

Variáveis e tipos de dados

Aprendendo mais sobre strings

Fatiamento

Fatiamento (slicing) é o recurso de extrair partes de uma string utilizando o índice de seus caracteres.

`string[x:y]`

O "X" indica a
posição de início
da fatia

O "Y" indica a
posição de fim da
fatia.

Fatiamento

A posição inicial (x) é incluída no resultado, porém, a posição final (y) não é incluída.

```
endereco = "www.python.org"
# www . python . org
# 012 3 456789 0 123
fatia = endereco[4:10]
print(fatia)
```

O resultado obtido com o print foi o texto "python". Observe os índices fornecidos:
4 = posição da letra "p"
10 = posição do ".", como o caractere da décima posição não é incluído, o valor retornado é do "p" até o "n", que é o nono caractere.

Fatiamento

Se os índices de limite inicial e final não forem informados, é realizada apenas uma cópia da string.

```
endereco = "www.python.org"  
# www . python . org  
# 012 3 456789 0 123  
fatia = endereco[:]  
print(fatia)
```

O conteúdo da variável fatia exibido com o print é www.python.org.

Fatiamento

Se for omitido o índice inicial será considerado à partir do primeiro caractere.

```
endereco = "www.python.org"
# www . python . org
# 012 3 456789 0 123
fatia = endereco[:3]
print(fatia)
```

O conteúdo da variável *fatia* exibido com o print é “www”, ou seja, do índice zero até o 3 (retornando até o índice 2, não considerando o índice 3).

Fatiamento

Se for omitido o índice final será considerado até o último caractere (considerando o último).

```
endereco = "www.python.org"  
# www . python . org  
# 012 3 456789 0 123  
fatia = endereco[11:]  
print(fatia)
```

O conteúdo da variável *fatia* exibido com o print é "org", ou seja, do índice 11 até o último caractere (retornando o 11, 12 e 13).

Fatiamento

É possível utilizar um índice negativo para iniciar no último caractere da string. O último caractere é o -1.

```
# Usando índice negativo para o final da string
#   w   w   w   .   p   y   t   h   o   n   .   o   r   g
# -14 -13 -12 -11 -10 -9  -8  -7  -6  -5  -4  -3  -2  -1
endereco = "www.python.org"
fatia = endereco[-10:-4]
print(fatia)
```

O conteúdo da variável fatia exibido com o print é "python".

Fatiamento

Exemplo prático 1: Obtendo o usuário de um endereço de e-mail.

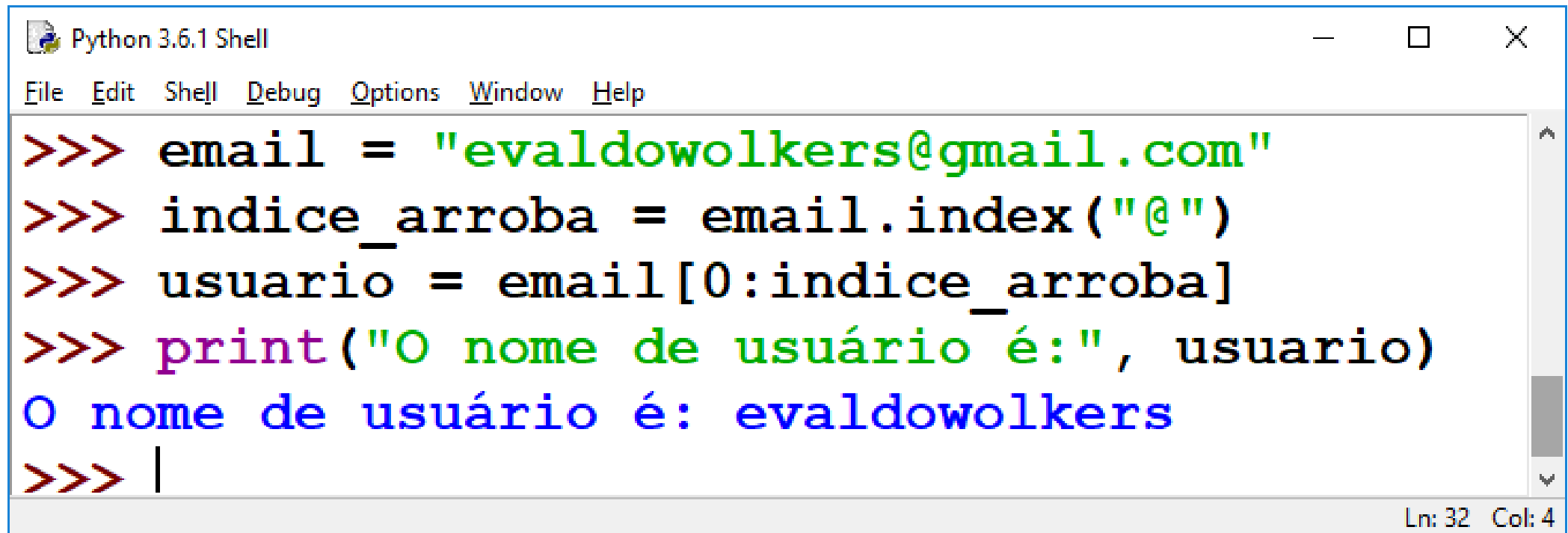
Como todo endereço de e-mail possui um arroba após o nome do usuário, podemos identificar a posição do arroba dentro do endereço de e-mail usando um método da string denominado “index”. Para encontrar a primeira ocorrência de um caractere ou texto dentro da string usamos:

```
string.index("a") # Caractere  
string.index("Python") # Texto
```


