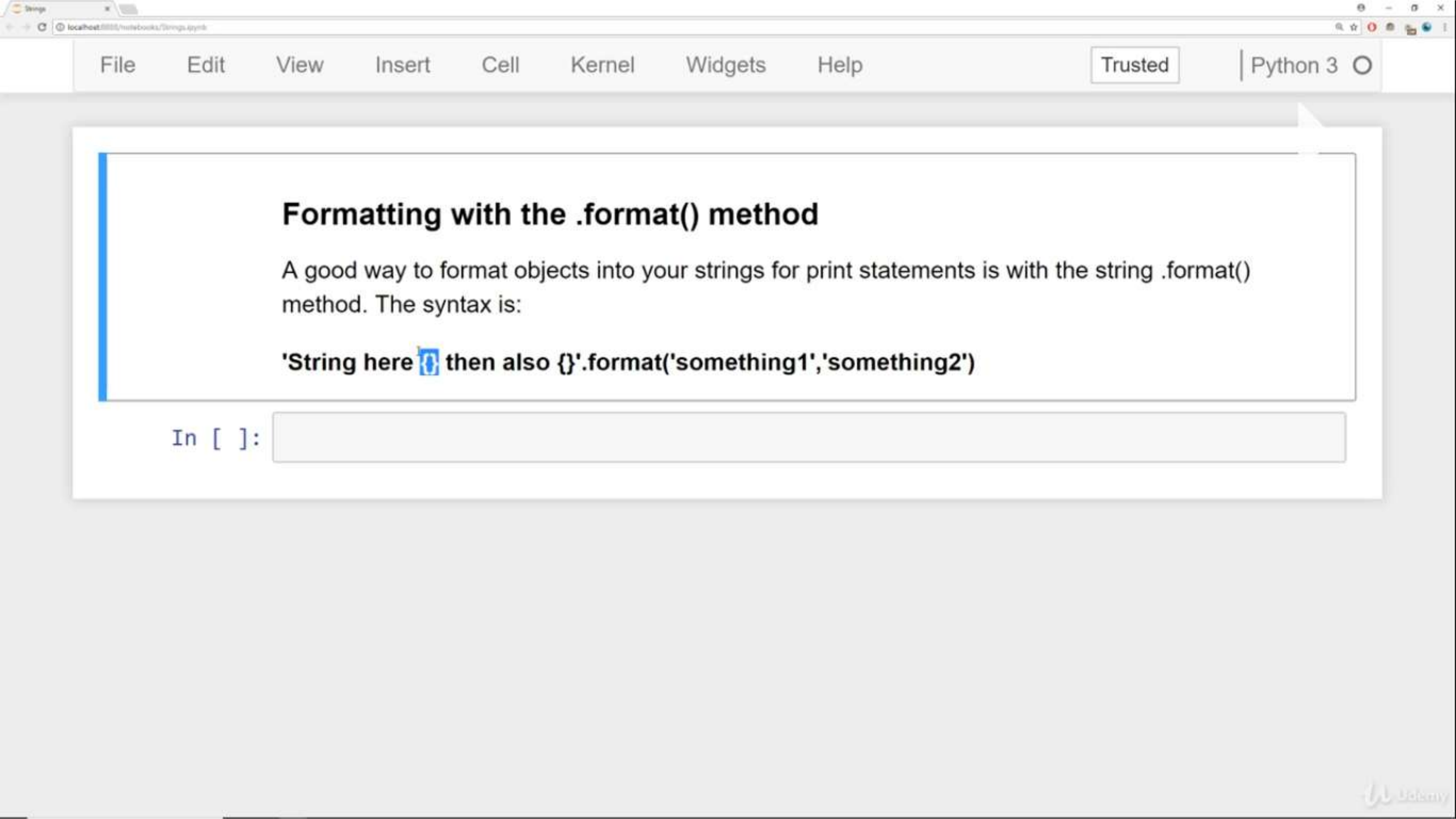
String Formatting for Printing

PIERIAN DATA

Complete Python 3 Bootcamp

- Often you will want to “inject” a variable into your string for printing. For example:
    - **my_name = “Jose”**
    - **print(“Hello ” + my_name)**
- There are multiple ways to format strings for printing variables in them.
- This is known as string interpolation.

