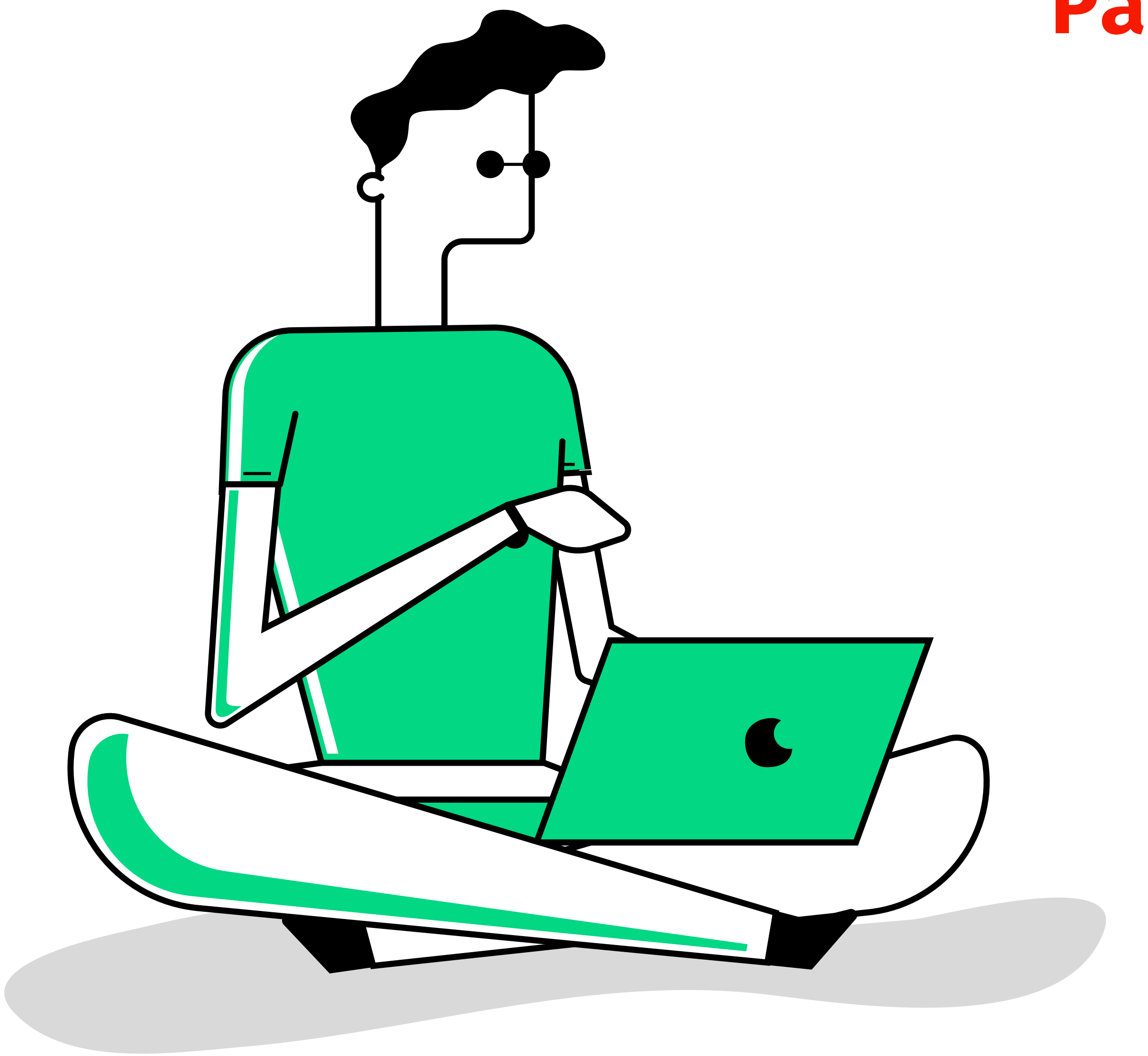
DevJunction.in

# Python One Liners

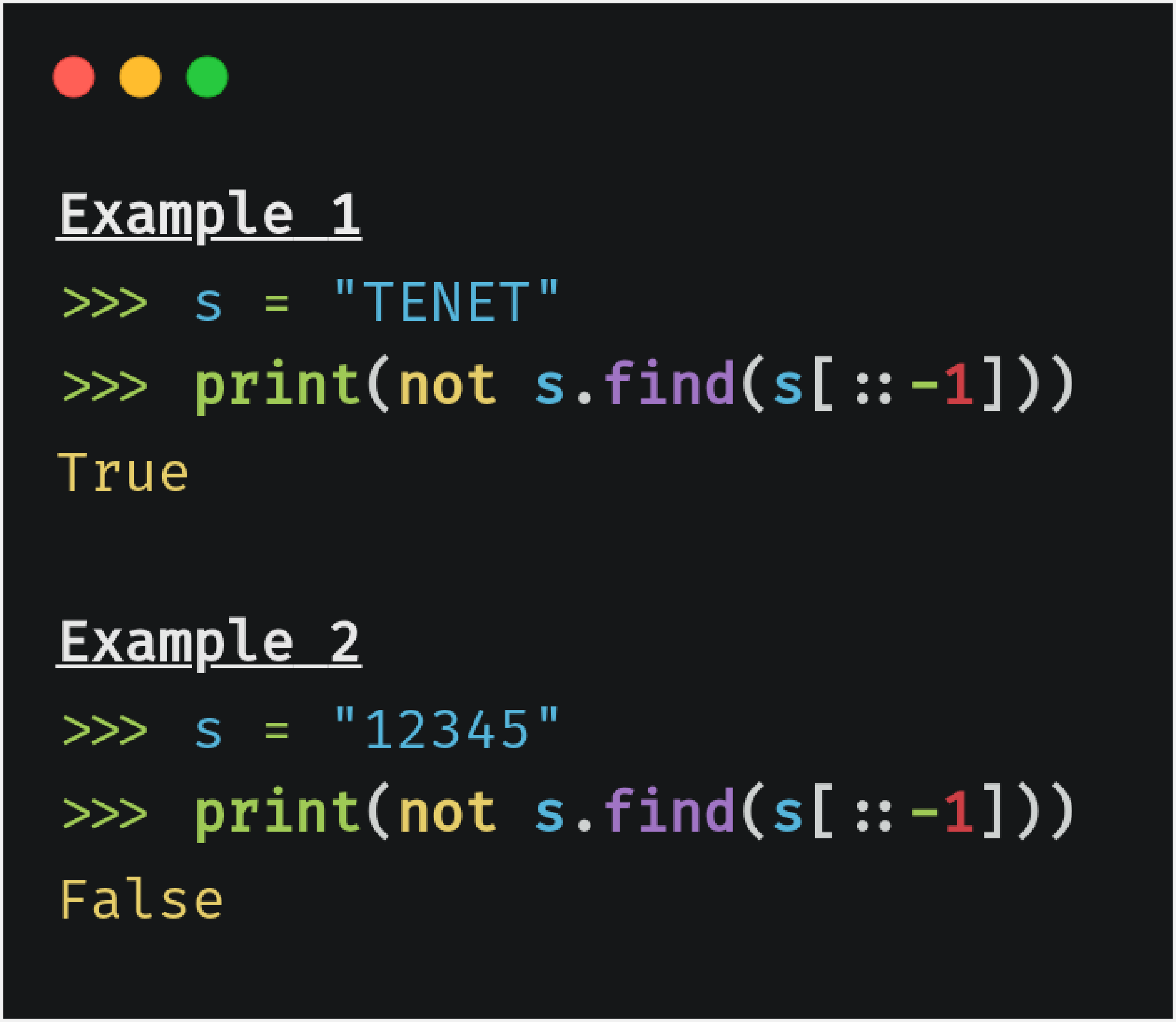