Fatiamento

```
email = "evaldowolkers@gmail.com"
        #01234567890123456789012
indice = email.index(".com")
print(indice)  # impresso 19
indice = email.index("@gmail")
print(indice)  # impresso 13
indice = email.index("@")
print(indice)  # impresso 13
```

Fatiamento



```
Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
>>> email = "evaldowolkers@gmail.com"
>>> indice_arroba = email.index("@")
>>> usuario = email[0:indice_arroba]
>>> print("O nome de usuário é:", usuario)
O nome de usuário é: evaldowolkers
>>> |
```

Ln: 32 Col: 4

Fatiamento

Exemplo prático 2: Obtendo o provedor de um endereço de e-mail.

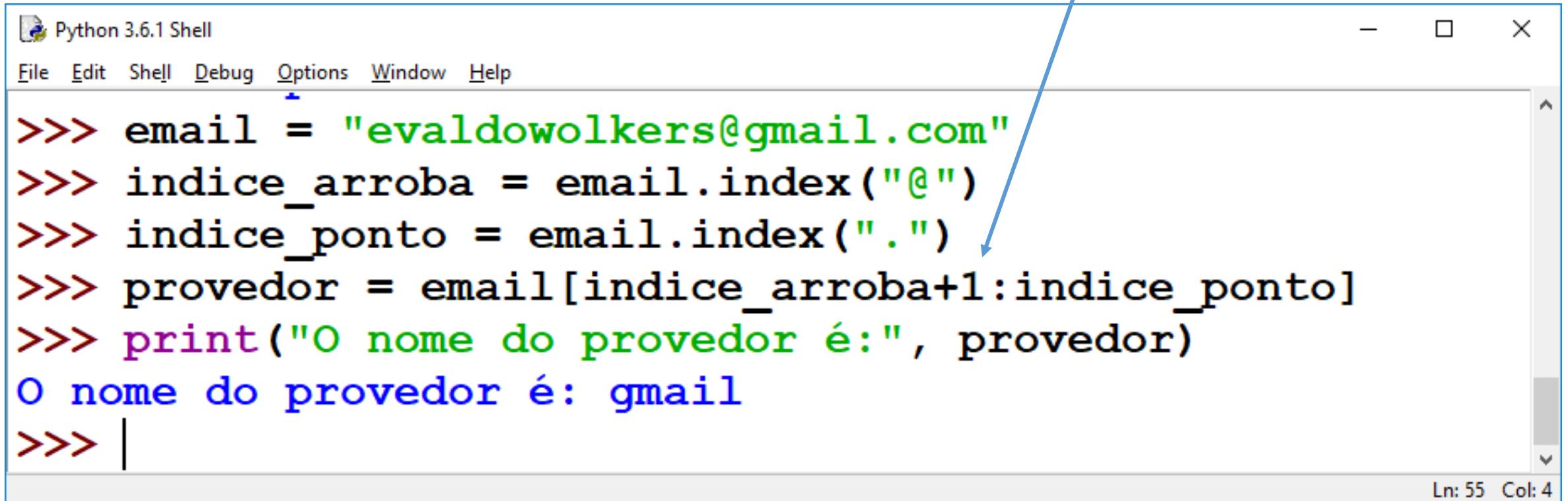
O provedor é a informação que fica entre o arroba e um ponto.