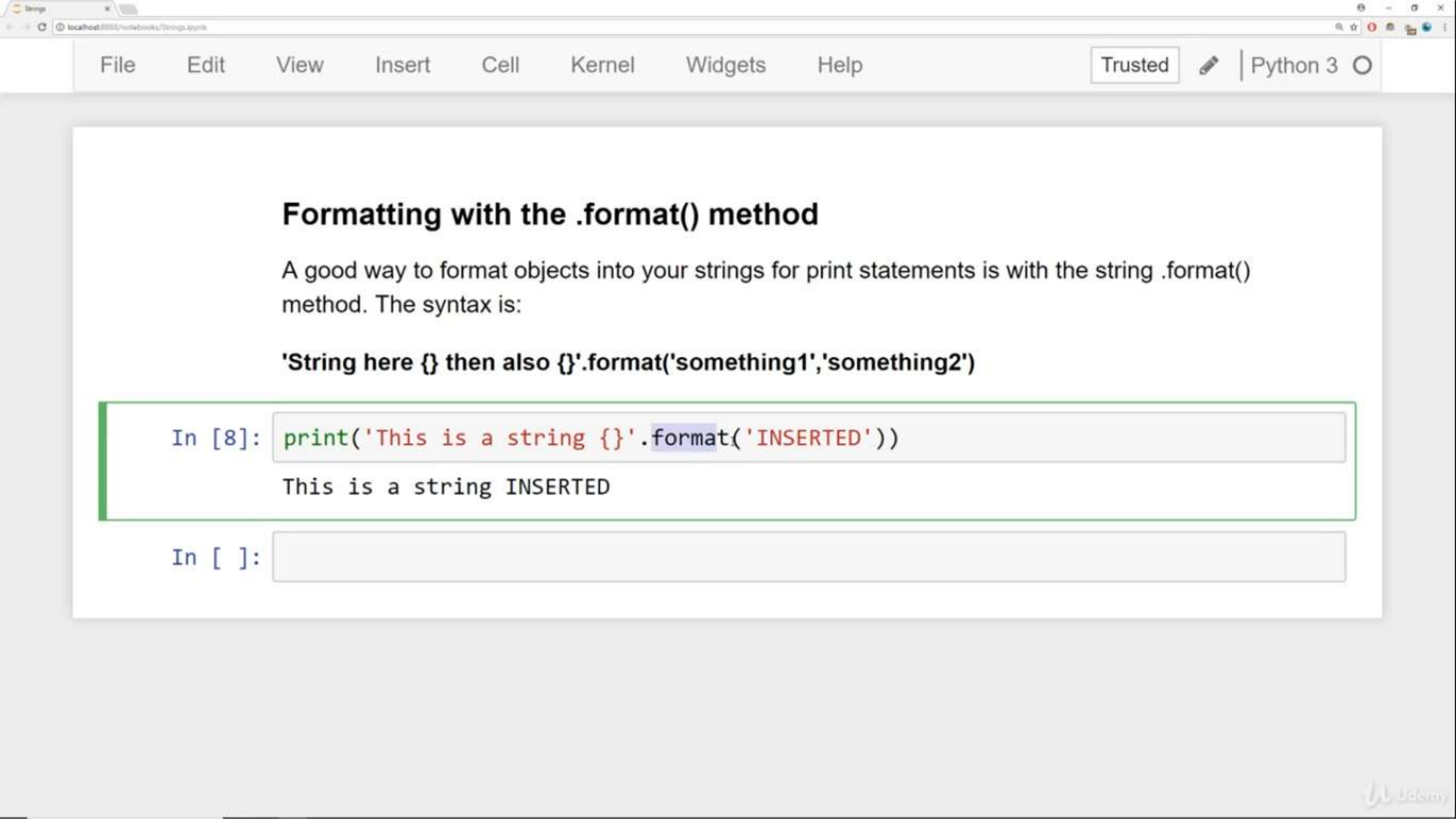
- Let’s explore two methods for this:
  - **.format()** method
  - **f-strings** (formatted string literals)



Formatting with the .format() method

A good way to format objects into your strings for print statements is with the string .format() method. The syntax is:

'String here {} then also {}'.format('something1','something2')