## Part-II



👍 ❤

Swipe →

# one line palindrome checker

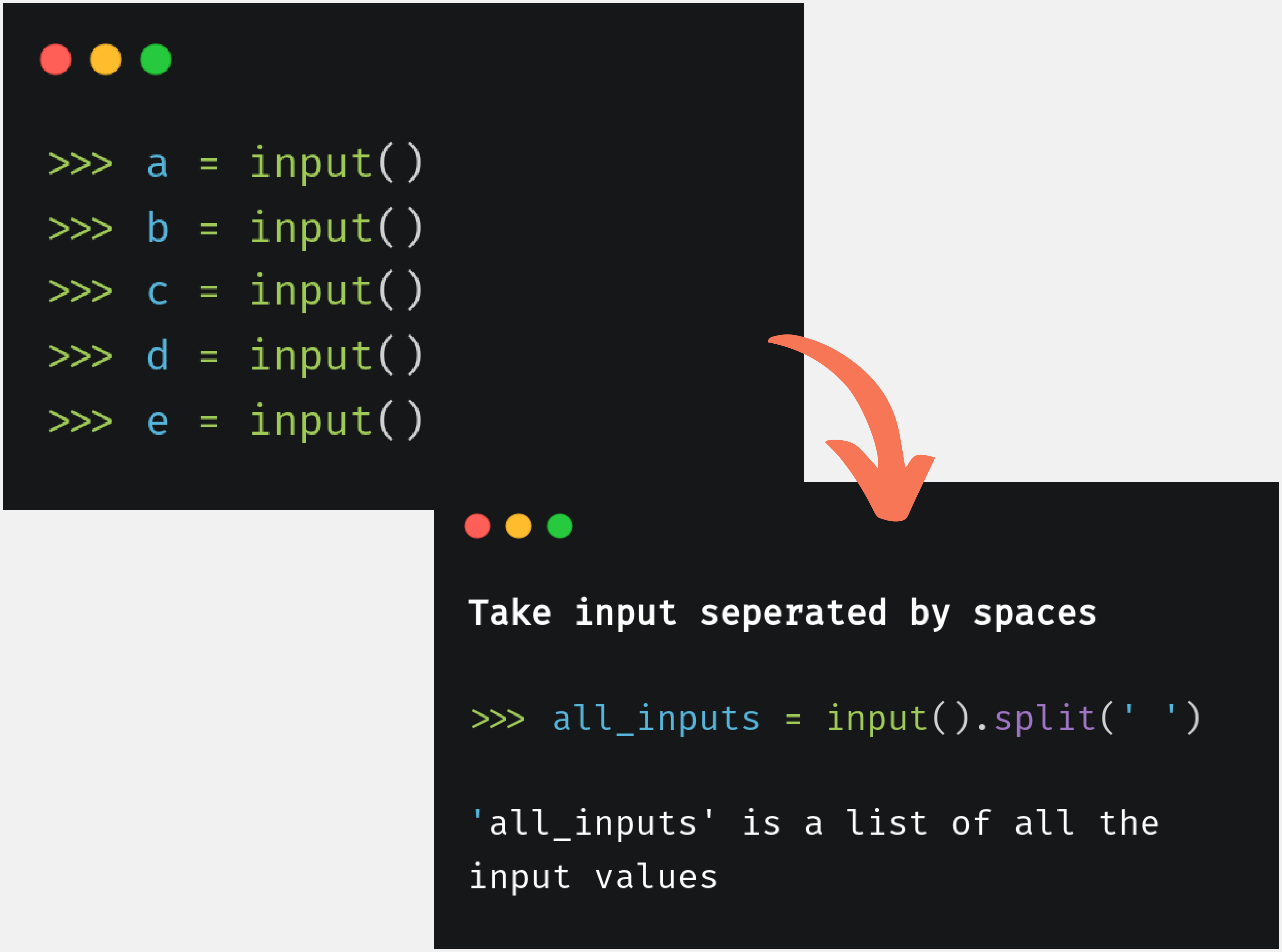


```
Example 1
>>> s = "TENET"
>>> print(not s.find(s[::-1]))
True

Example 2
>>> s = "12345"
>>> print(not s.find(s[::-1]))
False
```

