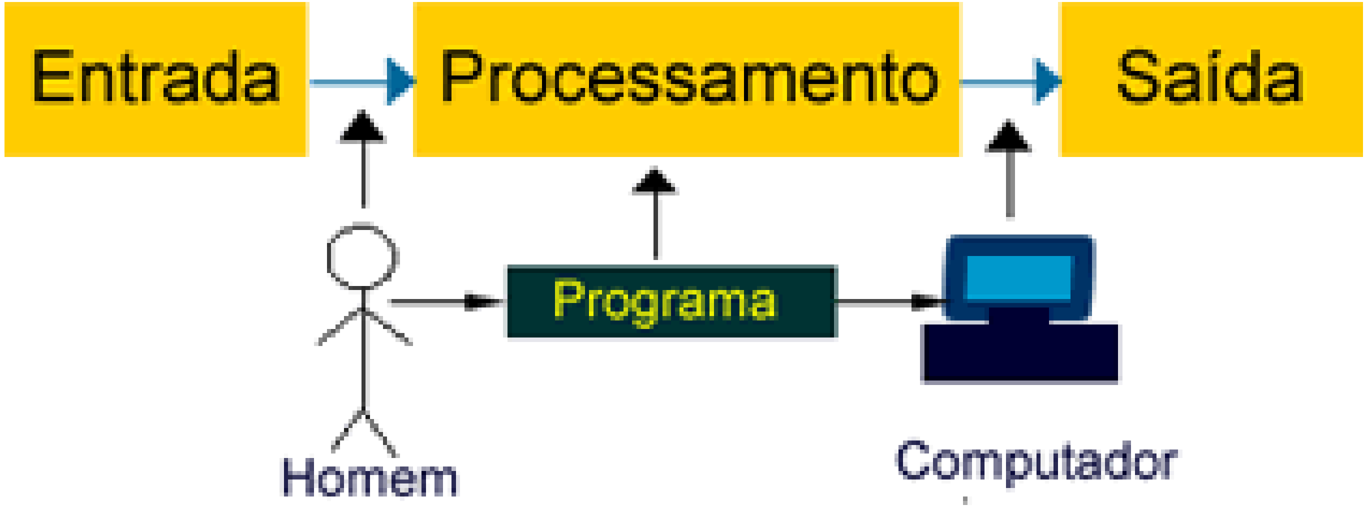
Curso Python e Django

Francisco André
fandrefh@gmail.com
Senac/PI

GO AHEAD!

