

Strings

- As **strings** no Python são **buitins** para armazenar texto;
- Como são imutáveis, não é possível adicionar, remover ou mesmo modificar algum caractere de uma **string**. Para realizar essas operações, o Python precisa criar um nova **string**.

Strings

- Para melhor entender o que é uma **string**, vamos pegar uma frase;
- Com esta frase vamos jogar dentro de uma variável do tipo string, lembre-se uma string sempre estará declarada entres “ ou ”.

```
frase = ' Aprendendo string '
```

Strings

- Quando fazemos este tipo de atribuição, o python coloca esta frase na memória do computador;
- Só que a frase não irá inteira, o que o python faz é criar micros espaços dentro da memória do computador;
- Ou seja, cada micros espaço destes, o python irá alocar uma letra.

Strings

A	p	r	e	n	d	e	n	d	o		s	t	r	i	n	g
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

- Cada um destes micro espaços recebe um índice, representado por um número sequencial iniciado pelo 0 até o número de letras contidas na string;
- Neste caso, então temos 17 caracteres alocados na memória.

Strings

Fatiamento de String

- A técnica de fatiamento é uma operação que consiste em pegar pedaços de uma string;
- Ex.: Vamos escrever **frase[9]**, o símbolo de colchetes é um identificador de uma estrutura de dados do python chamada lista¹.

A	p	r	e	n	d	e	n	d	o		s	t	r	i	n	g
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16



O número 9 é 'o' (minúsculo)

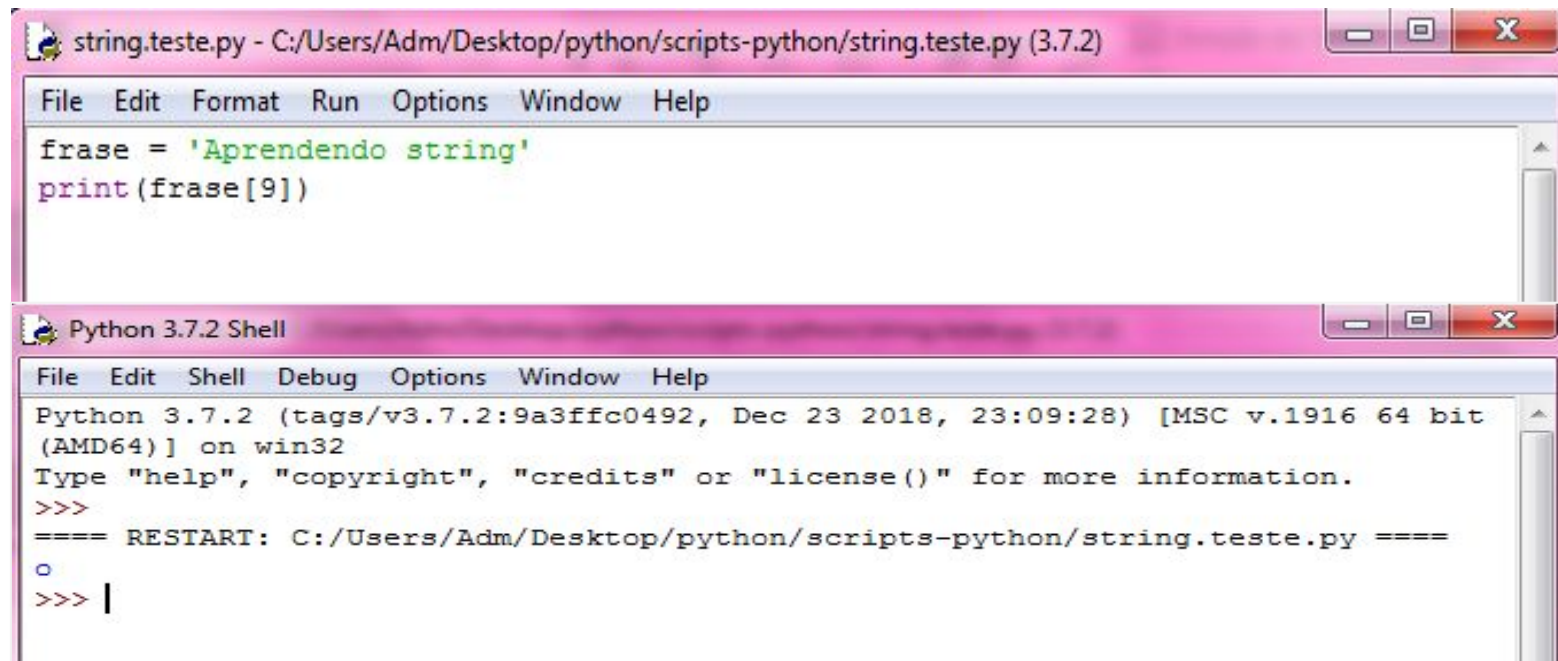


Lembre-se o Python diferencia letras maiúsculas de minúsculas

1. Lista: Conteúdo estudado em tópicos seguintes.

Strings

Fatiamento de String



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

```
frase = 'Aprendendo string'
print(frase[9])
```

The bottom window, titled 'Python 3.7.2 Shell', shows the output of the script. It displays the Python version and architecture, followed by a restart message and a prompt for user input.

```
Python 3.7.2 (tags/v3.7.2:9a3fffc0492, 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/string.teste.py ====
>>> |
```

Strings

Fatiamento de String

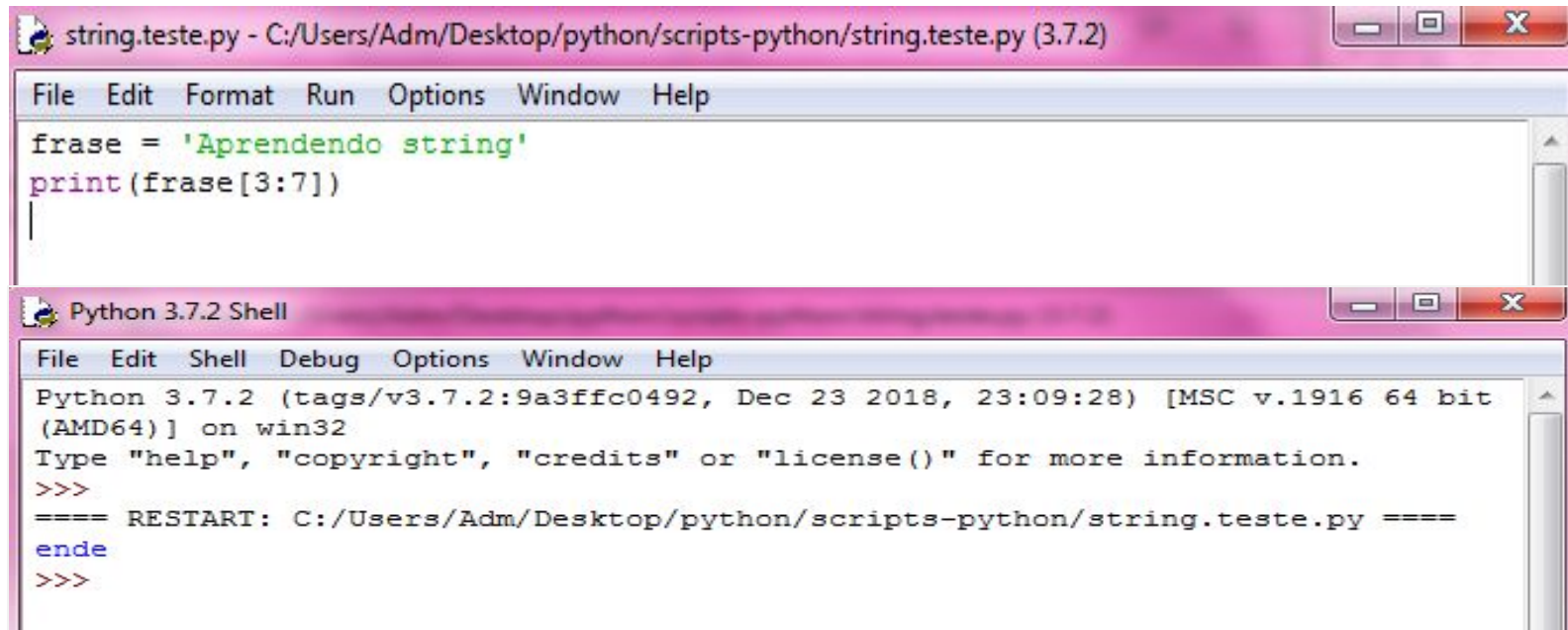
- Outro tipo de fatiamento é `frase[3:7]`

A	p	r	e	n	d	e	n	d	o		s	t	r	i	n	g
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

- Neste caso, o fatiamento seria do número 3 ao 7, mas em Python ele irá considerar do número 3 ao 6;
- Caso queira do 3 ao 7 terá que inserir `frase[3:8]`.

Strings

Fatiamento de String



The screenshot displays two windows from a Python IDE. The top window, titled 'string.teste.py - C:/Users/Adm/Desktop/python/scripts-python/string.teste.py (3.7.2)', contains the following Python code:

```
frase = 'Aprendendo string'
print(frase[3:7])
```

The bottom window, titled 'Python 3.7.2 Shell', shows the execution 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/string.teste.py ====
ende
>>>
```

Strings

Fatiamento de String

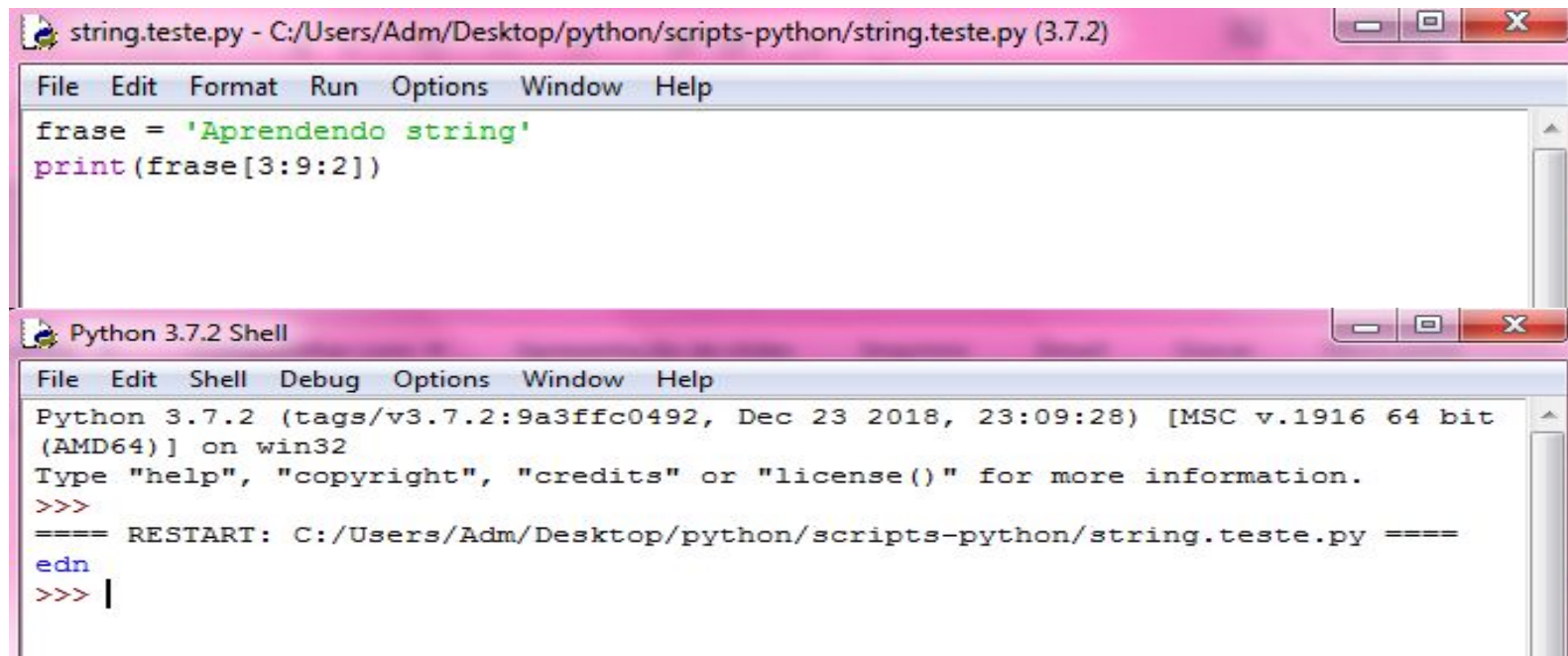
- Outro tipo de fatiamento é `frase[3:9:2]`

A	p	r	e	n	d	e	n	d	o		s	t	r	i	n	g
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

- Neste caso, o fatiamento seria do número 3 ao 9, mas ele terá que pular 2 no processo;
- Ou seja, o fatiamento começa no número **3(e)** e pula 2 casas **5(d)** e mais 2 casas **7(n)**.

Strings

Fatiamento de String



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

```
File Edit Format Run Options Window Help
frase = 'Aprendendo string'
print(frase[3:9:2])
```

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

```
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/string.teste.py ====
edn
>>> |
```

Strings

Fatiamento de String

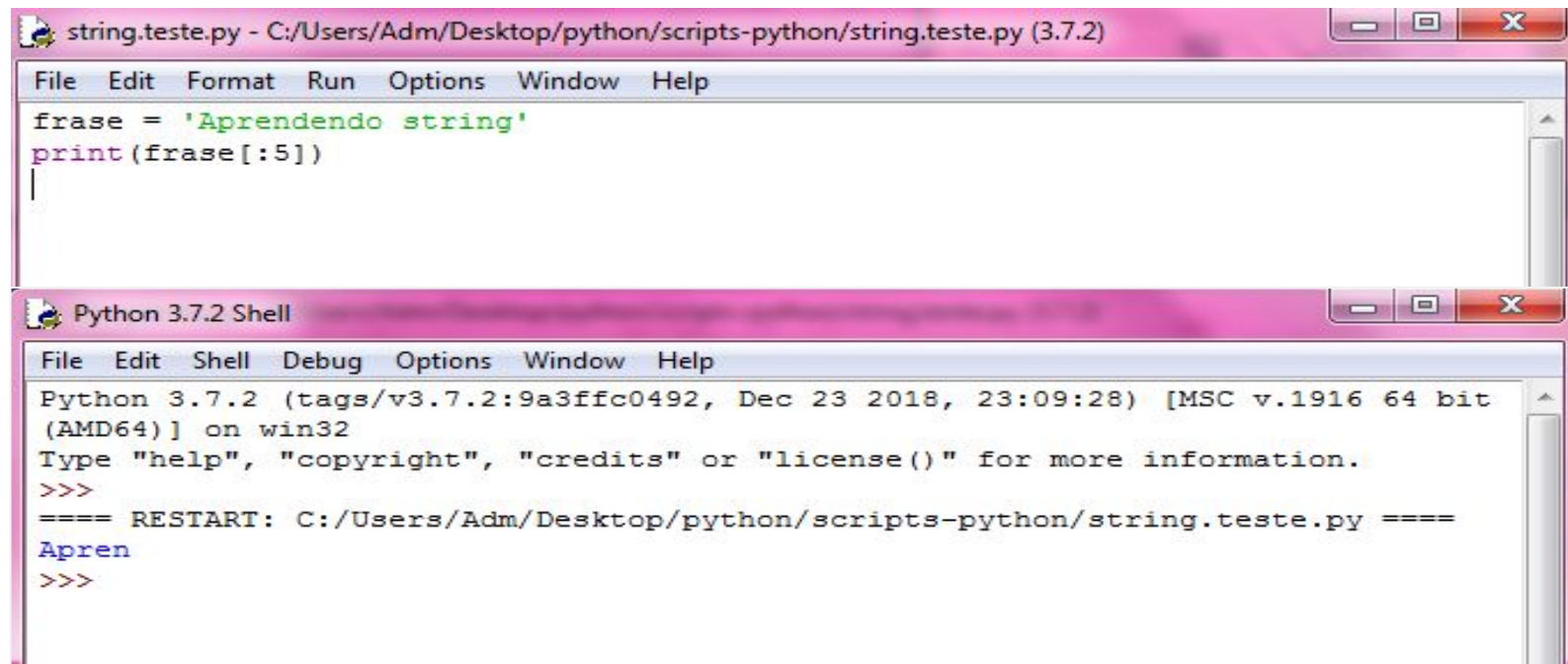
- Outro tipo de fatiamento é **frase[:5]**

A	p	r	e	n	d	e	n	d	o		s	t	r	i	n	g
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

- Neste caso, o fatiamento irá começar no número 0 pois não está indicando o seu início e irá até a número 5;
- Neste caso, até o número 4 como vimos anteriormente.

Strings

Fatiamento de String



The screenshot displays two windows from a Python IDE. The top window, titled 'string.teste.py - C:/Users/Adm/Desktop/python/scripts-python/string.teste.py (3.7.2)', contains the following Python code:

```
frase = 'Aprendendo string'
print(frase[:5])
```

The bottom window, titled 'Python 3.7.2 Shell', shows the execution 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/string.teste.py ====
Apren
>>>
```

Strings

Fatiamento de String

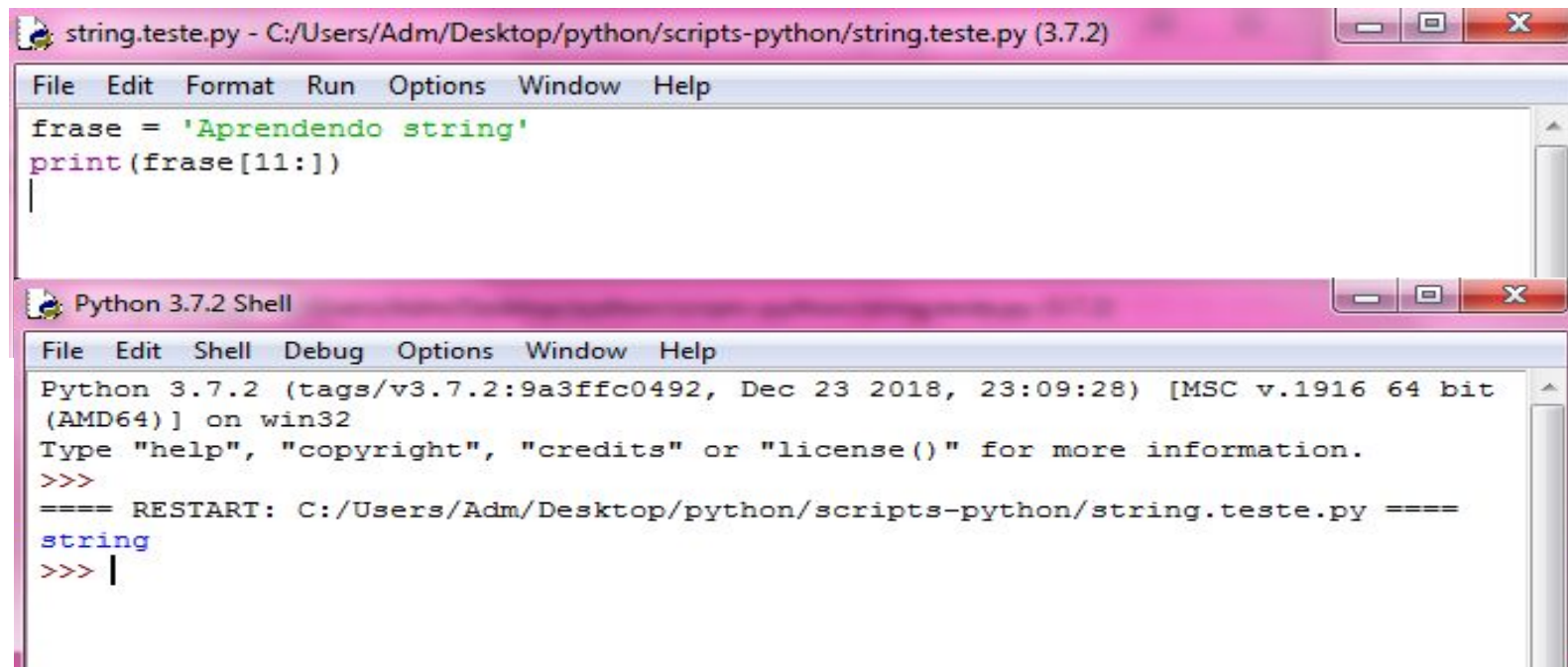
- Outro tipo de fatiamento é **frase[11:]**

A	p	r	e	n	d	e	n	d	o		s	t	r	i	n	g
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

- Neste caso, o fatiamento irá começar do número 11 até o final da frase;
- Neste caso, até o número 16.

Strings

Fatiamento de String



The screenshot displays two windows from a Python IDE. The top window, titled 'string.teste.py - C:/Users/Adm/Desktop/python/scripts-python/string.teste.py (3.7.2)', contains the following Python code:

```
frase = 'Aprendendo string'
print(frase[11:])
```

The bottom window, titled 'Python 3.7.2 Shell', shows the output of running the script. It includes the Python version information and the execution of the script, which prints the substring 'ndendo string'.

```
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/string.teste.py ====
string
>>> |
```

Strings

Fatiamiento de String

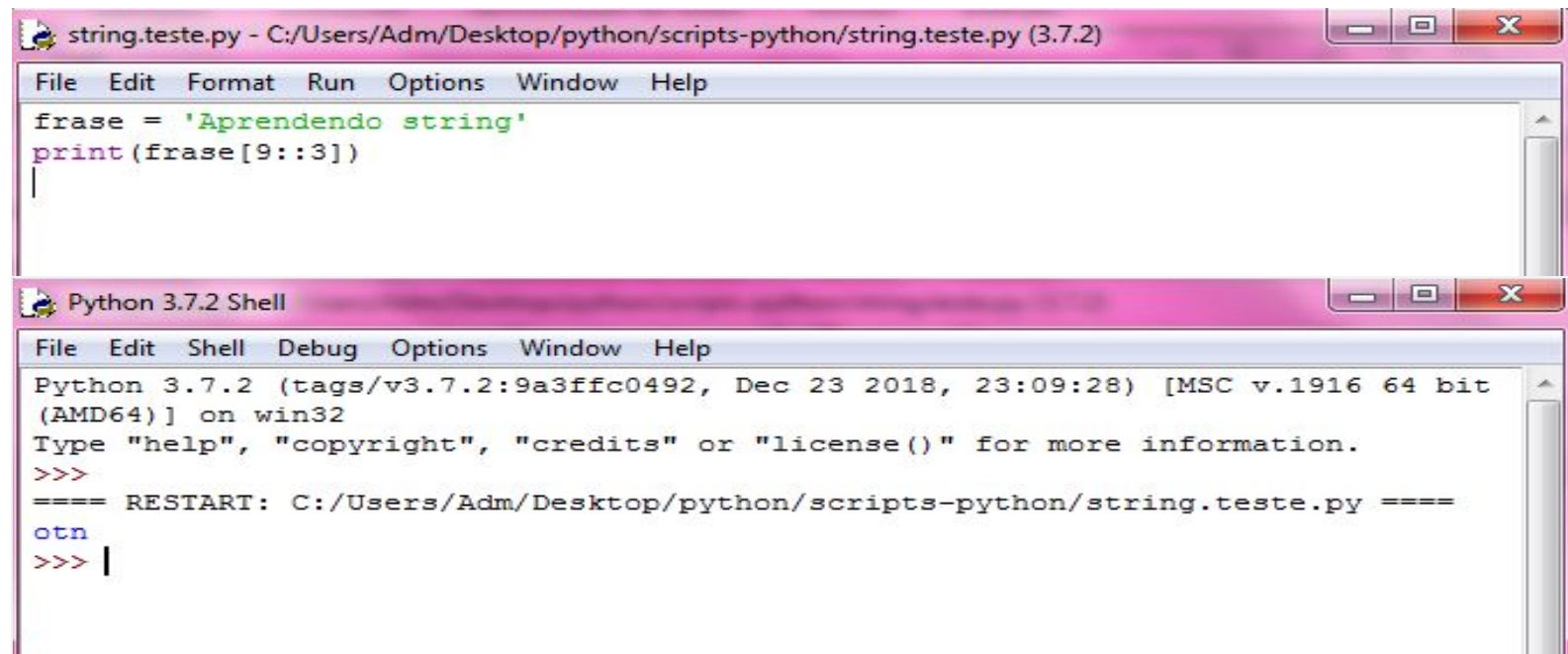
- Outro tipo de fatiamento é `frase[9::3]`

A	p	r	e	n	d	e	n	d	o	s	t	r	i	n	g
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

- Neste caso, o fatiamento irá começar do número 9 e ir até o final da frase, pois não temos nenhum número indicado entre os dois pontos (::);
- Porém, ele está pedindo para pular 3 casas, assim começando em **9(o)**, **12(t)** e parando em **15(n)**.