Swipe

# one line input



```
>>> a = input()
>>> b = input()
>>> c = input()
>>> d = input()
>>> e = input()
```

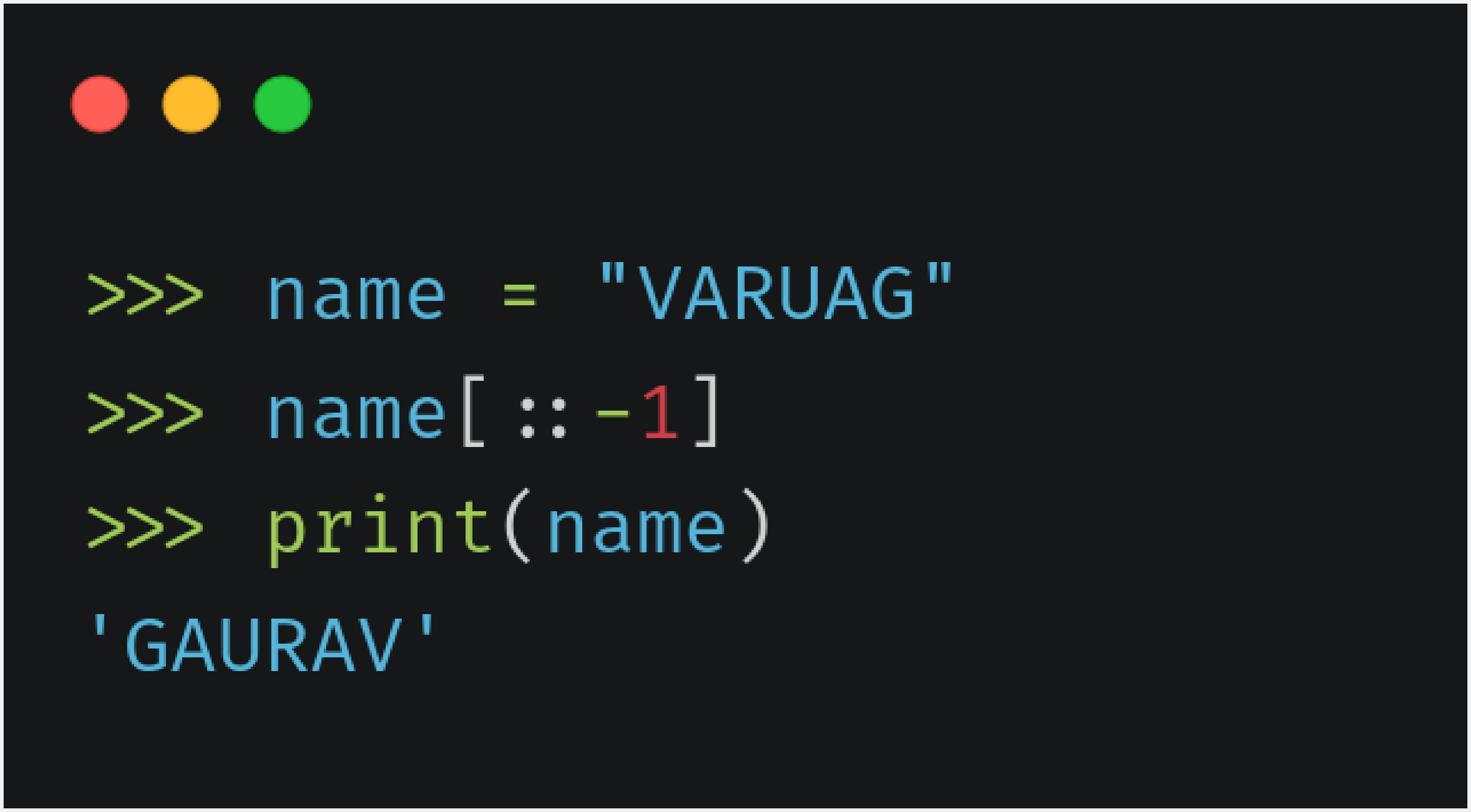
**Take input seperated by spaces**

```
>>> all_inputs = input().split(' ')
```

