

Aprendendo a programar com



Sumário

1.	O que é Python?	5
1.1	Porque utilizar Python?	6
1.2	Breve Histórico	7
2.	Instalando o Python	8
3.	Comentários (#)	21
4.	Tipos de Dados	23
5.	Quebra de linha (\n)	29
6.	Print e Input	32
	6.1 Exercícios Práticos	41
7.	Biblioteca Math	52
8.	Biblioteca Random	59



Sumário

8.1	<u>Exercícios Práticos</u>	65
9.	<u>Arquivos Externos</u>	70
10.	<u>Strings</u>	84
10.1	<u>Exercícios Práticos</u>	129
11.	<u>Módulo da Divisão</u>	134
12.	<u>Estruturas Condicionais</u>	141
12.1	<u>Exercícios Práticos</u>	170
13.	<u>Estrutura de Repetição</u>	184
13.1	<u>Exercícios Práticos</u>	199
14.	<u>Vetores</u>	211
15.	<u>Listas</u>	215



Sumário

16. Matrizes	238
16.1 Exercícios Práticos	243
17. Funções	254
17.1 Exercícios Práticos	269
18. Arquivos	278
18.1 Exercícios Práticos	299
19. Struct	311
19.1 Exercícios Práticos	323
19. Alocação de Memória	328
20. Árvores	348
20.1 Exercícios Práticos	381
21. Referências	386



O que é Python?

- Python é uma linguagem de programação interpretada, orientada a objetos, de alto nível e com semântica dinâmica;
- A simplicidade do Python reduz a manutenção de um programa;
- É uma das linguagens que mais tem crescido devido sua compatibilidade (roda na maioria dos sistemas operacionais) e capacidade de auxiliar outras linguagens.



Porque utilizar Python?

- O Python possui uma sintaxe clara e concisa, tornando a linguagem mais produtiva;
- A linguagem inclui diversas estruturas de alto nível (listas, dicionários, data / hora, complexos e outras) e uma vasta coleção de módulos prontos para uso;
- O Python também é muito utilizado como linguagem *script* em vários softwares, permitindo automatizar tarefas e adicionar novas funcionalidades, entre eles: *BrOffice.org*, *PostgreSQL*, *Blender*, *GIMP* e *Inkscape*.



Breve Histórico

- A linguagem foi criada por Guido Van Rossum em 1990;
- O Python foi concebido a partir de outra linguagem existente na época, chamada ABC;
- Hoje em dia a linguagem é bem aceita na indústrias por empresas de alta tecnologia, tais como: Google, yahoo, Microsoft, Nokia e Disney.



Instalando o Python

- Antes de iniciar a programar com Python, vamos fazer um teste e verificar se o seu sistema possui alguma versão do programa instalado.
- Para isso basta digitar em sua tela inicial do Windows **cmd** que logo irá abrir uma janela:



Instalando o Python

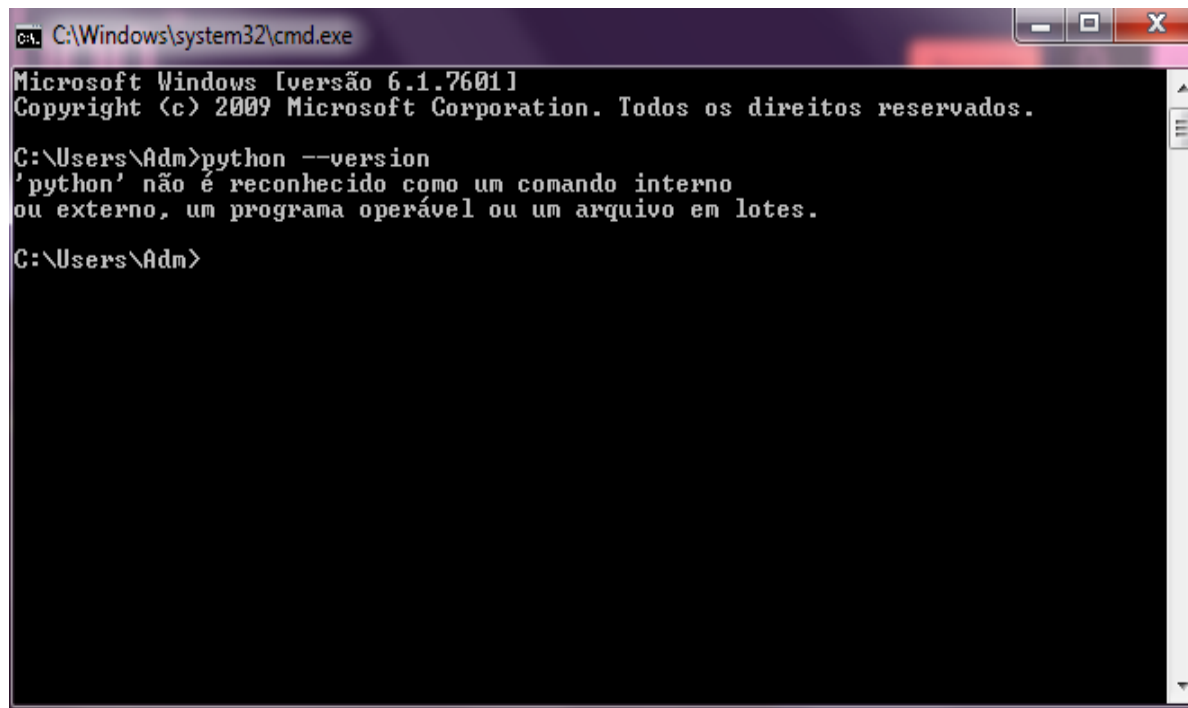


A screenshot of a Windows Command Prompt window. The title bar shows the path 'C:\Windows\system32\cmd.exe'. The window contains the following text: 'Microsoft Windows [versão 6.1.7601]', 'Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.', and the command prompt 'C:\Users\Adm>_'. The background of the command prompt is black with white text.



Instalando o Python

- Quando digitamos “ Python - - version ” é possível verificar que o programa não se encontra instalado neste sistema.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [versão 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

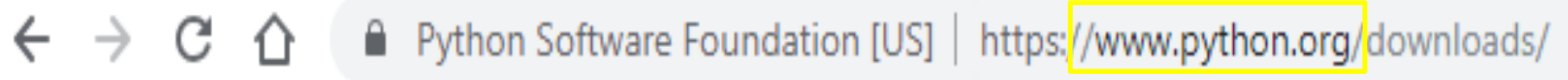
C:\Users\Adm>python --version
'python' não é reconhecido como um comando interno
ou externo, um programa operável ou um arquivo em lotes.