Formatting with the .format() method

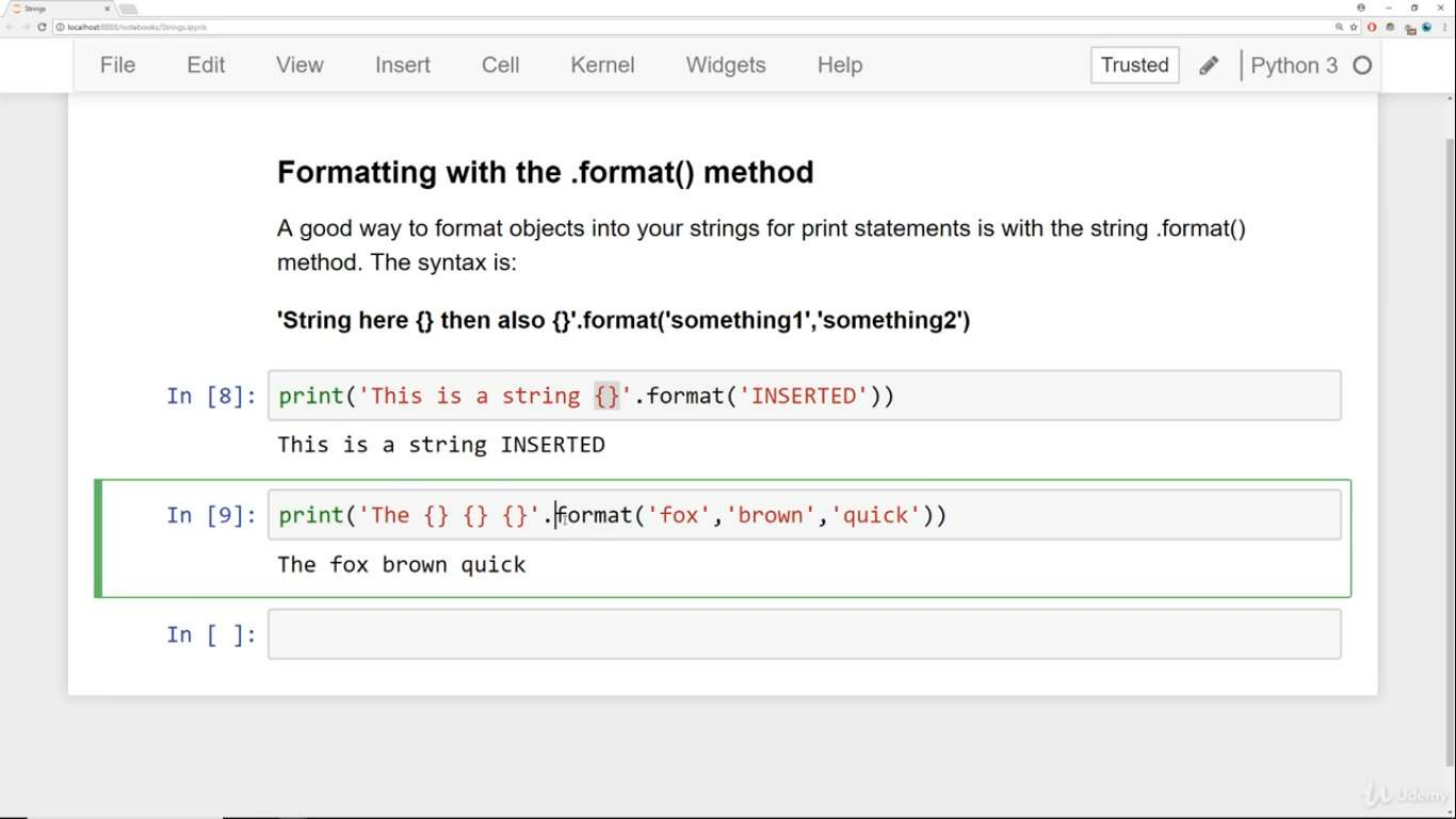
A good way to format objects into your strings for print statements is with the string .format() method. The syntax is:

'String here {} then also {}'.format('something1','something2')

In [8]:
```
print('This is a string {}'.format('INSERTED'))
```

```
This is a string INSERTED
```

In [ ]:



Formatting with the .format() method

A good way to format objects into your strings for print statements is with the string .format() method. The syntax is:

'String here {} then also {}'.format('something1','something2')