Strings

Fatiamento de String



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

```
File Edit Format Run Options Window Help
frase = 'Aprendendo string'
print(frase[9::3])
|
```

The bottom window, titled 'Python 3.7.2 Shell', shows the output 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/string.teste.py ====
otn
>>> |
```

Strings

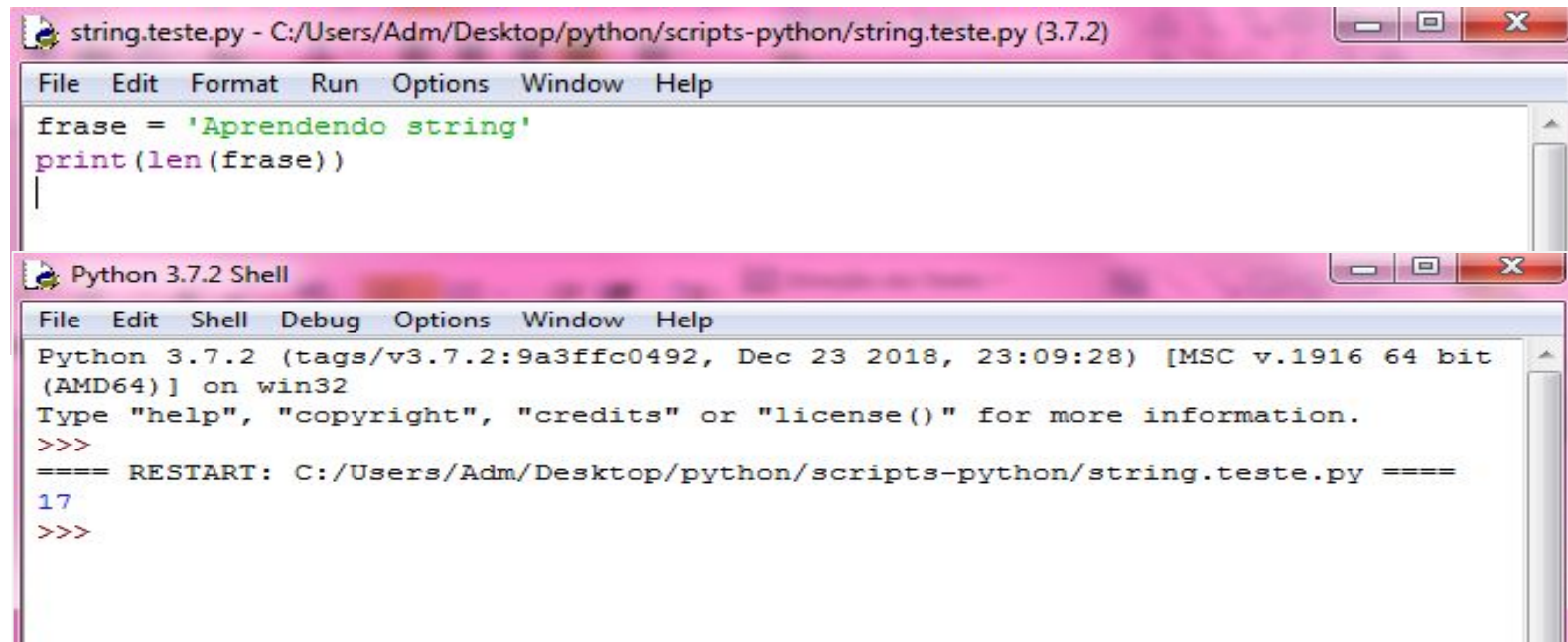
Análise em String

- **Len()** é uma função de análise dentro de uma string e tem objetivo de mostrar o comprimento da frase;
- Se escrevemos **len(frase)**, esta função irá contar quantos caracteres existem nesta frase, no nosso caso **17 caracteres**.

A	p	r	e	n	d	e	n	d	o		s	t	r	i	n	g
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Strings

Análise em String



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

```
frase = 'Aprendendo string'
print(len(frase))
```

The bottom window, titled 'Python 3.7.2 Shell', shows the execution 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/string.teste.py ====
17
>>>
```

Strings

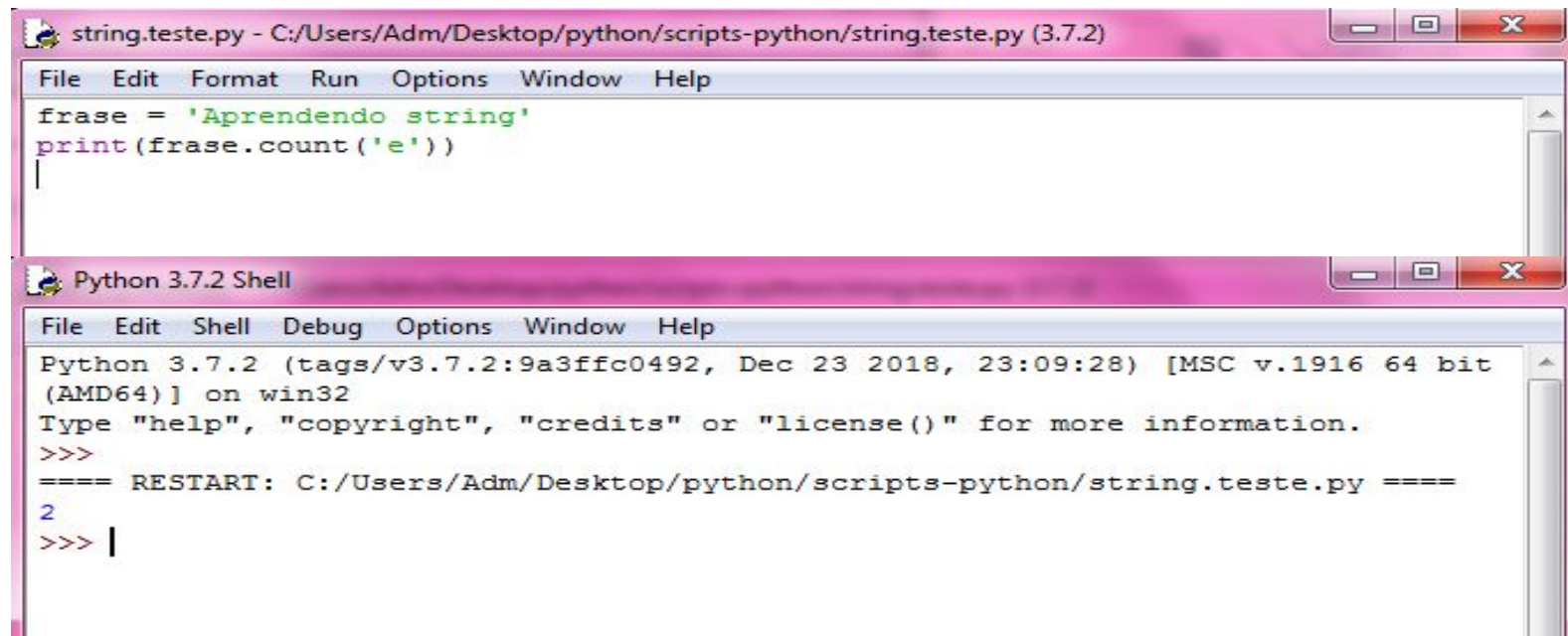
Análise em String

- **Count()** é uma função de análise dentro de uma string e tem objetivo de contar quantas vezes certa variável está contida na string.
- Se escrevemos **frase.count('e')**, esta função irá contar quantos vezes a letra 'e' aparecerá na string, neste caso 2 vezes.

A	p	r	e	n	d	e	n	d	o		s	t	r	i	n	g
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Strings

Análise em String



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

```
frase = 'Aprendendo string'
print(frase.count('e'))
|
```

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

```
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/string.teste.py ====
2
>>> |
```

Strings

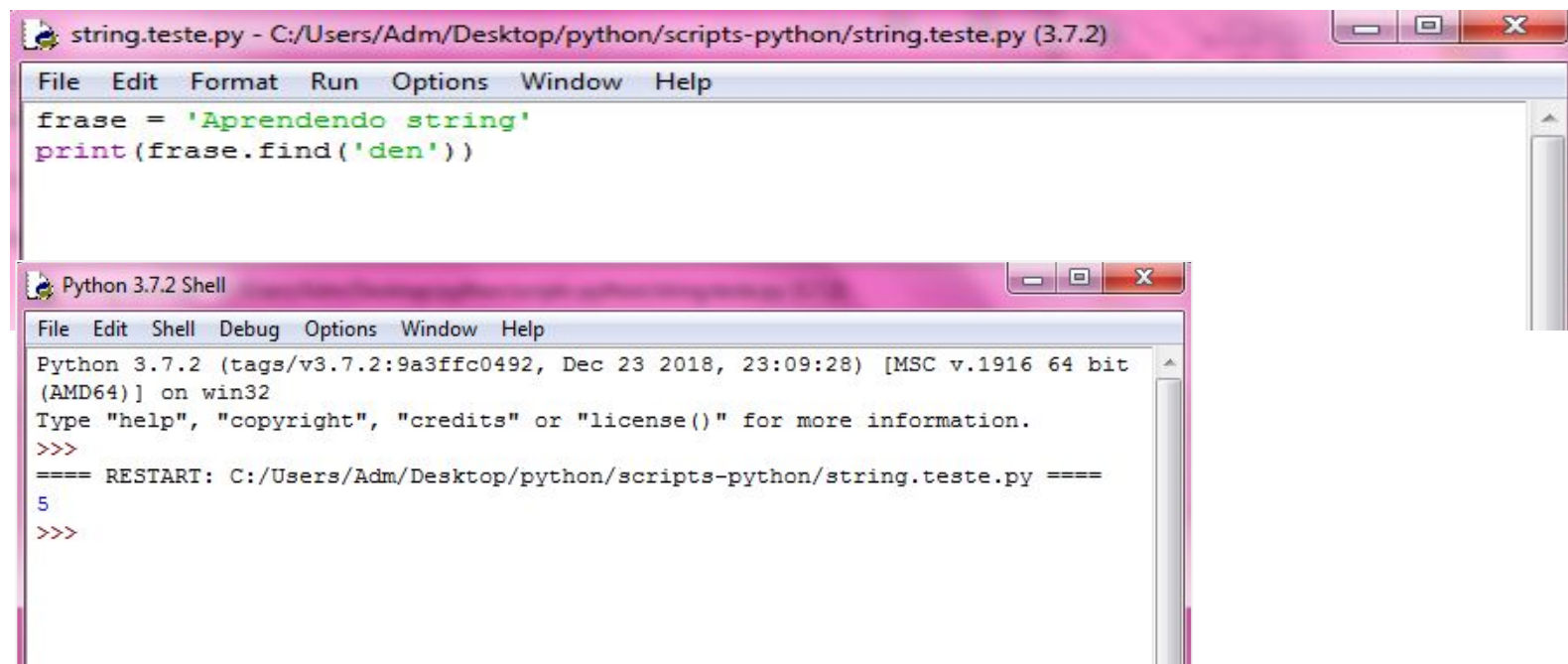
Análise em **String**

- **Find()** é uma função de análise dentro de uma string é tem objetivo de contar quantas vezes ele encontrou certa variável na string.
- Se escrevemos **frase.find('den')**, esta função irá te dizer em que momento começou o 'den', neste caso na posição 5;

A	p	r	e	n	d	e	n	d	o		s	t	r	i	n	g
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Strings

Análise em String



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

```
frase = 'Aprendendo string'
print(frase.find('den'))
```

The bottom window, titled 'Python 3.7.2 Shell', shows the execution 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/string.teste.py ====
5
>>>
```

Strings

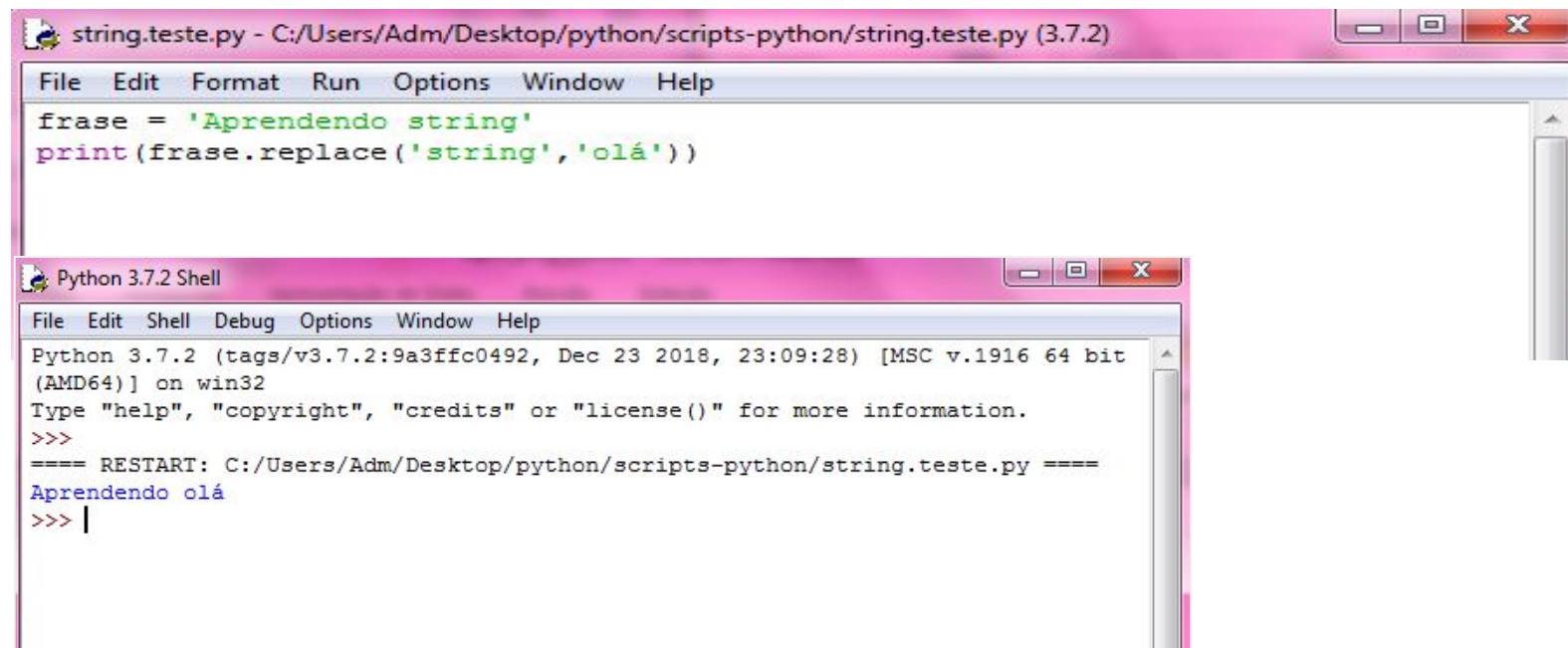
Transformação em **String**

- **Replace()** é uma função de transformação dentro de uma string e tem objetivo de trocar ou reposicionar certa variável;
- Se escrevemos **frase.replace('string','olá')**, esta função irá trocar 'string' por 'olá'.

A	p	r	e	n	d	e	n	d	o		o	l	á
0	1	2	3	4	5	6	7	8	9	10	11	12	13

Strings

Transformação em String



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

```
frase = 'Aprendendo string'
print(frase.replace('string', 'olá'))
```

The bottom window, titled 'Python 3.7.2 Shell', shows the execution 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/string.teste.py ====
Aprendendo olá
>>> |
```

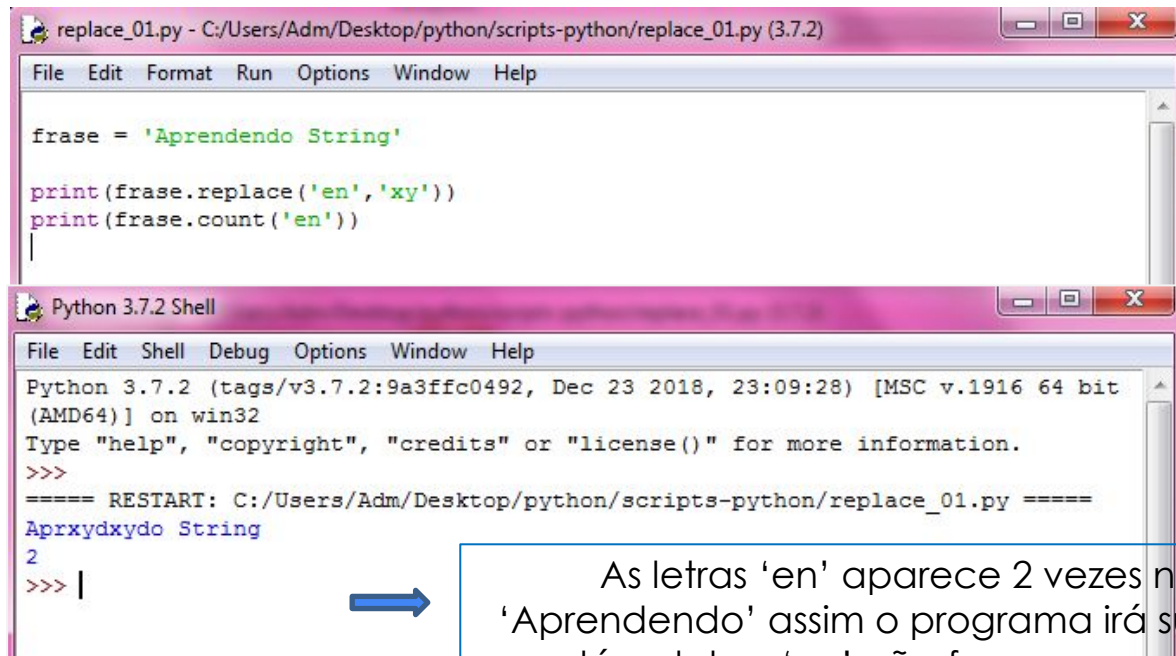
Strings

Transformação em **String**

- Continuando com o comando replace, vamos entender a seguinte situação:
- A partir da frase 'Aprendendo String' vamos substituir as letras 'en' por 'xy' e mostrar a quantidade de vezes que as letras 'en' aparece.

Strings

Transformação em String



The image shows a screenshot of a Python 3.7.2 IDE. The top window, titled 'replace_01.py - C:/Users/Adm/Desktop/python/scripts-python/replace_01.py (3.7.2)', contains the following code:

```
frase = 'Aprendendo String'

print(frase.replace('en', 'xy'))
print(frase.count('en'))
```

The bottom window, titled 'Python 3.7.2 Shell', shows the execution 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/replace_01.py =====
Aprxydxydo String
2
>>> |
```

A blue arrow points from the output '2' in the shell to the explanatory text box on the right.

As letras 'en' aparece 2 vezes na palavra 'Aprendendo' assim o programa irá substituir por 'xy' até as letras 'en' não forem encontradas mais.

Strings

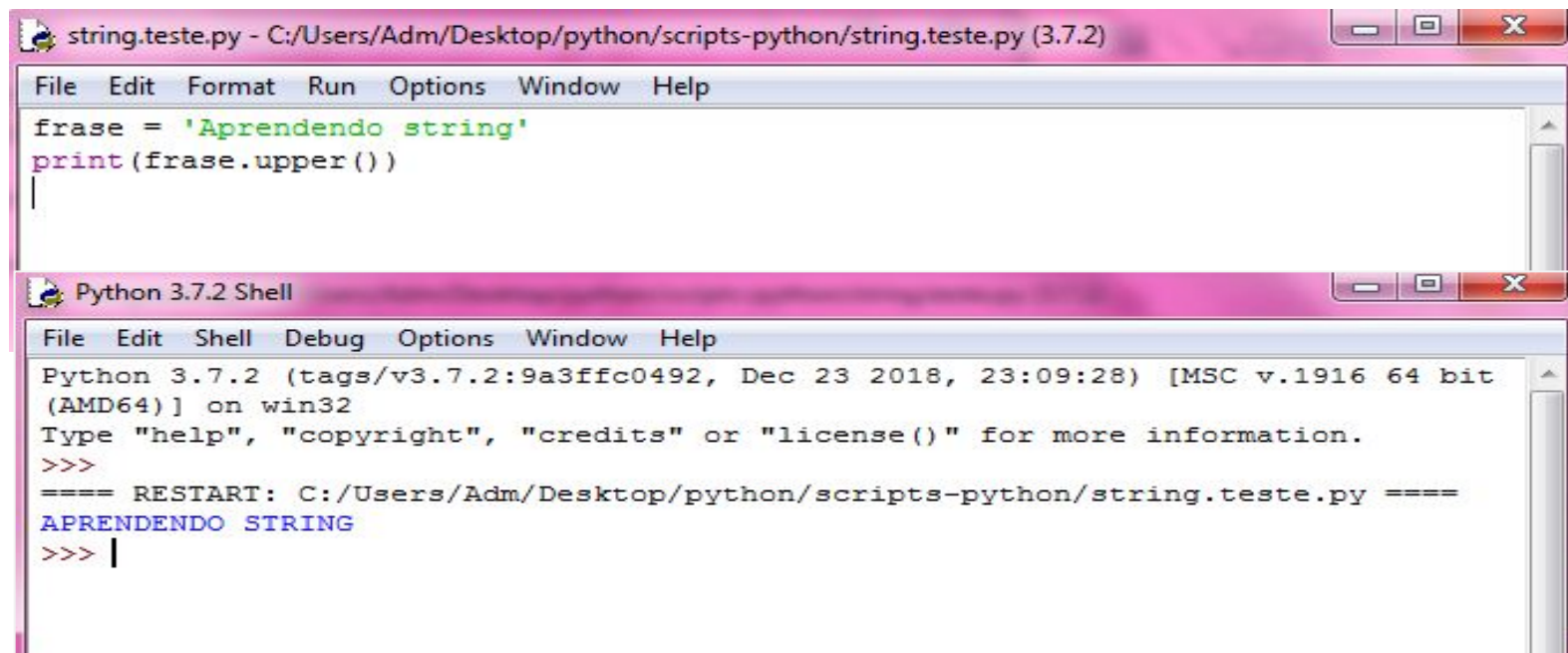
Transformação em **String**

- **Upper()** é uma função de transformação dentro de uma string e tem objetivo de trocar caracteres minúsculas por maiúsculas;
- Se escrevemos **frase.upper()**, o que já for maiúsculo ele mantém, o que for minúsculo ele irá transformar para maiúsculo.

A	P	R	E	N	D	E	N	D	O		S	T	R	I	N	G
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Strings

Transformação em String



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