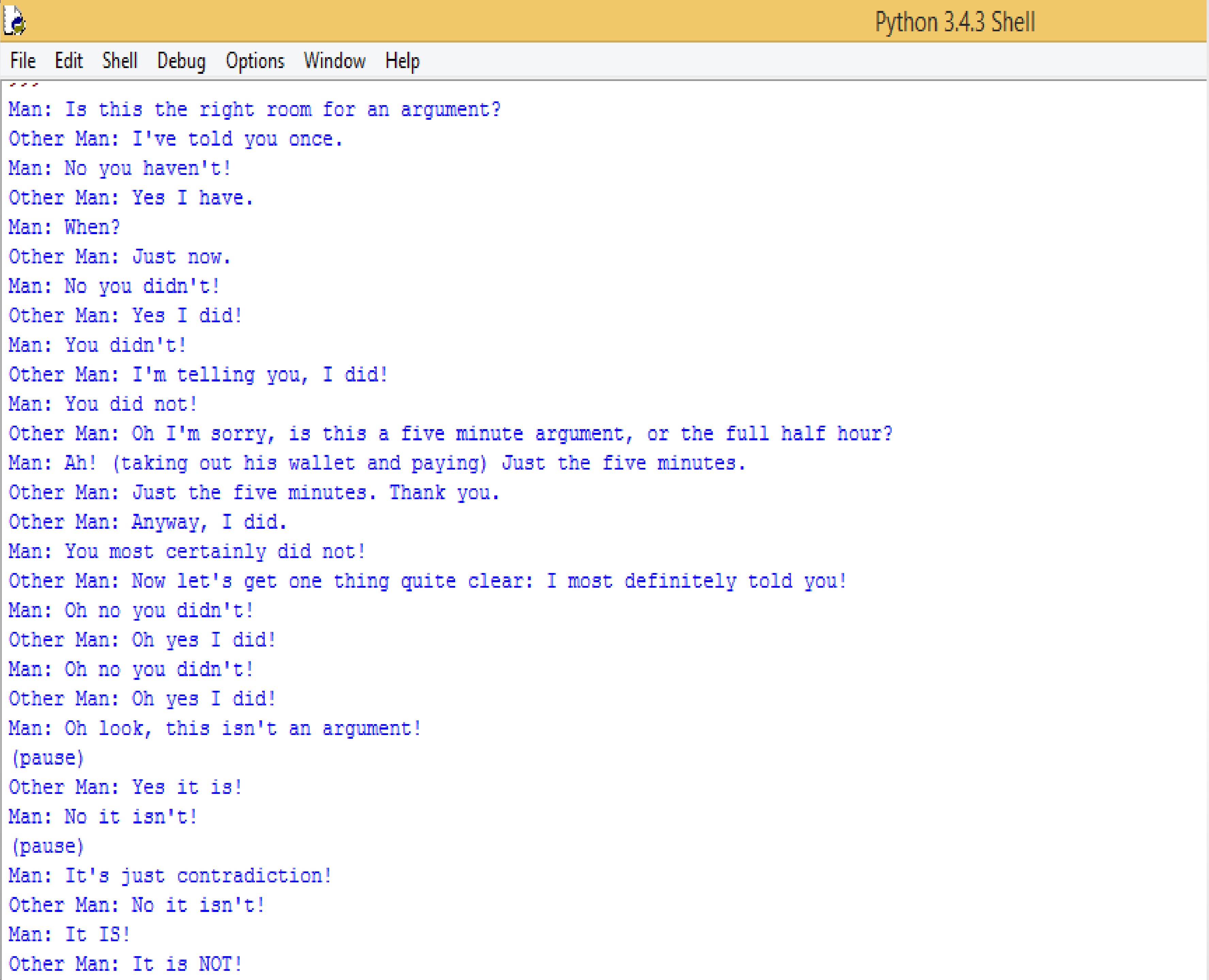
Tratamento de exceções

Tratamento de exceções significa prevê situações que possam interromper seu programa bruscamente. Isto acontece sempre que surge algo inesperado e seu programa simplesmente não sabe lidar com a situação.
Devemos então "ensinar" nossos programas a lidarem com estes erros, usando as *exceções*.


Entrada
Processamento
Saída
Programa
Homem
Computador

Lidando com os dados...