C:\Users\Adm>
```



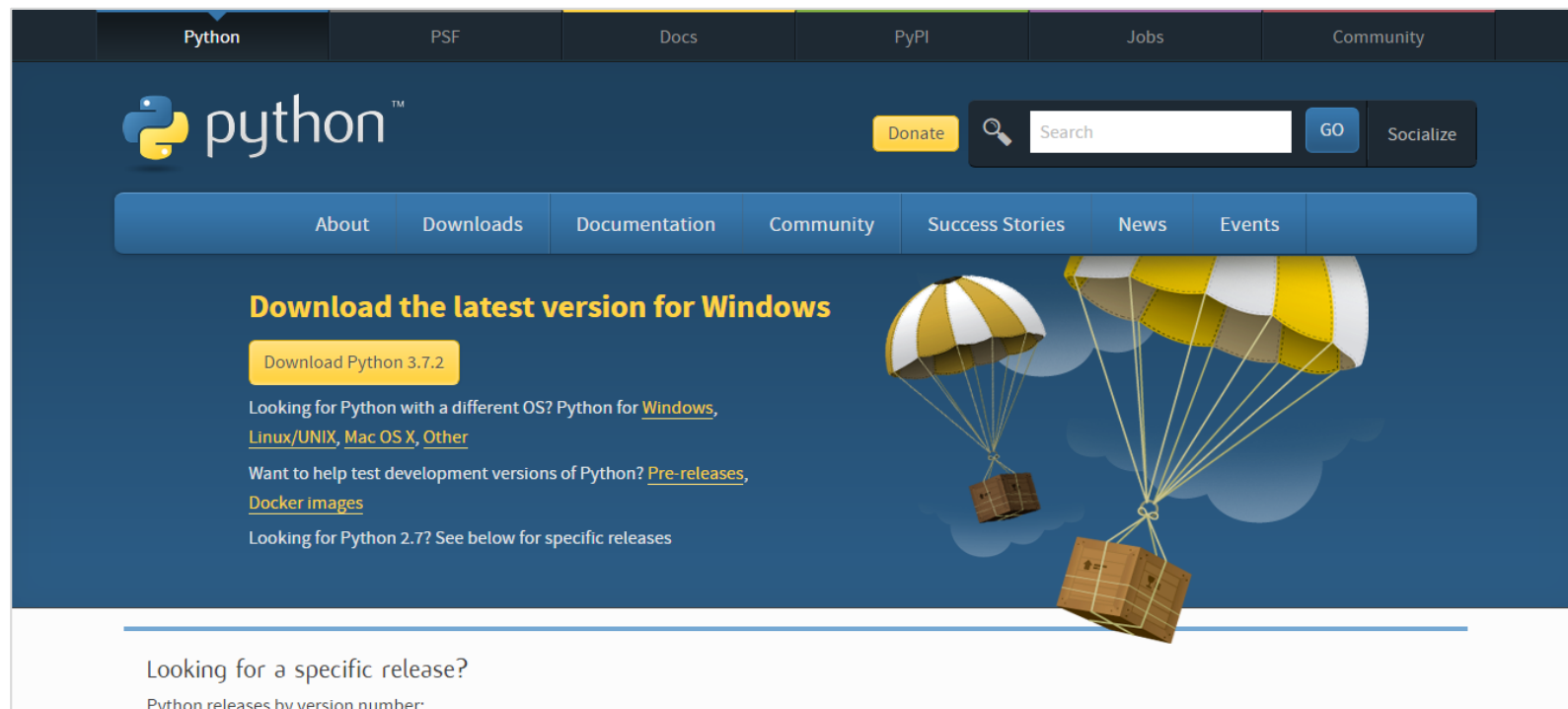
Instalando o Python

- Para fazer o download do Python é necessário acessar o site oficial que é disponibilizado no seguinte endereço:



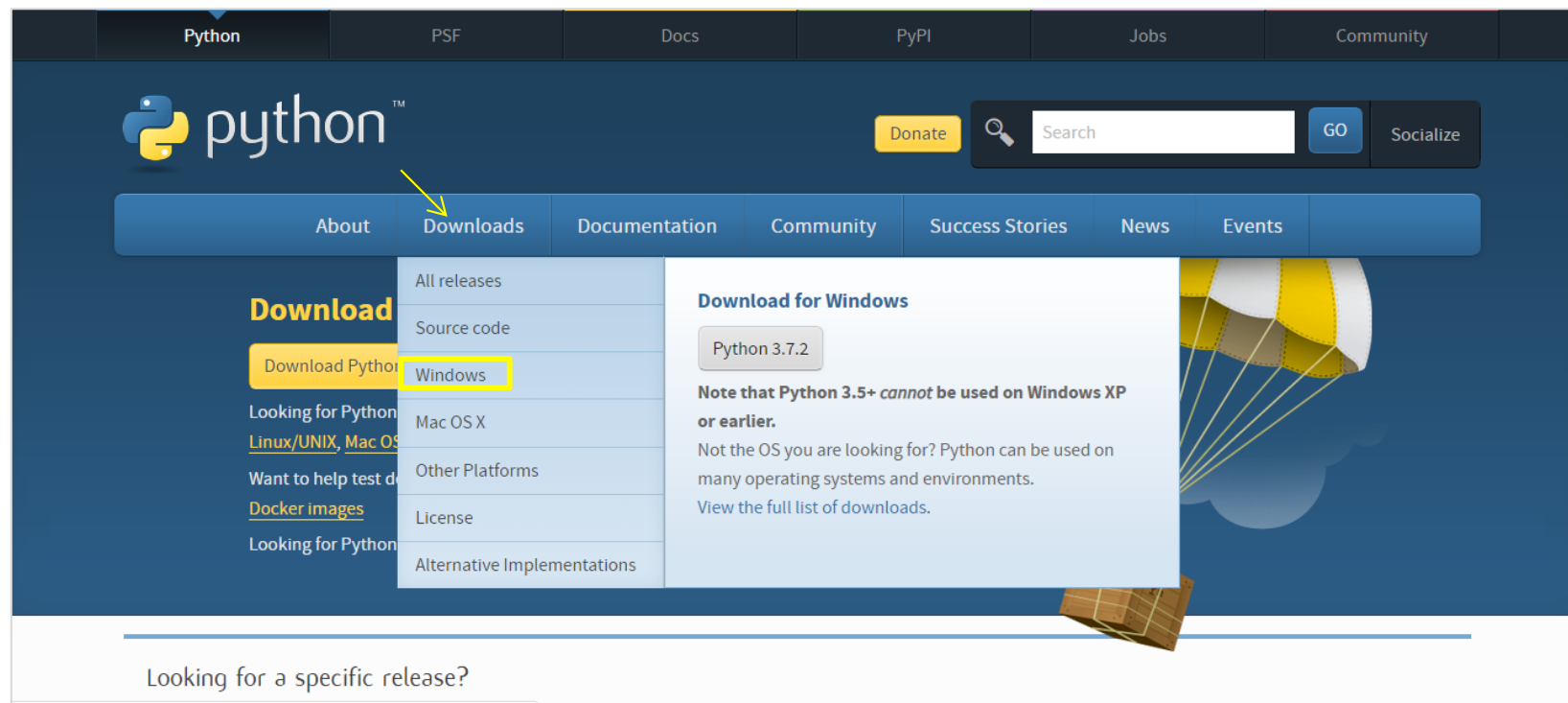
Instalando o Python

- Ao acessar o endereço irá abrir a seguinte página do site oficial do python.



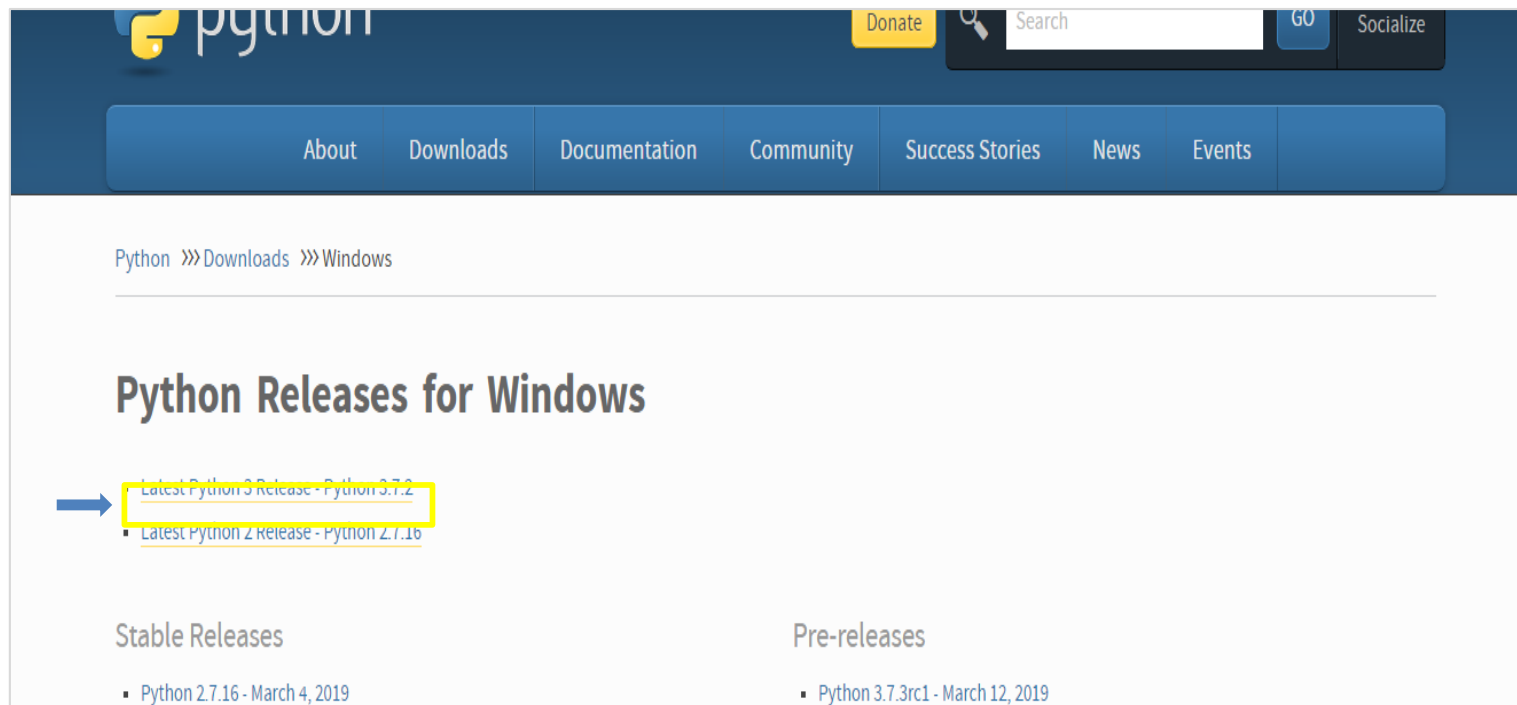
Instalando o Python

- Próximo passo é acessar a aba **downloads** e clicar na opção **Windows**.



Instalando o Python

- Agora vamos escolher a versão **Python 3.7.2** pois é a versão mais madura e mais recomendável para o USO.



Instalando o Python

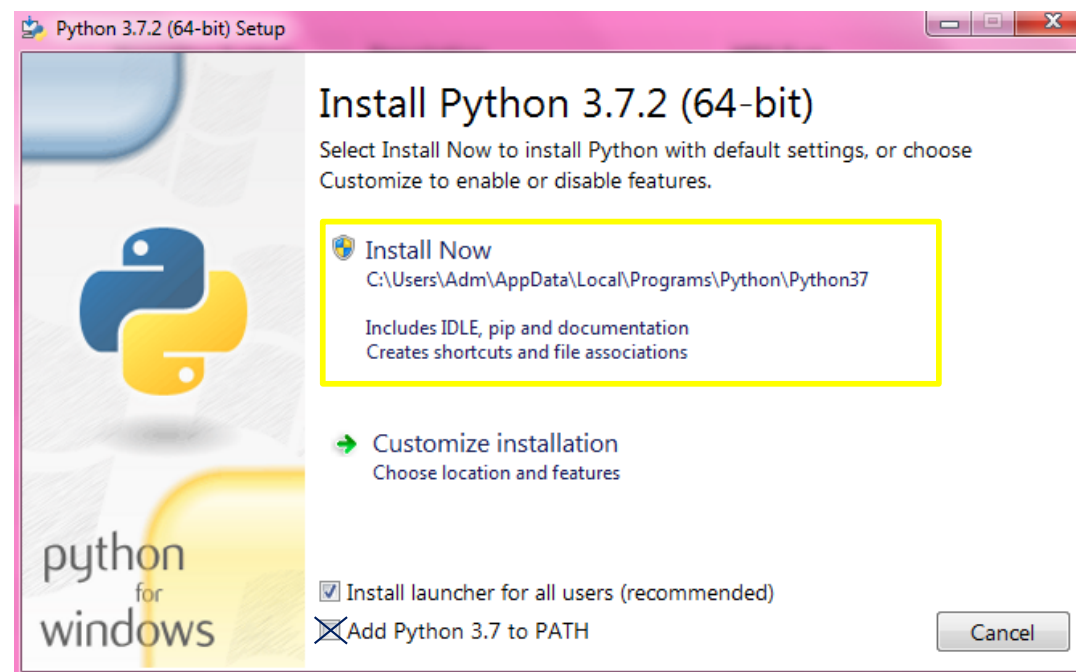
- Então é só clicar no link e realizar o download para Windows 64 bit, caso seu Windows seja mais antigo é só baixar a versão 32 bit.

Files					
Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		02a75015f7cd845e27b85192bb0ca4cb	22897802	SIG
XZ compressed source tarball	Source release		df6ec36011808205beda239c72f947cb	17042320	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	d8ff07973bc9c009de80c269fd7efcca	34405674	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	0fc95e9f6d6b4881f3b499da338a9a80	27766090	SIG
Windows help file	Windows		941b7d6279c0d4060a927a65dcab88c4	8092167	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	f81568590bef56e5997e63b434664d58	7025085	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	ff258093f0b3953c886192dec9f52763	26140976	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	8de2335249d84fe1eeb61ec25858bd82	1362888	SIG
Windows x86 embeddable zip file	Windows		26881045297dc1883a1d61baffeecafo	6533256	SIG
Windows x86 executable installer	Windows		38156b62c0cbcb03bfddeb86e66c3a0f	25365744	SIG
Windows x86 web-based installer	Windows		1e6c626514b72e21008f8cd53f945f10	1324648	SIG



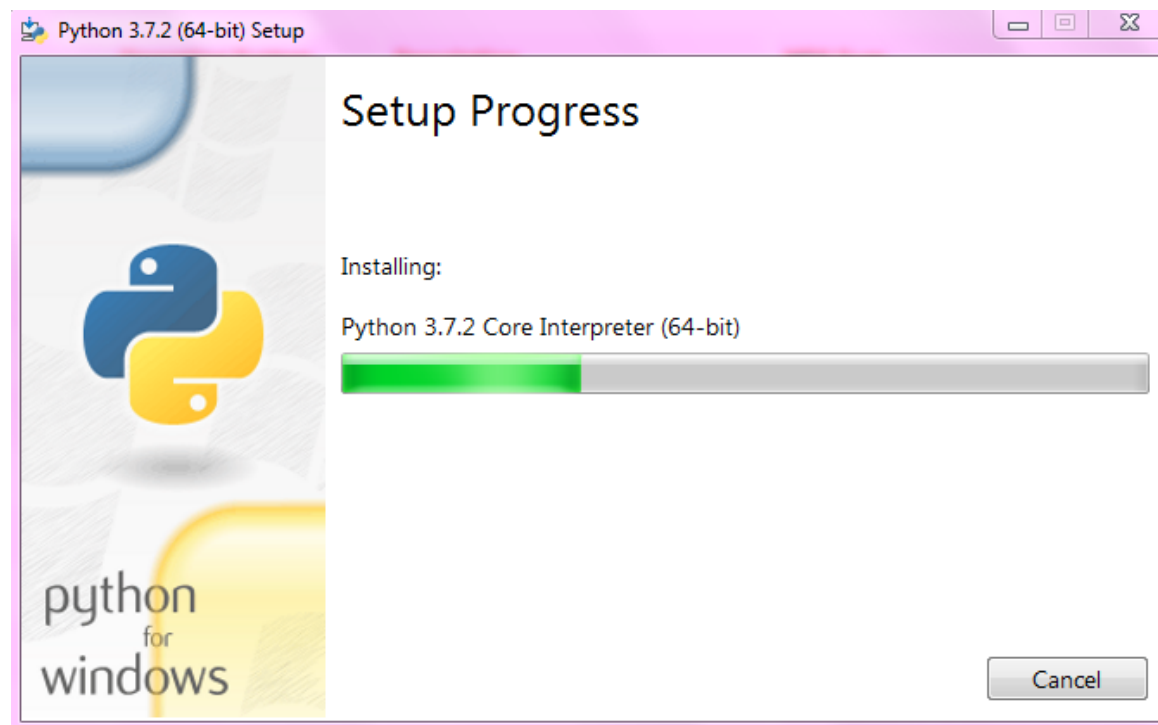
Instalando o Python

- Após o download do programa, a seguinte janela será aberta. Marque a opção “ Add Python 3.7 to PATH” e opta-se por “ Install now”.



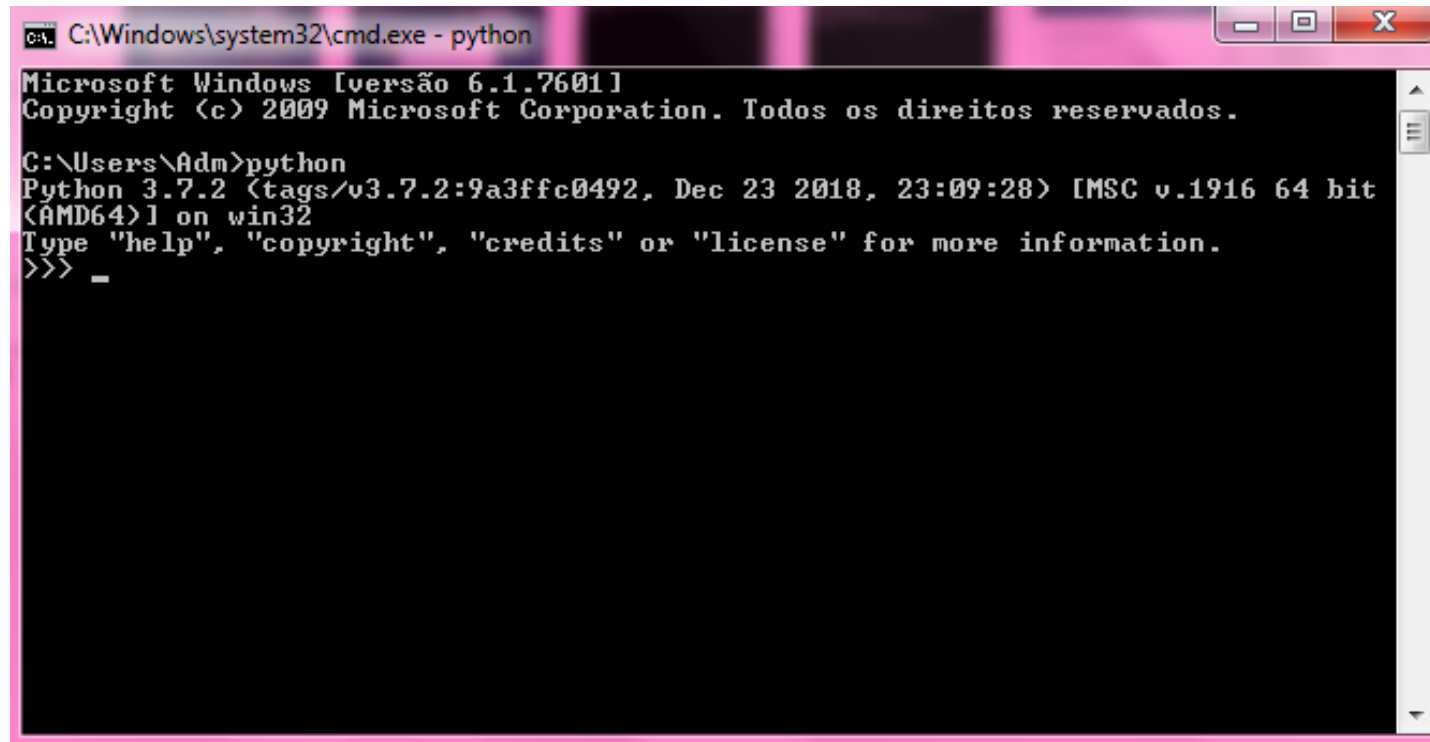
Instalando o Python

- Então o programa levará alguns minutos para instalar em seu sistema.



Instalando o Python

- Abra novamente a janela e verifique se o Python 3.7.2 foi instalado.



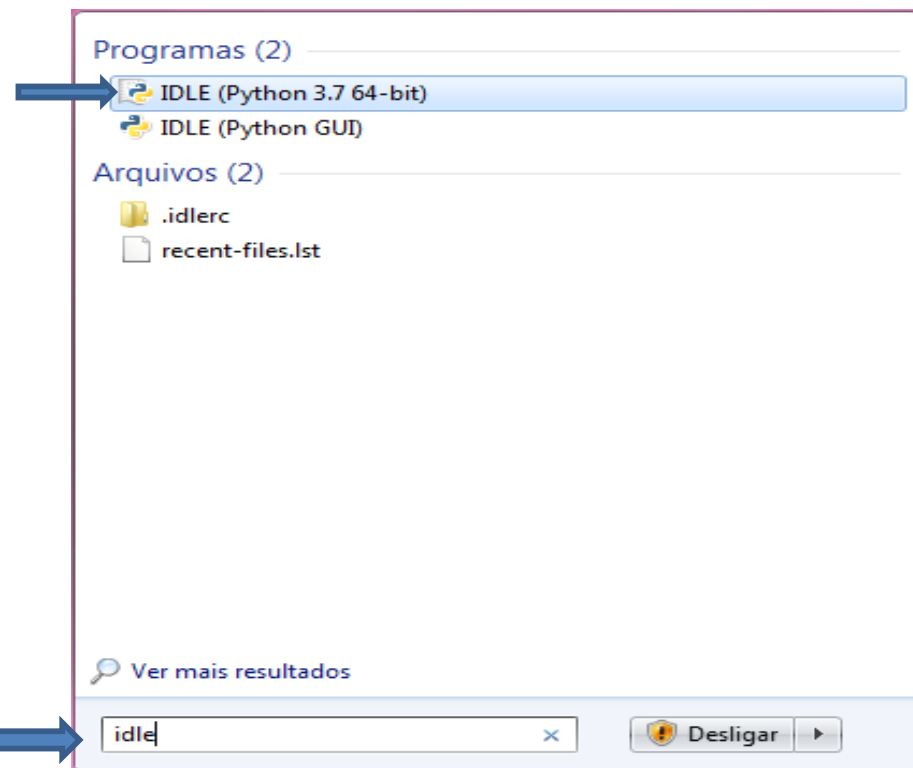
```
C:\Windows\system32\cmd.exe - python
Microsoft Windows [versão 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Adm>python
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```



Instalando o Python

- Para melhor trabalhar com Python 3.7.2 vamos utilizar o IDLE do programa.



Digite na menu inicial IDLE



Instalando o Python



The image shows a screenshot of the Python 3.7.2 Shell window. The window has a title bar that says "Python 3.7.2 Shell". 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 displays the following text: "Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32", "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 Col: 4".