'all_inputs' is a list of all the input values

Swipe

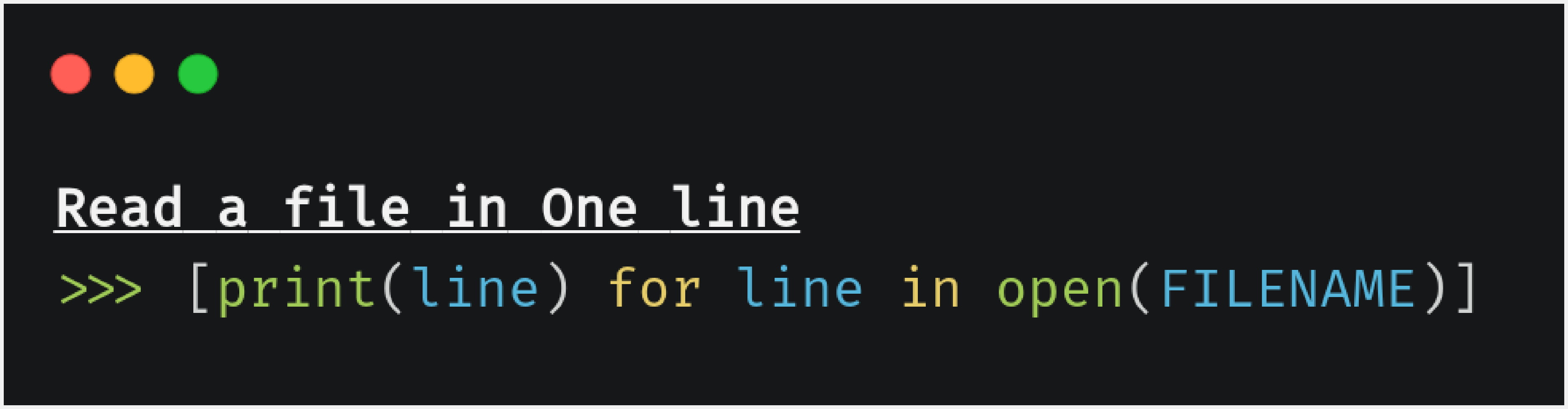
# reverse a string in one line



```
>>> name = "VARUAG"
>>> name[::-1]
>>> print(name)
'GAURAV'
```