```
frase = 'Aprendendo string'
print(frase.upper())
```

The bottom window, titled 'Python 3.7.2 Shell', shows the output of the script. It displays the Python version and environment information, followed by a restart message and the output of the script:

```
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/string.teste.py ====
APRENDENDO STRING
>>> |
```

Strings

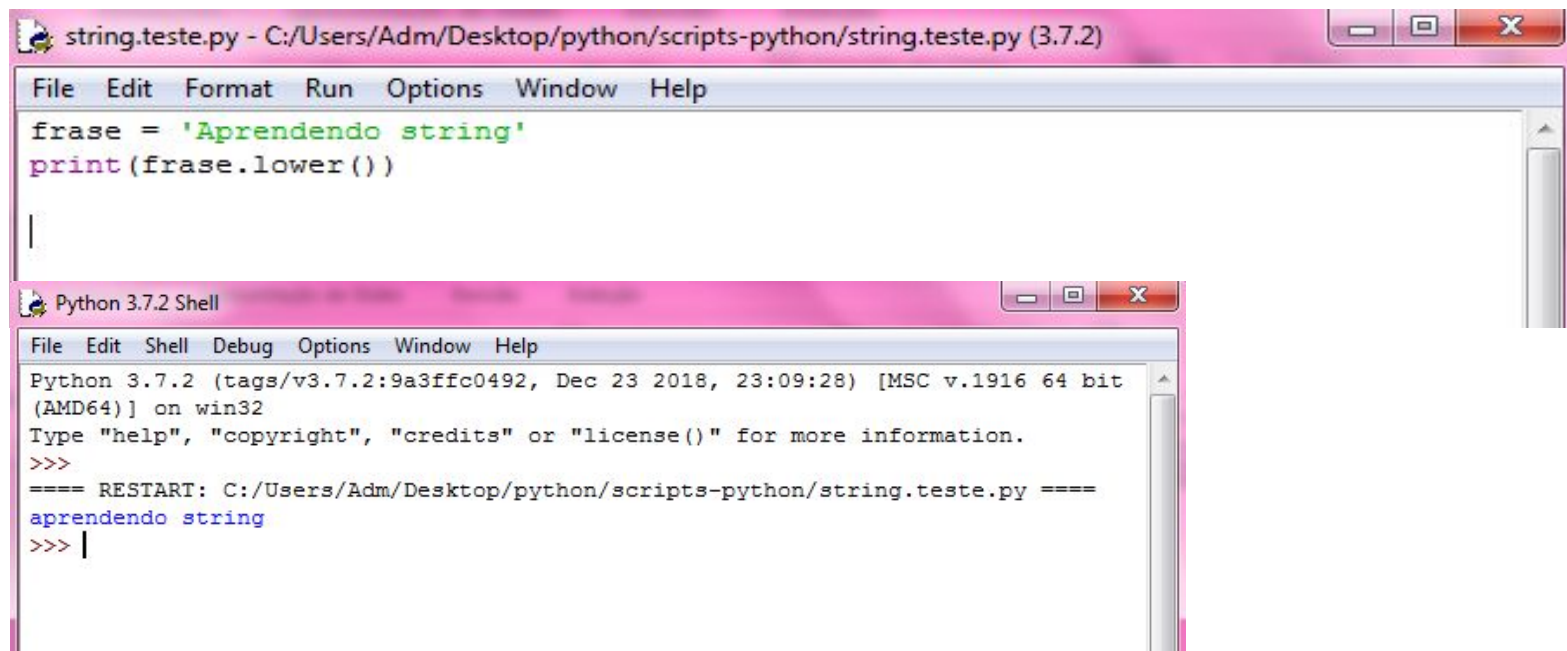
Transformação em **String**

- **Lower()** é uma função de transformação dentro de uma string e tem objetivo de trocar caracteres maiúsculas por minúsculas;
- Se escrevemos **frase.lower()**, o que já for minúsculo ele mantém, o que for maiúsculo ele irá transformar para minúsculo.

a	p	r	e	n	d	e	n	d	o		s	t	r	i	n	g
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Strings

Transformação em String



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

```
frase = 'Aprendendo string'
print(frase.lower())
```

The bottom window, titled 'Python 3.7.2 Shell', shows the output of running the script. It displays the Python version and architecture, followed by the execution of the script which prints the lowercase version of the string.

```
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/string.teste.py ====
aprendendo string
>>> |
```

Strings

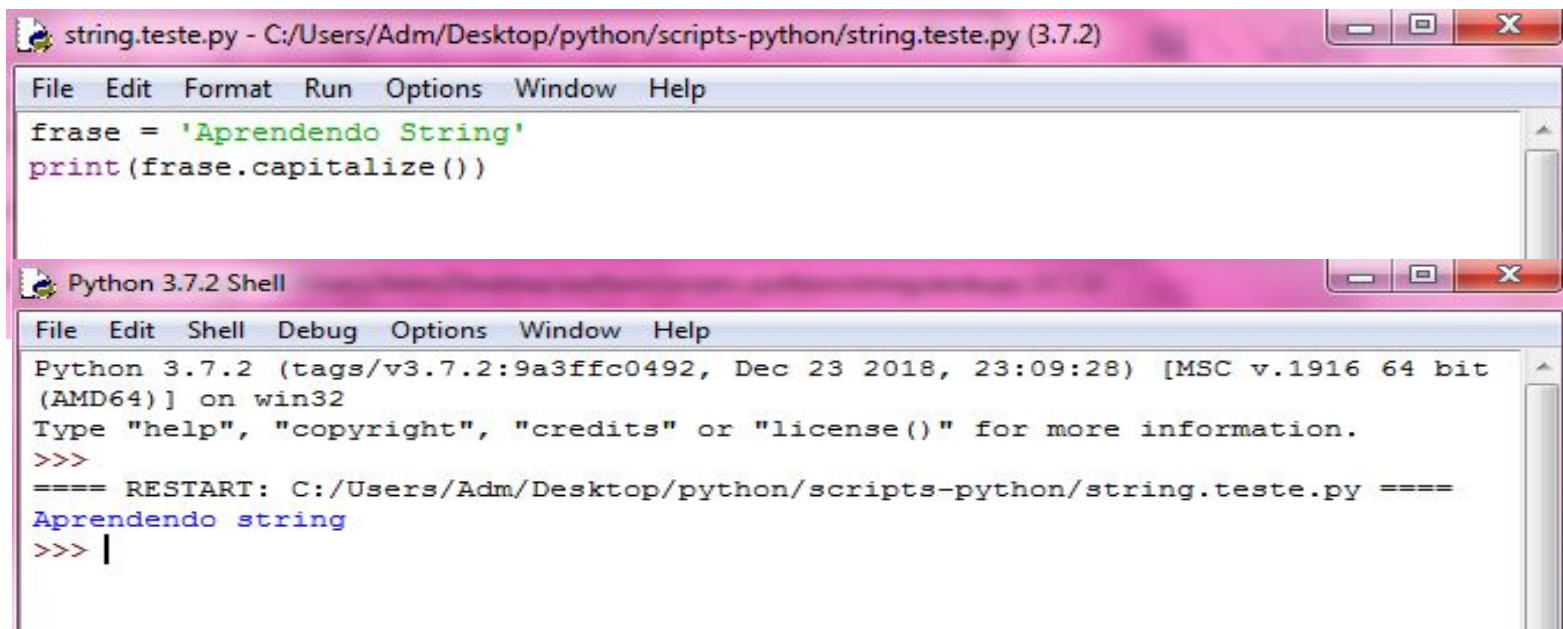
Transformação em **String**

- **Capitalize()** é uma função de transformação dentro de uma string e tem objetivo de trocar todos os caracteres para minúsculo e só manter o primeiro caractere em maiúsculo;
- Se escrevemos **frase.capitalize()**, o que já for minúsculo ele mantém, e transforma somente 'a' em maiúscula.

A	p	r	e	n	d	e	n	d	o		s	t	r	i	n	g
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Strings

Transformação em String



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

```
frase = 'Aprendendo String'
print(frase.capitalize())
```

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

```
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/string.teste.py ====
Aprendendo string
>>> |
```

Strings

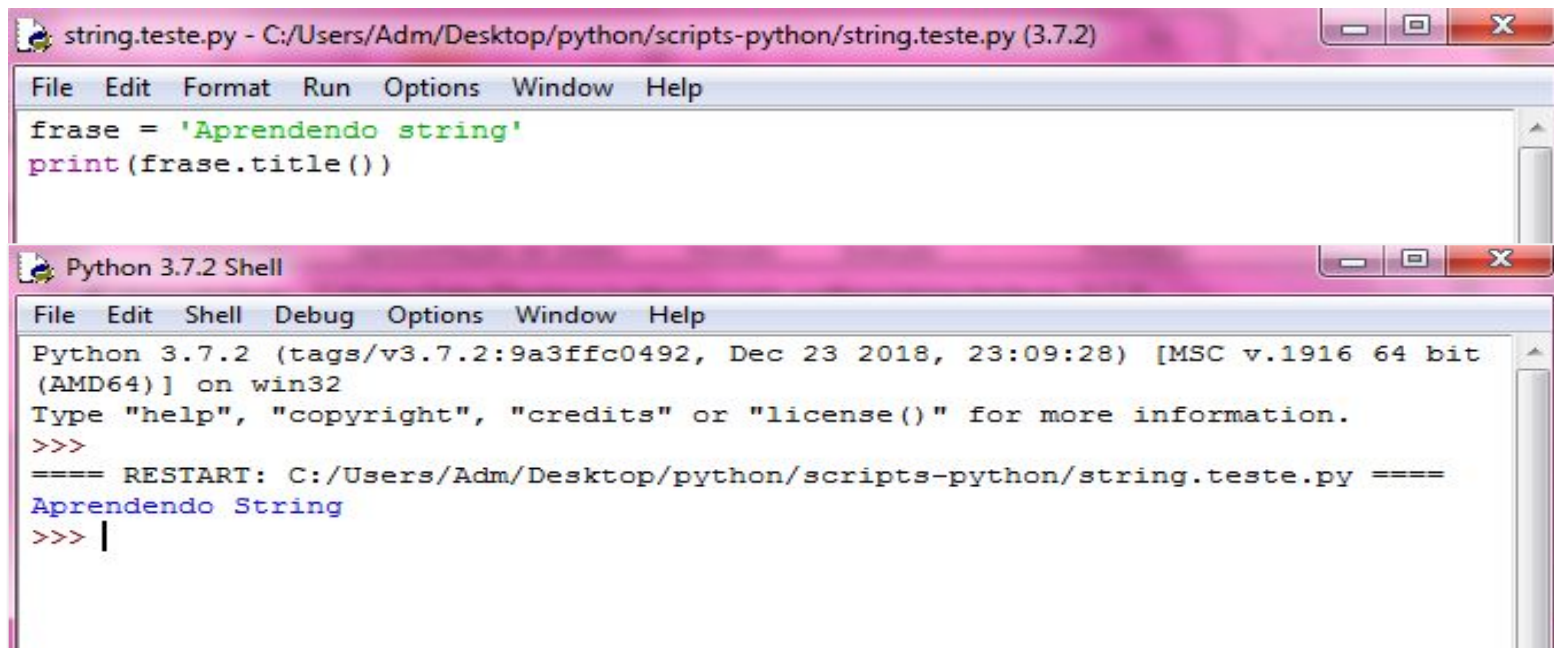
Transformação em **String**

- **Title()** é uma função de transformação dentro de uma string e tem objetivo de analisar quantas palavras tem em uma string e realizar uma capitalize em casa palavra;
- Se escrevemos **frase.title()**, ele irá analisar através dos espaços entre as palavras (quebra entre palavras) contando assim 2 palavras na frase e transformando minúscula em maiúscula.

A	p	r	e	n	d	e	n	d	o		S	t	r	i	n	g
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Strings

Transformação em **String**



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

```
frase = 'Aprendendo string'
print(frase.title())
```

The bottom window, titled 'Python 3.7.2 Shell', shows the execution 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/string.teste.py ====
Aprendendo String
>>> |
```

Strings

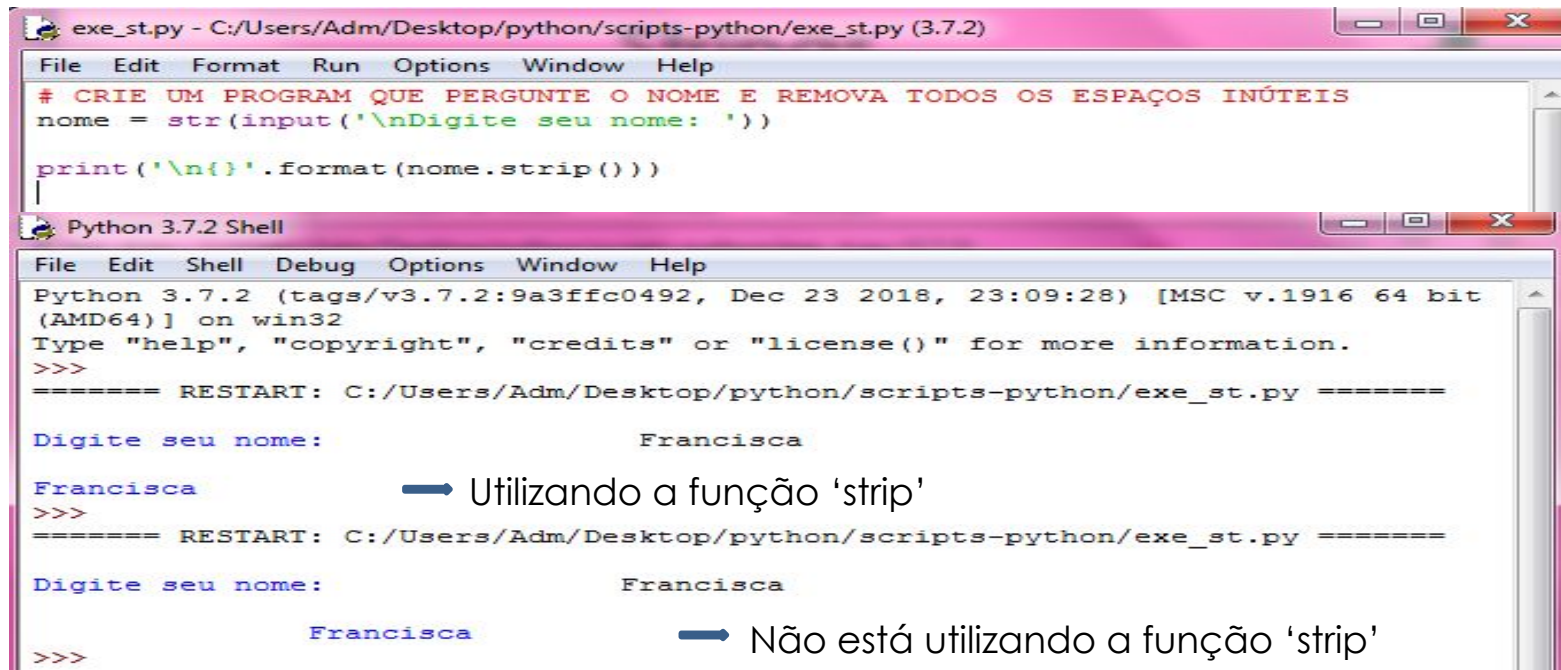
Transformação em **String**

- **Strip()** é uma função de transformação dentro de uma string e tem objetivo de remover todos os espaços inúteis em uma string;
- Se escrevemos **frase.strip()**, ele irá remover o espaço antes da letra 'A' e o espaço depois da letra 'g'.

	A	p	r	e	n	d	e	n	d	o		s	t	r	i	n	g	
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	

Strings

Transformação em String



```
exe_st.py - C:/Users/Adm/Desktop/python/scripts-python/exe_st.py (3.7.2)
File Edit Format Run Options Window Help
# CRIE UM PROGRAM QUE PERGUNTE O NOME E REMOVA TODOS OS ESPAÇOS INÚTEIS
nome = str(input('\nDigite seu nome: '))

print('\n{}'.format(nome.strip()))

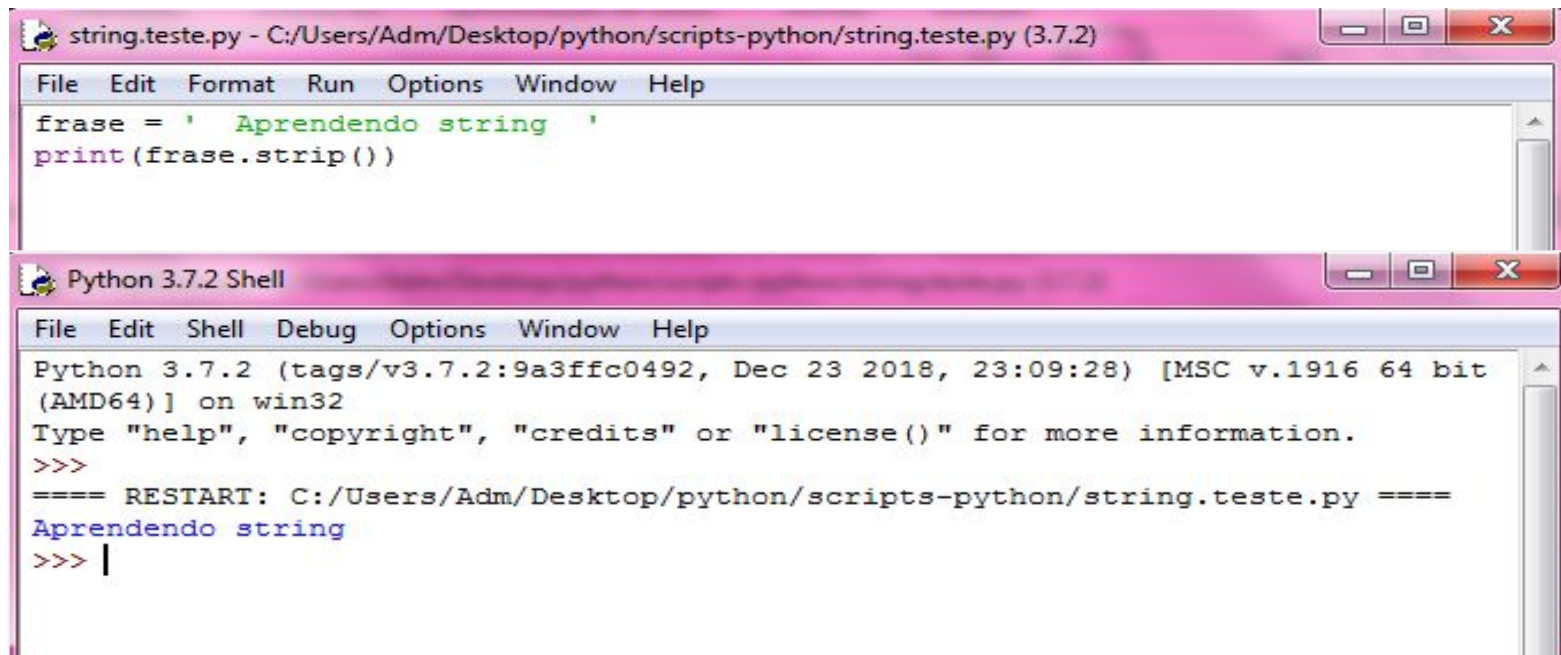
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/exe_st.py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

→ Utilizando a função 'strip'

→ Não está utilizando a função 'strip'

Strings

Transformação em String



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

```
frase = '  Aprendendo string  '
print(frase.strip())
```

The bottom window, titled 'Python 3.7.2 Shell', shows the output of running the script. It displays the Python version and architecture, followed by the execution of the script which prints the string 'Aprendendo string' with leading and trailing spaces removed.