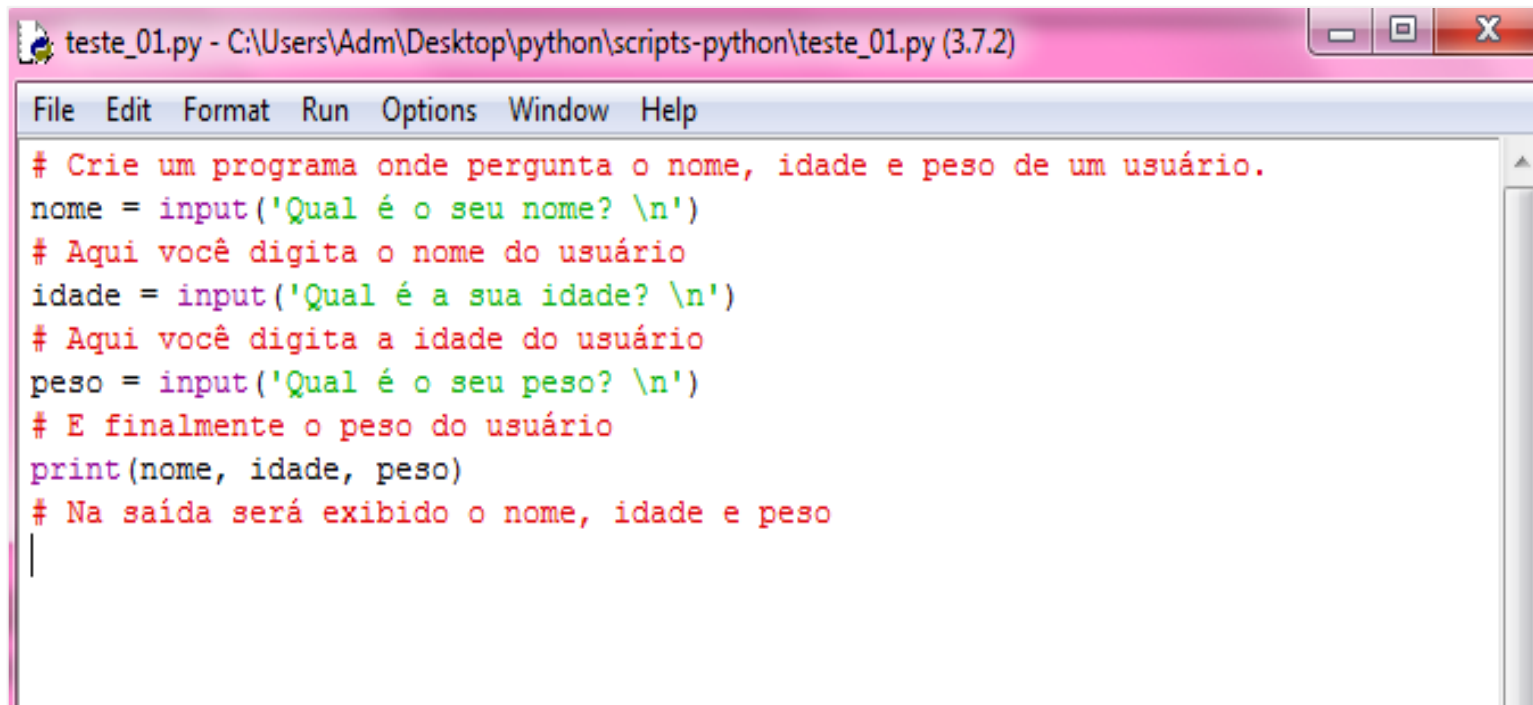
```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

Tela inicial do IDLE



Comentários (#)

- O caractere (#) marca o início de comentário. Qualquer texto depois do (#) será ignorado até o fim da linha.



```
File Edit Format Run Options Window Help
# Crie um programa onde pergunta o nome, idade e peso de um usuário.
nome = input('Qual é o seu nome? \n')
# Aqui você digita o nome do usuário
idade = input('Qual é a sua idade? \n')
# Aqui você digita a idade do usuário
peso = input('Qual é o seu peso? \n')
# E finalmente o peso do usuário
print(nome, idade, peso)
# Na saída será exibido o nome, idade e peso
|
```

Comentários (#)



ATENÇÃO !

- ↳ Lembre-se o comentário não será executado e interpretado, apenas serve como auxiliar para que você possa comentar algo sobre o programa que está sendo criado.



Tipos de dados em Python

- Em Python trabalhamos com os **Tipos Primitivos** que são os tipos de dados mais simples, isto é, a informação em sua forma mais primitiva;
- Os tipos primitivos são chamados de tipos **built-ins**, onde esta nomenclatura é utilizada para indicar que estamos utilizando informações que foram definidas, por padrão, através de classes dentro da Máquina Virtual do Python.

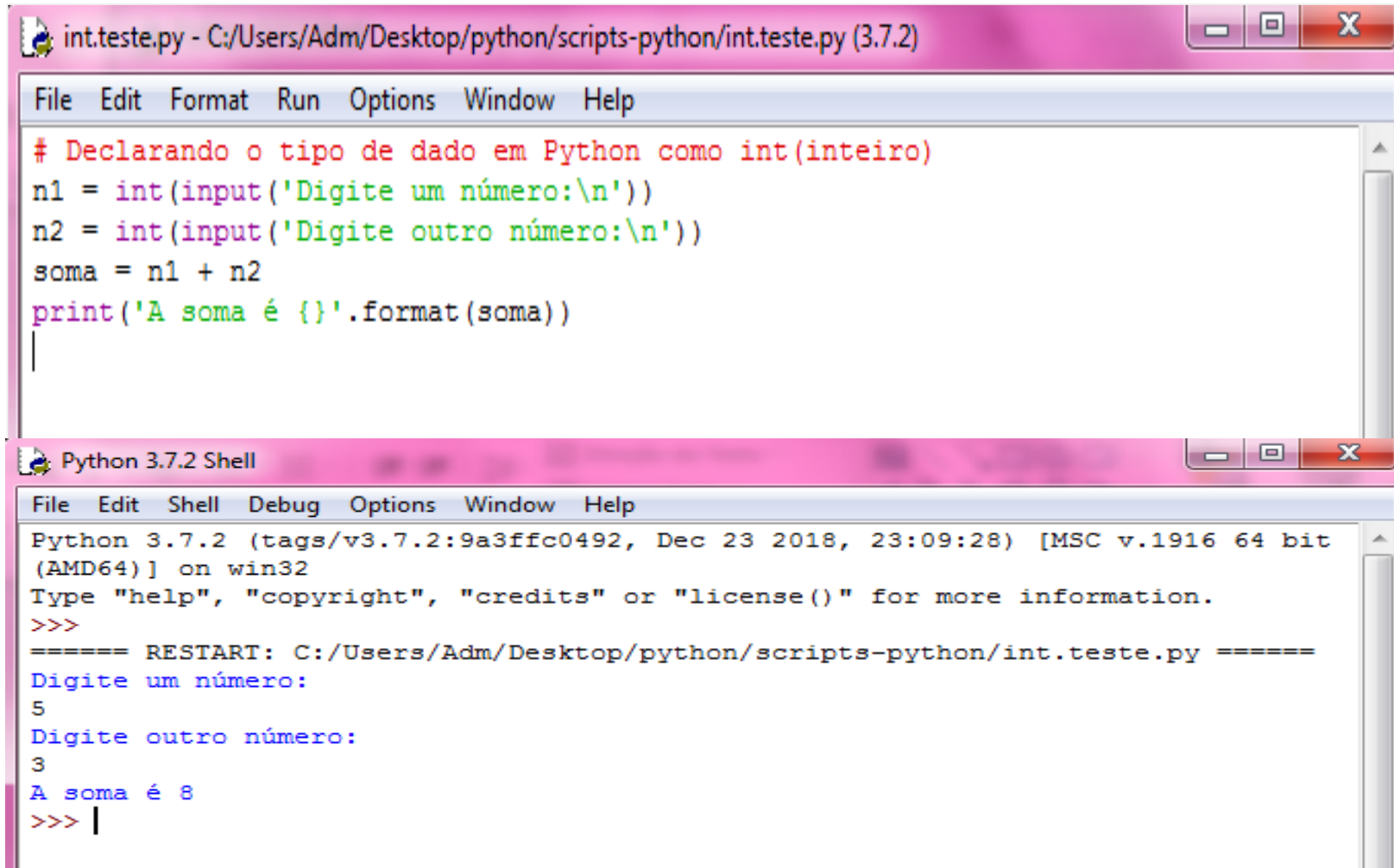


Tipos de dados em Python

- A seguir, temos a lista dos principais tipos built-ins da linguagem Python:
 - **int** - para números inteiros;
 - **Float** – para números decimais;
 - **str** - para conjunto de caracteres;
 - **bool** - armazena True ou False;
 - **list** - para agrupar um conjunto de elementos.



Tipos de dados em Python



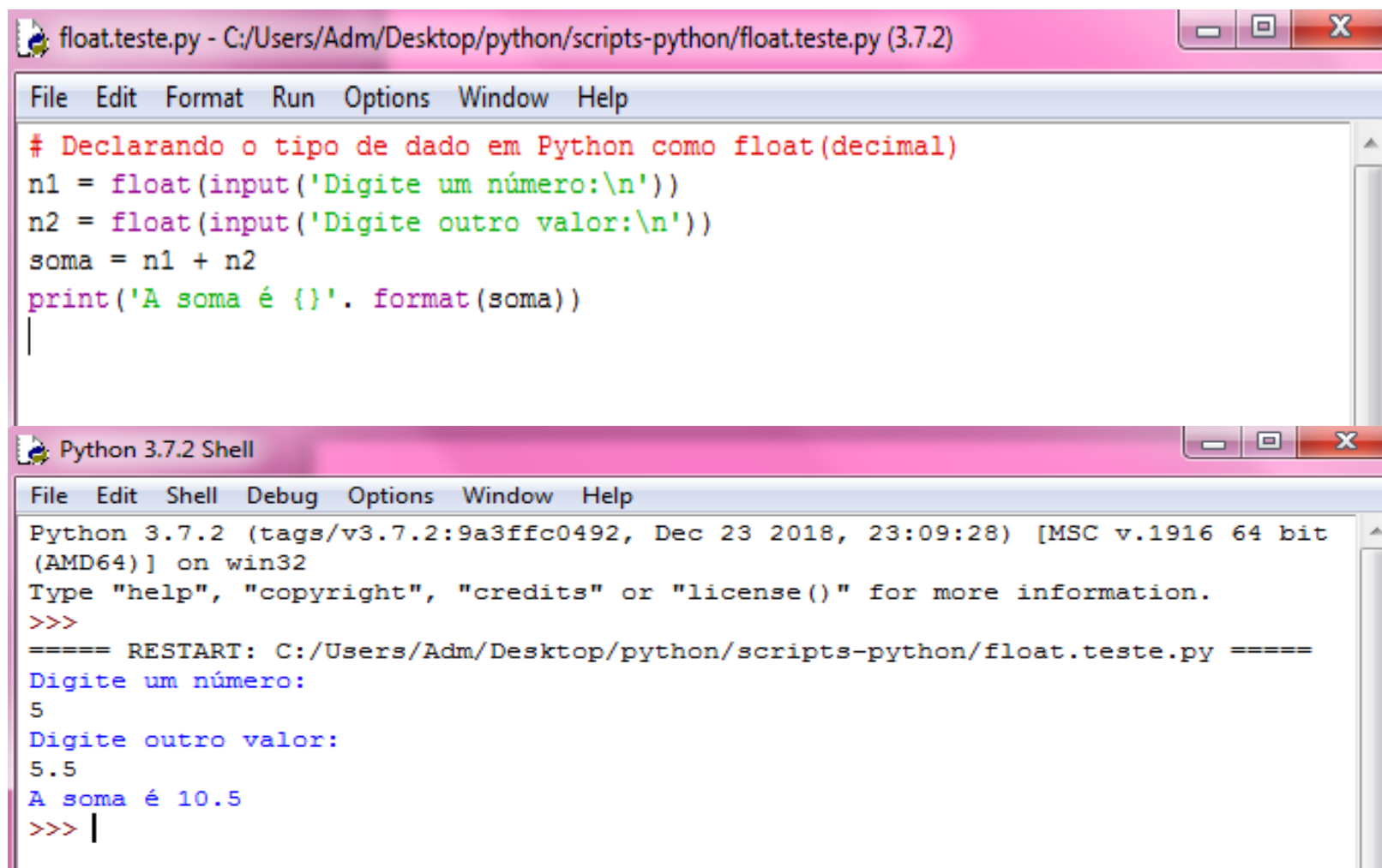
The image shows a screenshot of a Python IDE with two windows. The top window, titled 'int.teste.py - C:/Users/Adm/Desktop/python/scripts-python/int.teste.py (3.7.2)', contains the following Python code:

```
File Edit Format Run Options Window Help
# Declarando o tipo de dado em Python como int(inteiro)
n1 = int(input('Digite um número:\n'))
n2 = int(input('Digite outro número:\n'))
soma = n1 + n2
print('A soma é {}'.format(soma))
|
```

The bottom window, titled 'Python 3.7.2 Shell', shows the execution of the script:

```
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/int.teste.py =====
Digite um número:
5
Digite outro número:
3
A soma é 8
>>> |
```

Tipos de dados em Python



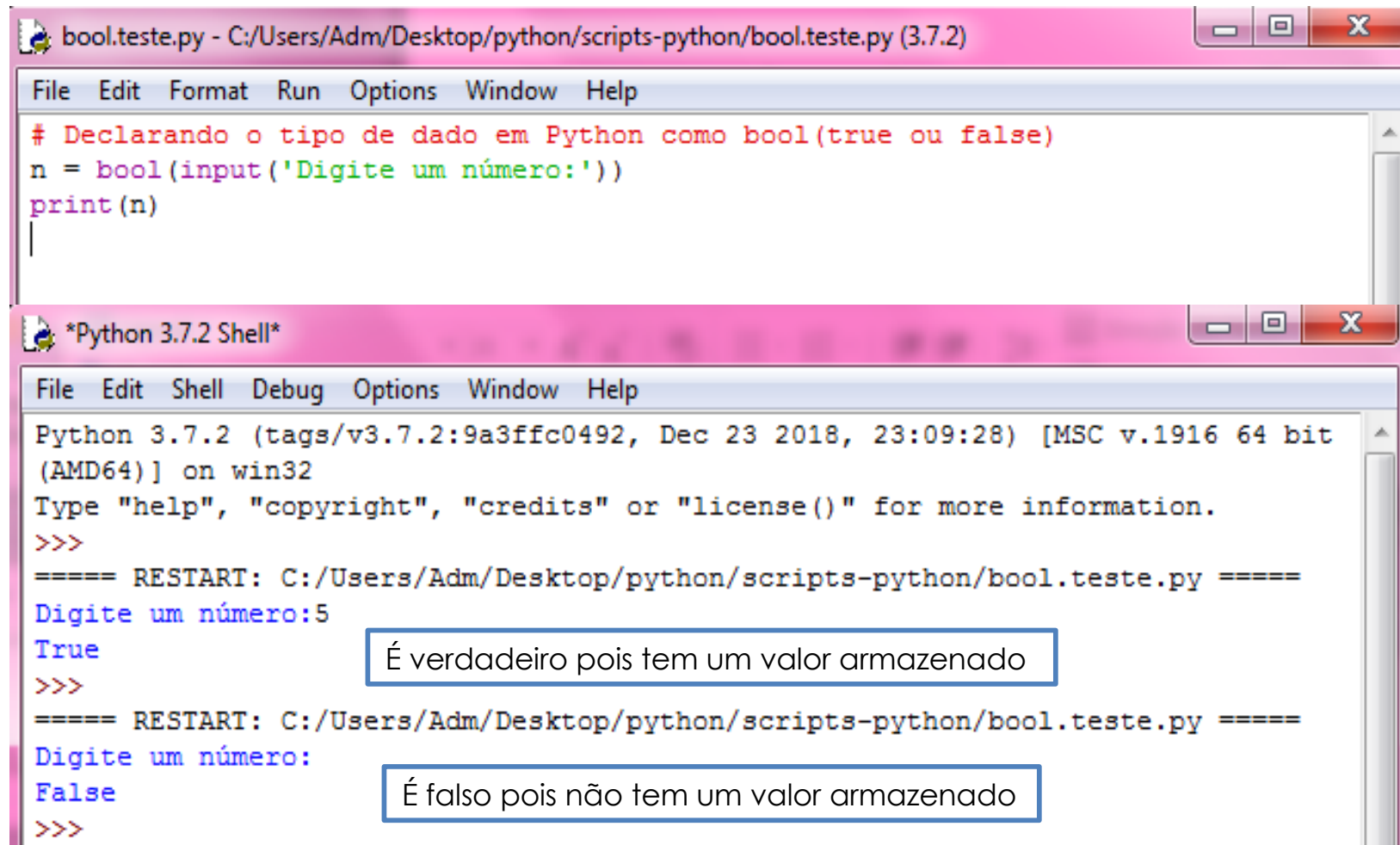
The image shows a screenshot of a Python IDE with two windows. The top window, titled 'float.teste.py - C:/Users/Adm/Desktop/python/scripts-python/float.teste.py (3.7.2)', contains the following Python code:

```
File Edit Format Run Options Window Help
# Declarando o tipo de dado em Python como float(decimal)
n1 = float(input('Digite um número:\n'))
n2 = float(input('Digite outro valor:\n'))
soma = n1 + n2
print('A soma é {}'.format(soma))
|
```

The bottom window, titled 'Python 3.7.2 Shell', shows the execution of the script:

```
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/float.teste.py =====
Digite um número:
5
Digite outro valor:
5.5
A soma é 10.5
>>> |
```

Tipos de dados em Python



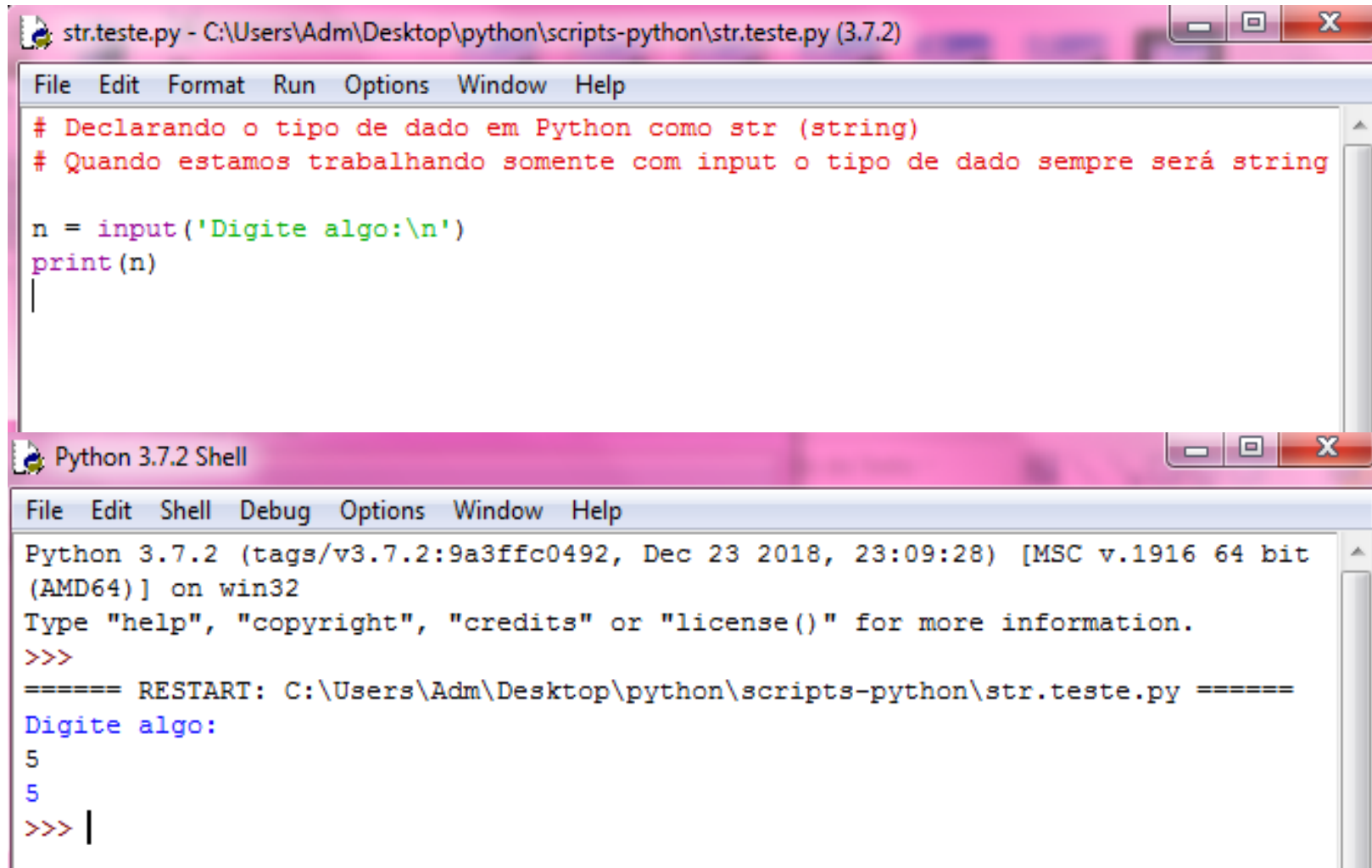
The image shows a screenshot of a Python IDE with two windows. The top window, titled 'bool.teste.py - C:/Users/Adm/Desktop/python/scripts-python/bool.teste.py (3.7.2)', contains the following code:

```
File Edit Format Run Options Window Help
# Declarando o tipo de dado em Python como bool(true ou false)
n = bool(input('Digite um número:'))
print(n)
```

The bottom window, titled '*Python 3.7.2 Shell*', shows the execution of the script. It displays the Python version and architecture, followed by a restart of the script. The user is prompted to enter a number. In the first run, the user enters '5', and the output is 'True'. In the second run, the user enters an empty string, and the output is 'False'. Two blue boxes with white text provide explanations for the results:

- For the first run (input '5'), the box says: "É verdadeiro pois tem um valor armazenado".
- For the second run (input ''), the box says: "É falso pois não tem um valor armazenado".

Tipos de dados em Python



The image shows a screenshot of a Python IDE window titled "str.teste.py - C:\Users\Adm\Desktop\python\scripts-python\str.teste.py (3.7.2)". The window contains a menu bar (File, Edit, Format, Run, Options, Window, Help) and a code editor with the following Python code:

```
# Declarando o tipo de dado em Python como str (string)
# Quando estamos trabalhando somente com input o tipo de dado sempre será string

n = input('Digite algo:\n')
print(n)
```

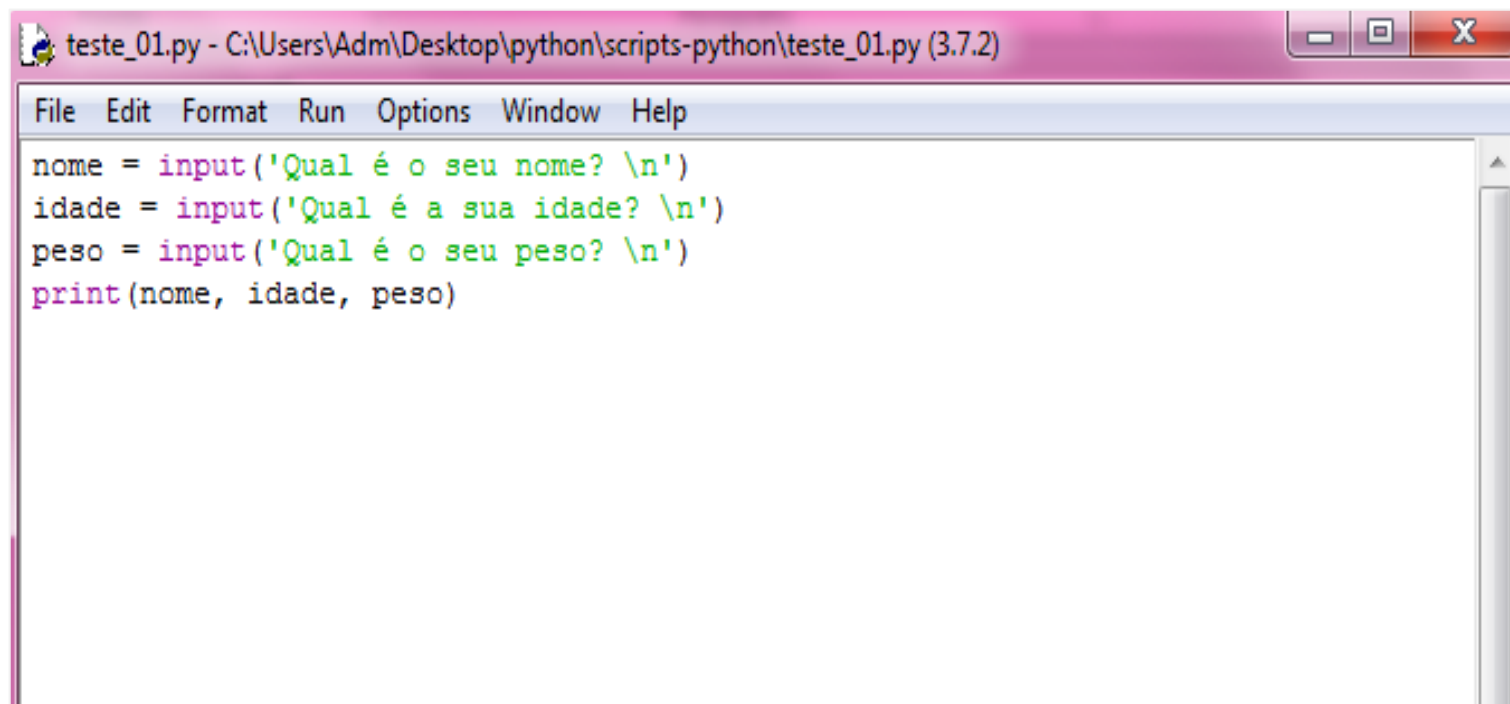
Below the code editor is a "Python 3.7.2 Shell" window. It has a menu bar (File, Edit, Shell, Debug, Options, Window, Help) and displays the following output:

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Adm\Desktop\python\scripts-python\str.teste.py =====
Digite algo:
5
5
>>> |
```



Quebra de linhas (\n)

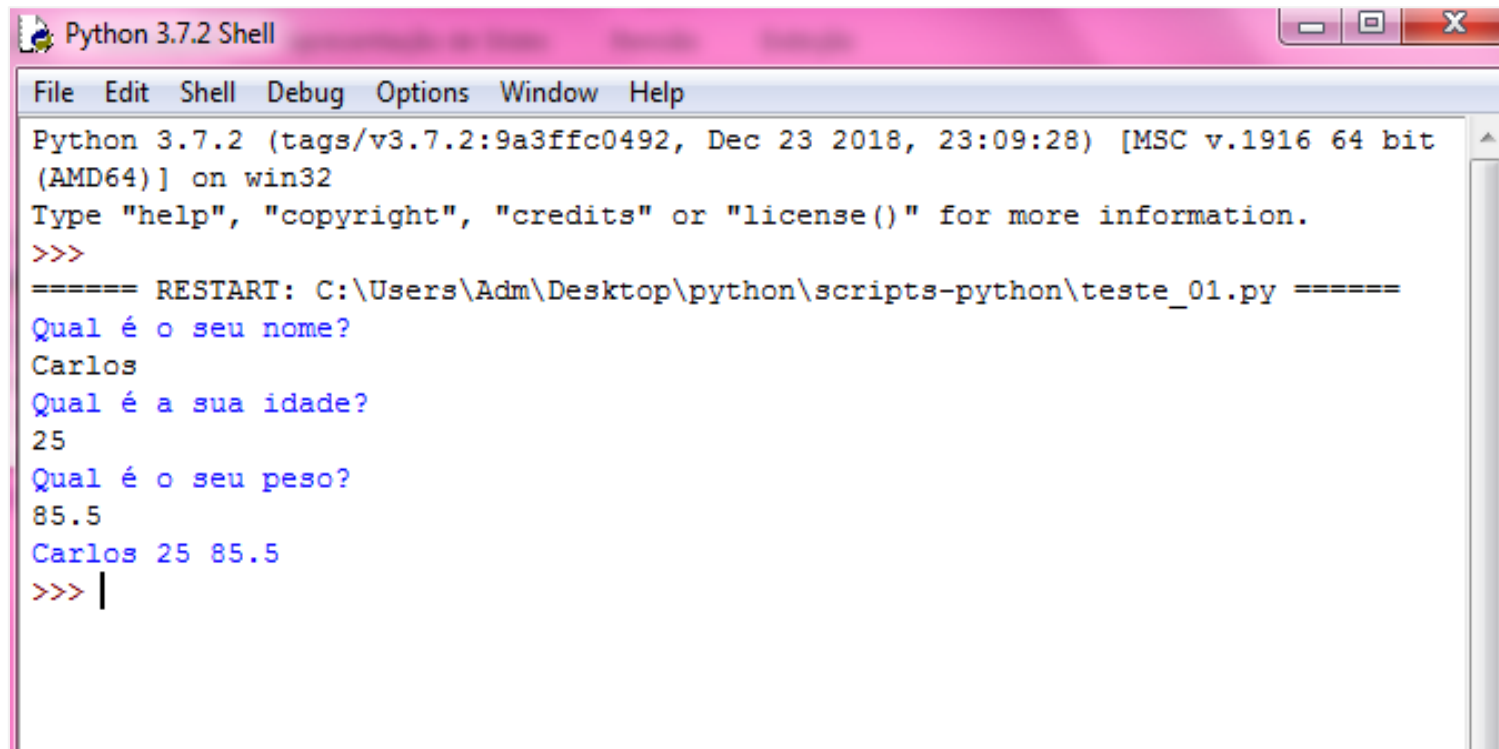
- Um programa feito em Python é constituído de linhas, que podem continuar nas linhas seguintes, pelo uso do caractere de barra invertida (\n).



```
teste_01.py - C:\Users\Adm\Desktop\python\scripts-python\teste_01.py (3.7.2)
File Edit Format Run Options Window Help
nome = input('Qual é o seu nome? \n')
idade = input('Qual é a sua idade? \n')
peso = input('Qual é o seu peso? \n')
print(nome, idade, peso)
```

Quebra de linhas (\n)

- Assim ocorrerá a quebra de linha entre as linhas que possuem o (\n).