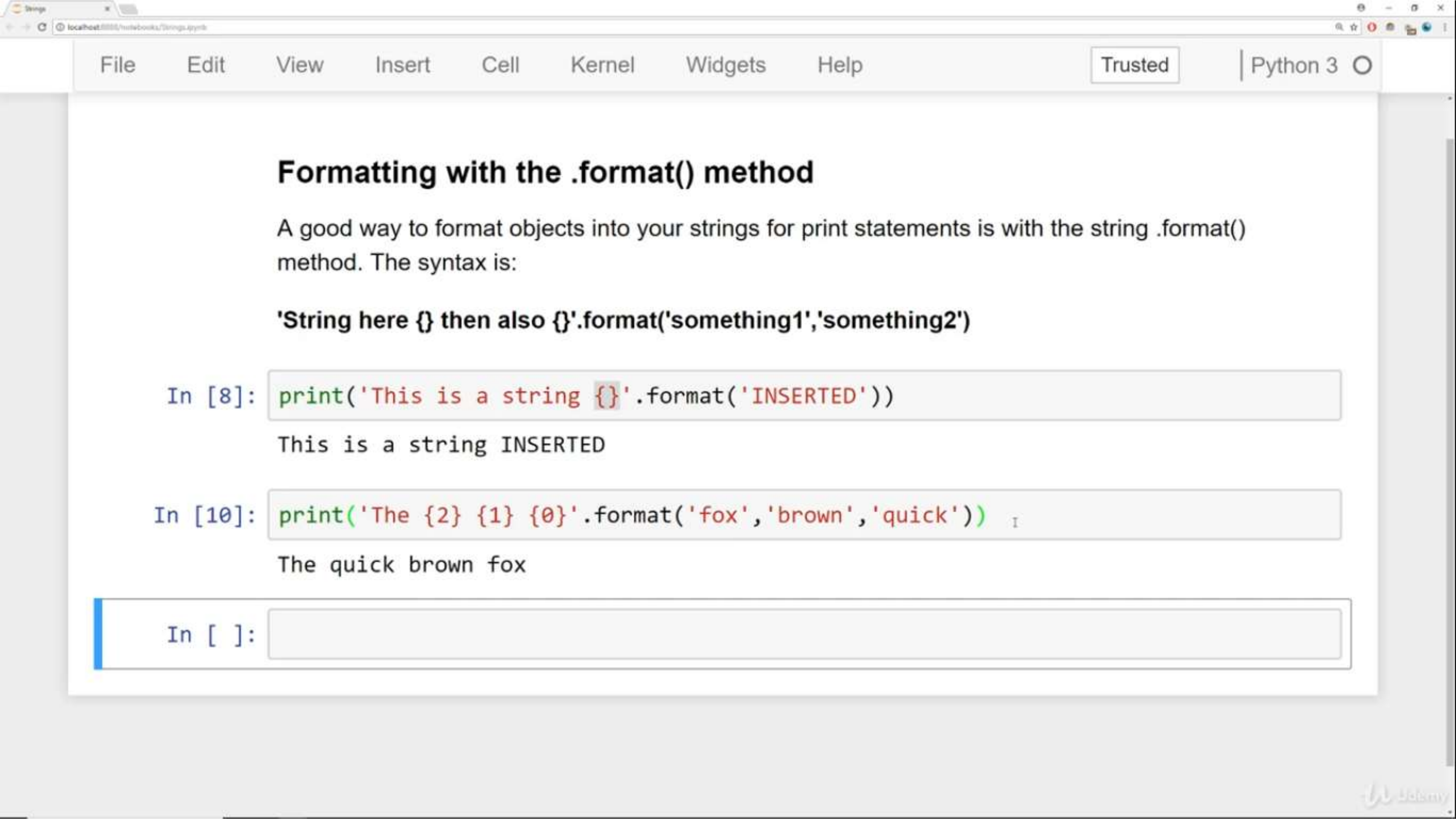
In [8]:
```
print('This is a string {}'.format('INSERTED'))
```

```
This is a string INSERTED
```

In [9]:
```
print('The {} {} {}'.format('fox','brown','quick'))
```

```
The fox brown quick
```

In [ ]:



Formatting with the .format() method

A good way to format objects into your strings for print statements is with the string .format() method. The syntax is:

'String here {} then also {}'.format('something1','something2')