```
Python 3.7.2 (tags/v3.7.2:9a3fffc0492, 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/string.teste.py ====
Aprendendo string
>>> |
```

Strings

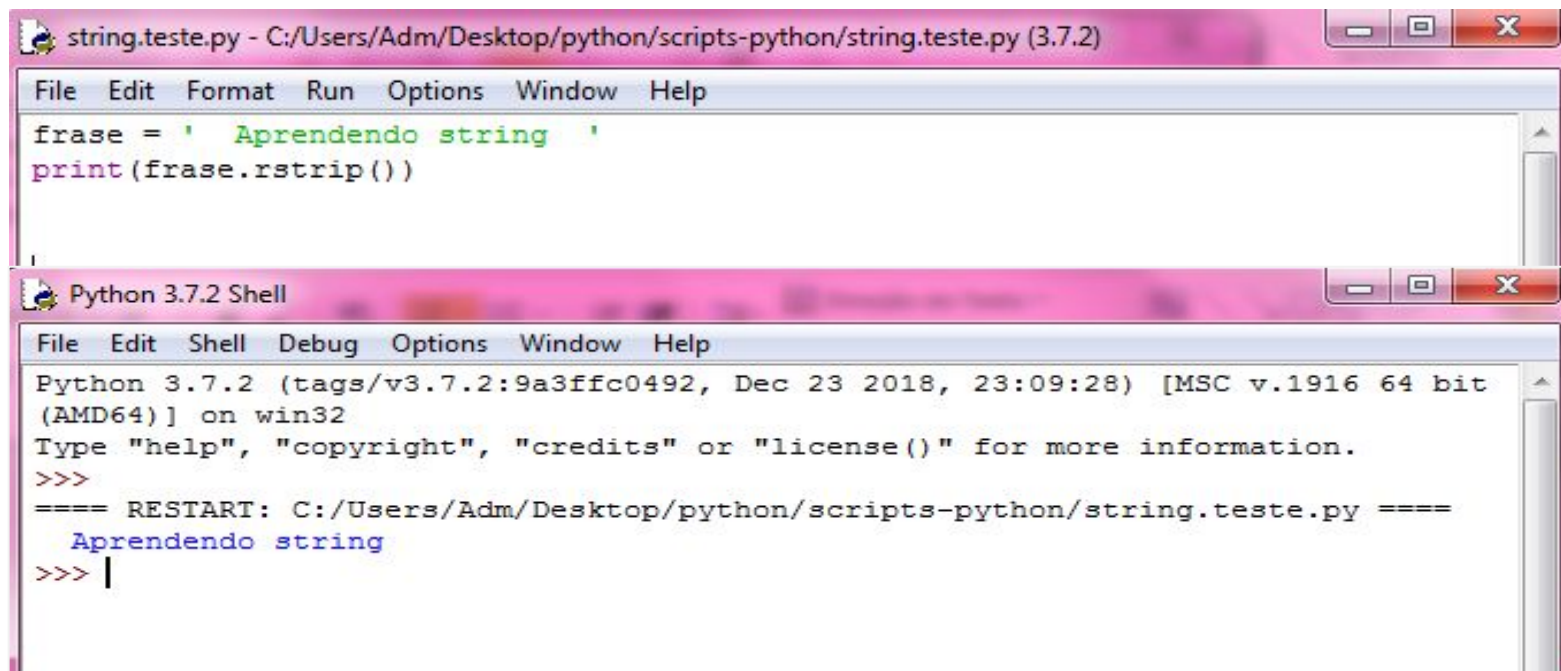
Transformação em **String**

- **Rstrip()** é uma função de transformação dentro de uma string e tem objetivo de remover somente o espaço inútil do lado direito em uma string;
- Se escrevemos **frase.rstrip()**, ele irá remover o espaço depois da posição '16'.

	A	p	r	e	n	d	e	n	d	o		s	t	r	i	n	g	
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	

Strings

Transformação em String



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

```
frase = '  Aprendendo string  '
print(frase.rstrip())
```

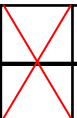
The bottom window, titled 'Python 3.7.2 Shell', shows the execution output:

```
Python 3.7.2 (tags/v3.7.2:9a3fffc0492, 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/string.teste.py ====
  Aprendendo string
>>> |
```

Strings

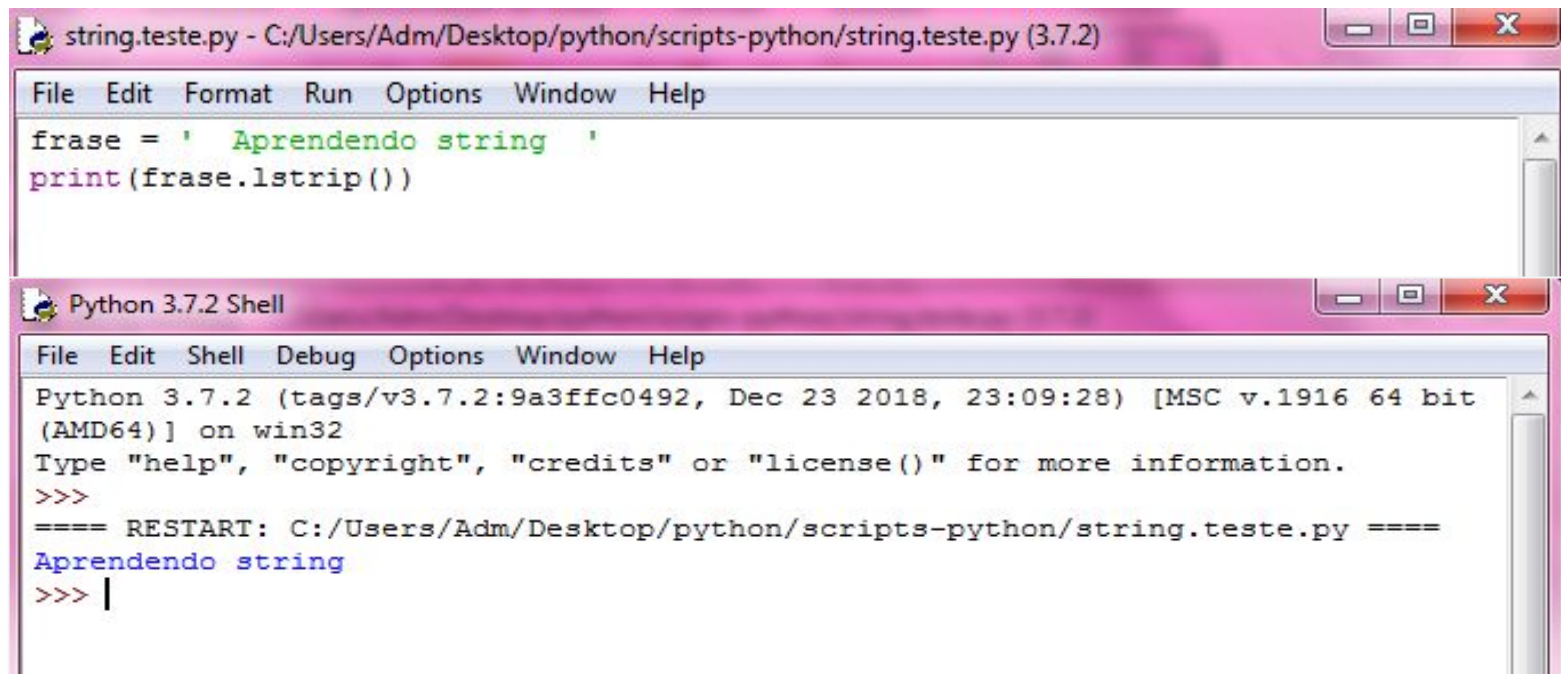
Transformação em **String**

- **Lstrip()** é uma função de transformação dentro de uma string e tem objetivo de remover somente o espaço inútil do lado esquerdo em uma string;
- Se escrevemos **frase.lstrip()**, ele irá remover o espaço antes da posição '0'.

	A	p	r	e	n	d	e	n	d	o		s	t	r	i	n	g	
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	

Strings

Transformação em String



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

```
File Edit Format Run Options Window Help
frase = '  Aprendendo string  '
print(frase.lstrip())
```

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

```
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3fffc0492, 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/string.teste.py ====
Aprendendo string
>>> |
```

Strings

Divisão em String

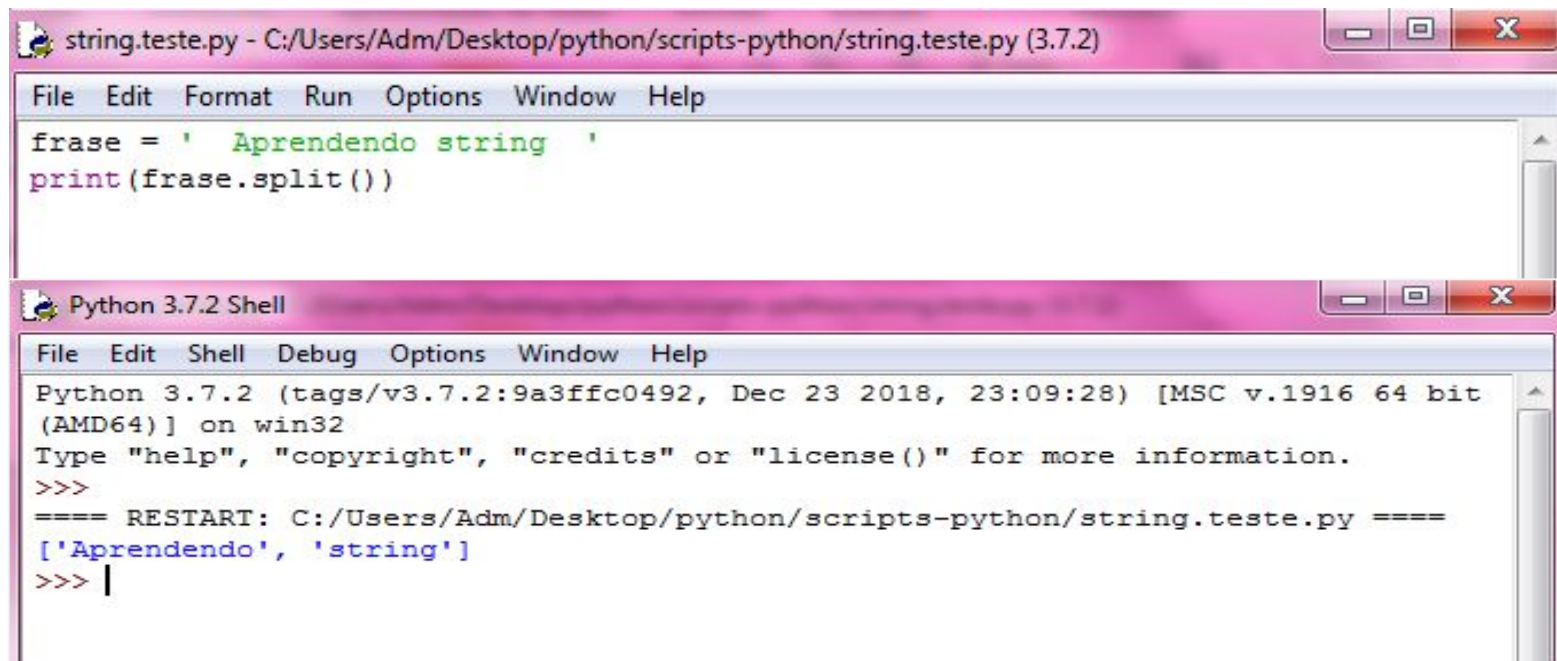
- **Split()** é uma função de divisão dentro de uma string e tem objetivo de fornecer uma divisão entre as strings considerando os seus espaços;
- Se escrevemos **frase.split()**, ele irá ver onde contém um espaço (neste caso na posição 10) e irá realizar a divisão gerando novas posições e novas strings.

A	p	r	e	n	d	e	n	d	o
0	1	2	3	4	5	6	7	8	9

s	t	r	i	n	g
0	1	2	3	4	5

Strings

Divisão em String



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

```
frase = '  Aprendendo string  '
print(frase.split())
```

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

```
Python 3.7.2 (tags/v3.7.2:9a3fffc0492, 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/string.teste.py ====
['Aprendendo', 'string']
>>> |
```

Strings

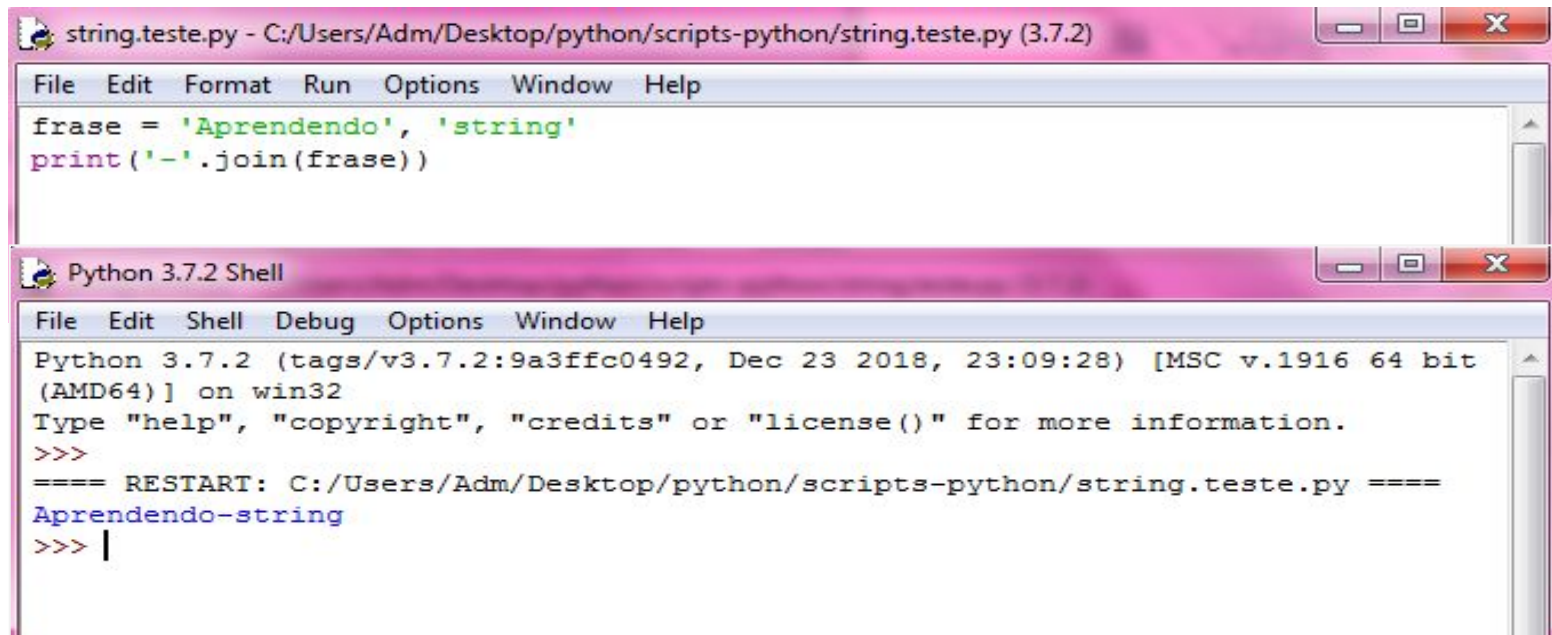
Junção em String

- **Join()** é uma função de junção dentro de uma string e tem objetivo de juntar uma divisão entre as strings;
- Se escrevemos `'-'.join(frase)`, ele irá juntar todos os elementos da frase e irá colocar este separador '-' que declaramos na string.

A	p	r	e	n	d	e	n	d	o	-	s	t	r	i	n	g
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Strings

Junção em String



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

```
frase = 'Aprendendo', 'string'  
print('-'.join(frase))
```

The bottom window, titled 'Python 3.7.2 Shell', shows the execution 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/string.teste.py ====  
Aprendendo-string  
>>> |
```

Exercícios Práticos

- Exercício 17:
- ✓ Faça um programa que leia o nome completo de uma pessoa e mostre:
 - O nome com todas as letras maiúsculas e minúsculas;
 - Quantas letras ao todo(sem considerar espaços);
 - Quantas letras tem o primeiro nome.

Exercícios Práticos

- Exercício 18:
- ✓ Faça um programa que leia o nome de uma cidade e diga se ela começa ou não com o nome 'Santo'.

Exercícios Práticos

- Exercício 19:
- ✓ Faça um programa que leia uma frase pelo teclado e mostre quantas vezes aparece a letra “A”, em que posição aparece a primeira vez e em que posição aparece a última vez.

Exercícios Práticos

- Exercício 20:
- ✓ Faça um programa que leia o nome completo de uma pessoa, mostrando em seguida o primeiro e o último nome separadamente.

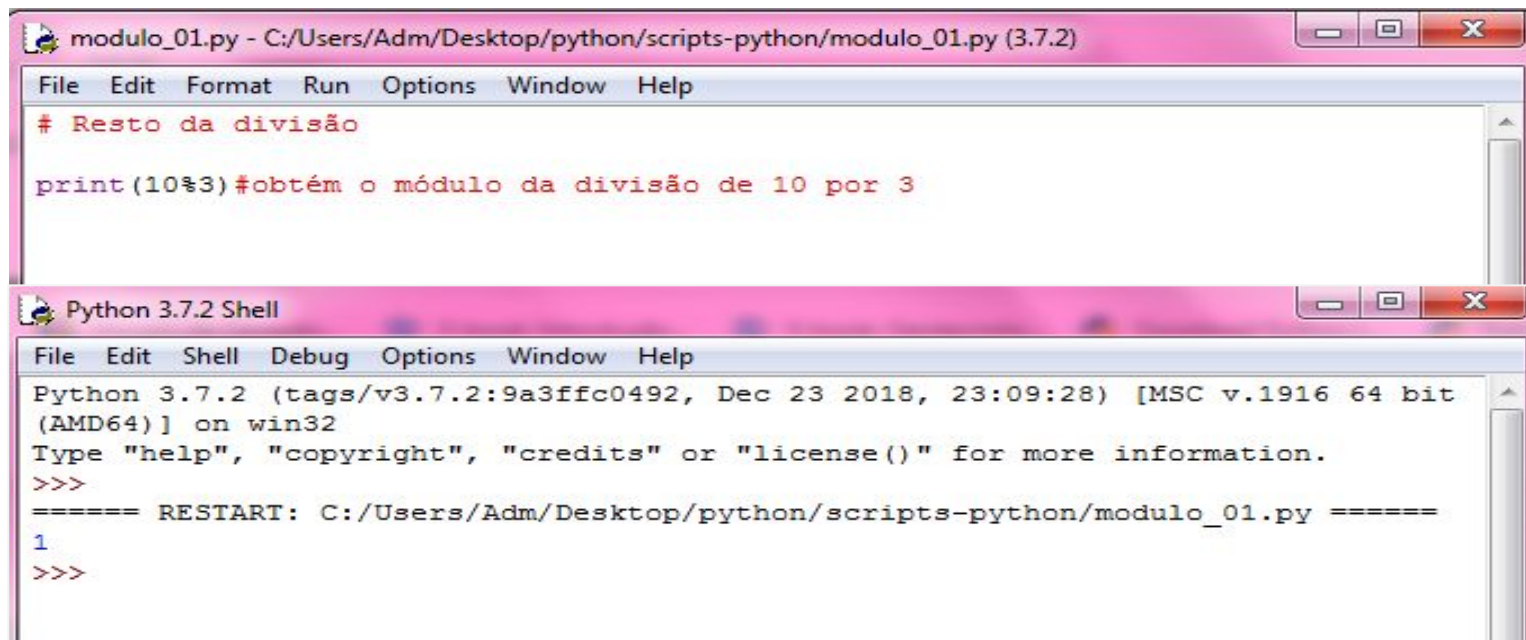
Exercícios Práticos

- Exercício 21:
- ✓ Faça um programa que leia o nome completo de uma pessoa, mostrando em seguida somente as iniciais em maiúsculas.

Módulo da Divisão

- O **Módulo da Divisão** é o operador que extraí o resto da divisão. Na linguagem em Python, bem como, a maioria das linguagens de programação, disponibilizam um operador para este fim;
- O sinal de porcentagem % é o operador módulo. No qual é considerado quase que um padrão entre as linguagens de programação.

Módulo da Divisão



The image shows a screenshot of a Python IDE with two windows. The top window, titled 'modulo_01.py - C:/Users/Adm/Desktop/python/scripts-python/modulo_01.py (3.7.2)', contains the following Python code:

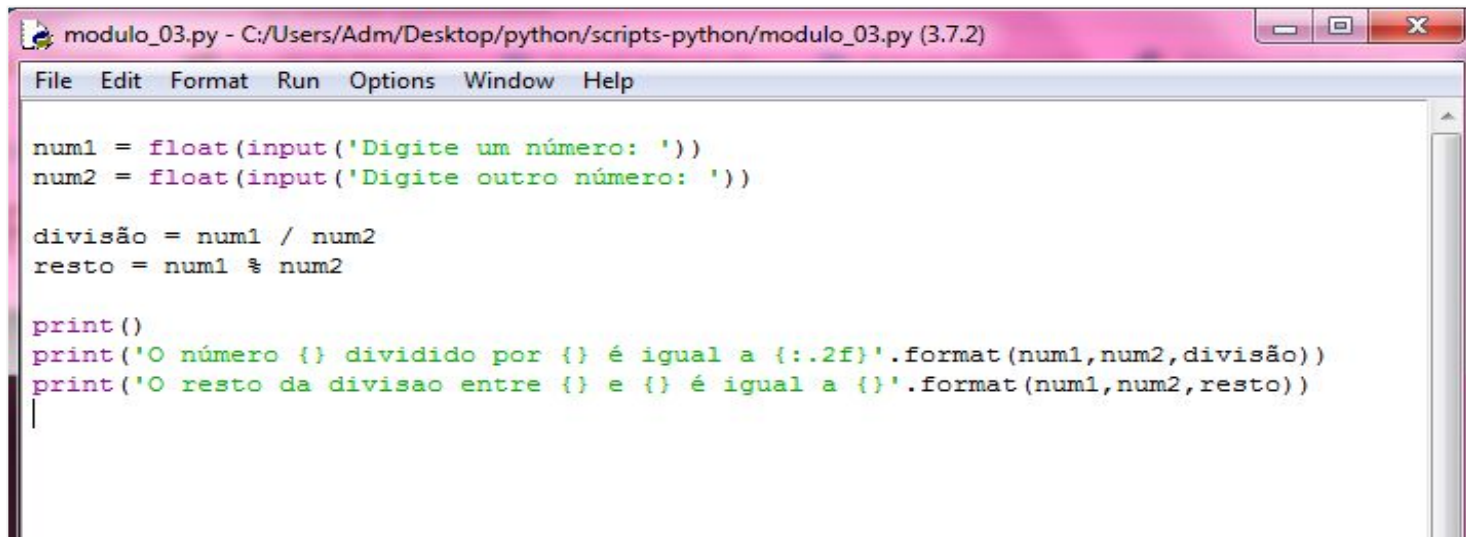
```
File Edit Format Run Options Window Help
# Resto da divisão

print(10%3)#obtem o módulo da divisão de 10 por 3
```

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

```
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/modulo_01.py =====
1
>>>
```

Módulo da Divisão



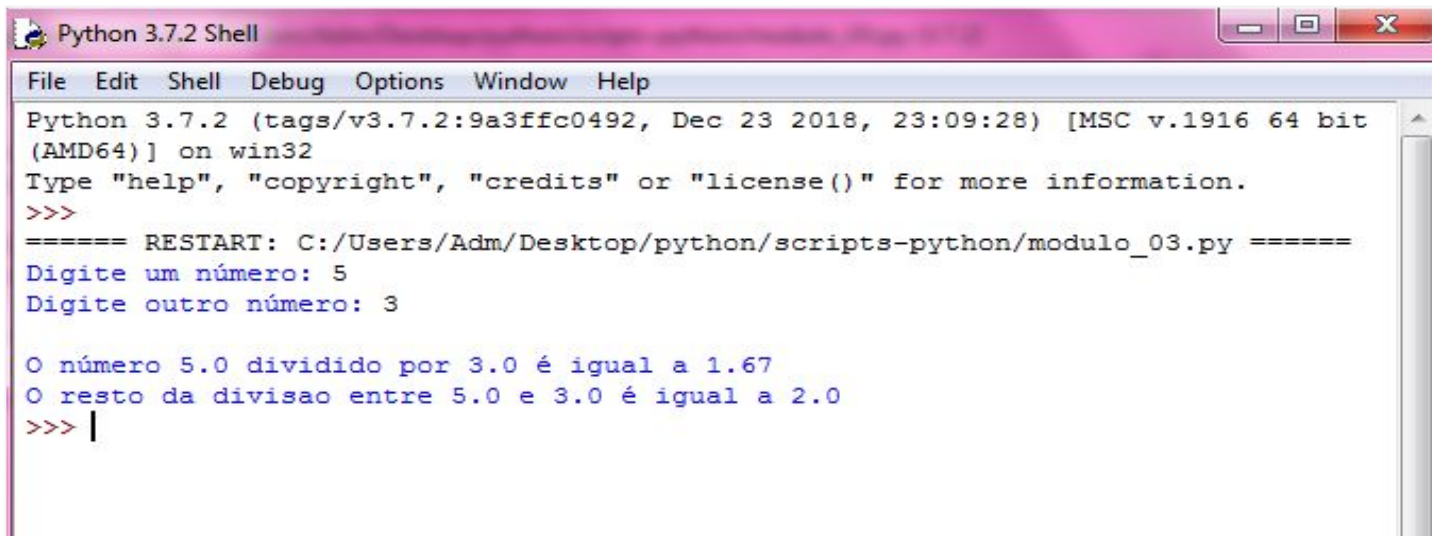
The image shows a screenshot of a Python IDE window titled "modulo_03.py - C:/Users/Adm/Desktop/python/scripts-python/modulo_03.py (3.7.2)". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The main text area contains the following Python code:

```
num1 = float(input('Digite um número: '))
num2 = float(input('Digite outro número: '))

divisão = num1 / num2
resto = num1 % num2

print()
print('O número {} dividido por {} é igual a {:.2f}'.format(num1, num2, divisão))
print('O resto da divisao entre {} e {} é igual a {}'.format(num1, num2, resto))
|
```

Módulo da Divisão



```
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/modulo_03.py =====
Digite um número: 5
Digite outro número: 3

O número 5.0 dividido por 3.0 é igual a 1.67
O resto da divisao entre 5.0 e 3.0 é igual a 2.0
>>> |
```

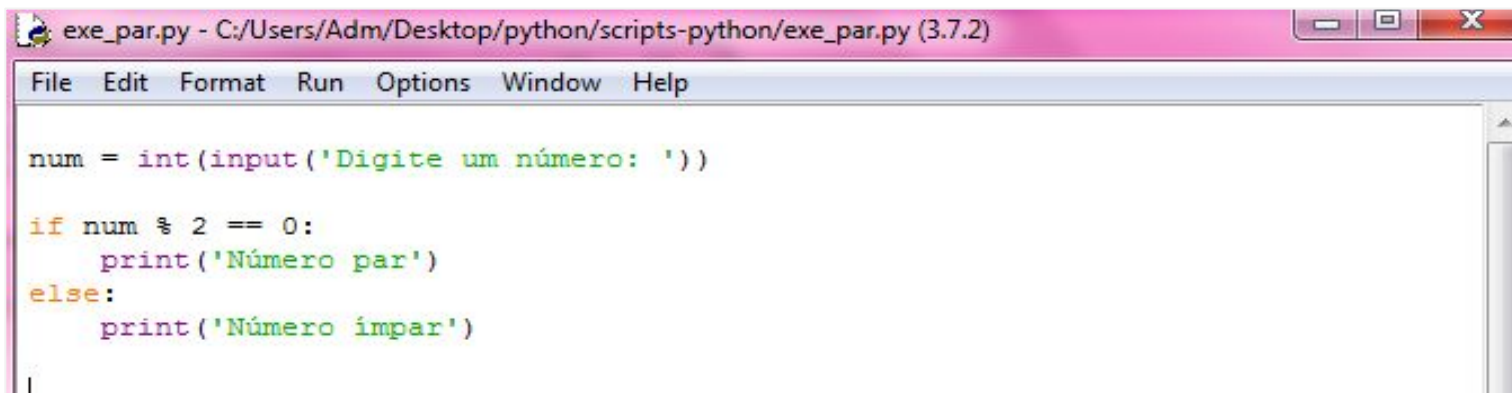
Módulo da Divisão

Números Pares

- Podemos saber se um número é par com o uso do operador módulo, para isto, basta verificarmos se a divisão possui resto, ou melhor, se o valor retornado é diferente de zero.