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Adm\Desktop\python\scripts-python\teste_01.py =====
Qual é o seu nome?
Carlos
Qual é a sua idade?
25
Qual é o seu peso?
85.5
Carlos 25 85.5
>>> |
```

Quebra de linhas (\n)



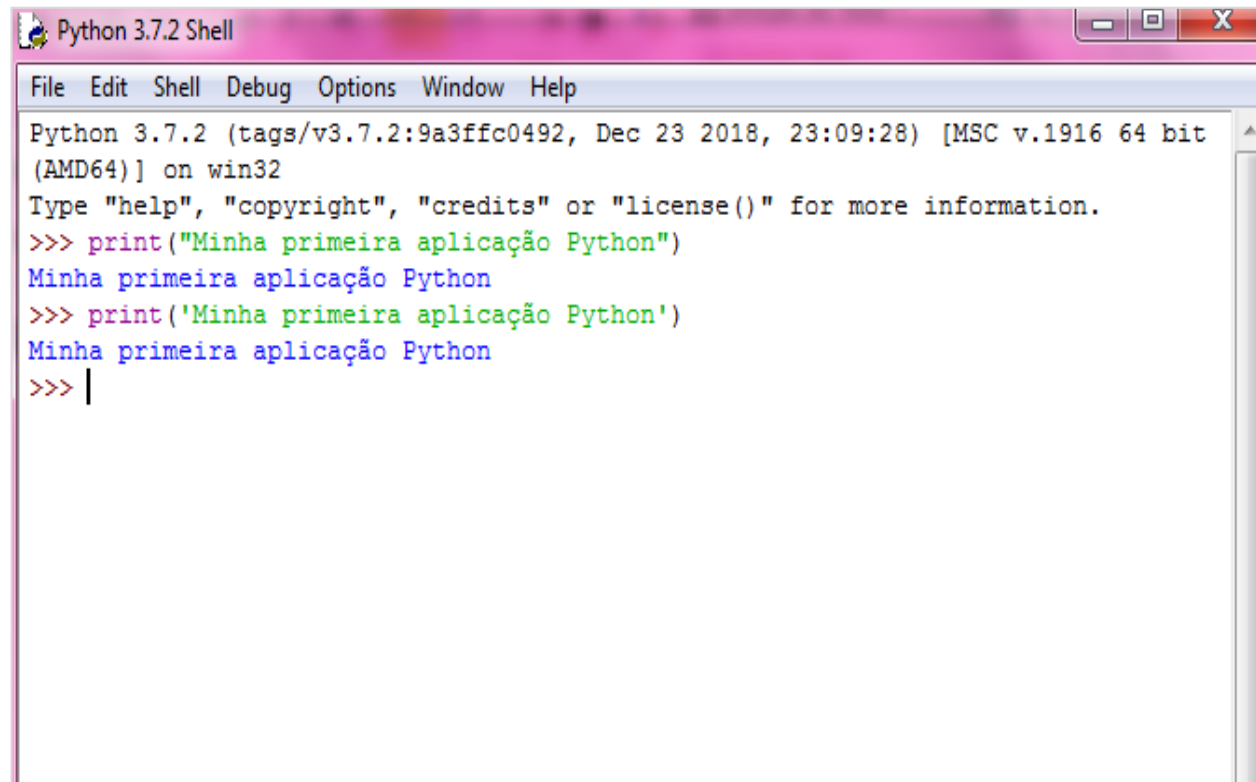
ATENÇÃO !

↳ Lembre-se a quebra de linhas (\n) deve-se ser implementada somente em '`\n`' ou "`\n`", caso contrário o programa não irá ler o comando e assim não irá executá-lo.



Funções Print e Input

- Para o nosso primeiro código vamos imprimir uma mensagem bem simples:



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Minha primeira aplicação Python")
Minha primeira aplicação Python
>>> print('Minha primeira aplicação Python')
Minha primeira aplicação Python
>>> |
```

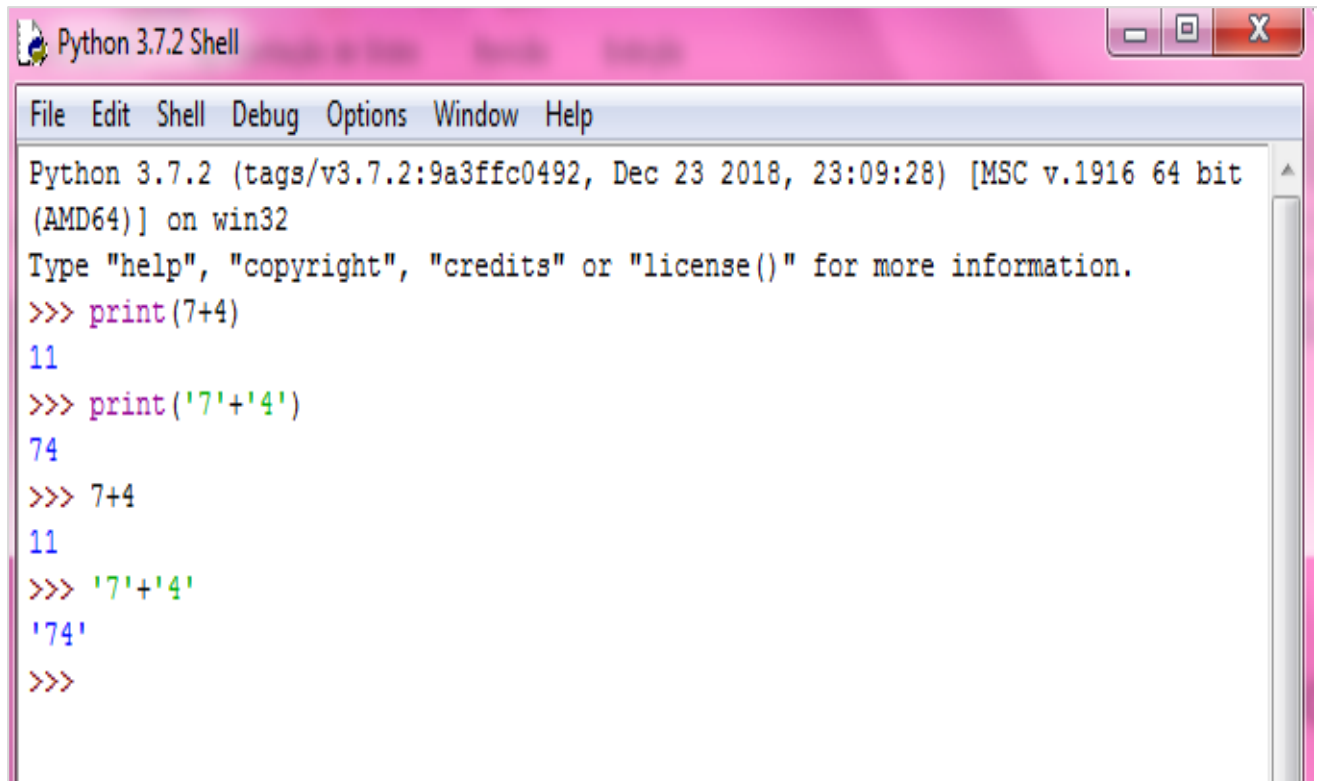

Funções **Print** e **Input**

- O **print()** é uma **função** do Python utilizada para imprimir alguma mensagem na tela;
- Uma mensagem deve estar delimitada entre aspas simples (") ou duplas (""");
- O interpretador, no modo interativo, já vai mostrar a saída deste comando no console, logo abaixo dele.



Funções Print e Input

- Vamos fazer uma operação matemática de soma bem simples:



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print(7+4)
11
>>> print('7'+ '4')
74
>>> 7+4
11
>>> '7'+ '4'
'74'
>>>
```



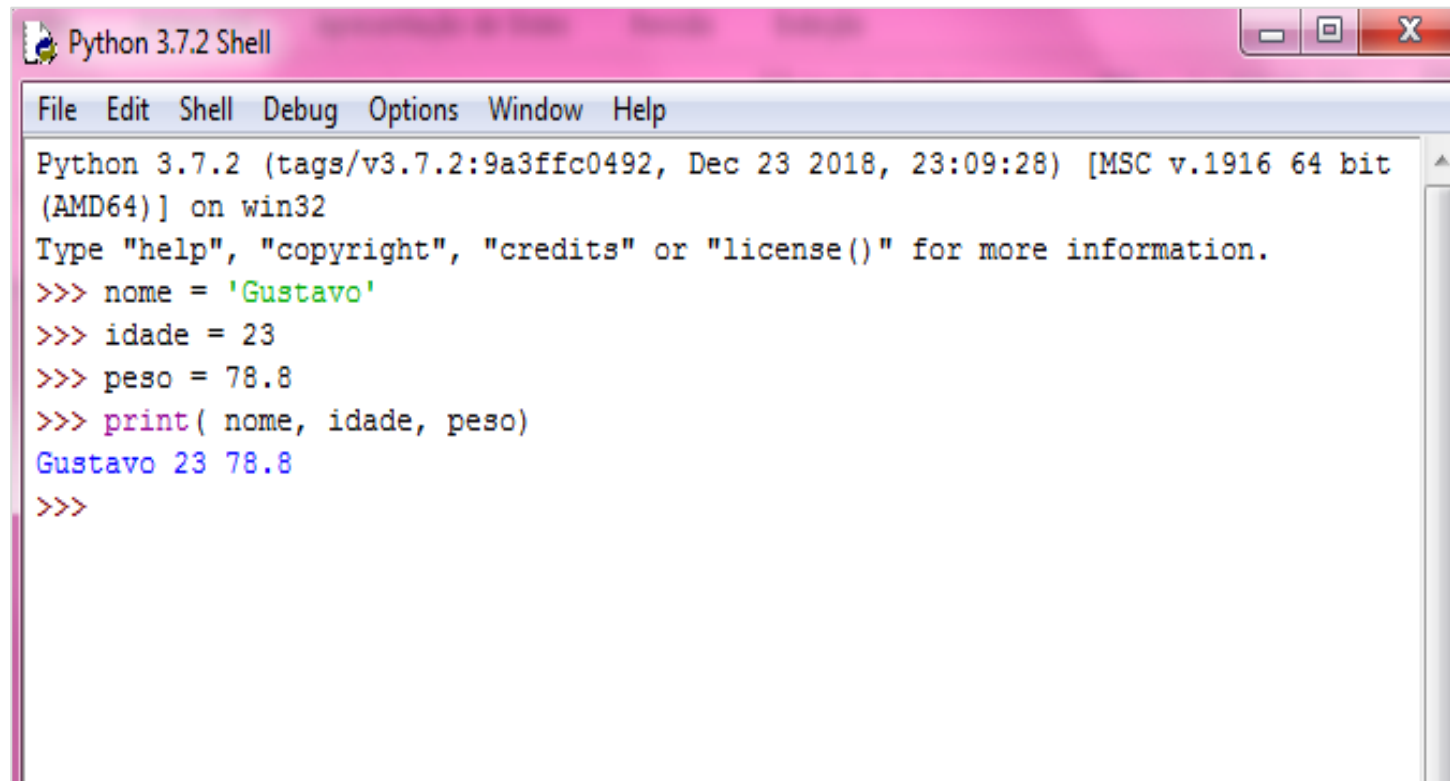
Funções **Print** e **Input**

- Como estamos trabalhando com IDLE podemos programar de uma maneira mais interativa digitando apenas `>>>7+4` sem a necessidade de `print` ;
- Quando digitamos `>>> '7' + '4'`, o programa irá entender que o número digitado entre aspas simples é uma palavra, assim fazendo com que a saída seja a união destes números e não fornecendo a soma.



Funções Print e Input

- Vamos imprimir em tela o nome, idade e peso de um usuário:



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> nome = 'Gustavo'
>>> idade = 23
>>> peso = 78.8
>>> print( nome, idade, peso)
Gustavo 23 78.8
>>>
```

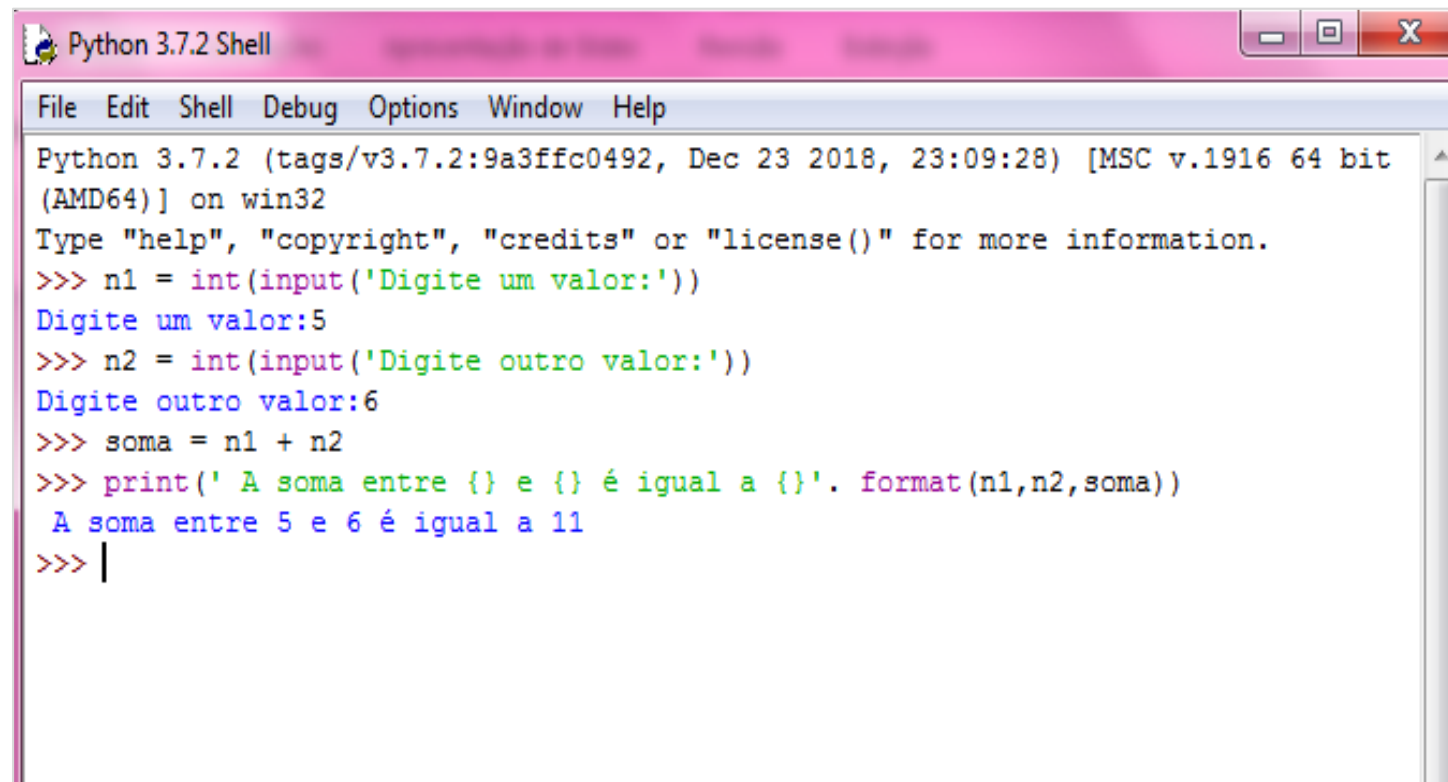
Funções **Print** e **Input**

- Vamos imprimir em tela o mesmo exercício anterior mas utilizando a função **input**.
- O **input()** é uma **função** do Python utilizada para ler alguma mensagem na tela;