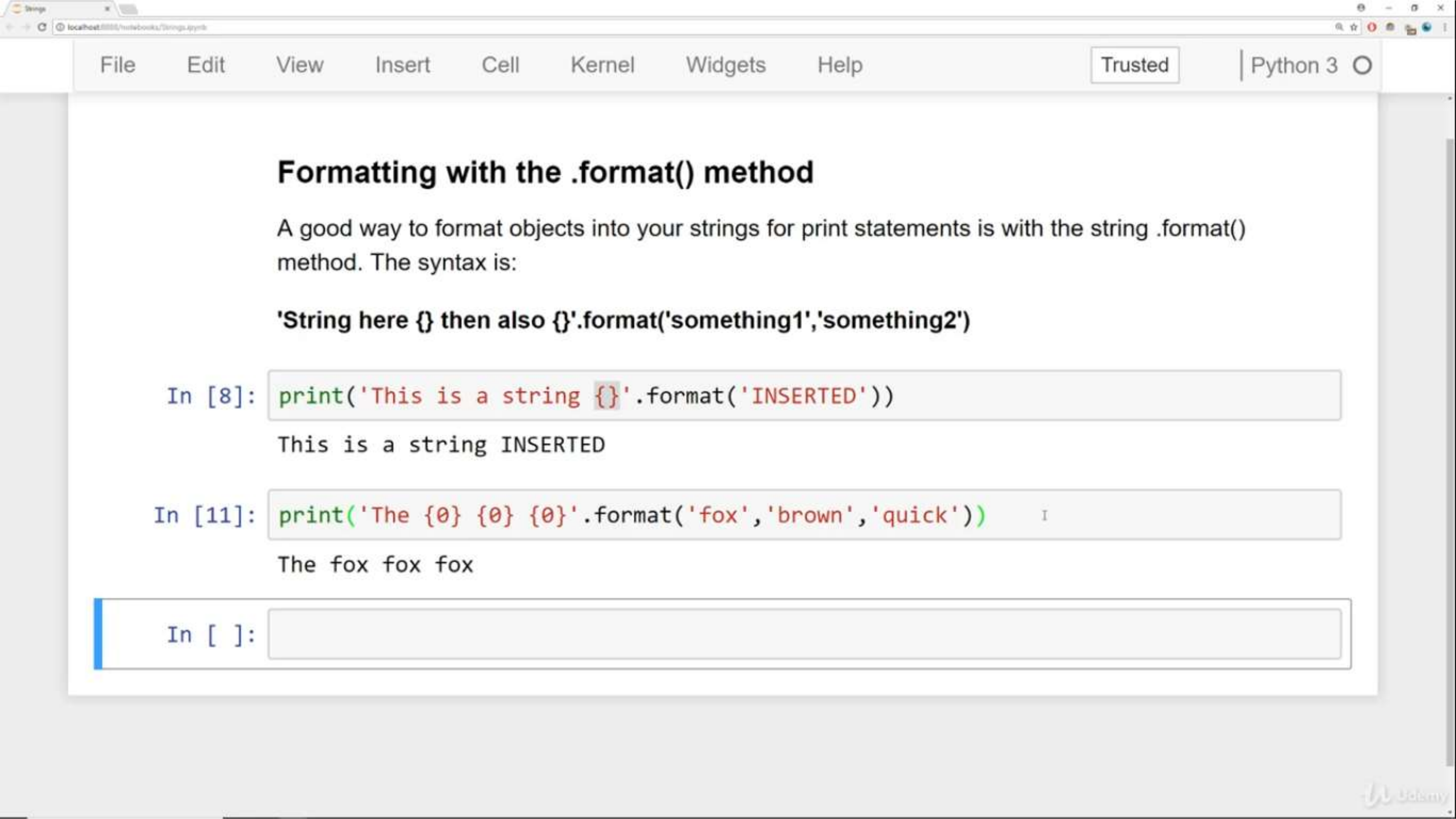
```
In [8]: print('This is a string {}'.format('INSERTED'))

This is a string INSERTED

In [10]: print('The {2} {1} {0}'.format('fox','brown','quick'))

The quick brown fox

In [ ]:
```



Formatting with the .format() method

A good way to format objects into your strings for print statements is with the string .format() method. The syntax is:

'String here {} then also {}'.format('something1','something2')