Módulo da Divisão

Números Pares



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

num = int(input('Digite um número: '))

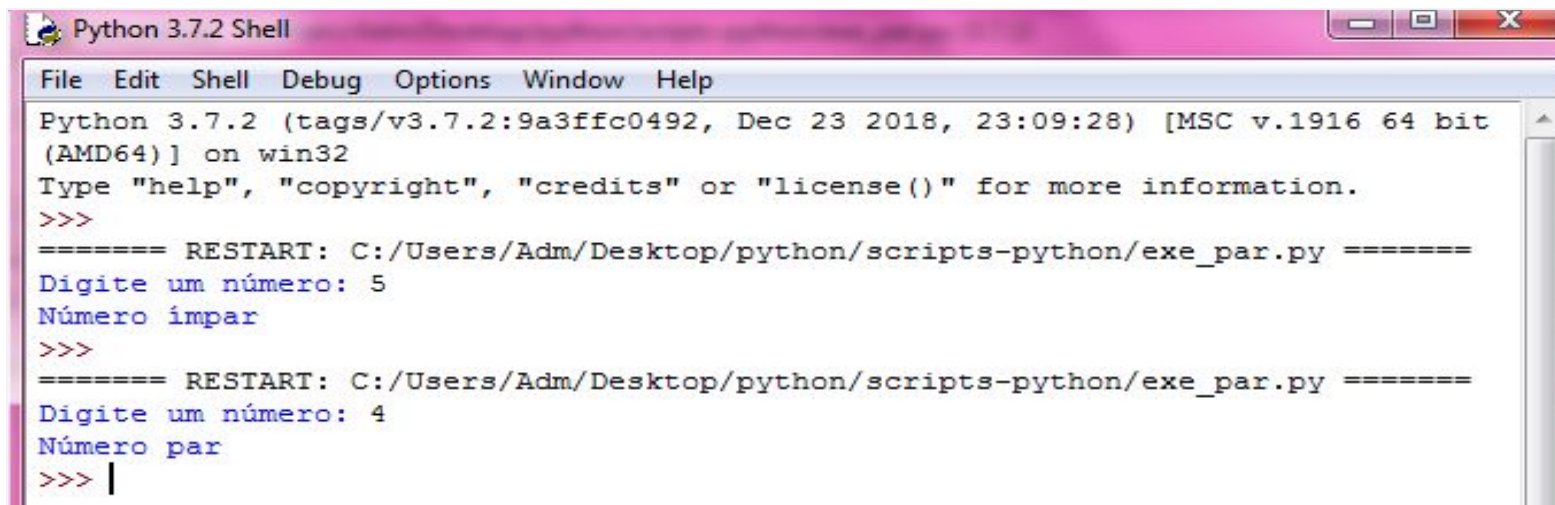
if num % 2 == 0:
    print('Número par')
else:
    print('Número ímpar')
```



Neste caso, estamos utilizando estruturas condicionais que estudaremos a seguir!

Módulo da Divisão

Números Pares



```
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/exe_par.py =====
Digite um número: 5
Número ímpar
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/exe_par.py =====
Digite um número: 4
Número par
>>> |
```

Estruturas Condicionais

- Uma estrutura de condição, verifica a condição de argumentos passados e, executa um comando caso a condição seja verdadeira;
- Antes de entrarmos na análise da estrutura em Python vamos conhecer os **Operadores Condicionais do Python**.

Estruturas Condicionais

- Os Operadores Condicionais são utilizados para fazer as comparações dos valores que são passados e retornam o valor Verdadeiro ou Falso.

Operador	Tipo	Valor
==	Igualdade	Verifica a igualdade entre dois valores.
!=	Diferença	Verifica a diferença entre dois valores.
>	Comparação	Verificar se o valor A é maior que o valor B.
<	Comparação	Verifica se o valor A é menor que o valor B.
>=	Comparação	Verifica se o valor A é maior ou igual ao valor B.
<=	Comparação	Verifica se o valor A é menor ou igual ao valor B.

Estruturas Condicionais

If

- O **IF** é uma **estrutura de condição simples** que permite avaliar uma expressão e, de acordo com seu resultado, executar uma determinada ação.

Estruturas Condicionais

If

- A sintaxe do IF em Python é o seguinte:

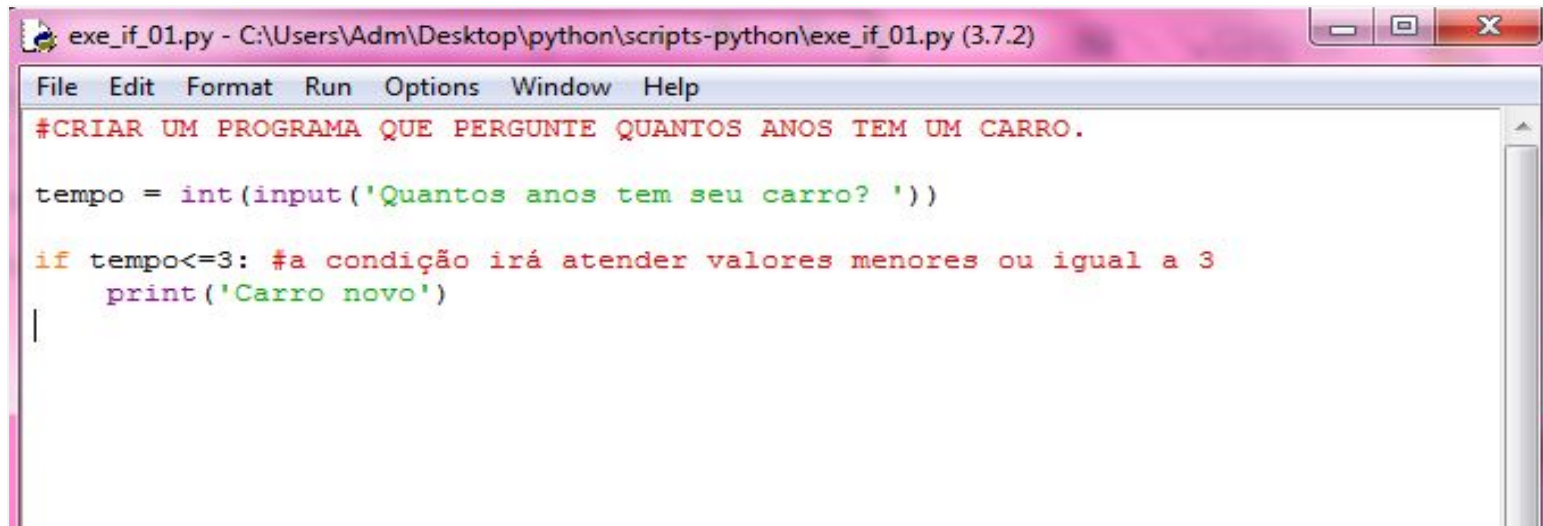
if < condição >:
->bloco de código

If num<10:
 print('Menor que 10')

Em Python ao final de cada condição, utilizamos " : "

Estruturas Condicionais

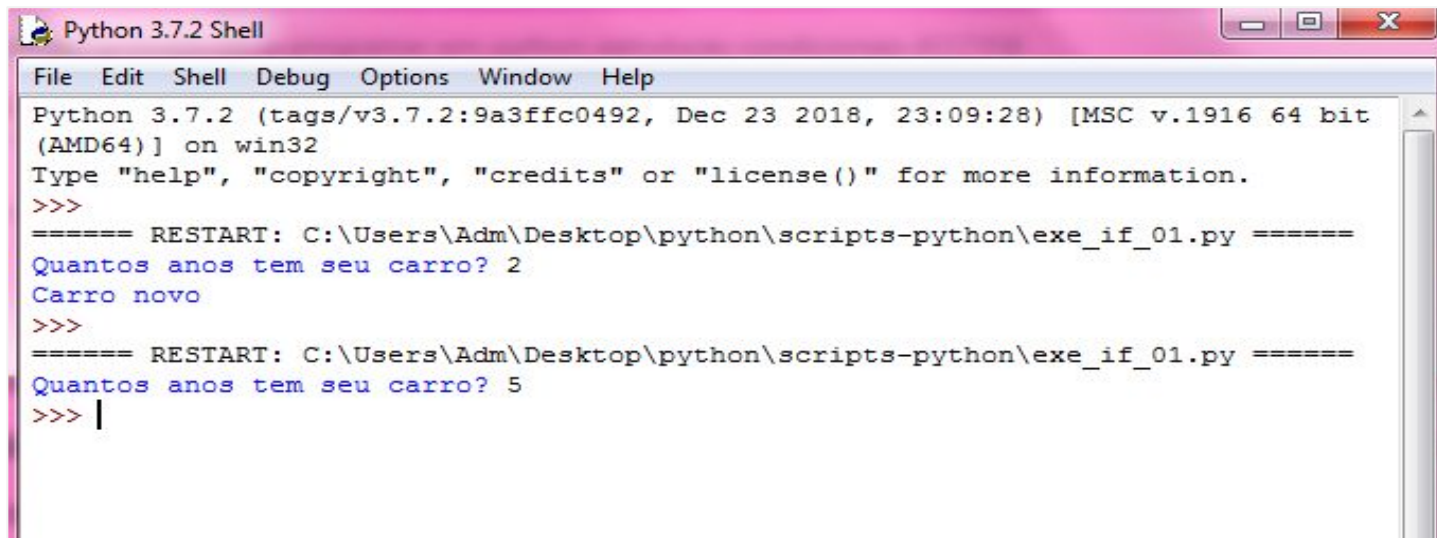
If

A screenshot of a Python IDE window titled "exe_if_01.py - C:\Users\Adm\Desktop\python\scripts-python\exe_if_01.py (3.7.2)". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code editor contains the following Python code:

```
#CRIAR UM PROGRAMA QUE PERGUNTE QUANTOS ANOS TEM UM CARRO.  
  
tempo = int(input('Quantos anos tem seu carro? '))  
  
if tempo<=3: #a condição irá atender valores menores ou igual a 3  
    print('Carro novo')  
|
```

Estruturas Condicionais

If



```
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\exe_if_01.py =====
Quantos anos tem seu carro? 2
Carro novo
>>>
===== RESTART: C:\Users\Adm\Desktop\python\scripts-python\exe_if_01.py =====
Quantos anos tem seu carro? 5
>>> |
```

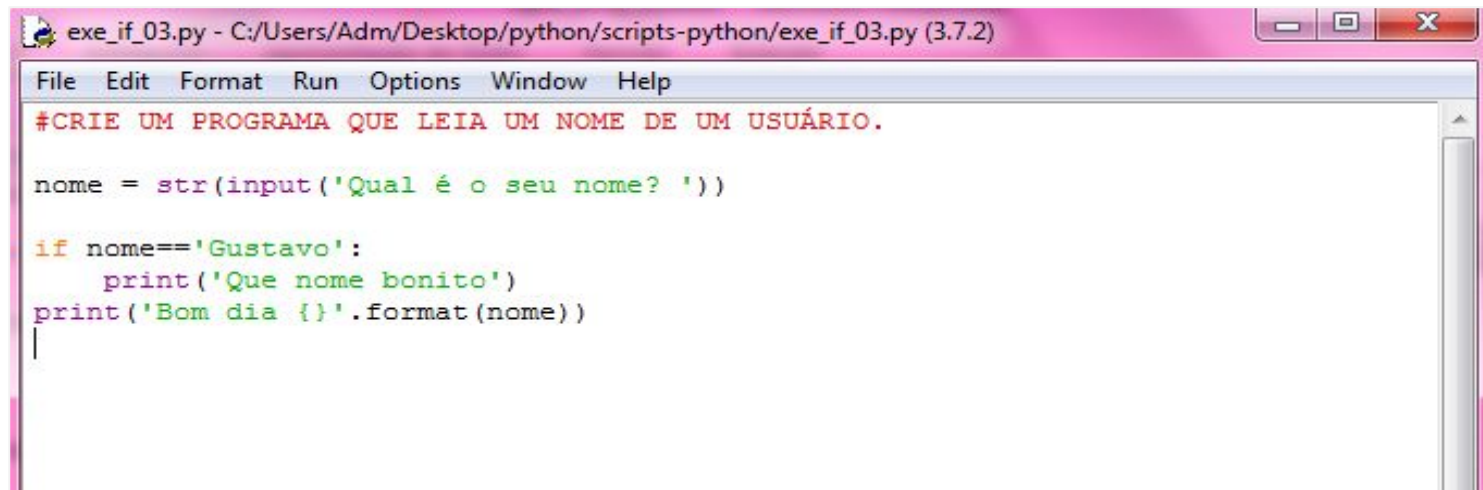
Estruturas Condicionais

If

- No exemplo utilizando **if**, ele irá atender apenas uma condição, caso você digite um número que não atenda a esta condição, você então deverá trabalhar com uma condição que atenda os dois casos, que seria o comando **if-else**.

Estruturas Condicionais

If



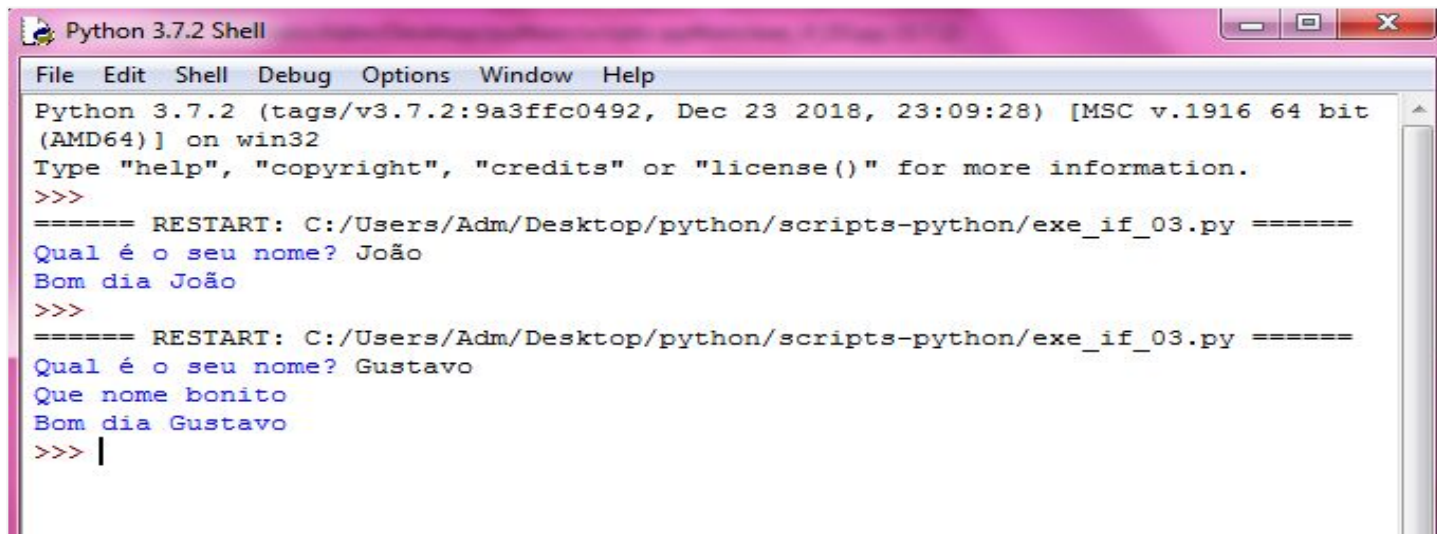
```
exe_if_03.py - C:/Users/Adm/Desktop/python/scripts-python/exe_if_03.py (3.7.2)
File Edit Format Run Options Window Help
#CRIE UM PROGRAMA QUE LEIA UM NOME DE UM USUÁRIO.

nome = str(input('Qual é o seu nome? '))

if nome=='Gustavo':
    print('Que nome bonito')
print('Bom dia {}'.format(nome))
|
```

Estruturas Condicionais

If



```
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/exe_if_03.py =====
Qual é o seu nome? João
Bom dia João
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/exe_if_03.py =====
Qual é o seu nome? Gustavo
Que nome bonito
Bom dia Gustavo
>>> |
```

Estruturas Condicionais

If-else

- Quando a condição não é atendida com o comando **if** é necessário utilizarmos o comando **else**;
- Neste caso, estamos trabalhando com uma **estrutura condicional composta** que executa um comando quando a condição for verdadeira e outra condição for falsa.

Estruturas Condicionais

If-else

- A sintaxe do IF-ELSE em Python é o seguinte:

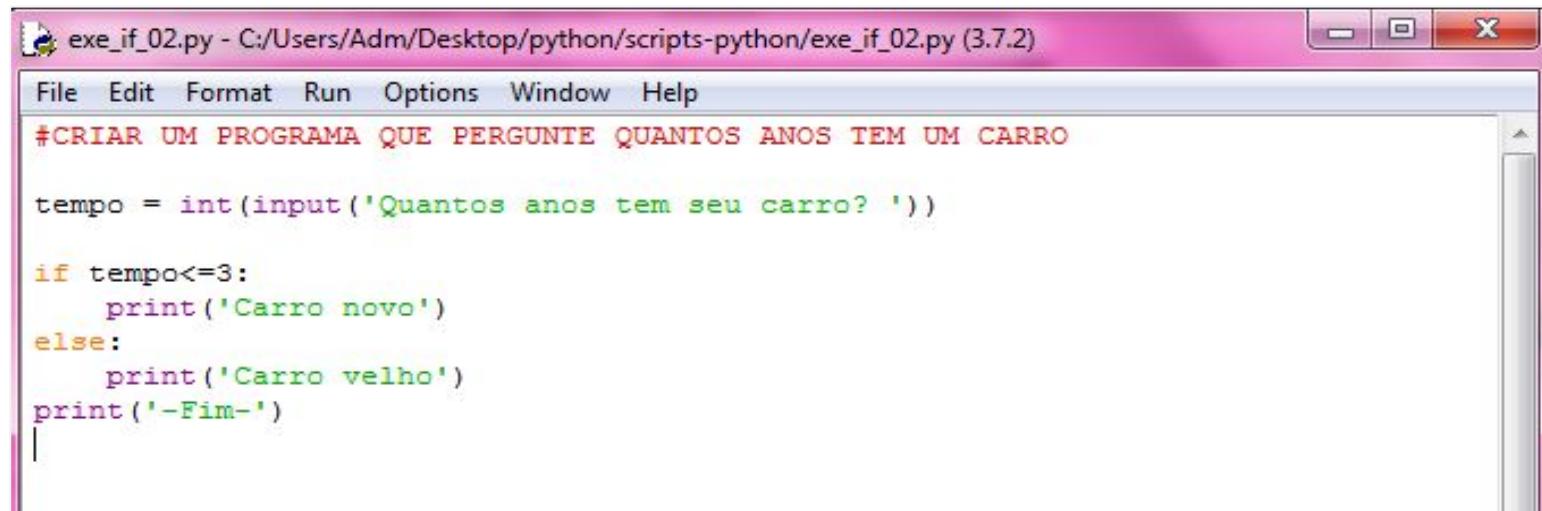
if < condição > :
->bloco de código
else:
->bloco de código

if num < 10:
 print('Menor que 10')
else:
 print('Maior que 10')



Estruturas Condicionais

If-else

A screenshot of a Python IDE window titled 'exe_if_02.py - C:/Users/Adm/Desktop/python/scripts-python/exe_if_02.py (3.7.2)'. The window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The code editor contains the following Python code:

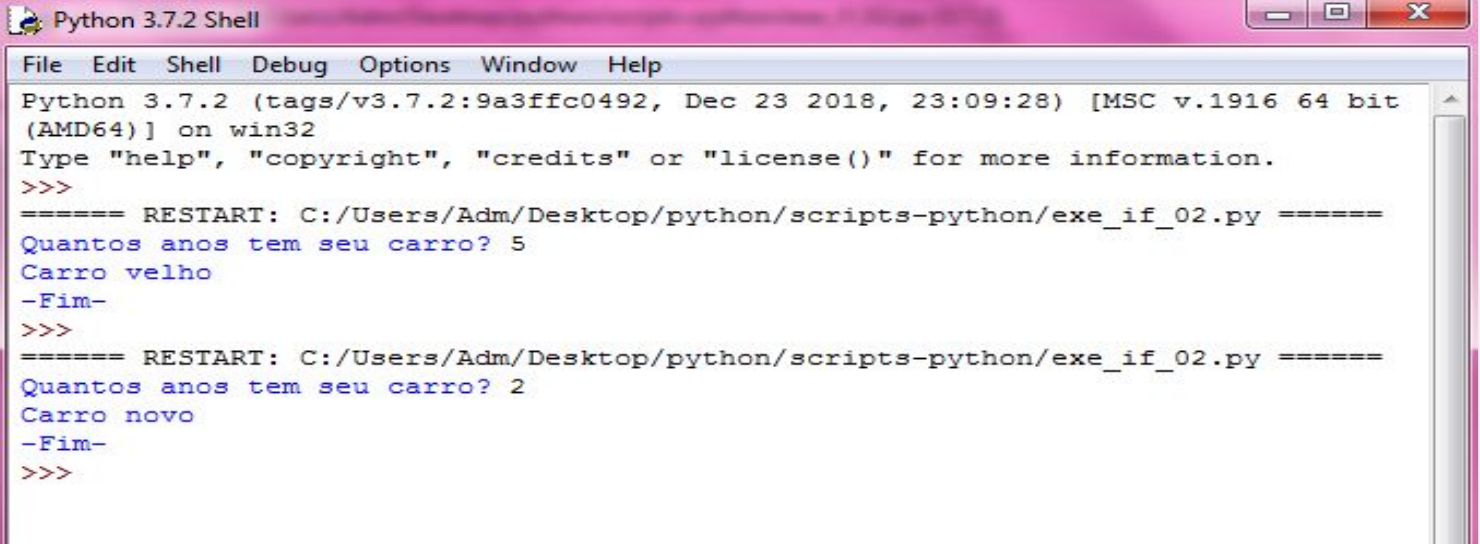
```
#CRIAR UM PROGRAMA QUE PERGUNTE QUANTOS ANOS TEM UM CARRO

tempo = int(input('Quantos anos tem seu carro? '))

if tempo<=3:
    print('Carro novo')
else:
    print('Carro velho')
print('-Fim-')
```

Estruturas Condicionais

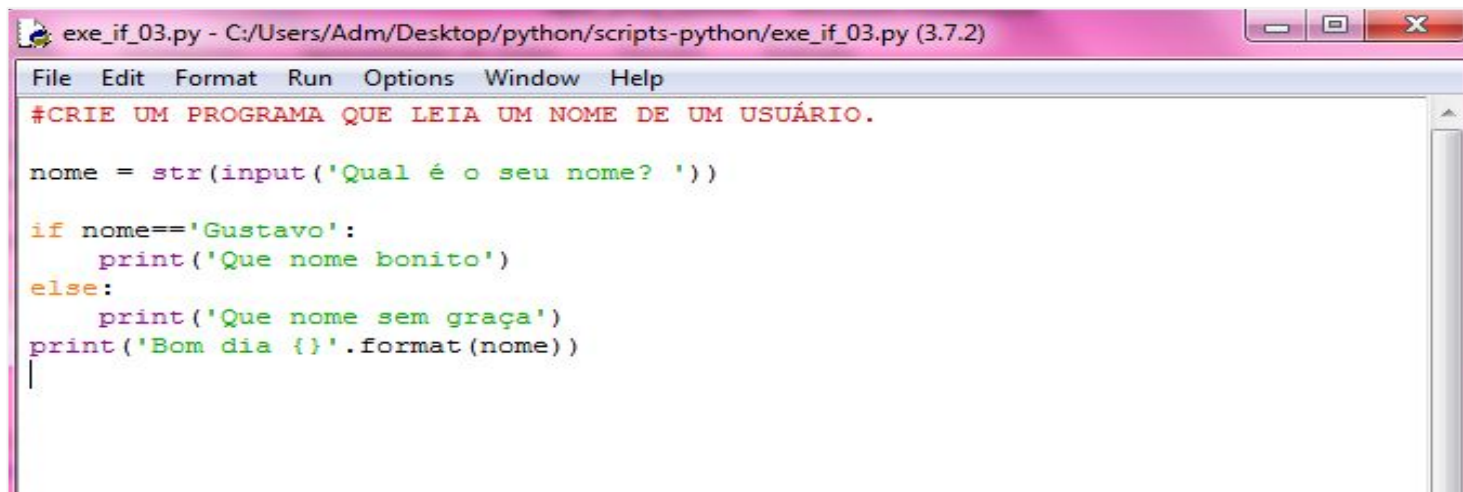
If-else