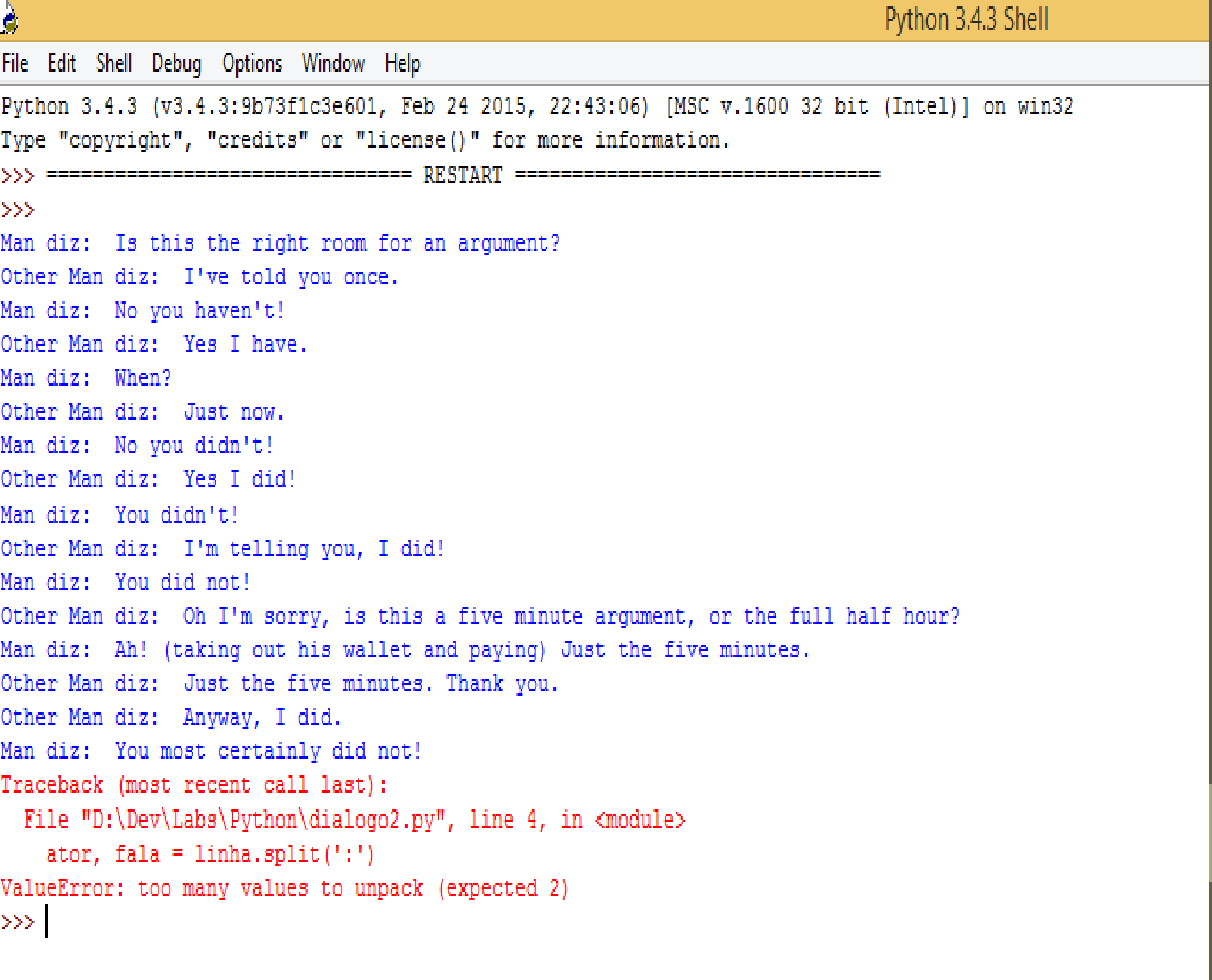
```
dialogo = open('dialogo.txt')
dialogo.seek(0)
for linha in dialogo:
   print(linha, end='')
dialogo.close( )
```



```
'''
Man: Is this the right room for an argument?
Other Man: I've told you once.
Man: No you haven't!
Other Man: Yes I have.
Man: When?
Other Man: Just now.
Man: No you didn't!
Other Man: Yes I did!
Man: You didn't!
Other Man: I'm telling you, I did!
Man: You did not!
Other Man: Oh I'm sorry, is this a five minute argument, or the full half hour?
Man: Ah! (taking out his wallet and paying) Just the five minutes.
Other Man: Just the five minutes. Thank you.
Other Man: Anyway, I did.
Man: You most certainly did not!
Other Man: Now let's get one thing quite clear: I most definitely told you!
Man: Oh no you didn't!
Other Man: Oh yes I did!
Man: Oh no you didn't!
Other Man: Oh yes I did!
Man: Oh look, this isn't an argument!
(pause)
Other Man: Yes it is!
Man: No it isn't!
(pause)
Man: It's just contradiction!
Other Man: No it isn't!
Man: It IS!
Other Man: It is NOT!
```

Lidando com os dados...

```
dialogo = open('dialogo.txt')
dialogo.seek(0)
for linha in dialogo:
    ator, fala = linha.split(':')
    print(ator, end='')
    print(' diz: ', end='')
    print(fala, end='')
dialogo.close( )
```



```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>> 
Man diz:  Is this the right room for an argument?
Other Man diz:  I've told you once.
Man diz:  No you haven't!
Other Man diz:  Yes I have.
Man diz:  When?
Other Man diz:  Just now.
Man diz:  No you didn't!
Other Man diz:  Yes I did!
Man diz:  You didn't!
Other Man diz:  I'm telling you, I did!
Man diz:  You did not!
Other Man diz:  Oh I'm sorry, is this a five minute argument, or the full half hour?
Man diz:  Ah! (taking out his wallet and paying) Just the five minutes.
Other Man diz:  Just the five minutes. Thank you.
Other Man diz:  Anyway, I did.
Man diz:  You most certainly did not!
Traceback (most recent call last):
  File "D:\Dev\Labs\Python\dialogo2.py", line 4, in <module>
    ator, fala = linha.split(':')
ValueError: too many values to unpack (expected 2)
>>> 
```

Lidando com os dados...

```
>>> texto = 'string'
>>> dir(texto)
[... 'split', ....]
```

Lidando com os dados...