Funções Print e Input

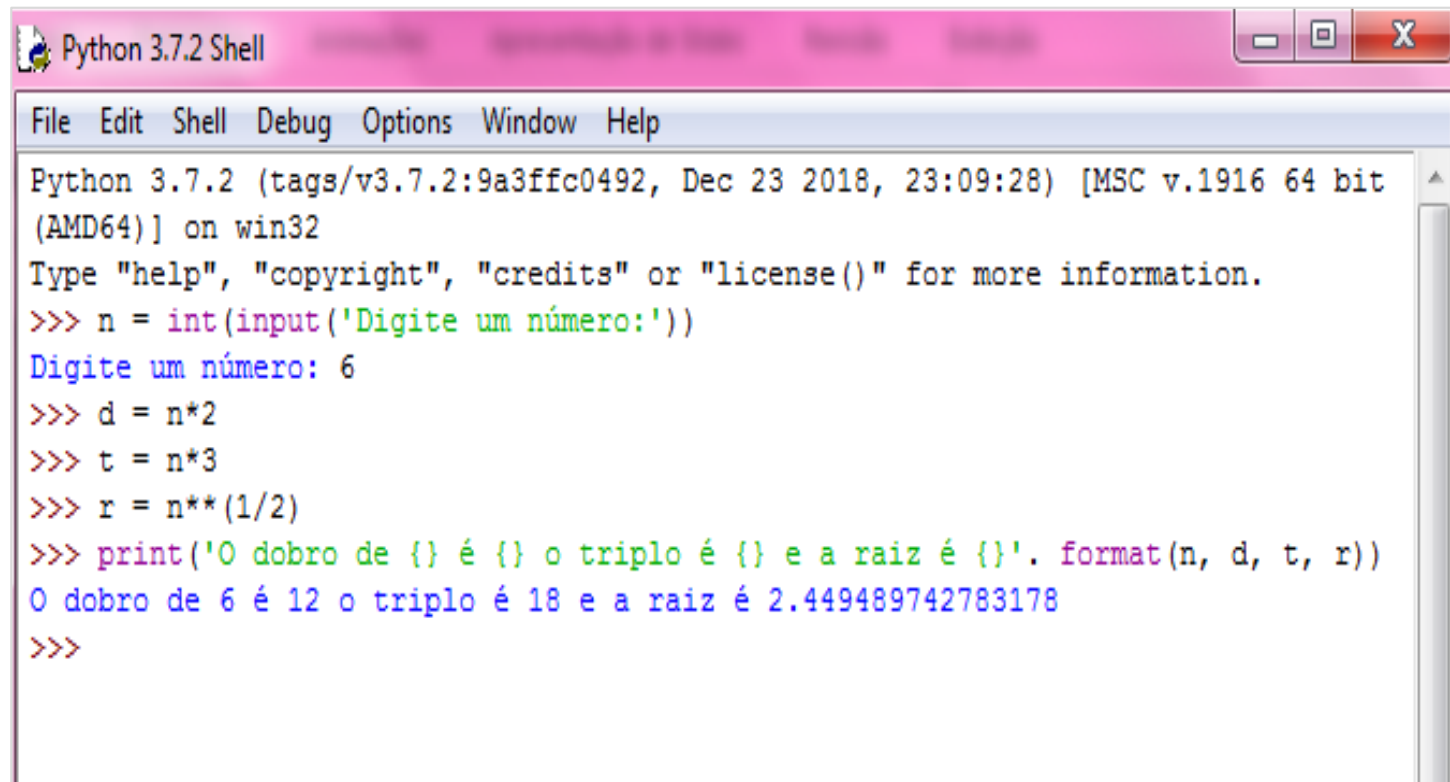
- Crie um programa que leia dois número e mostre a soma entre eles:



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> n1 = int(input('Digite um valor:'))
Digite um valor:5
>>> n2 = int(input('Digite outro valor:'))
Digite outro valor:6
>>> soma = n1 + n2
>>> print(' A soma entre {} e {} é igual a {}'.format(n1,n2,soma))
A soma entre 5 e 6 é igual a 11
>>> |
```

Funções Print e Input

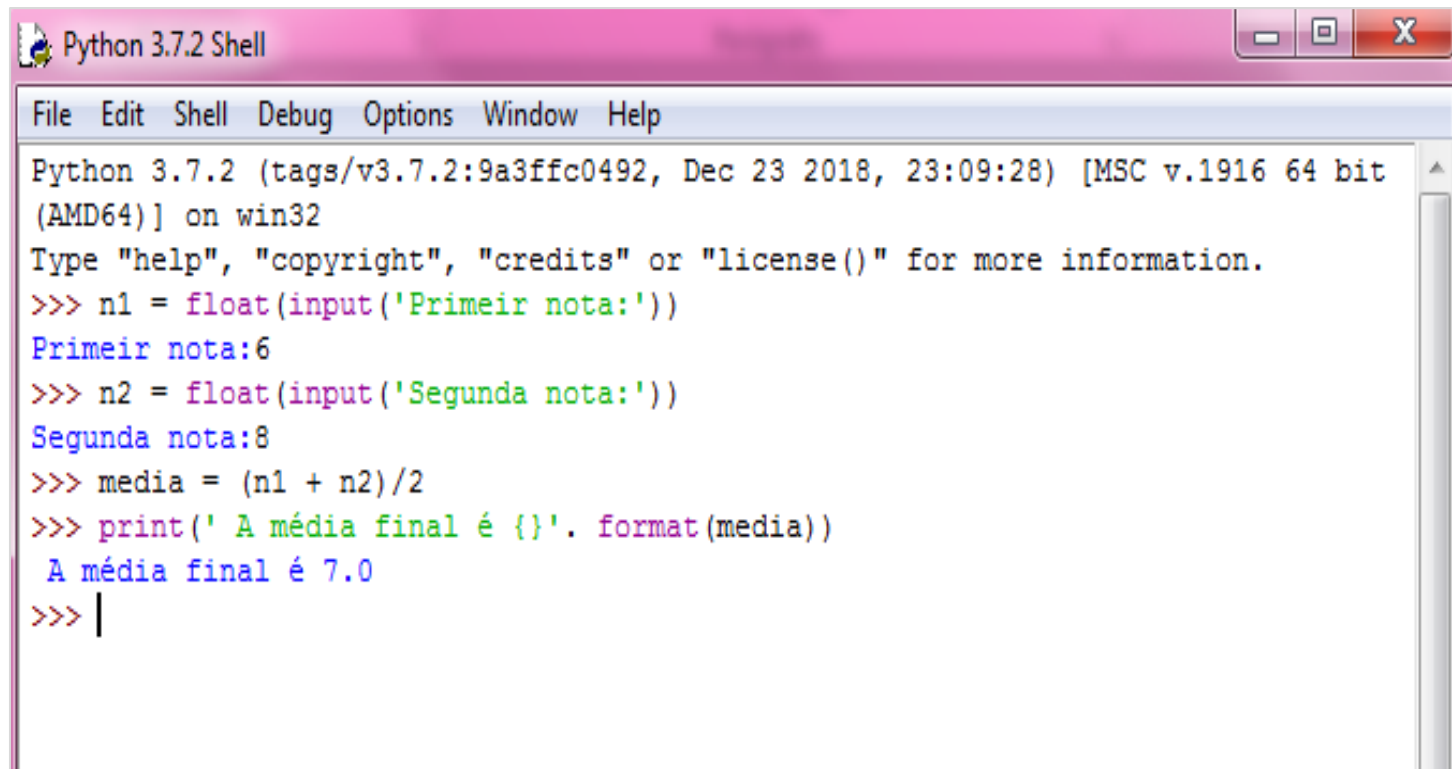
- Crie um algoritmo que leia um número e mostre o seu dobro, triplo e raiz quadrada:



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> n = int(input('Digite um número:'))
Digite um número: 6
>>> d = n*2
>>> t = n*3
>>> r = n**(1/2)
>>> print('O dobro de {} é {} o triplo é {} e a raiz é {}'.format(n, d, t, r))
O dobro de 6 é 12 o triplo é 18 e a raiz é 2.449489742783178
>>>
```

Funções Print e Input

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



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> n1 = float(input('Primeir nota:'))
Primeir nota:6
>>> n2 = float(input('Segunda nota:'))
Segunda nota:8
>>> media = (n1 + n2)/2
>>> print(' A média final é {}'.format(media))
A média final é 7.0
>>> |
```


Exercícios Práticos

- Exercício 1:
 - ✓ Faça um programa que peça um número e então mostre a mensagem “O número informado foi [número]”.



Exercícios Práticos

- Exercício 2:
 - ✓ Faça um programa que peça dois números e imprima a soma , subtração, multiplicação e divisão.



Exercícios Práticos

- Exercício 3:
 - ✓ Faça um programa que peça 4 notas e mostre a média.



Exercícios Práticos

- Exercício 4:
 - ✓ Faça um programa que leia um valor em metros e o exiba convertido em centímetros e milímetros.



Exercícios Práticos

- Exercício 5:
 - ✓ Faça um programa que leia um número inteiro qualquer e mostre na tela sua tabuada do 10.



Exercícios Práticos

- Exercício 6:
 - ✓ Faça um programa que leia quanto dinheiro uma pessoa tem na carteira e mostre quantos dólares ela pode comprar.
 - * Considere US\$1.00=R\$ 3.87.



Exercícios Práticos

- Exercício 7:
 - ✓ Faça um programa que leia a largura e a altura de uma parede em metros, calcule a sua área e a quantidade de tinta necessária para pintá-la, sabendo que cada litro de tinta pinta uma área de 2 metros quadrados.



Exercícios Práticos

- Exercício 8:
 - ✓ Faça um algoritmo que leia o preço de um produto e mostre seu novo preço, com 5% de desconto.



Exercícios Práticos

- Exercício 9:
 - ✓ Faça um algoritmo que leia o salário de um funcionário e mostre seu novo salário, com 15% de aumento.



Exercícios Práticos

- Exercício 10:
 - ✓ Faça um programa que converta uma temperatura digitando em graus Celsius e converta para graus Fahrenheit.
 - * $f = ((9 * c) / 5) + 32$.



Exercícios Práticos

- Exercício 11:
 - ✓ Faça um programa que pergunte a quantidade de Km percorridos por um carro alugado e a quantidade de dias pelos quais ele foi alugado. Calcule o preço a pagar, sabendo que o carro custa R\$60 por dia e R\$0,15 Km rodado.



Utilizando módulos

Biblioteca Math

- É comum dizer que o Python vem com “baterias inclusas”, em referência a vasta biblioteca de módulos e pacotes que é distribuída com o interpretador;
- O módulo **math** define funções logarítmicas, de exponenciação, trigonométricas, hiperbólicas e conversões angulares, entre outras.



Utilizando módulos

Biblioteca Math

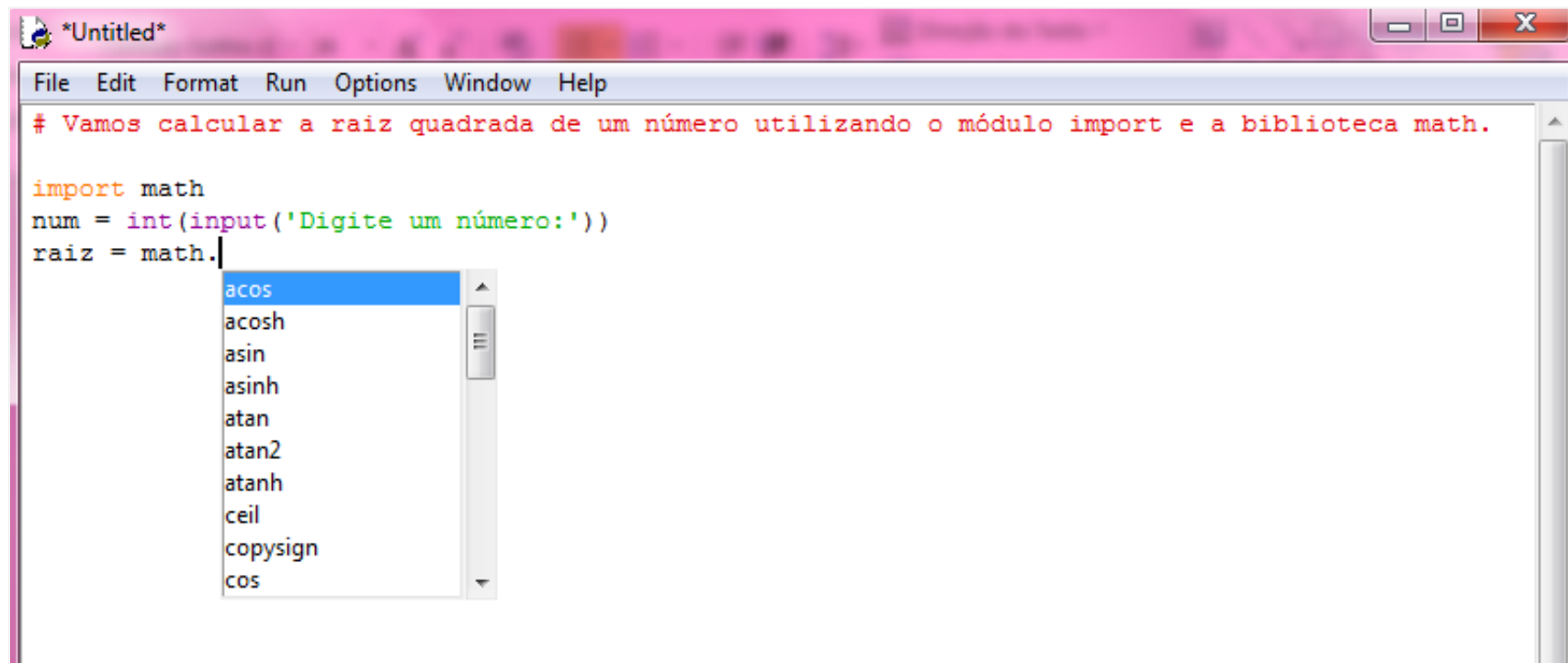
- A biblioteca math nos oferece várias funções descritas abaixo:
 - **math.sqrt(numero)**: Retorna a raiz quadrada do número.
 - **math.cos(numero)**: Retorna o cosseno do número em radiano.
 - **math.sin(numero)**: Retorna o seno do número em radiano.
 - **math.tan(numero)**: Retorna a tangente do número em radiano.
 - **math.radians(numero)**: Converte o angulo 'numero' de graus para radiano.
 - **math.pi**: Não é bem uma função, está mais para uma constante com o número pi (3.1415926535897931).
 - **math.hypot(x, y)**: Retorna a hipotenusa dos números (catetos) fornecidos.



Utilizando módulos

Biblioteca Math

- Note que quando digitamos “*math.*” as opções disponíveis da biblioteca será disponibilizada porque utilizamos o “*import math*”.



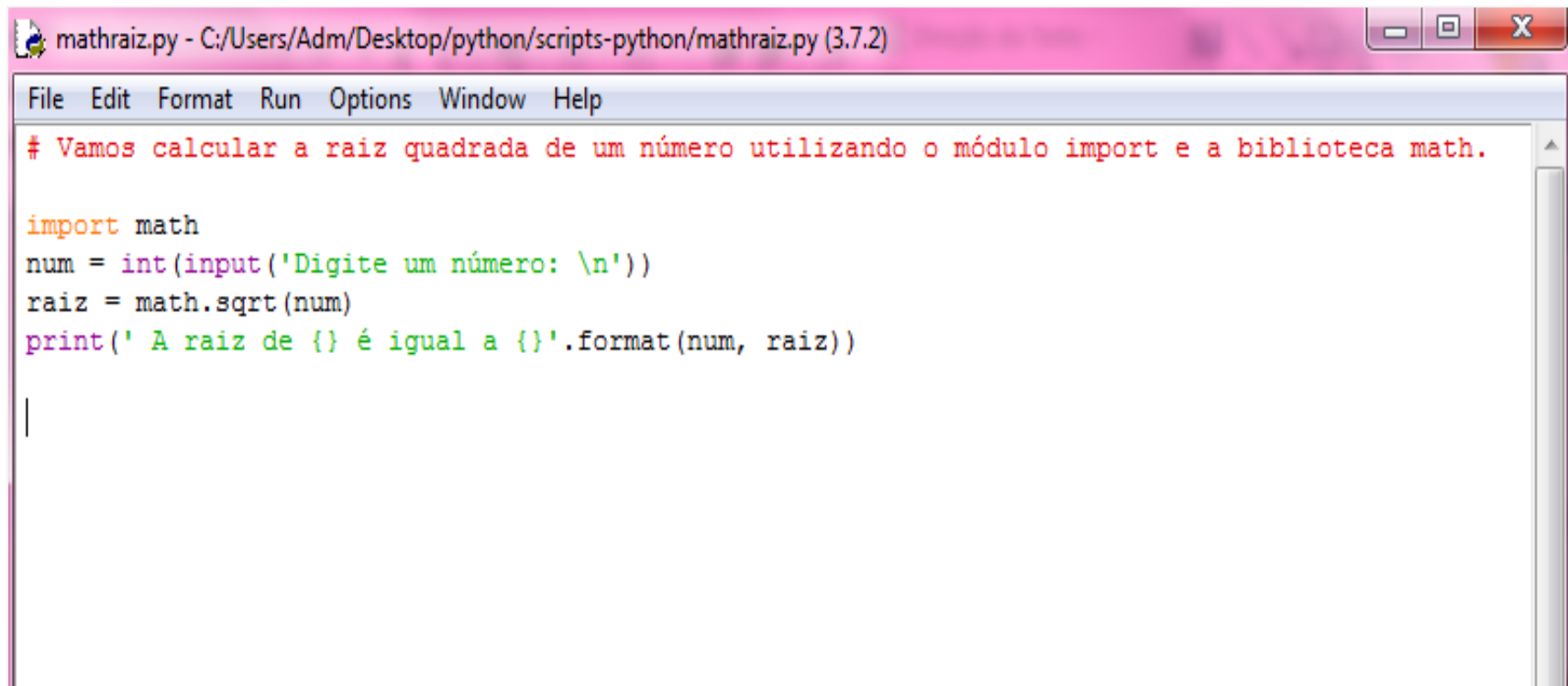
```
*Untitled*
File Edit Format Run Options Window Help
# Vamos calcular a raiz quadrada de um número utilizando o módulo import e a biblioteca math.