```
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/exe_if_02.py =====
Quantos anos tem seu carro? 5
Carro velho
-Fim-
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/exe_if_02.py =====
Quantos anos tem seu carro? 2
Carro novo
-Fim-
>>>
```

Estruturas Condicionais

If-else

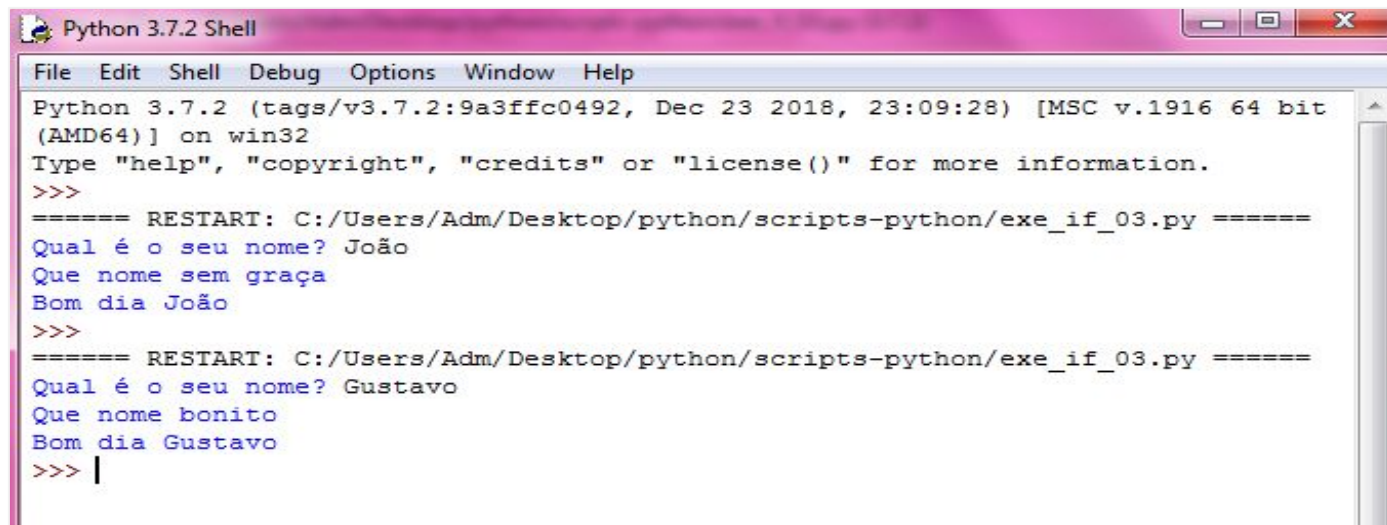


The image shows a screenshot of a Python IDE window titled "exe_if_03.py - C:/Users/Adm/Desktop/python/scripts-python/exe_if_03.py (3.7.2)". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code inside the window is as follows:

```
#CRIE UM PROGRAMA QUE LEIA UM NOME DE UM USUÁRIO.  
  
nome = str(input('Qual é o seu nome? '))  
  
if nome=='Gustavo':  
    print('Que nome bonito')  
else:  
    print('Que nome sem graça')  
print('Bom dia {}'.format(nome))
```

Estruturas Condicionais

If-else



```
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/exe_if_03.py =====
Qual é o seu nome? João
Que nome sem graça
Bom dia João
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/exe_if_03.py =====
Qual é o seu nome? Gustavo
Que nome bonito
Bom dia Gustavo
>>> |
```

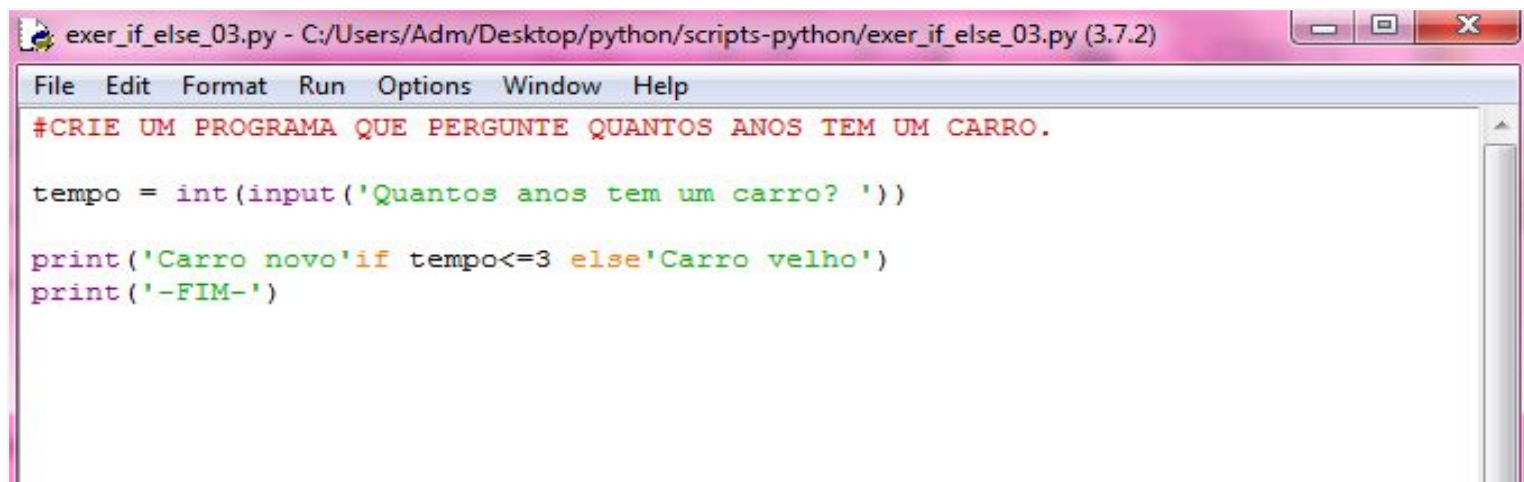
Estruturas Condicionais

If-else

- Podemos trabalhar de uma maneira mais simples em Python, utilizando a **condição simplificada**;
- Esta condição irá simplificar em somente uma linha no Python a estrutura condicional if-else.

Estruturas Condicionais

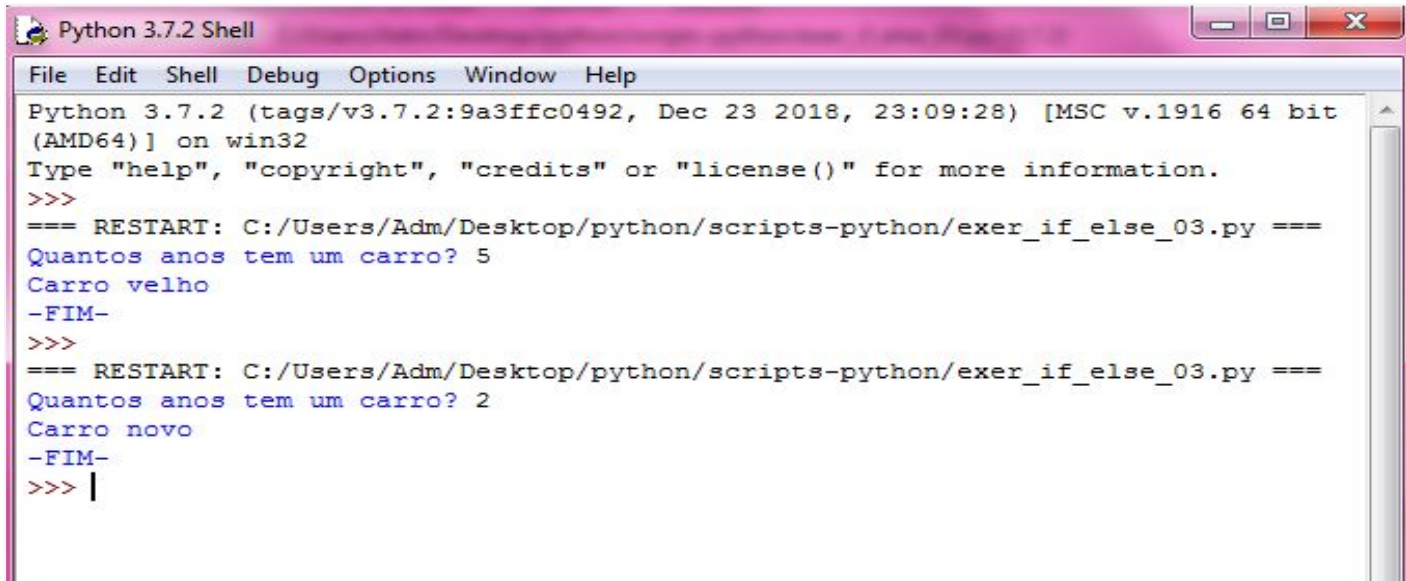
If-else

A screenshot of a Python IDE window titled "exer_if_else_03.py - C:/Users/Adm/Desktop/python/scripts-python/exer_if_else_03.py (3.7.2)". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code editor contains the following Python code:

```
#CRIE UM PROGRAMA QUE PERGUNTE QUANTOS ANOS TEM UM CARRO.  
  
tempo = int(input('Quantos anos tem um carro? '))  
  
print('Carro novo' if tempo <= 3 else 'Carro velho')  
print('-FIM-')
```

Estruturas Condicionais

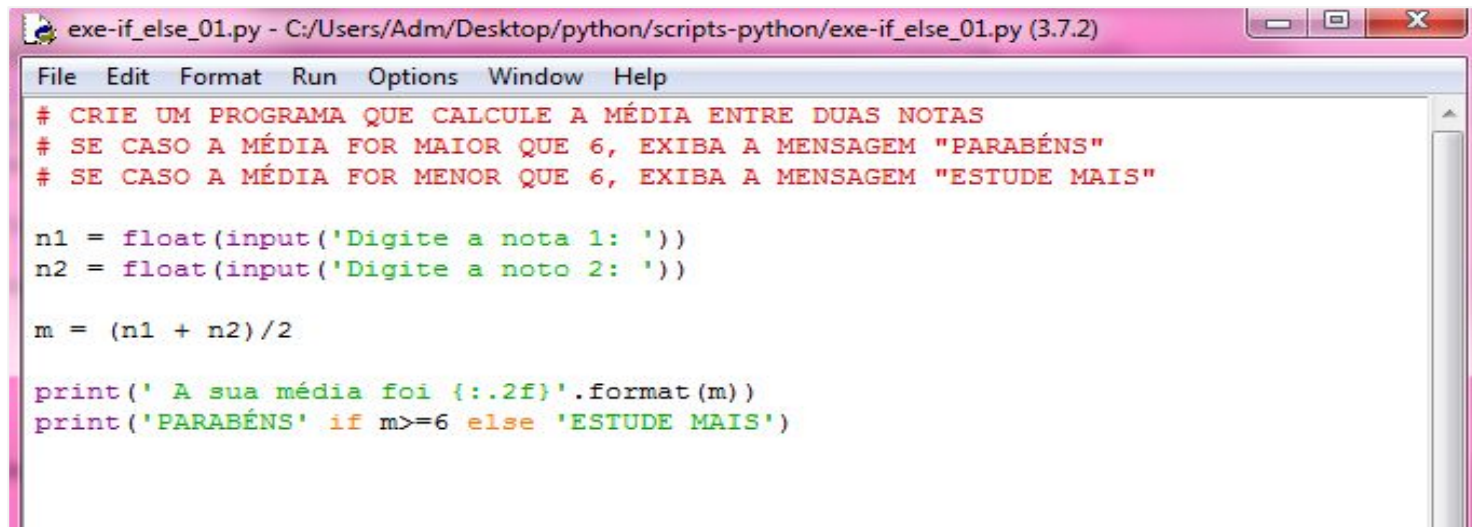
If-else



```
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/exer_if_else_03.py ===
Quantos anos tem um carro? 5
Carro velho
-FIM-
>>>
=== RESTART: C:/Users/Adm/Desktop/python/scripts-python/exer_if_else_03.py ===
Quantos anos tem um carro? 2
Carro novo
-FIM-
>>> |
```

Estruturas Condicionais

If-else



```
exe-if_else_01.py - C:/Users/Adm/Desktop/python/scripts-python/exe-if_else_01.py (3.7.2)
File Edit Format Run Options Window Help
# CRIE UM PROGRAMA QUE CALCULE A MÉDIA ENTRE DUAS NOTAS
# SE CASO A MÉDIA FOR MAIOR QUE 6, EXIBA A MENSAGEM "PARABÉNS"
# SE CASO A MÉDIA FOR MENOR QUE 6, EXIBA A MENSAGEM "ESTUDE MAIS"

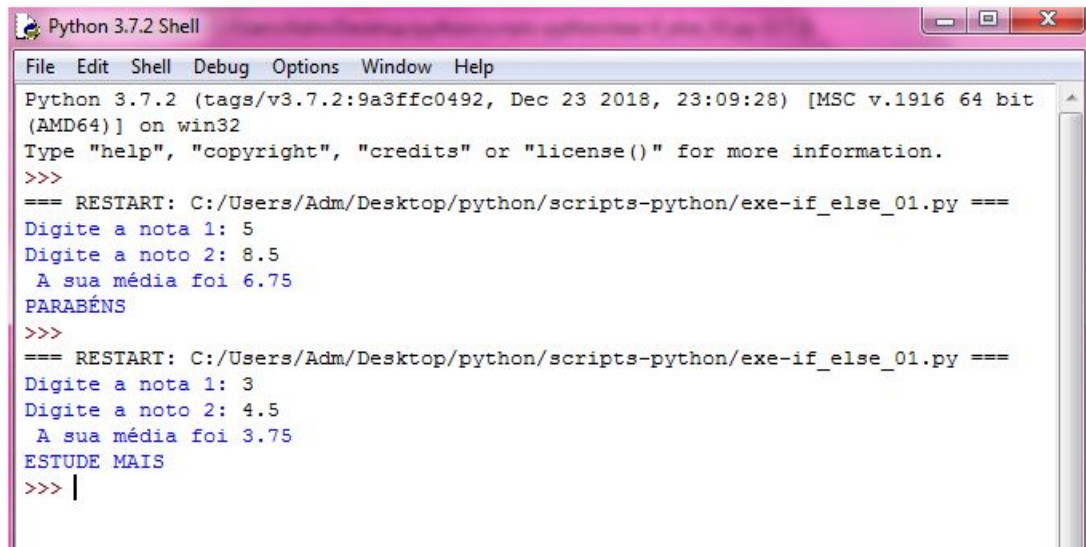
n1 = float(input('Digite a nota 1: '))
n2 = float(input('Digite a nota 2: '))

m = (n1 + n2)/2

print(' A sua média foi {:.2f}'.format(m))
print('PARABÉNS' if m>=6 else 'ESTUDE MAIS')
```

Estruturas Condicionais

If-else



```
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/exe-if_else_01.py ===
Digite a nota 1: 5
Digite a nota 2: 8.5
A sua média foi 6.75
PARABÉNS
>>>
=== RESTART: C:/Users/Adm/Desktop/python/scripts-python/exe-if_else_01.py ===
Digite a nota 1: 3
Digite a nota 2: 4.5
A sua média foi 3.75
ESTUDE MAIS
>>> |
```

Estruturas Condicionais

If-elif-else

- Se existir mais de uma condição alternativa que precisa ser verificada, devemos utilizar o **elif** para avaliar as expressões intermediárias antes de usar o **else**;
- Neste caso, estamos trabalhando com **estruturas condicionais aninhadas** que são várias condições em cascatas, ou seja, um **if** dentro de outro **if**.

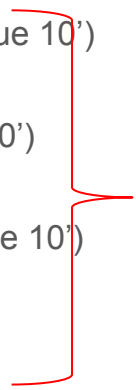
Estruturas Condicionais

If-elif-else

- A sintaxe do IF-ELIF-ELSE em Python é o seguinte:

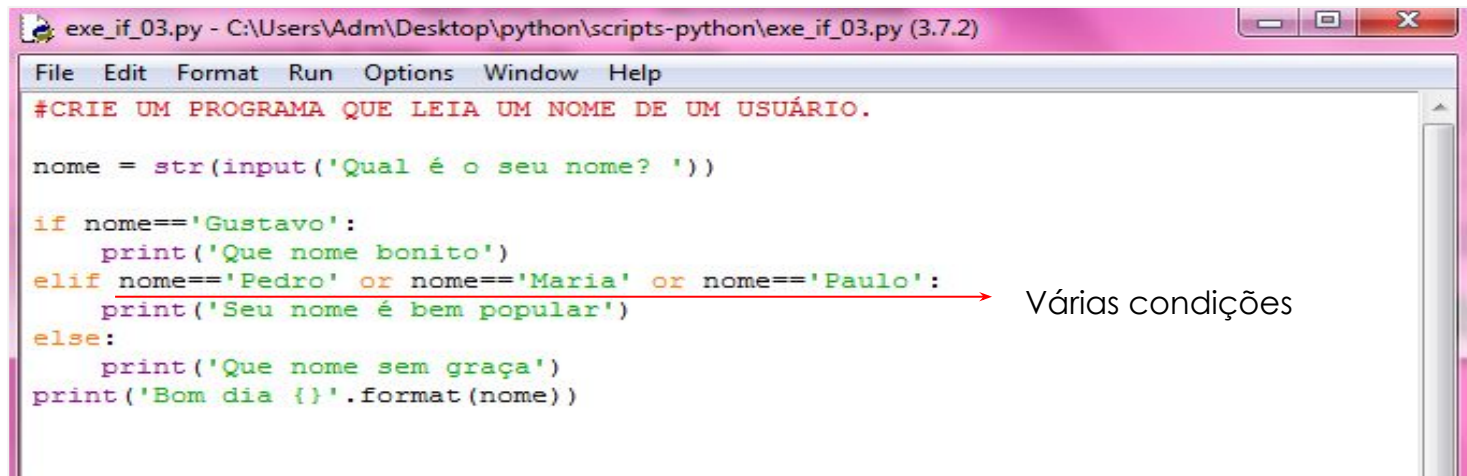
if < condição > :
->bloco de código
elif < condição >:
->bloco de código
else:
->bloco de código

if num <10:
 print('Menor que 10')
elif num==10:
 print('Igual a 10')
else:
 print('Maior que 10')



Estruturas Condicionais

If-elif-else



```
exe_if_03.py - C:\Users\Adm\Desktop\python\scripts-python\exe_if_03.py (3.7.2)
File Edit Format Run Options Window Help
#CRIE UM PROGRAMA QUE LEIA UM NOME DE UM USUÁRIO.