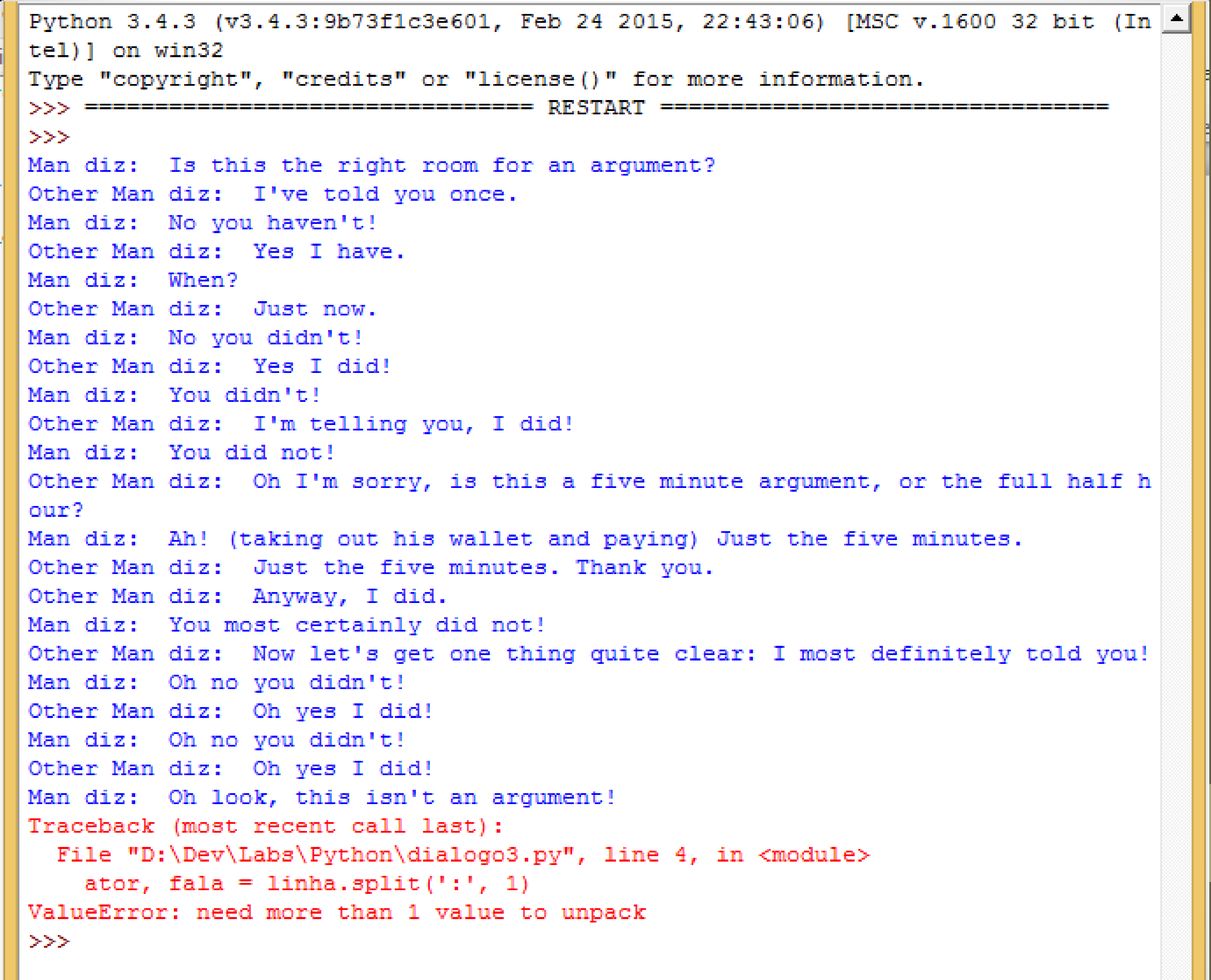
```
>>> help(texto.split)
Help on built-in function split:

split(...) method of builtins.str instance
    S.split(sep=None, maxsplit=-1) -> list of strings

    Return a list of the words in S, using sep as the
    delimiter string.  If maxsplit is given, at most maxsplit
    splits are done. If sep is not specified or is None, any
    whitespace string is a separator and empty strings are
    removed from the result.
```

