

code in Python 3.6

```
x = 'apple'  
  
def fun1():  
    x = 'mango'  
  
fun1()  
print(x)
```

(drag lower right corner to resize code editor)

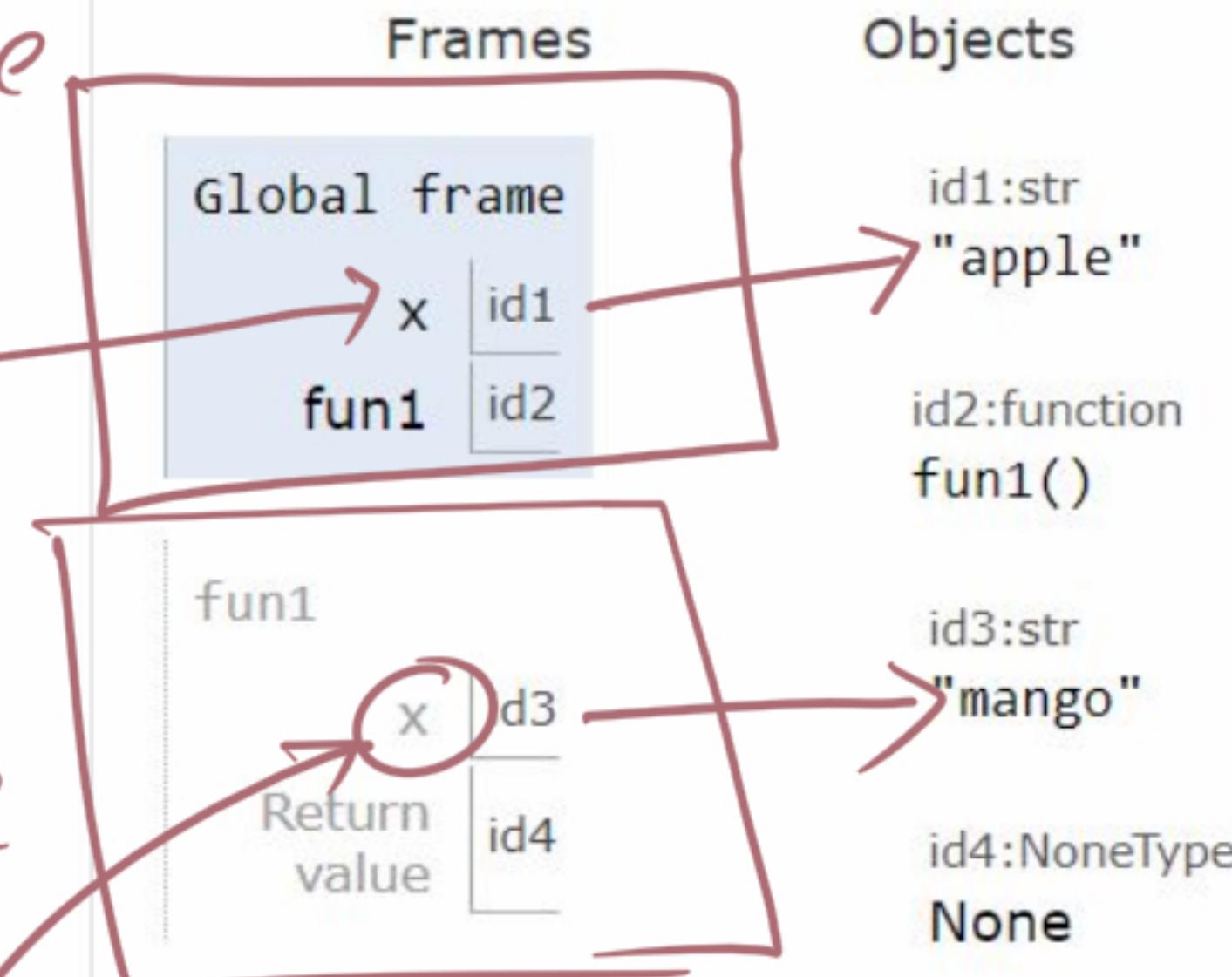
(drag lower right corner to resize code editor)