Exemplo:

[fulano@gmail.com](#) (o provedor é o **gmail**)

Fatiamento

Foi incrementado 1 para não pegar o arroba.

A screenshot of a Python 3.6.1 Shell window. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main area contains a Python script that defines an email string, finds the indices of '@' and '.', and then slices the string to extract the provider name. The output shows 'O nome do provedor é: gmail'. A blue arrow points from the text 'Foi incrementado 1 para não pegar o arroba.' to the '+1' in the slicing operation. The status bar at the bottom right shows 'Ln: 55 Col: 4'.

```
>>> email = "evaldowolkers@gmail.com"
>>> indice_arroba = email.index("@")
>>> indice_ponto = email.index(".")
>>> provedor = email[indice_arroba+1:indice_ponto]
>>> print("O nome do provedor é:", provedor)
O nome do provedor é: gmail
>>> |
```

Fatiamento

E se o e-mail for assim:

joao.silva@gmail.com

?

Fatiamento

```
Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
>>> email = "joao.silva@gmail.com"
>>> indice_arroba = email.index("@")
>>> indice_ponto = email.index(".")
>>> provedor = email[indice_arroba+1:indice_ponto]
>>> print("O nome do provedor é:", provedor)
O nome do provedor é:
>>> |
```

Ln: 49 Col: 4

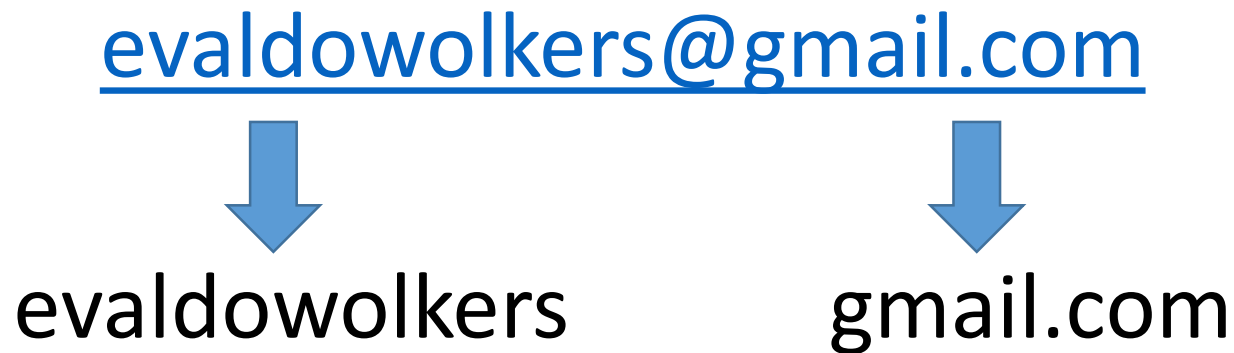
Nada foi impresso porque o índice final ficou menor que o índice inicial. email[10:4]
Vamos ver como resolver isso.

Fatiamento

Que tal aprendermos um novo método para “quebrar” uma string à partir de um determinado caractere.

Veja:

evaldowolkers@gmail.com



evaldowolkers gmail.com

Fatiamento

Para isso podemos usar o método split.

O método split retorna uma lista de palavras usando um separador informado por parâmetro.

Obs.: Em aulas futuras irei falar sobre listas em Python.
Nesta aula, entenda que o e-mail será quebrado em
das informações.

