.

Variáveis

- O **nome de uma variável** é um identificador.
- Podem ter qualquer quantidade de caracteres.

Exemplo: `notaAlunoUERJPrimeiroSemestre2020`

- **Tipagem dinâmica:** não precisam ser declaradas.
- **Tipagem forte:** a partir da atribuição de um valor elas passam a ser do tipo do valor atribuído.

Tipos de dados simples

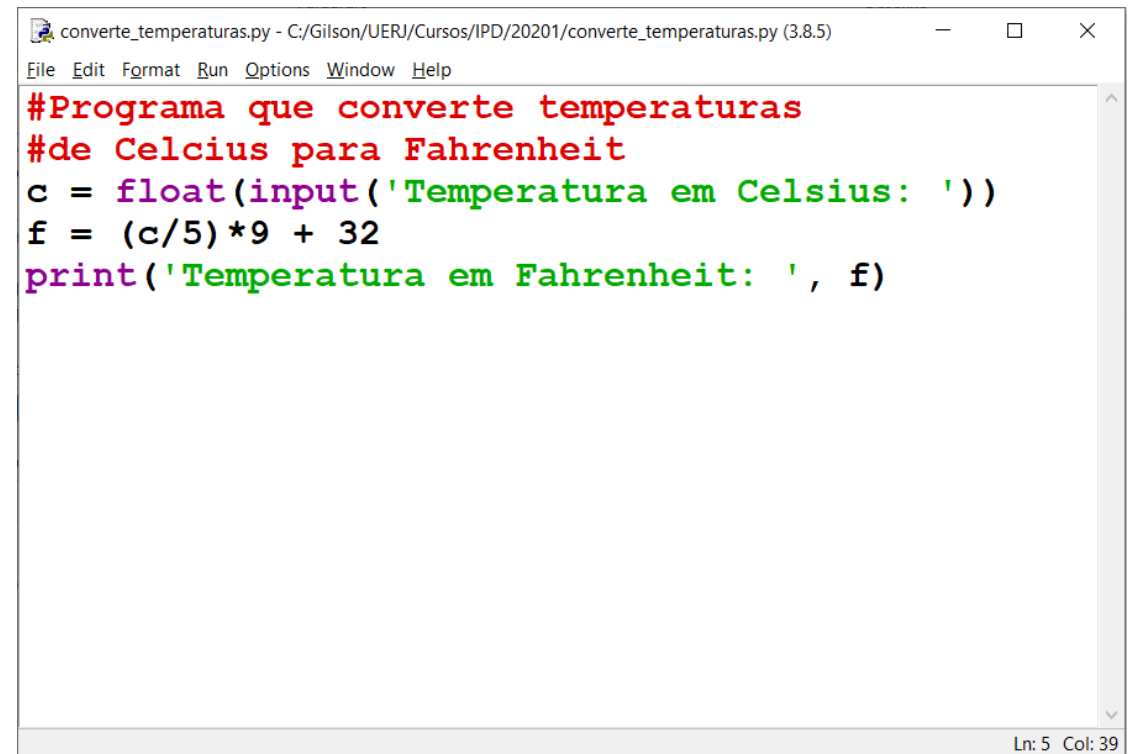
- Inteiros: *int* (4 bytes)
- Inteiros longos: *long* (8 bytes)
- Reais: *float* (8 bytes)
- Números complexos: *complex* (8 bytes)
- Literais: *str* (um byte por caractere)
- Lógicos: *bool* (1 byte)

Indentação

- Os **blocos de comandos** são delimitados em Python pela **indentação**.
- Significa **recuo** (derivado da palavra em inglês *indentation*, também grafado nas formas *identação* e *endentação*)
- O número de espaços no recuo é variável, mas todas as instruções dentro de um bloco têm de ser recuadas na mesma quantidade de espaços.
- Em Python a **indentação é obrigatória!**

Comentários

- São muito importantes para dar **clareza aos programas**.
- São ignorados pelo interpretador: é como se não existissem.
- São precedidos pelo caractere **#** (tralha ou *hash*).



The screenshot shows a window titled "converte_temperaturas.py - C:/Gilson/UERJ/Cursos/IPD/20201/converte_temperaturas.py (3.8.5)". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code is as follows:

```
#Programa que converte temperaturas
#de Celcius para Fahrenheit
c = float(input('Temperatura em Celsius: '))
f = (c/5)*9 + 32
print('Temperatura em Fahrenheit: ', f)
```

The status bar at the bottom right indicates "Ln: 5 Col: 39".