nome = str(input('Qual é o seu nome? '))

if nome=='Gustavo':
    print('Que nome bonito')
elif nome=='Pedro' or nome=='Maria' or nome=='Paulo':
    print('Seu nome é bem popular')
else:
    print('Que nome sem graça')
print('Bom dia {}'.format(nome))
```

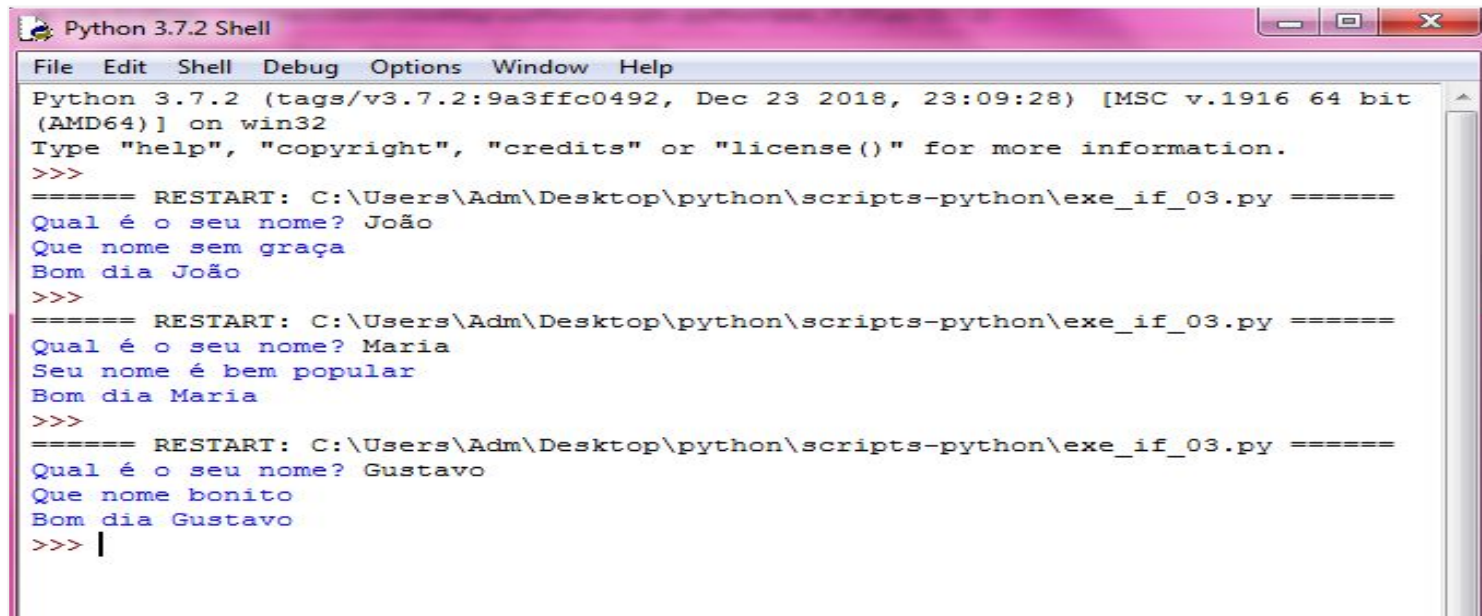
Várias condições



O **elif** deve ser sempre utilizado junto com o **if**

Estruturas Condicionais

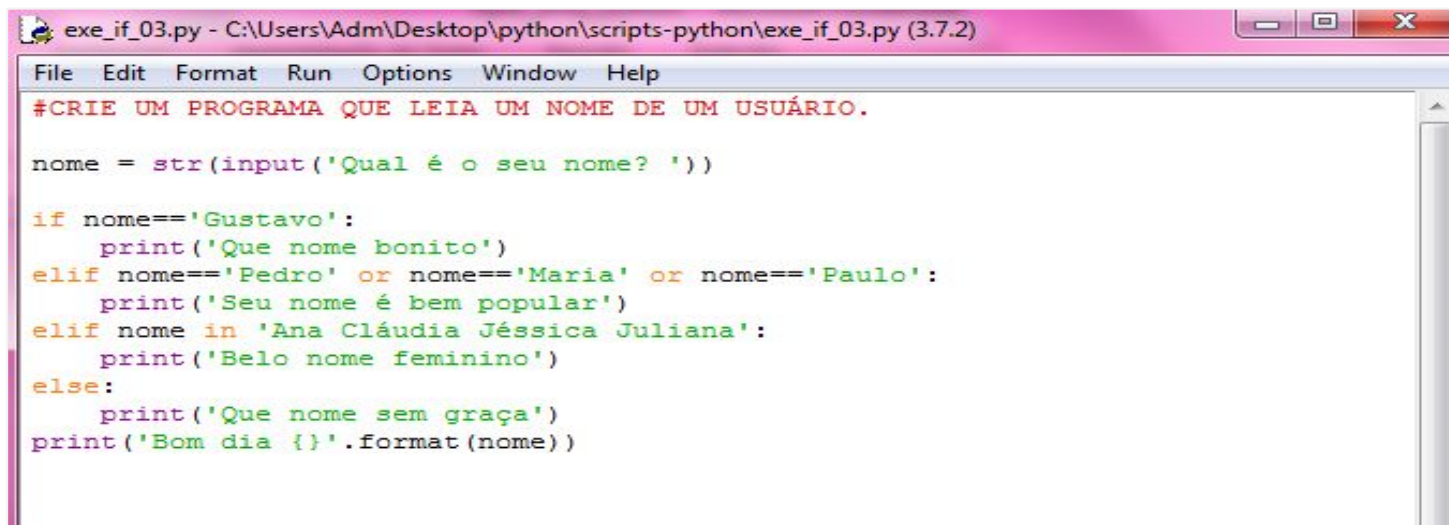
If-elif-else



```
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\exe_if_03.py =====
Qual é o seu nome? João
Que nome sem graça
Bom dia João
>>>
===== RESTART: C:\Users\Adm\Desktop\python\scripts-python\exe_if_03.py =====
Qual é o seu nome? Maria
Seu nome é bem popular
Bom dia Maria
>>>
===== RESTART: C:\Users\Adm\Desktop\python\scripts-python\exe_if_03.py =====
Qual é o seu nome? Gustavo
Que nome bonito
Bom dia Gustavo
>>> |
```

Estruturas Condicionais

If-elif-else



```
exe_if_03.py - C:\Users\Adm\Desktop\python\scripts-python\exe_if_03.py (3.7.2)
File Edit Format Run Options Window Help
#CRIE UM PROGRAMA QUE LEIA UM NOME DE UM USUÁRIO.

nome = str(input('Qual é o seu nome? '))

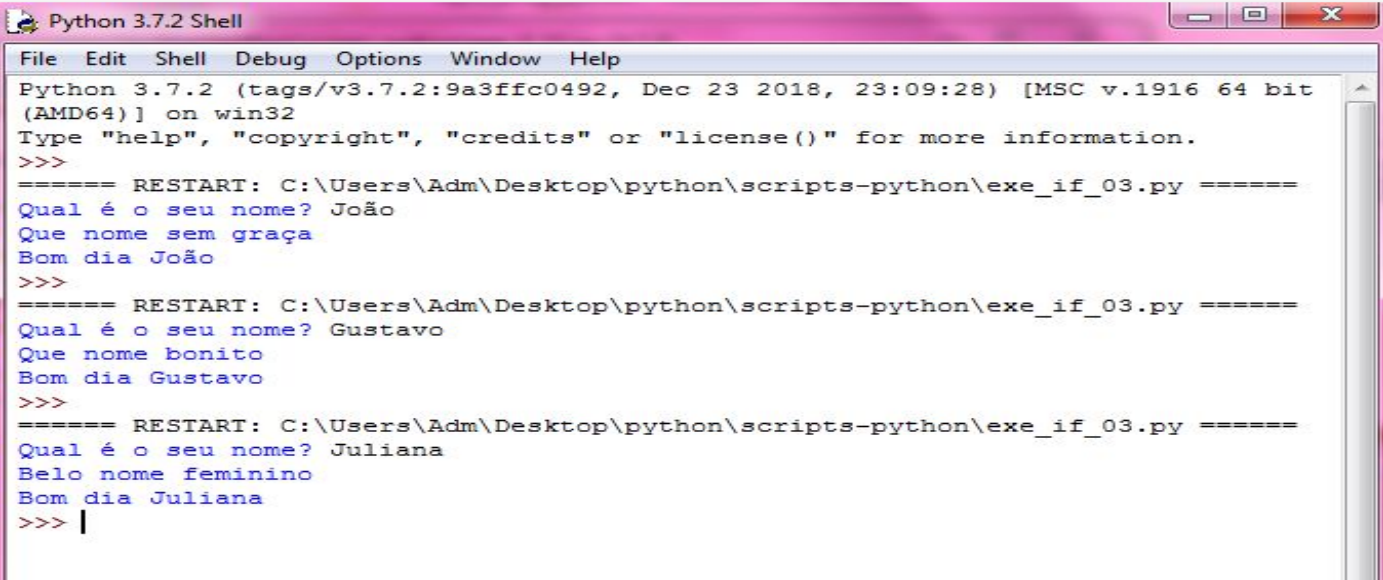
if nome=='Gustavo':
    print('Que nome bonito')
elif nome=='Pedro' or nome=='Maria' or nome=='Paulo':
    print('Seu nome é bem popular')
elif nome in 'Ana Cláudia Jéssica Juliana':
    print('Belo nome feminino')
else:
    print('Que nome sem graça')
print('Bom dia {}'.format(nome))
```



Pode-se colocar quantos **elif** for necessário

Estruturas Condicionais

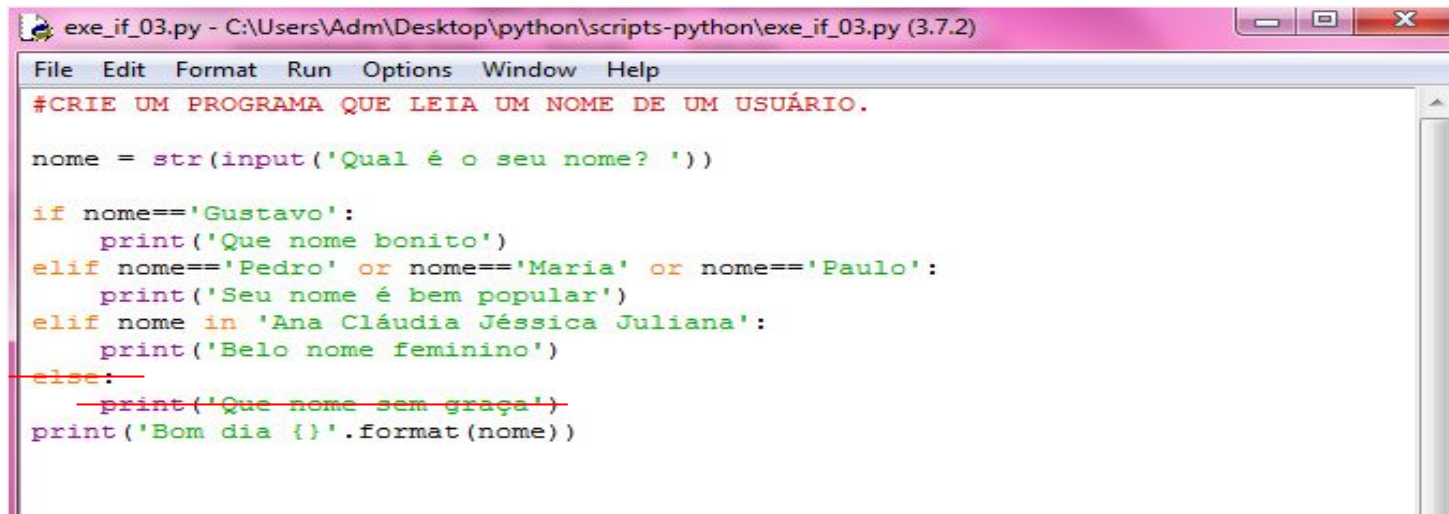
If-elif-else



```
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\exe_if_03.py =====
Qual é o seu nome? João
Que nome sem graça
Bom dia João
>>>
===== RESTART: C:\Users\Adm\Desktop\python\scripts-python\exe_if_03.py =====
Qual é o seu nome? Gustavo
Que nome bonito
Bom dia Gustavo
>>>
===== RESTART: C:\Users\Adm\Desktop\python\scripts-python\exe_if_03.py =====
Qual é o seu nome? Juliana
Belo nome feminino
Bom dia Juliana
>>> |
```

Estruturas Condicionais

If-elif-else



```
exe_if_03.py - C:\Users\Adm\Desktop\python\scripts-python\exe_if_03.py (3.7.2)
File Edit Format Run Options Window Help
#CRIE UM PROGRAMA QUE LEIA UM NOME DE UM USUÁRIO.

nome = str(input('Qual é o seu nome? '))

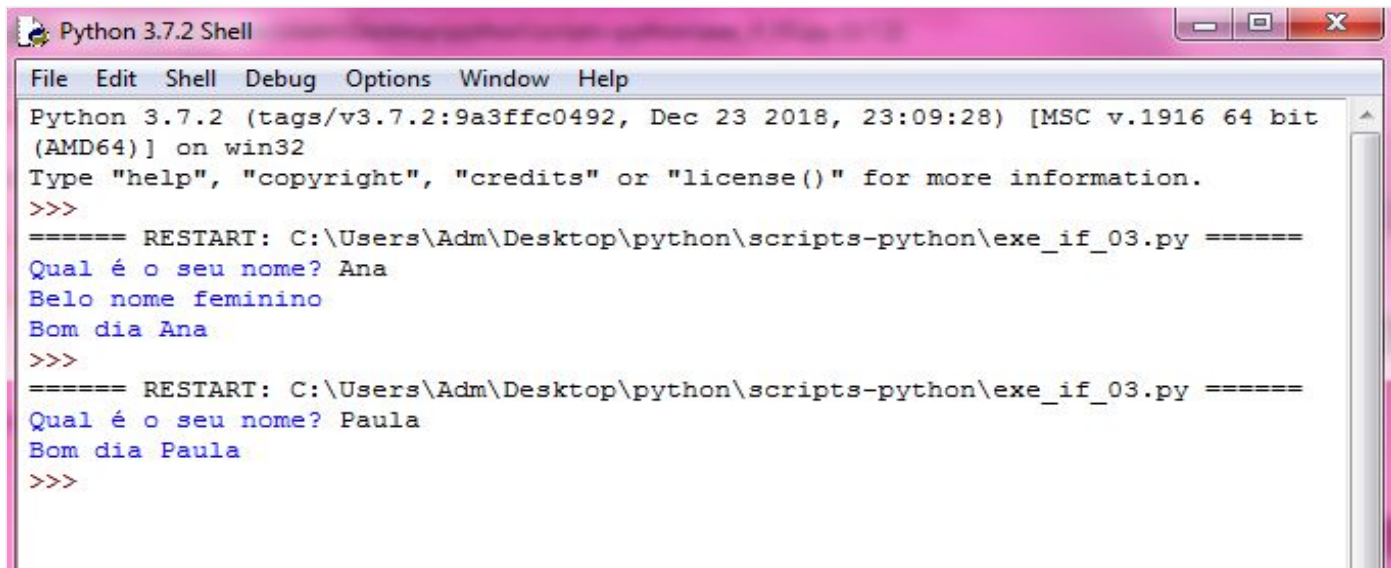
if nome=='Gustavo':
    print('Que nome bonito')
elif nome=='Pedro' or nome=='Maria' or nome=='Paulo':
    print('Seu nome é bem popular')
elif nome in 'Ana Cláudia Jéssica Juliana':
    print('Belo nome feminino')
else:
    print('Que nome sem graça')
print('Bom dia {}'.format(nome))
```



O else então se torna opcional, vamos rodar o programa sem utiliza-lo

Estruturas Condicionais

If-elif-else



```
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\exe_if_03.py =====
Qual é o seu nome? Ana
Belo nome feminino
Bom dia Ana
>>>
===== RESTART: C:\Users\Adm\Desktop\python\scripts-python\exe_if_03.py =====
Qual é o seu nome? Paula
Bom dia Paula
>>>
```

Estruturas Condicionais

If-elif-else



ATENÇÃO !

Quando estamos trabalhando em **C**, utilizamos a estrutura condicional **Switch**, porém em **Python**, ele não possui o comando **switch**, a maneira mais comum de fazer um código similar é usando **if-elif-else**.

Exercícios Práticos

- Exercício 1:
- ✓ Escreva um programa que faça o computador "pensar" em um número inteiro entre 0 e 5 e peça para o usuário tentar descobrir qual foi o número escolhido pelo computador. O programa deverá escrever na tela se o usuário venceu ou perdeu.

Exercícios Práticos

- Exercício 2:
- ✓ Faça um programa que leia a velocidade de um carro. Se ele ultrapassar 80Km/h, mostre uma mensagem dizendo que ele foi multado. A multa vai custar R\$7,00 por cada Km acima do limite.

Exercícios Práticos

- Exercício 3:
- ✓ Crie um programa que leia um número inteiro e mostre na tela se ele é PAR ou ÍMPAR.

Exercícios Práticos

- Exercício 4:
- ✓ Desenvolva um programa que pergunte a distância de uma viagem em Km. Calcule o preço da passagem, cobrando R\$0,50 por Km para viagens de até 200Km e R\$0,45 para viagens mais longas.

Exercícios Práticos

- Exercício 5:
- ✓ Faça um programa que leia um ano qualquer e mostre se ele é bissexto.

Exercícios Práticos

- Exercício 6:
- ✓ Faça um programa que leia três números e mostre qual é o maior e qual é o menor.

Exercícios Práticos

- Exercício 7:
- ✓ Escreva um programa que pergunte o salário de um funcionário e calcule o valor do seu aumento. Para salários superiores a R\$1250,00, calcule um aumento de 10%. Para os inferiores ou iguais, o aumento é de 15%.

Exercícios Práticos

- Exercício 8:
- ✓ Desenvolva um programa que leia o comprimento de três retas e diga ao usuário se elas podem ou não formar um triângulo.

Exercícios Práticos

- Exercício 9:
- ✓ Escreva um programa para aprovar o empréstimo bancário para a compra de uma casa. Pergunte o valor da casa, o salário do comprador e em quantos anos ele vai pagar. A prestação mensal não pode exceder 30% do salário ou então o empréstimo será negado.

Exercícios Práticos

- Exercício 10:
- ✓ Escreva um programa em Python que leia um número inteiro qualquer e peça para o usuário escolher qual será a base de conversão: 1 para binário, 2 para octal e 3 para hexadecimal.

Exercícios Práticos

- Exercício 11:
- ✓ Escreva um programa que leia dois números inteiros e compare-os. mostrando na tela uma mensagem:
 - O primeiro valor é maior
 - O segundo valor é maior
 - Não existe valor maior, os dois são iguais

Exercícios Práticos

- Exercício 12:
- ✓ Crie um programa que leia duas notas de um aluno e calcule sua média, mostrando uma mensagem no final, de acordo com a média atingida:
 - Média abaixo de 5.0: REPROVADO
 - Média entre 5.0 e 6.0: RECUPERAÇÃO
 - Média 6.0 ou superior: APROVADO

Exercícios Práticos

- Exercício 13:
- ✓ Crie um programa que peça 3 lados de um triângulo e diga que tipo de triângulo será formado:
 - EQUILÁTERO: todos os lados iguais
 - ISÓSCELES: dois lados iguais, um diferente
 - ESCALENO: todos os lados são diferentes.

Exercícios Práticos

- Exercício 14:
- ✓ Escreva um programa em Python que leia um número inteiro qualquer e peça para o usuário escolher qual será a base de conversão: 1 para binário, 2 para octal e 3 para hexadecimal.

Estrutura de Repetição

(Laços de Repetição)

- Laços (**loops**) são estruturas de repetição, geralmente usados para processar coleções de dados, tais como linhas de um arquivo ou registros de um banco de dados, que precisam ser processados por um mesmo bloco de código.

Estrutura de Repetição

For

- É a estrutura de repetição mais usada no Python. A instrução aceita não só sequências estáticas, mas também sequências geradas por **iteradores**;
- **Iteradores** são estruturas que permitem iterações, ou seja, acesso aos itens de uma coleção de elementos, de forma sequencial.

Estrutura de Repetição

For

- A sintaxe do laço FOR em Python é o seguinte:

for < referência > **in** < sequência >



for **x** in **range**(1,100)



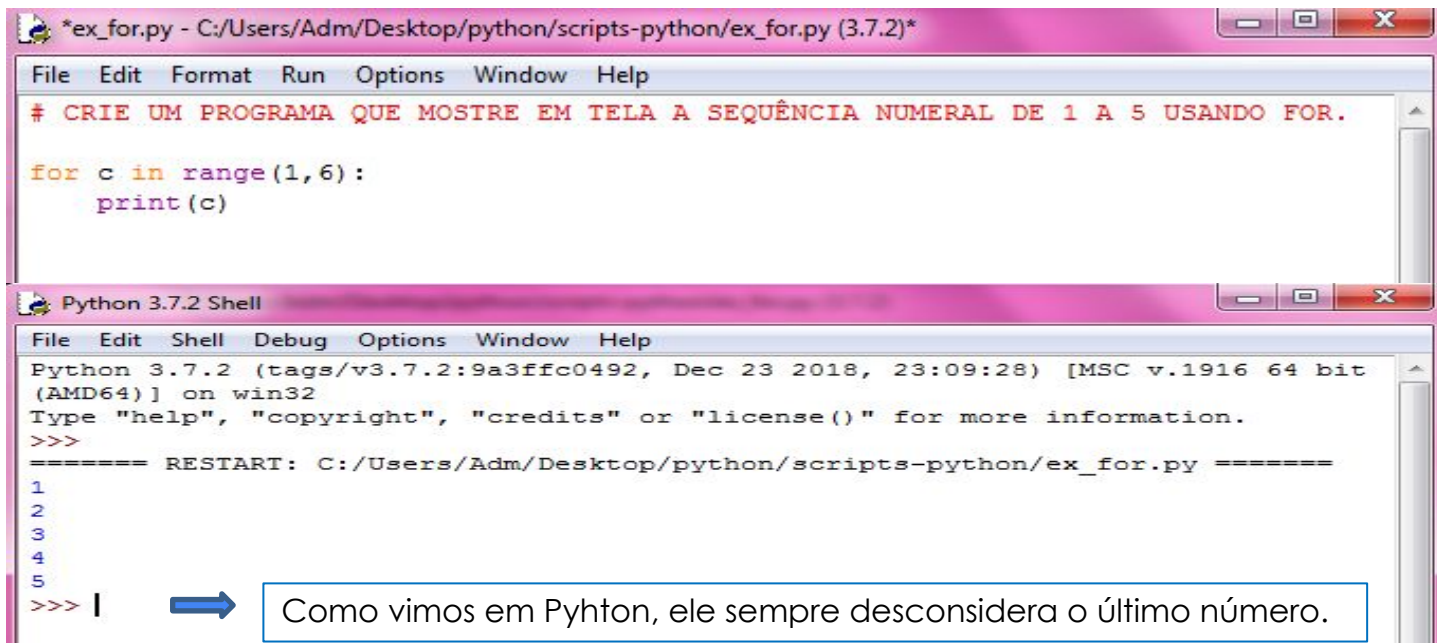
O que é range??

Estrutura de Repetição

For

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

Estrutura de Repetição For



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

```
# CRIE UM PROGRAMA QUE MOSTRE EM TELA A SEQUÊNCIA NUMERAL DE 1 A 5 USANDO FOR.  
  
for c in range(1,6):  
    print(c)
```

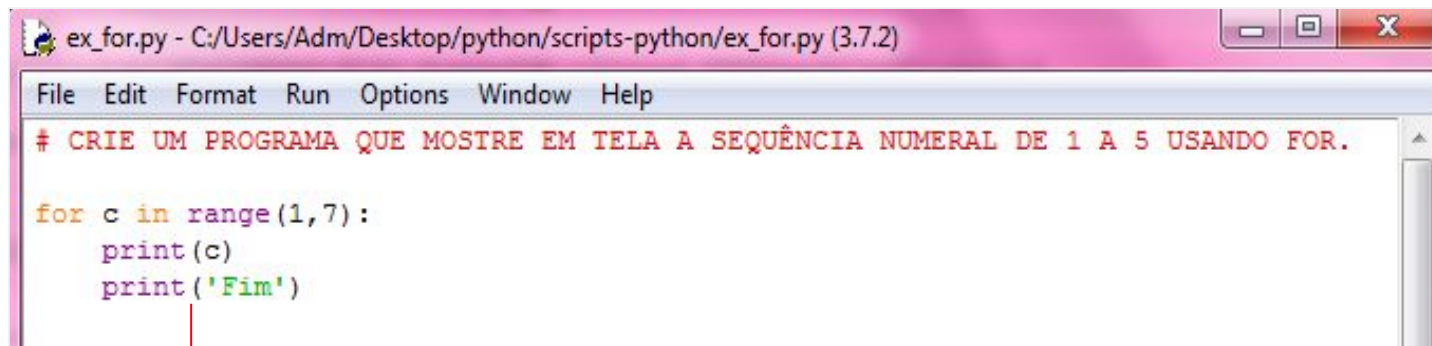
The bottom window, titled 'Python 3.7.2 Shell', shows the output of the program after execution:

```
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_for.py =====  
1  
2  
3  
4  
5  
>>> |
```

A blue arrow points from the prompt '>>> |' to a text box containing the explanation: 'Como vimos em Python, ele sempre desconsidera o último número.'

Estrutura de Repetição

For



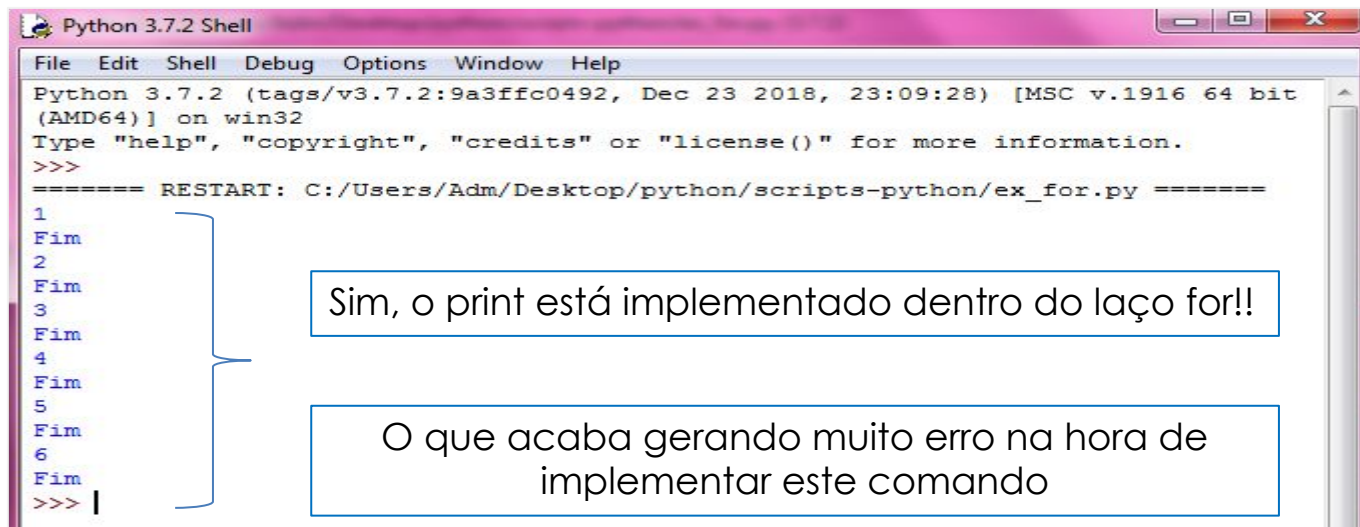
```
ex_for.py - C:/Users/Adm/Desktop/python/scripts-python/ex_for.py (3.7.2)
File Edit Format Run Options Window Help
# CRIE UM PROGRAMA QUE MOSTRE EM TELA A SEQUÊNCIA NUMERAL DE 1 A 5 USANDO FOR.

for c in range(1,7):
    print(c)
    print('Fim')
```

Este print está dentro do laço
for ?

Estrutura de Repetição

For

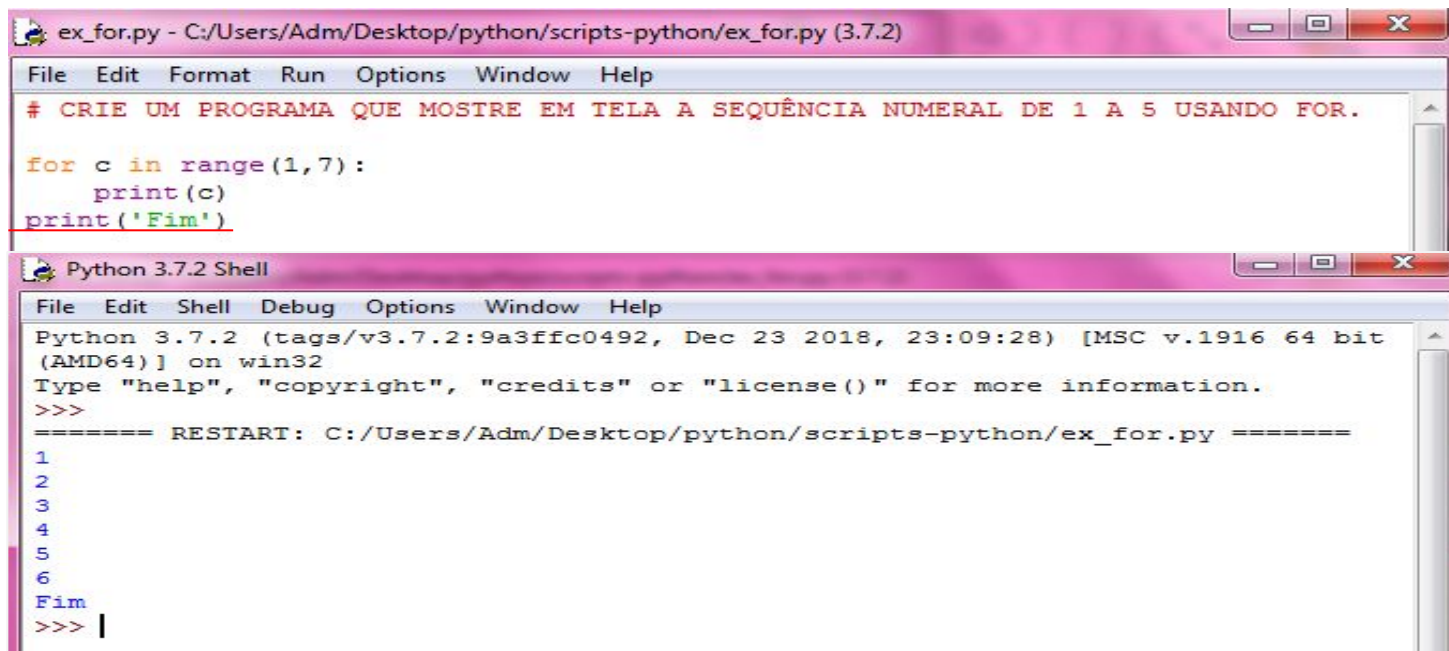


```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/ex_for.py =====
1
Fim
2
Fim
3
Fim
4
Fim
5
Fim
6
Fim
>>> |
```

Sim, o print está implementado dentro do laço for!!

O que acaba gerando muito erro na hora de implementar este comando

Estrutura de Repetição For



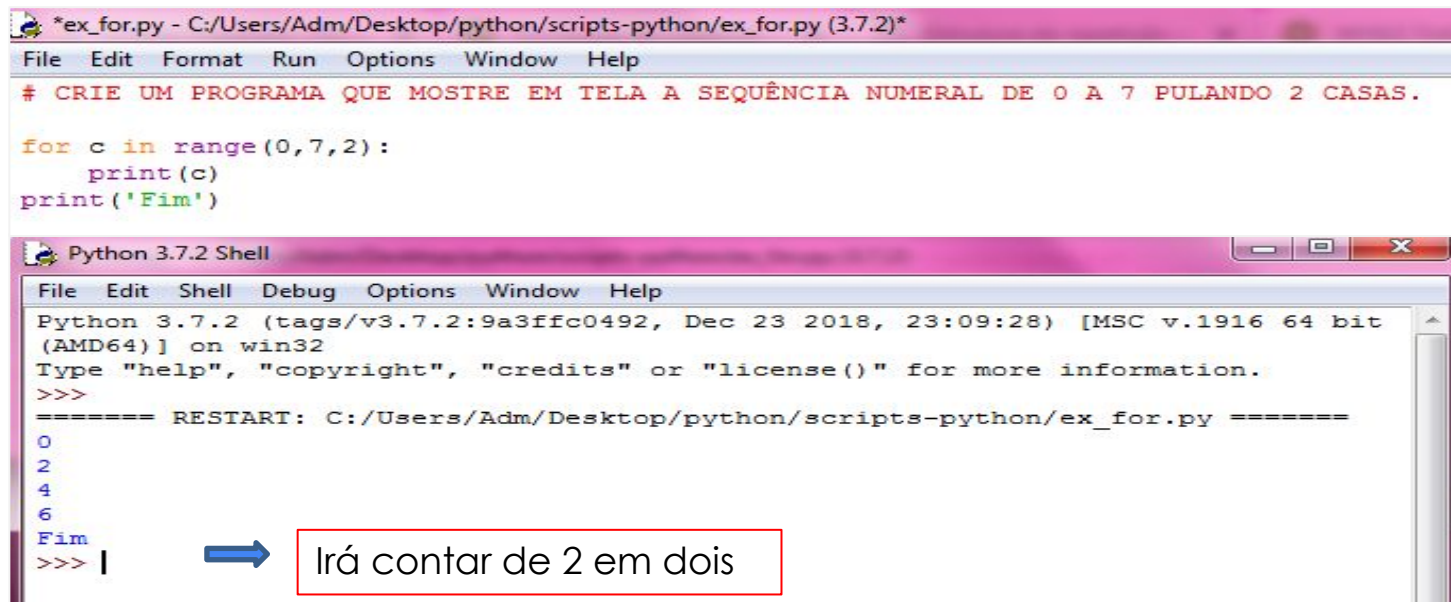
The image shows a screenshot of a Python IDE with two windows. The top window, titled 'ex_for.py - C:/Users/Adm/Desktop/python/scripts-python/ex_for.py (3.7.2)', contains the following Python code:

```
# CRIE UM PROGRAMA QUE MOSTRE EM TELA A SEQUÊNCIA NUMERAL DE 1 A 5 USANDO FOR.  
  
for c in range(1,7):  
    print(c)  
print('Fim')
```

The bottom window, titled 'Python 3.7.2 Shell', shows the output of the program after execution:

```
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_for.py =====  
1  
2  
3  
4  
5  
6  
Fim  
>>> |
```

Estrutura de Repetição For



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

```
# CRIE UM PROGRAMA QUE MOSTRE EM TELA A SEQUÊNCIA NUMERAL DE 0 A 7 PULANDO 2 CASAS.  
  
for c in range(0,7,2):  
    print(c)  
print('Fim')
```

The bottom window, titled 'Python 3.7.2 Shell', shows the output of the program after execution:

```
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_for.py =====  
0  
2  
4  
6  
Fim  
>>> |
```

An annotation is present below the shell output: a blue arrow points from the vertical bar at the end of the prompt '>>> |' to a red-bordered box containing the text 'Irá contar de 2 em dois'.

Estrutura de Repetição

While

- O comando **while** faz com que um conjunto de instruções seja executado enquanto uma condição é atendida. Quando o resultado dessa condição passa a ser falso, a execução do loop é interrompida.

Estrutura de Repetição

While

- A sintaxe do laço WHILE em Python é o seguinte:

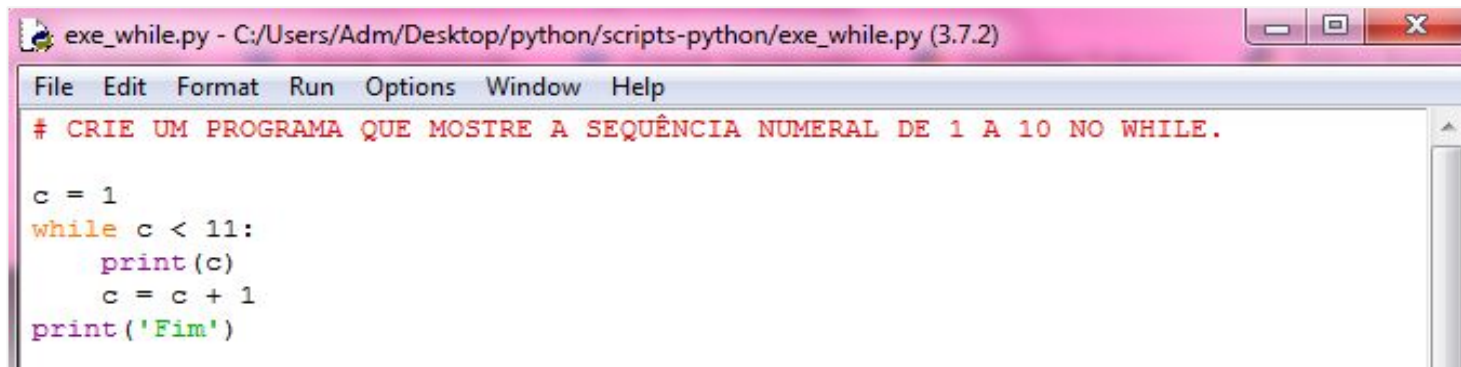
while < condição >



while x < 100

Estrutura de Repetição

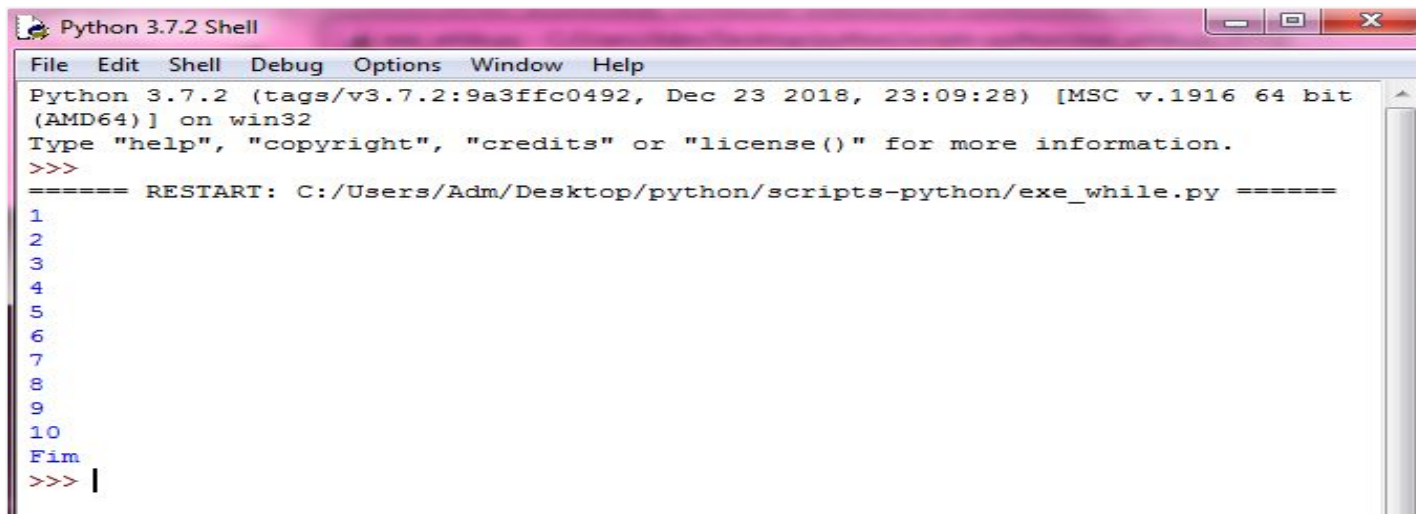
While

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

```
# CRIE UM PROGRAMA QUE MOSTRE A SEQUÊNCIA NUMERAL DE 1 A 10 NO WHILE.  
  
c = 1  
while c < 11:  
    print(c)  
    c = c + 1  
print('Fim')
```

Estrutura de Repetição

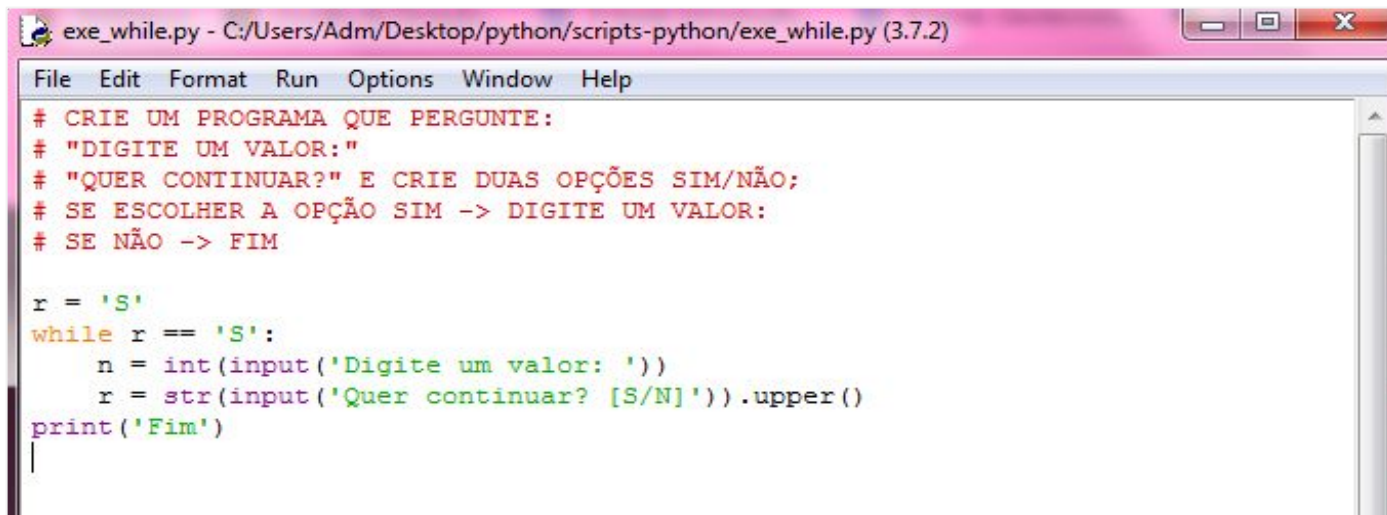
While



```
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/exe_while.py =====
1
2
3
4
5
6
7
8
9
10
Fim
>>> |
```

Estrutura de Repetição

While

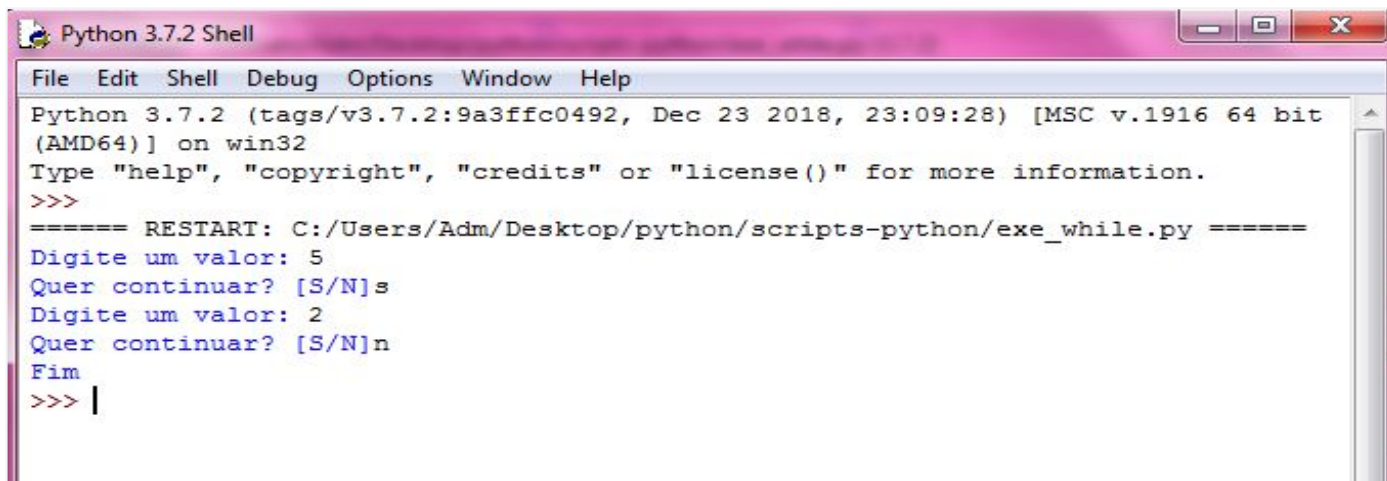


The screenshot shows a Python IDE window titled "exe_while.py - C:/Users/Adm/Desktop/python/scripts-python/exe_while.py (3.7.2)". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code editor contains the following Python code:

```
# CRIE UM PROGRAMA QUE PERGUNTE:  
# "DIGITE UM VALOR:"  
# "QUER CONTINUAR?" E CRIE DUAS OPÇÕES SIM/NÃO;  
# SE ESCOLHER A OPÇÃO SIM -> DIGITE UM VALOR:  
# SE NÃO -> FIM  
  
r = 'S'  
while r == 'S':  
    n = int(input('Digite um valor: '))  
    r = str(input('Quer continuar? [S/N]')).upper()  
print('Fim')
```

Estrutura de Repetição

While



```
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/exe_while.py =====
Digite um valor: 5
Quer continuar? [S/N)s
Digite um valor: 2
Quer continuar? [S/N)n
Fim
>>> |
```

Exercícios Práticos

- Exercício 15:
- ✓ Faça um programa que mostre na tela uma contagem numeral de 10 até 0.

Exercícios Práticos

- Exercício 16:
- ✓ Crie um programa que mostre na tela todos os números pares que estão no intervalo entre 1 e 50.

Exercícios Práticos

- Exercício 17:
- ✓ Faça um programa que calcule a soma entre todos os números que são múltiplos de três e que se encontram no intervalo de 1 até 500.

Exercícios Práticos

- Exercício 18:
- ✓ Faça um programa que mostre a tabuada de um número que o usuário escolher utilizando um laço for.

Exercícios Práticos

- Exercício 19:
- ✓ Desenvolva um programa que leia seis números inteiros e mostre a soma apenas daqueles que forem pares. Se o valor digitado for ímpar, desconsidere-o.

Exercícios Práticos

- Exercício 20:
- ✓ Faça um programa que leia um número inteiro e diga se ele é ou não um número primo.

Exercícios Práticos

- Exercício 21:
- ✓ Faça um programa que leia o peso de cinco pessoas. No final, mostre qual foi o maior e o menor peso lidos.

Exercícios Práticos

- Exercício 22:
- ✓ Desenvolva um programa que leia o nome, idade e sexo de 4 pessoas. No final do programa, mostre: a média de idade do grupo, qual é o nome do homem mais velho e quantas mulheres têm menos de 20 anos.

Exercícios Práticos

- Exercício 23:
- ✓ Faça um programa que leia o sexo de uma pessoa, mas só aceite valores 'Masculino' ou 'Feminino'. Caso esteja errado apareça a opção 'inválida'.

Exercícios Práticos

- Exercício 24:

✓ Crie um programa que leia dois valores e mostre um menu na tela:

- [1] somar
- [2] multiplicar
- [3] maior
- [4] novos números
- [5] sair do programa

Seu programa deverá realizar a operação solicitada em cada caso.

Exercícios Práticos

- Exercício 25:

- ✓ Faça um programa que leia um número qualquer e mostre o seu fatorial.

Ex: $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

Exercícios Práticos

- Exercício 26:
- ✓ Escreva um programa que leia um número N inteiro qualquer e mostre na tela os N primeiros elementos de uma Sequência de Fibonacci.