Fatiamento

Veja:

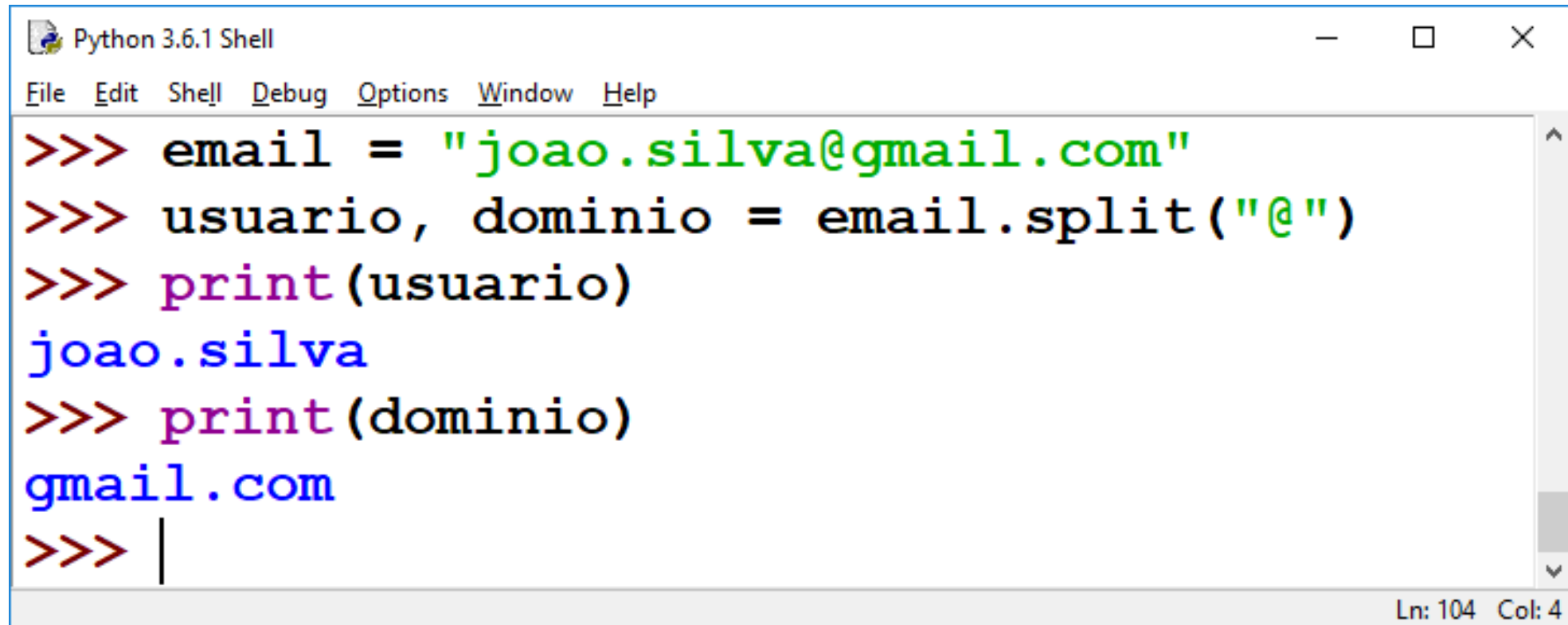
```
string.split("separador")
```

No nosso caso usaremos assim:

```
"joao.silva@gmail.com".split("@")
```

Serão retornadas duas strings, uma contendo "joao.silva" e a outra contendo "gmail.com".