Operadores Aritméticos

+	(soma)
-	(subtração)
*	(multiplicação)
/	(divisão)
//	(divisão inteira)
%	(resto/módulo da divisão inteira)
**	(exponenciação)

Testar no Python...

Nesta aula vamos usar o *Programiz* para executar comandos e escrever programas Python.

<https://www.programiz.com/python-programming/online-compiler/>

Operadores Aritméticos

Testar no Python:

`4 + 3 - 2`

`2 * 3`

`7 / 2`

`7 // 2`

`7 % 2`

`2 ** 3`

`a = a + 1`

`a += 1 # a = a + 1`

Precedência de Operadores

1. ()
2. **
3. *, /, //, %
4. +, -

Em caso de mesma hierarquia resolve-se da esquerda para direita:

$$4 + (3^{**}2)//3 - 4\%3 * 5 - 2$$

Operadores Relacionais

== (igual)

!= (diferente)

> (maior que)

>= (maior ou igual)

< (menor que)

<= (menor ou igual)

Um operador relacional **retorna sempre um valor lógico** *False* ou *True*.

Operadores Relacionais

Testar no Python:

```
a = 2
```

```
b = 3
```

```
a == b
```

```
a != b
```

```
a > b
```

Operadores Lógicos

and (e lógico)

or (ou lógico)

not (não lógico)

Um operador lógico **retorna sempre um valor lógico** *False* ou *True*.

Operadores Lógicos

Testar no Python:

```
a = 2
```

```
b = 3
```

```
a == b
```

```
a == b and a != b
```

```
a == b or a != b
```

```
not (a == b and a != b)
```

Comandos de Entrada e Saída

- Comandos de **entrada e saída** são usados para enviar dados para o programa e mostrar os resultados do programa.
- Comandos básicos (funções de entrada e saída):
 - Entrada: `input(mensagem)`
 - Saída: `print(lista_de_variaveis)`

Comandos de Entrada e Saída

- **Sintaxe** do comando de saída:

`print(lista_de_variaveis)`

- *lista_de_variaveis* contém uma lista das variáveis (ou constantes) a ser impressa (apresentada na tela).
- Por exemplo:
`print(a)`
`print(b)`
`print(a, b)`
`print('Temperaturas:', c, 'Celsius ==', f, 'Fahrenheit')`

Comandos de Entrada e Saída

- **Sintaxe** do comando de entrada:

variavel = input('mensagem')

- *variavel* é o nome da variável a ter um valor atribuído pelo comando.
- *mensagem* é um texto para auxiliar o usuário (para saber o que ele tem que fazer).
- A mensagem é opcional:

variavel = input()

Comandos de Entrada e Saída

Testar no Python:

```
a = input('Entre com um número: ')\nb = input('Entre com outro número: ')\nc = a + b\nprint('A soma dos dois números é:', c)
```

Como assim?

Comandos de Entrada e Saída

- O comando `input()` **só lê *strings*** (cadeias de caracteres).
- Você deve **transformar a saída** do comando, indicando o tipo de dado desejado.
- Por exemplo:

```
a = int(input('Entre com um número inteiro: '))  
b = float(input('Entre com um número real: '))
```
- **Cuidado:** se o usuário entrar com um valor de tipo errado, seu programa vai dar erro!

Vamos programar...

Fazer um programa para ler 2 números e imprimir a soma.

```
a = int(input('n1: '))  
b = int(input('n2: '))  
c = a + b  
print('soma: ', c)
```

Vamos programar...

Fazer um programa para ler 4 números e imprimir a média.

```
a = float ('N1: ')
b = float ('N2: ')
c = float ('N3: ')
d = float ('N4: ')
result = (a+b+c+d)/4
print('soma: ', result)
```

Vamos programar...

Fazer um programa para ler o valor do tempo em segundos e imprimir em hora, minuto e segundos. Ex: 4000s = 1h 6min 40s

```
totalSeg = int(input('tempo em segundos: '))
hora = totalSeg // 3600
minuto = (totalSeg % 3600) // 60
segundo = (totalSeg % 3600) % 60
print(hora,':',minuto,':',segundo)
```



Introdução ao Processamento de Dados

Turma 3 (2020.1)



Introdução ao Python

Gilson. A. O. P. Costa (IME/UERJ)

gilson.costa@ime.uerj.br