apple

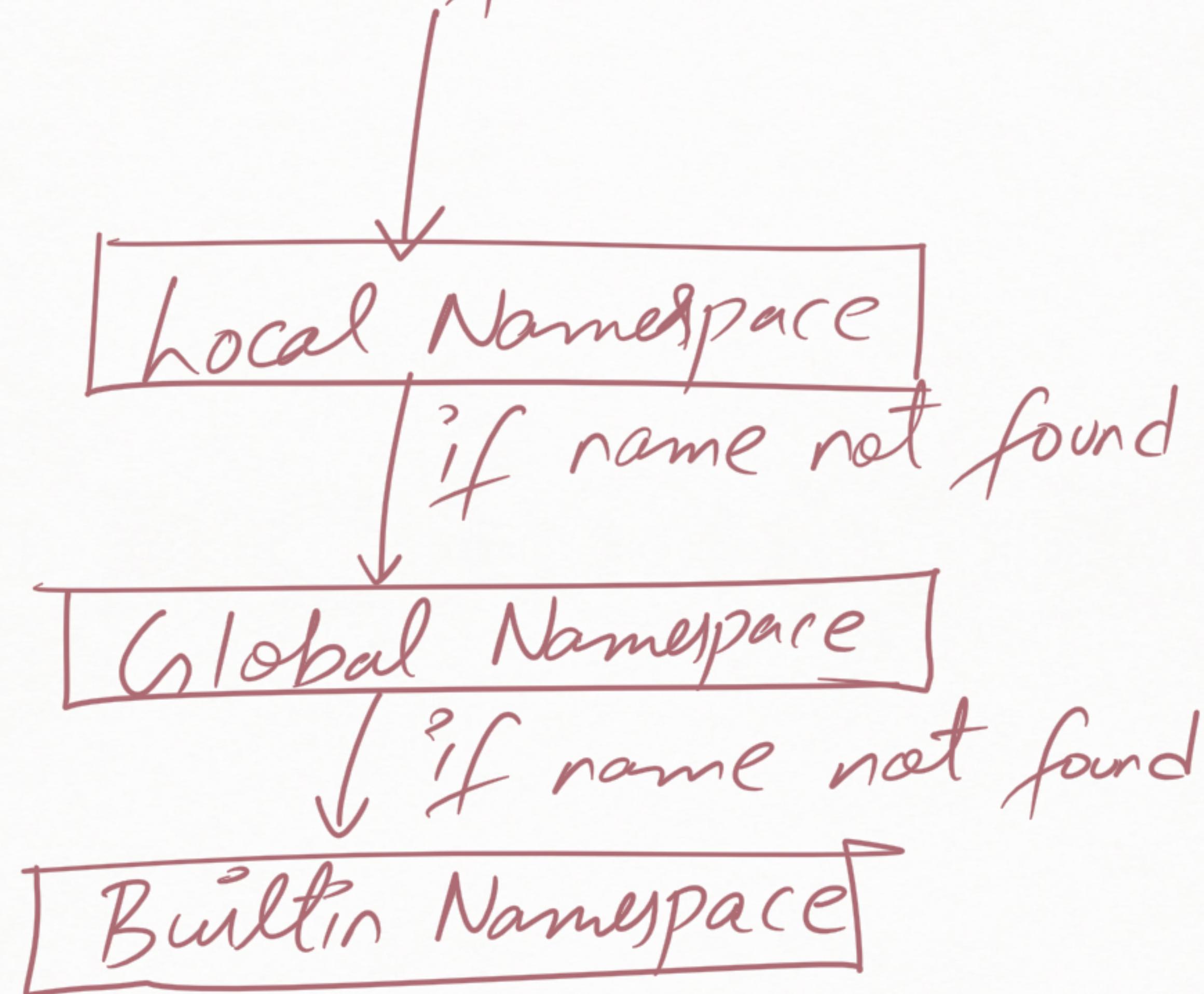
Global Namespace  
global x

local Namespace

local x  
of fun1



When we search for a names



In [6]:

```
x = 'apple'  
def fun1():  
    print(x)  
fun1()  
apple
```

global ✓  
Local Namespace X  
Global Namespace ✓

In [7]:

```
x = 'apple'  
def fun1():  
    x = 'mango'  
    print(x)  
fun1()  
mango
```

local Namespace ✓

In [9]:

```
def fun1():
    a = [1, 2, 3, 4, 5]
    x = len(a)
    print(x)
fun1()
```

5

Built-in Namespace.