Fatiamento

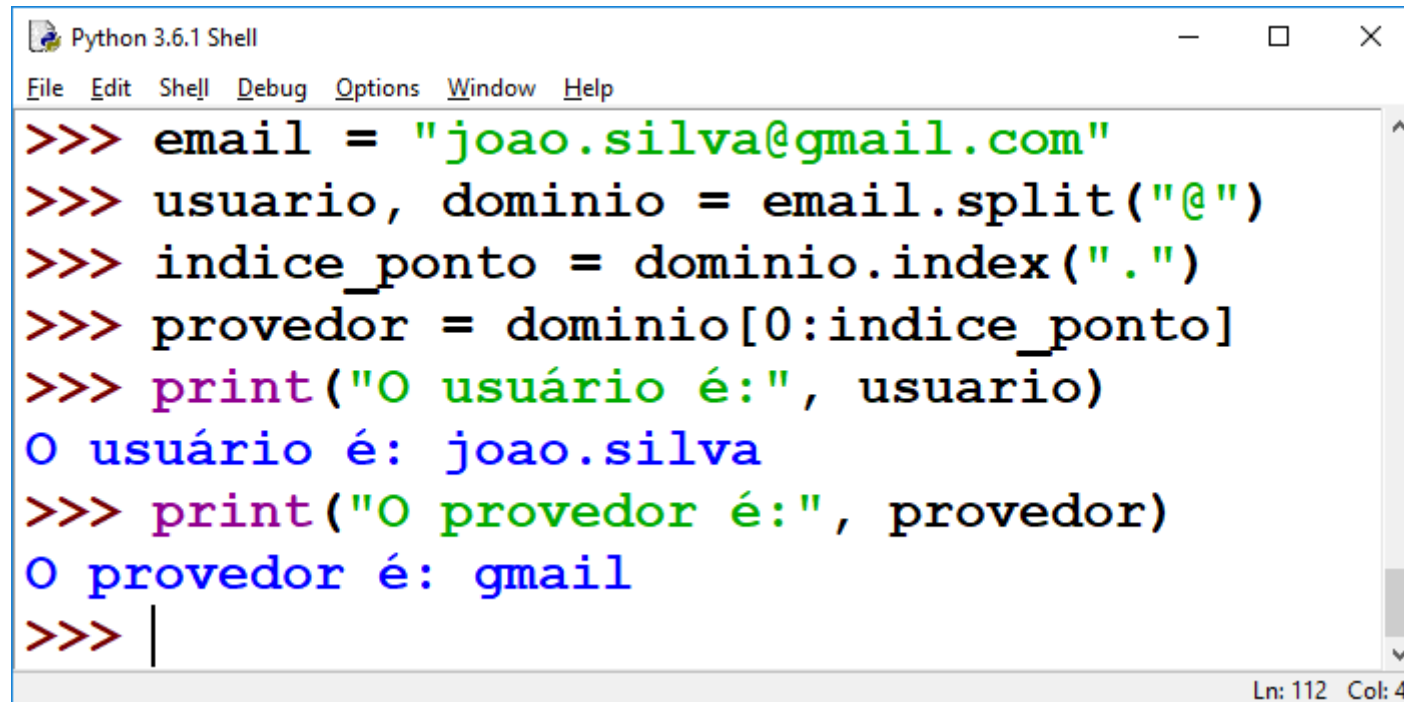
A screenshot of a Python 3.6.1 Shell window. The window has a title bar with the text "Python 3.6.1 Shell" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with options: File, Edit, Shell, Debug, Options, Window, and Help. The main area of the window contains a Python script. The script defines a variable 'email' with the value "joao.silva@gmail.com", then splits it at the '@' symbol into 'usuario' and 'dominio'. It then prints 'usuario', which outputs 'joao.silva', and prints 'dominio', which outputs 'gmail.com'. The cursor is at the end of the last line of code. The status bar at the bottom right shows "Ln: 104 Col: 4".

```
Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
>>> email = "joao.silva@gmail.com"
>>> usuario, dominio = email.split("@")
>>> print(usuario)
joao.silva
>>> print(dominio)
gmail.com
>>> |
```

Ln: 104 Col: 4

Fatiamento

Agora que temos o domínio separado podemos procurar o “.” para indentificá-lo e separar o provedor.

A screenshot of a Python 3.6.1 Shell window. The window has a title bar with the text "Python 3.6.1 Shell" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with options: File, Edit, Shell, Debug, Options, Window, and Help. The main area of the window contains a Python script. The script defines a variable 'email' with the value "joao.silva@gmail.com". It then splits 'email' at the '@' symbol into 'usuario' and 'dominio'. It finds the index of the first '.' in 'dominio' and uses it to slice 'dominio' into 'provedor'. Finally, it prints the user and provider names. The output shows "O usuário é: joao.silva" and "O provedor é: gmail". The prompt ">>>" is followed by a vertical bar, indicating the cursor is at the end of the last line.

```
>>> email = "joao.silva@gmail.com"
>>> usuario, dominio = email.split("@")
>>> indice_ponto = dominio.index(".")
>>> provedor = dominio[0:indice_ponto]
>>> print("O usuário é:", usuario)
O usuário é: joao.silva
>>> print("O provedor é:", provedor)
O provedor é: gmail
>>> |
```

Ln: 112 Col: 4

FIM