import math
num = int(input('Digite um número:'))
raiz = math. |
acos
acosh
asin
asinh
atan
atan2
atanh
ceil
copysign
cos
```

Utilizando módulos

Biblioteca Math

- Na biblioteca math para calcular a raiz de um número deve-se chamar a função “sqrt”.



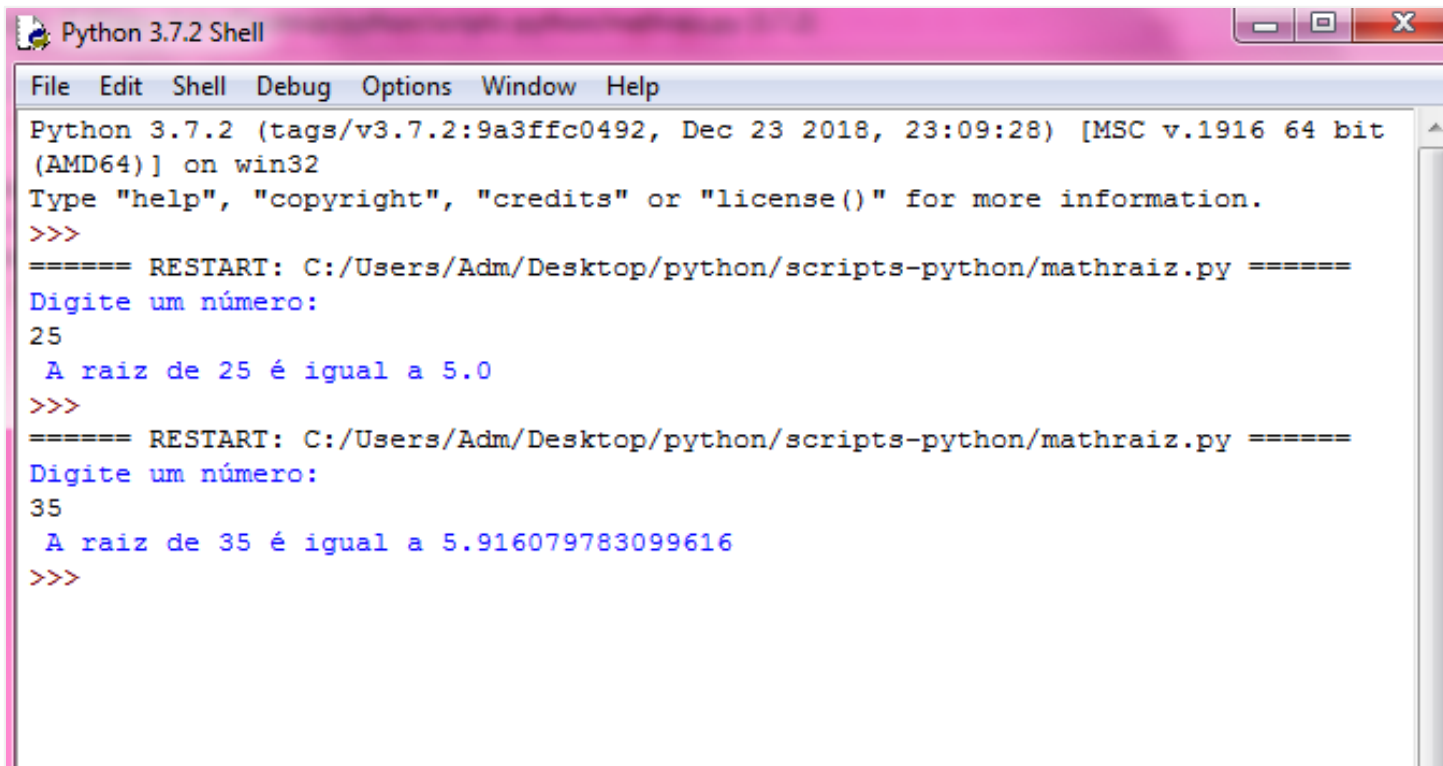
```
mathraiz.py - C:/Users/Adm/Desktop/python/scripts-python/mathraiz.py (3.7.2)
File Edit Format Run Options Window Help
# Vamos calcular a raiz quadrada de um número utilizando o módulo import e a biblioteca math.

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

Utilizando módulos

Biblioteca Math

- Assim o programa irá calcular a raiz do número digitado.

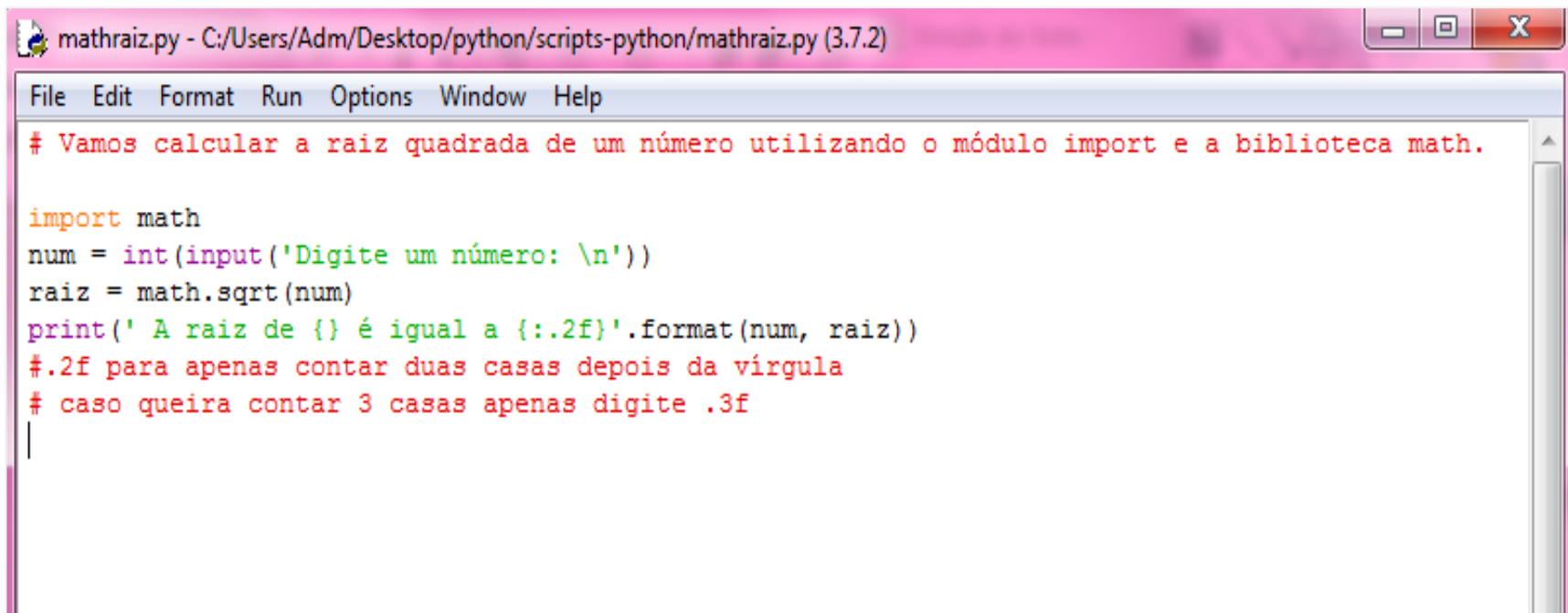


```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/mathraiz.py =====
Digite um número:
25
A raiz de 25 é igual a 5.0
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/mathraiz.py =====
Digite um número:
35
A raiz de 35 é igual a 5.916079783099616
>>>
```


Utilizando módulos

Biblioteca Math

- Utiliza-se o “.2f” para contar duas casas depois da vírgula em caso de resultados com muitas casas decimais.



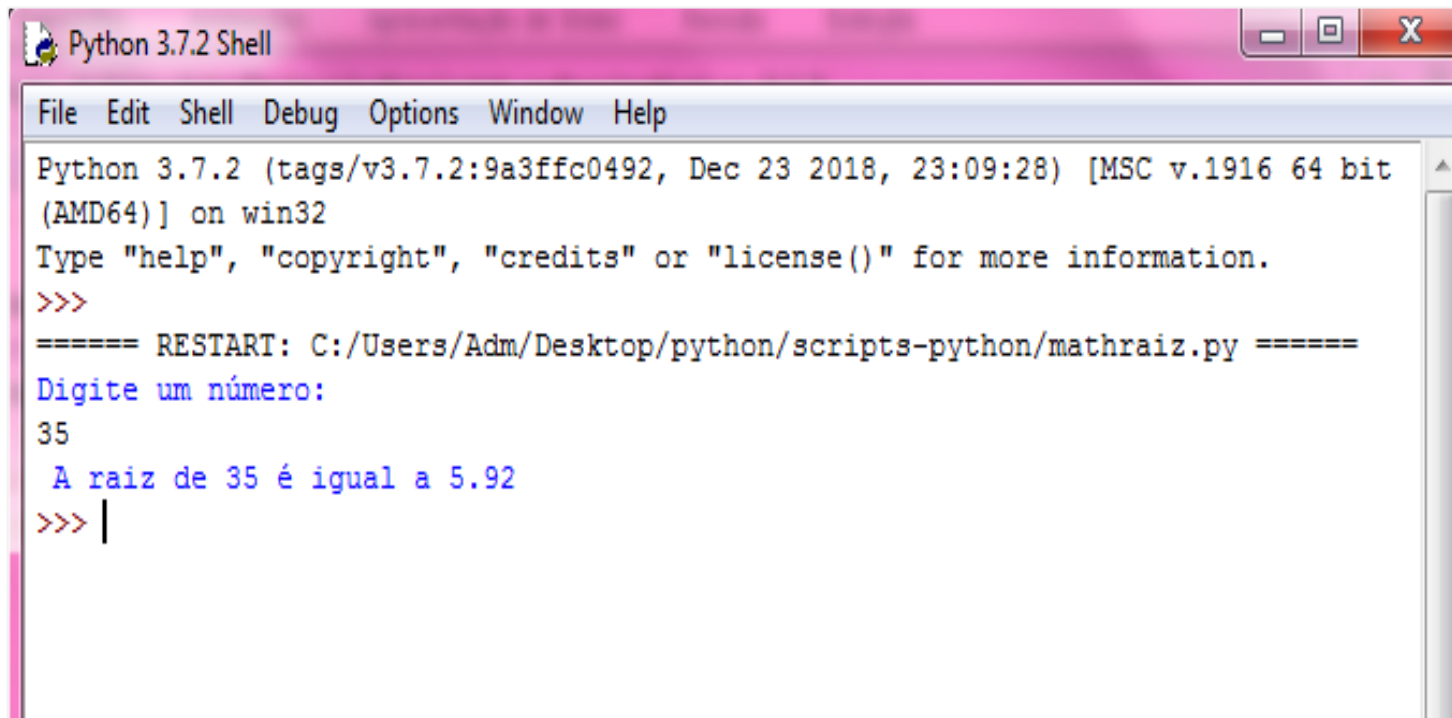
```
mathraiz.py - C:/Users/Adm/Desktop/python/scripts-python/mathraiz.py (3.7.2)
File Edit Format Run Options Window Help
# Vamos calcular a raiz quadrada de um número utilizando o módulo import e a biblioteca math.

import math
num = int(input('Digite um número: \n'))
raiz = math.sqrt(num)
print(' A raiz de {} é igual a {:.2f}'.format(num, raiz))
# .2f para apenas contar duas casas depois da vírgula
# caso queira contar 3 casas apenas digite .3f
|
```

Utilizando módulos

Biblioteca Math

- O resultado como saída irá executar com duas casas decimais após a vírgula.