It searches for len name  
in fun1 local Namespace

Do NOT Find

Global Namespace

Not found!!

Built-in Namespace ✓

In [17]:

```
def print(x):  
    return x+2  
  
def len(x):  
    return x+5
```

In [18]:

```
print(1)
```

Out[18]:

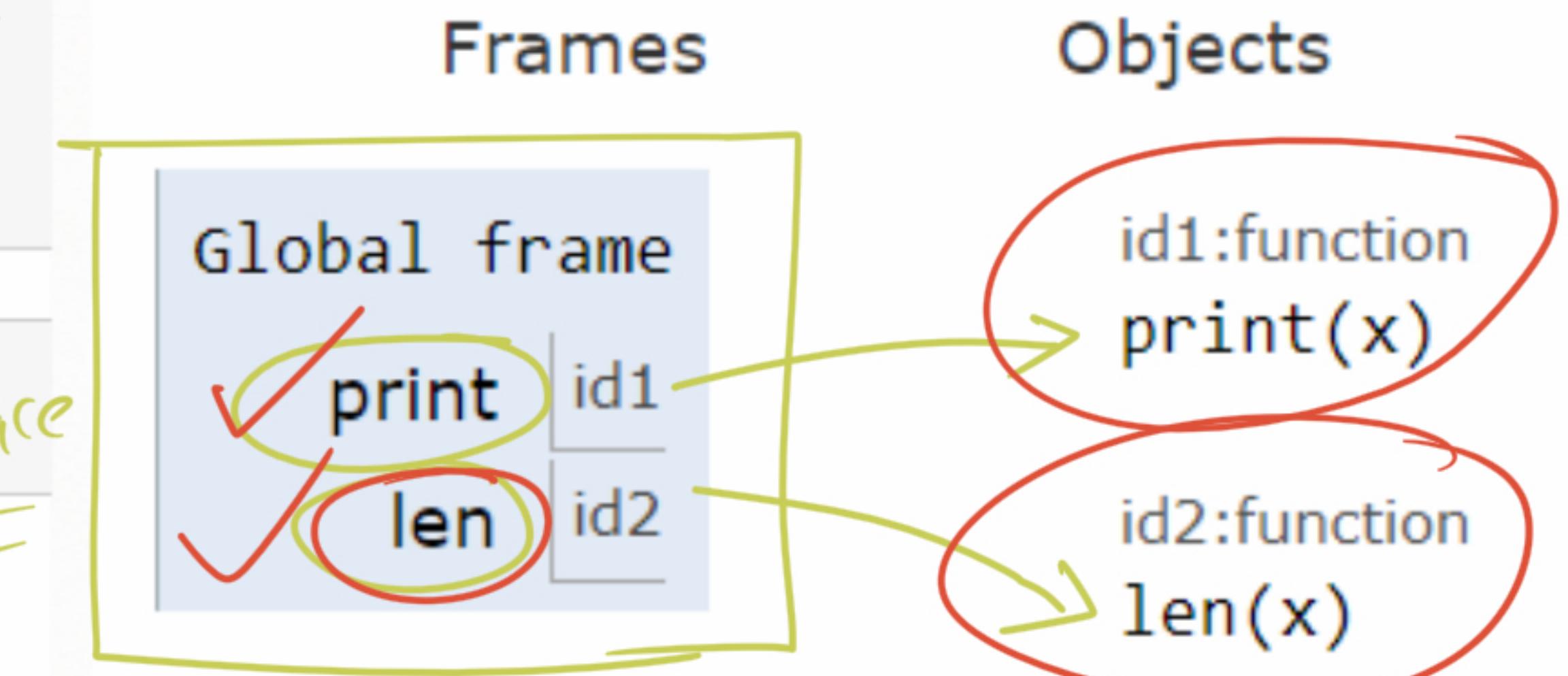
```
3
```

In [19]:

```
len(10)
```

Out[19]:

```
15
```



Local Namespace X

Global Namespace ✓

```
def fun1(*a):  
    print(a)
```

fun1(1,2)  
fun1(1,2,3,4,5)  
fun1("apple")

a automatically becomes a tuple containing all the passed elements.

a = (1,2)  
a = (1,2,3,4,5)  
a = ("apple",)