Lidando com os dados...

```
dialogo = open('dialogo.txt')
dialogo.seek(0)
for linha in dialogo:
    ator, fala = linha.split(':', 1)
    print(ator, end='')
    print(' diz: ', end='')
    print(fala, end='')
dialogo.close( )
```



```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>> 
Man diz:  Is this the right room for an argument?
Other Man diz:  I've told you once.
Man diz:  No you haven't!
Other Man diz:  Yes I have.
Man diz:  When?
Other Man diz:  Just now.
Man diz:  No you didn't!
Other Man diz:  Yes I did!
Man diz:  You didn't!
Other Man diz:  I'm telling you, I did!
Man diz:  You did not!
Other Man diz:  Oh I'm sorry, is this a five minute argument, or the full half h
our?
Man diz:  Ah! (taking out his wallet and paying) Just the five minutes.
Other Man diz:  Just the five minutes. Thank you.
Other Man diz:  Anyway, I did.
Man diz:  You most certainly did not!
Other Man diz:  Now let's get one thing quite clear: I most definitely told you!
Man diz:  Oh no you didn't!
Other Man diz:  Oh yes I did!
Man diz:  Oh no you didn't!
Other Man diz:  Oh yes I did!
Man diz:  Oh look, this isn't an argument!
Traceback (most recent call last):
  File "D:\Dev\Labs\Python\dialogo3.py", line 4, in <module>
    ator, fala = linha.split(':', 1)
ValueError: need more than 1 value to unpack
>>> 
```

Lidando com os dados...

```
>>> texto = 'string'
>>> dir(texto)
[... ‘find', ....]
```

Lidando com os dados...

```
>>> help(texto.find)
Help on built-in function find:

find(...) method of builtins.str instance
    S.find(sub[, start[, end]]) -> int

    Return the lowest index in S where substring sub is found,
    such that sub is contained within S[start:end].  Optional
    arguments start and end are interpreted as in slice notation.

    Return -1 on failure.
```

Lidando com os dados...

```
>>> frase = 'Eu disse a você, que não existe nada igual a Python!'
>>> frase.find(':')
-1
```

Lidando com os dados...

```
>>> frase = 'Eu disse a você: que não existe nada
igual a Python!'
>>> frase.find(':')
15
```

Lidando com os dados...

```
dialogo = open('dialogo.txt')
dialogo.seek(0)
for linha in dialogo:
    if not linha.find(':') == -1:
        ator, fala = linha.split(':', 1)
        print(ator, end='')
        print(' diz: ', end='')
        print(fala, end='')
dialogo.close( )
```

Tente primeiro, depois recupere...

Ao invés de adicionar lógica extras e tentar prevenir os mais diversos tipos de erros e situações inesperadas, podemos usar a abordagem de “tentar”, e se tudo estiver certo, “seguir” em frente, caso algo esteja errado, “trate” os erros e só então “prossiga” com o fluxo do programa.

try/except...

A instrução *try* existe para fornecer uma maneira sistemática de lidar com as exceções e evitar paradas bruscas no programa.

Sintaxe geral:

```
try:
   #código a ser “experimentado”...
except:
   #código a ser executado caso o “experimento” não tenha dado certo
```

E se o arquivo não existir?

```
dialogo = open('dialogoss.txt')
dialogo.seek(0)
for linha in dialogo:
    try:
        ator, fala = linha.split(':', 1)
        print(ator, end='')
        print(' diz: ', end='')
        print(fala, end='')
    except:
        pass
dialogo.close( )
```

Você ganha um FileNotFound...

```
>>>
Traceback (most recent call last):
  File "D:/Dev/Labs/Python/dialogo5.py", line 1, in
<module>
    dialogo = open('dialogoss.txt')
FileNotFoundError: [Errno 2] No such file or
directory: 'dialogoss.txt'
```

Então você pensa...

Eu conheço o módulo *os*!

```
import os
if os.path.exists('dialogoss.txt'):
   dialogo = open('dialogoss.txt')
   dialogo.seek(0)
   for linha in dialogo:
      try:
         ator, fala = linha.split(':', 1)
         print(ator, end='')
         print(' diz: ', end='')
         print(fala, end='')
      except:
         pass
   dialogo.close()
else:
   print('O arquivo não existe.')
```

Ou...

Uma camada extra de try/except…

```
try:
    dialogo = open('dialogoss.txt')
    dialogo.seek(0)
    for linha in dialogo:
        try:
            ator, fala = linha.split(':', 1)
            print(ator, end='')
            print(' diz: ', end='')
            print(fala, end='')
        except:
            pass
    dialogo.close()
except:
    print('O arquivo não existe.')
```

Seja mais específico...

```
try:
    dialogo = open('dialogoss.txt')
    dialogo.seek(0)
    for linha in dialogo:
        try:
            ator, fala = linha.split(':', 1)
            print(ator, end='')
            print(' diz: ', end='')
            print(fala, end='')
        except ValueError:
            pass
    dialogo.close( )
except FileNotFoundError: #Ou IOError
    print('O arquivo não existe.')
```

That's all Folks!