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/mathraiz.py =====
Digite um número:
35
  A raiz de 35 é igual a 5.92
>>> |
```

Utilizando módulos

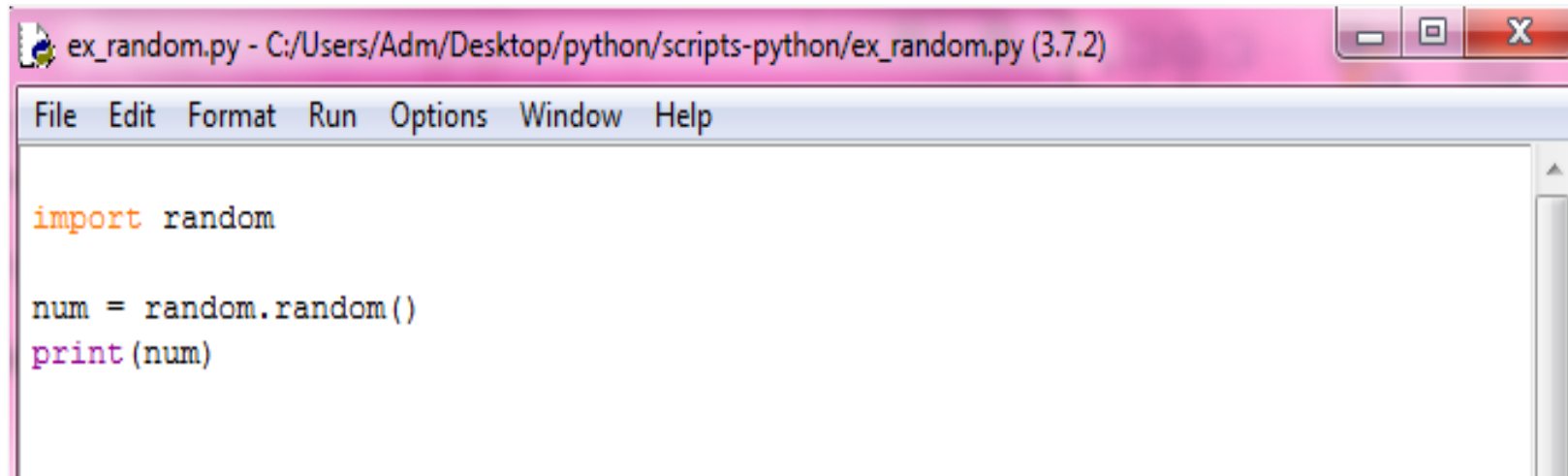
Biblioteca Random

- Ao lançarmos uma moeda ou um dado, nunca temos certeza do resultado final. Essa imprevisibilidade tem muitas aplicações como na escolha dos vencedores em sorteios ou geração de casos de teste para experimentos com valores aleatórios com base em algoritmos. Tendo essa utilidade em mente, Python provê-nos o módulo random;
- O módulo **random** traz funções para a geração de números aleatórios.



Utilizando módulos

Biblioteca Random



```
ex_random.py - C:/Users/Adm/Desktop/python/scripts-python/ex_random.py (3.7.2)
File Edit Format Run Options Window Help

import random

num = random.random()
print(num)
```

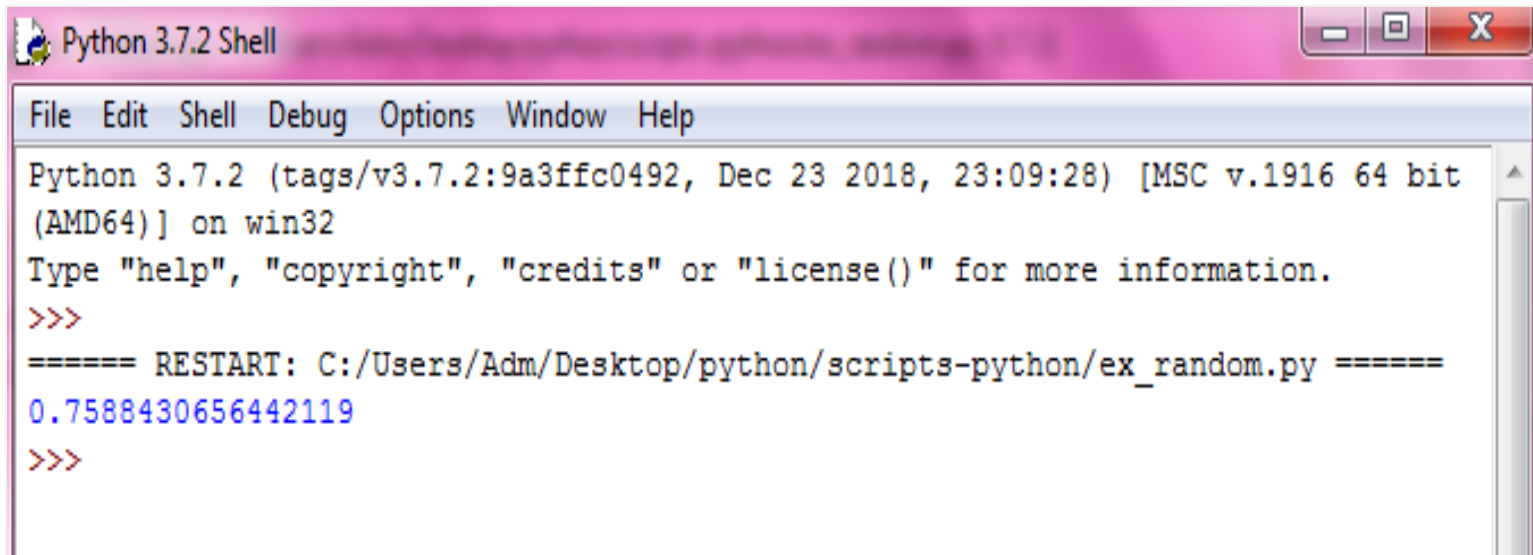


O método Random da classe Random, gera um número aleatório entre 0 e 1 (números reais)



Utilizando módulos

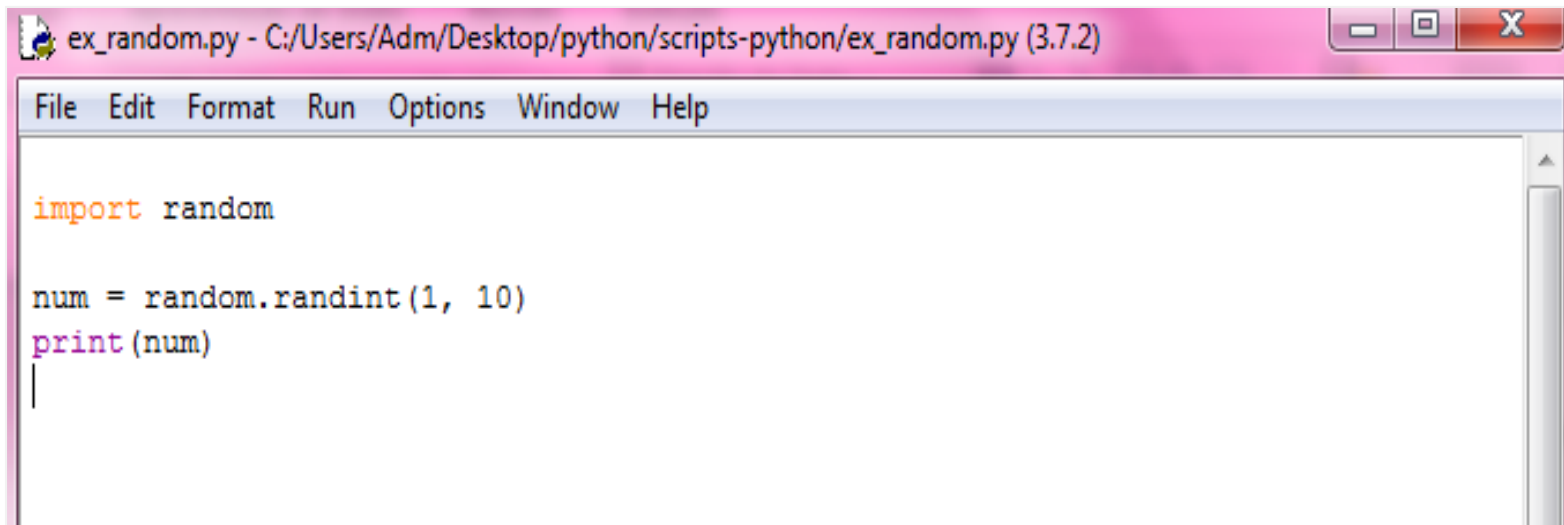
Biblioteca Random



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/ex_random.py =====
0.7588430656442119
>>>
```

Utilizando módulos

Biblioteca Random



```
ex_random.py - C:/Users/Adm/Desktop/python/scripts-python/ex_random.py (3.7.2)
File Edit Format Run Options Window Help

import random

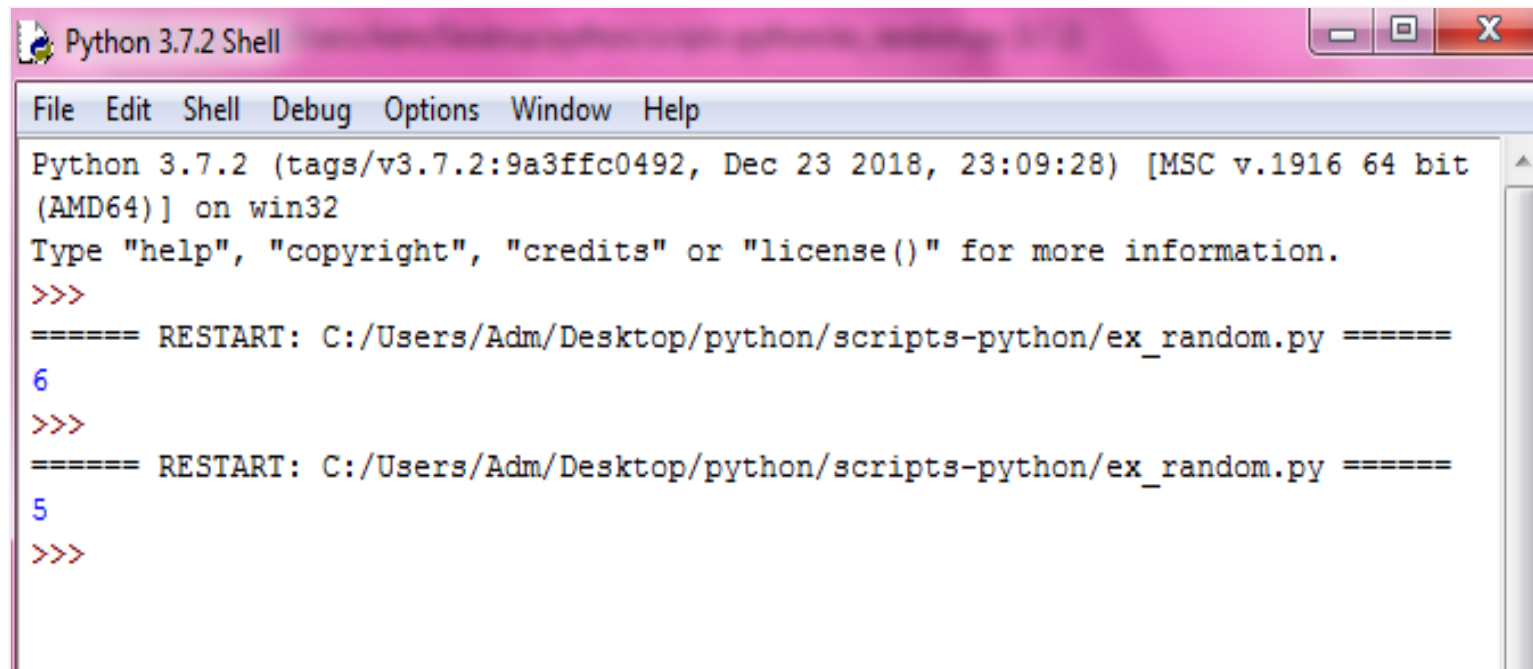
num = random.randint(1, 10)
print(num)
```

➔ O método Random da classe Randint, gera um número aleatório num intervalo entre 2 números específicos(números inteiros)



Utilizando módulos

Biblioteca Random



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/ex_random.py =====
6
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/ex_random.py =====
5
>>>
```



Utilizando módulos

Biblioteca Random

- Existe outras classes que podemos trabalhar dentro da biblioteca Random como:

Classe	Objetivo
<code>random.seed()</code>	Gera a sequência de números criados uma única vez
<code>random.randrange()</code>	Gera números inteiros menores que uma certa variável
<code>random.choice()</code>	Utilizado para selecionar um elemento aleatório de uma sequência não vazia
<code>random.shuffle()</code>	Utilizado para misturar uma certa sequência de números



Exercícios Práticos

- Exercício 12:
 - ✓ Faça um programa que leia um número Real qualquer pelo teclado e mostre na tela a sua porção inteira.



Exercícios Práticos

- Exercício 13:
 - ✓ Faça um programa que leia o comprimento do cateto oposto e do cateto adjacente de um triângulo retângulo. Calcule e mostre o comprimento da hipotenusa.



Exercícios Práticos

- Exercício 14:
 - ✓ Faça um programa que leia um ângulo qualquer e mostre na tela o valor de seno, cosseno e tangente desse ângulo.



Exercícios Práticos

- Exercício 15:
 - ✓ Um professor quer sortear um dos seus quatro alunos para apagar o quadro. Faça um programa que ajude ele, lendo o nome dos alunos e escrevendo na tela o nome do escolhido.



Exercícios Práticos

- Exercício 16:
 - ✓ O mesmo professor do desafio 019 quer sortear a ordem de apresentação de trabalhos dos alunos. Faça um programa que leia o nome dos quatro alunos e mostre a ordem sorteada.