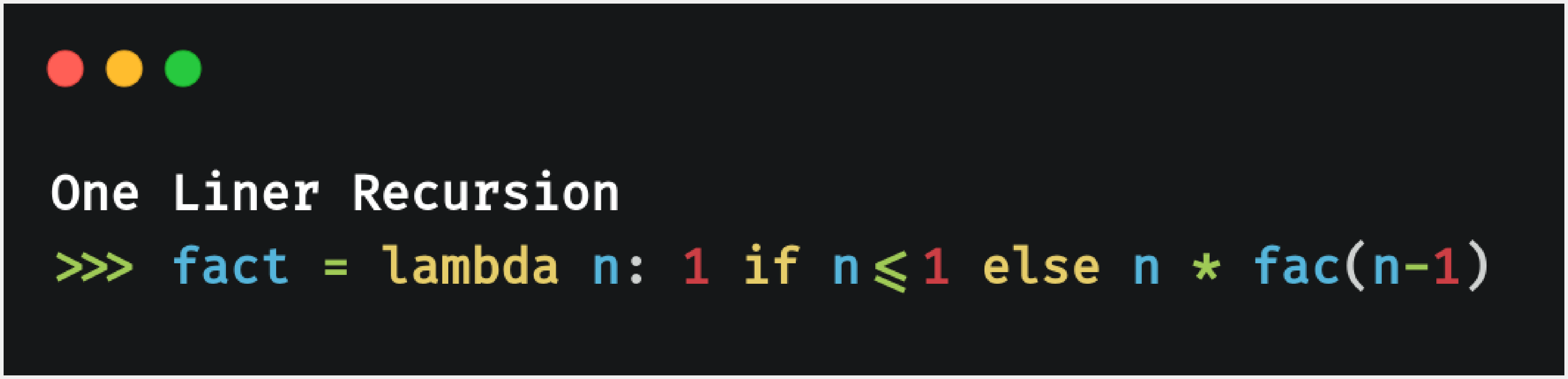
Swipe →

# read a file in one line code



```
Read a file in One line
>>> [print(line) for line in open(FILENAME)]
```

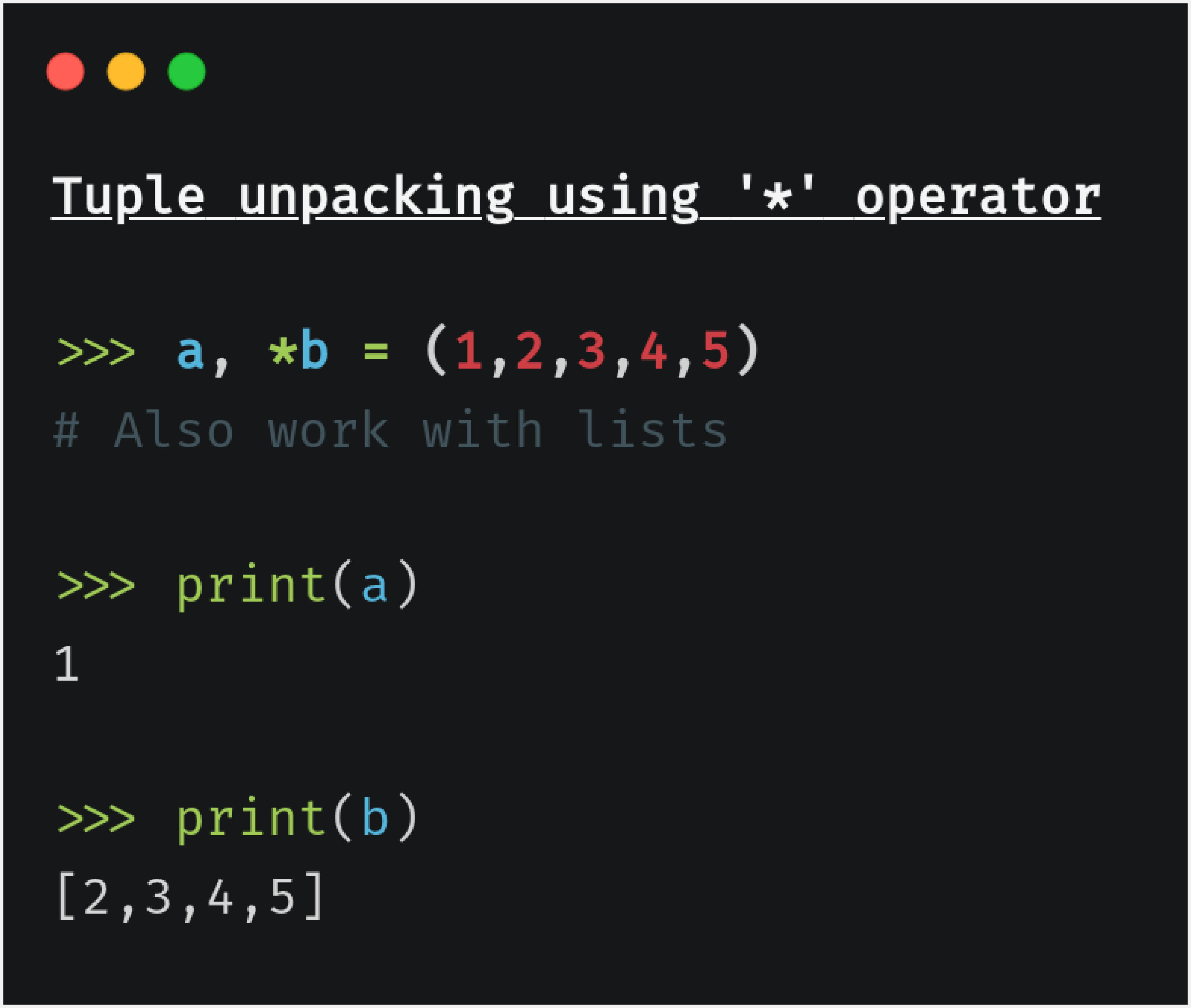
# one line recursion using lambda



```
One Liner Recursion
>>> fact = lambda n: 1 if n<=1 else n * fac(n-1)
```

Swipe

# tuple unpacking using *



Tuple unpacking using '*' operator

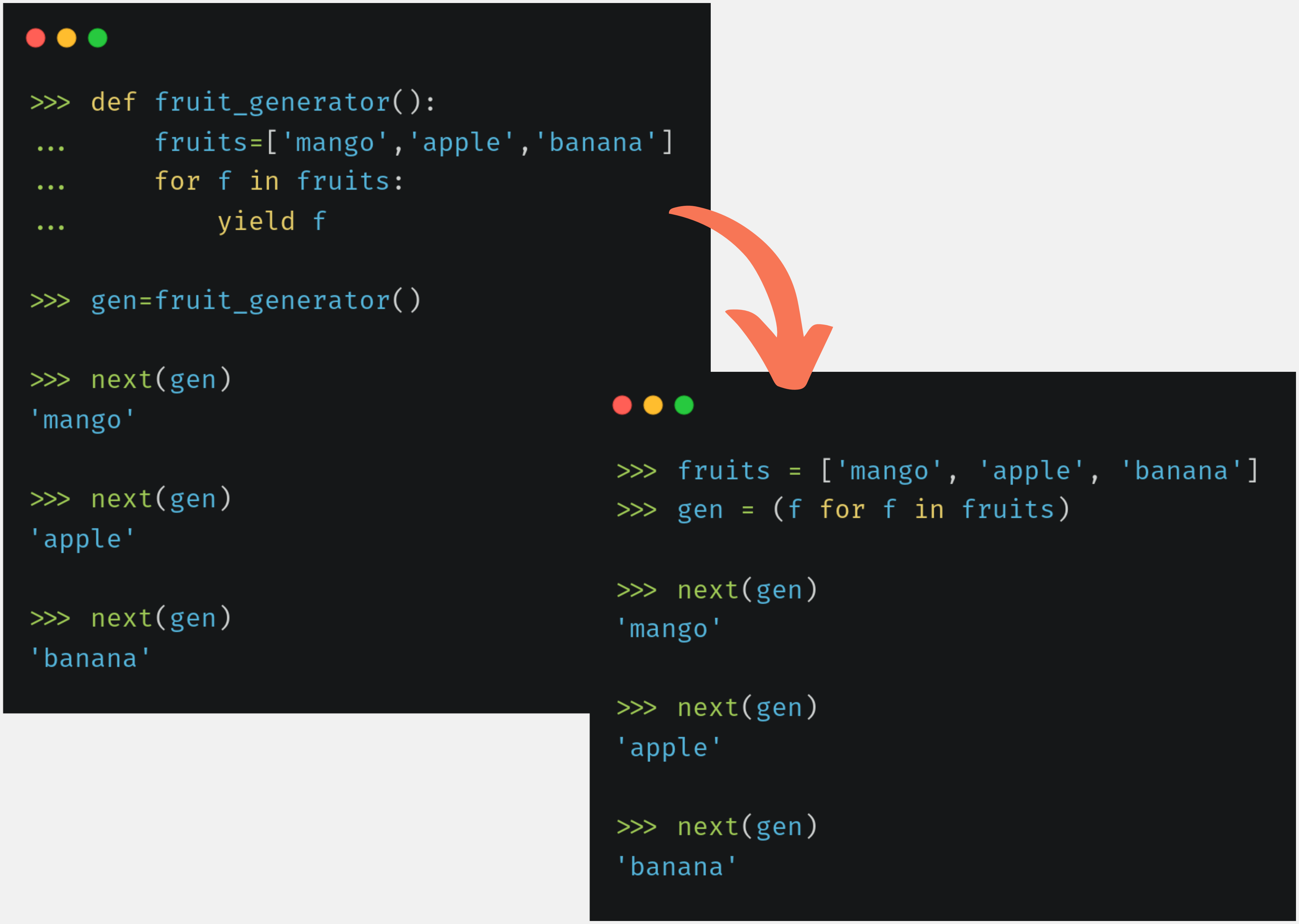
```
>>> a, *b = (1,2,3,4,5)
# Also work with lists

>>> print(a)
1

>>> print(b)
[2,3,4,5]
```

Swipe

# generator expression



```
>>> def fruit_generator():
...     fruits=['mango','apple','banana']
...     for f in fruits:
...         yield f

>>> gen=fruit_generator()

>>> next(gen)
'mango'

>>> next(gen)
'apple'

>>> next(gen)
'banana'
```

```
>>> fruits = ['mango', 'apple', 'banana']
>>> gen = (f for f in fruits)

>>> next(gen)
'mango'

>>> next(gen)
'apple'

>>> next(gen)
'banana'
```

Swipe

# THANKS FOR READING TILL THE END

Instagram - @DevJunction.in

LinkedIn - @DevJunction

YouTube - DevJunction