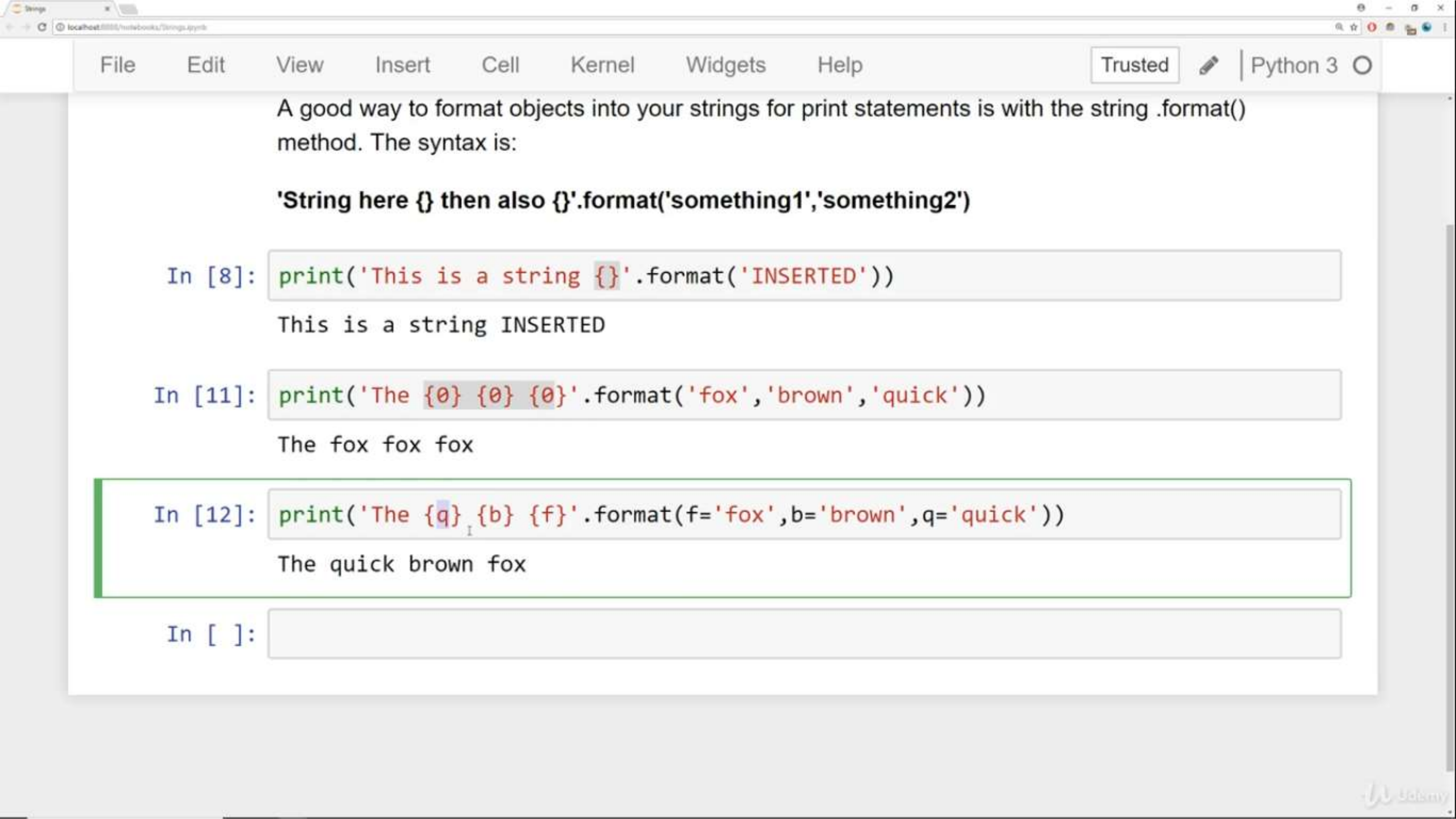
```
In [8]: print('This is a string {}'.format('INSERTED'))
```

```
This is a string INSERTED
```

```
In [11]: print('The {0} {0} {0}'.format('fox','brown','quick'))
```

```
The fox fox fox
```

```
In [ ]:
```



A good way to format objects into your strings for print statements is with the string .format() method. The syntax is:

'String here {} then also {}'.format('something1','something2')

```
In [8]: print('This is a string {}'.format('INSERTED'))
```

```
This is a string INSERTED
```

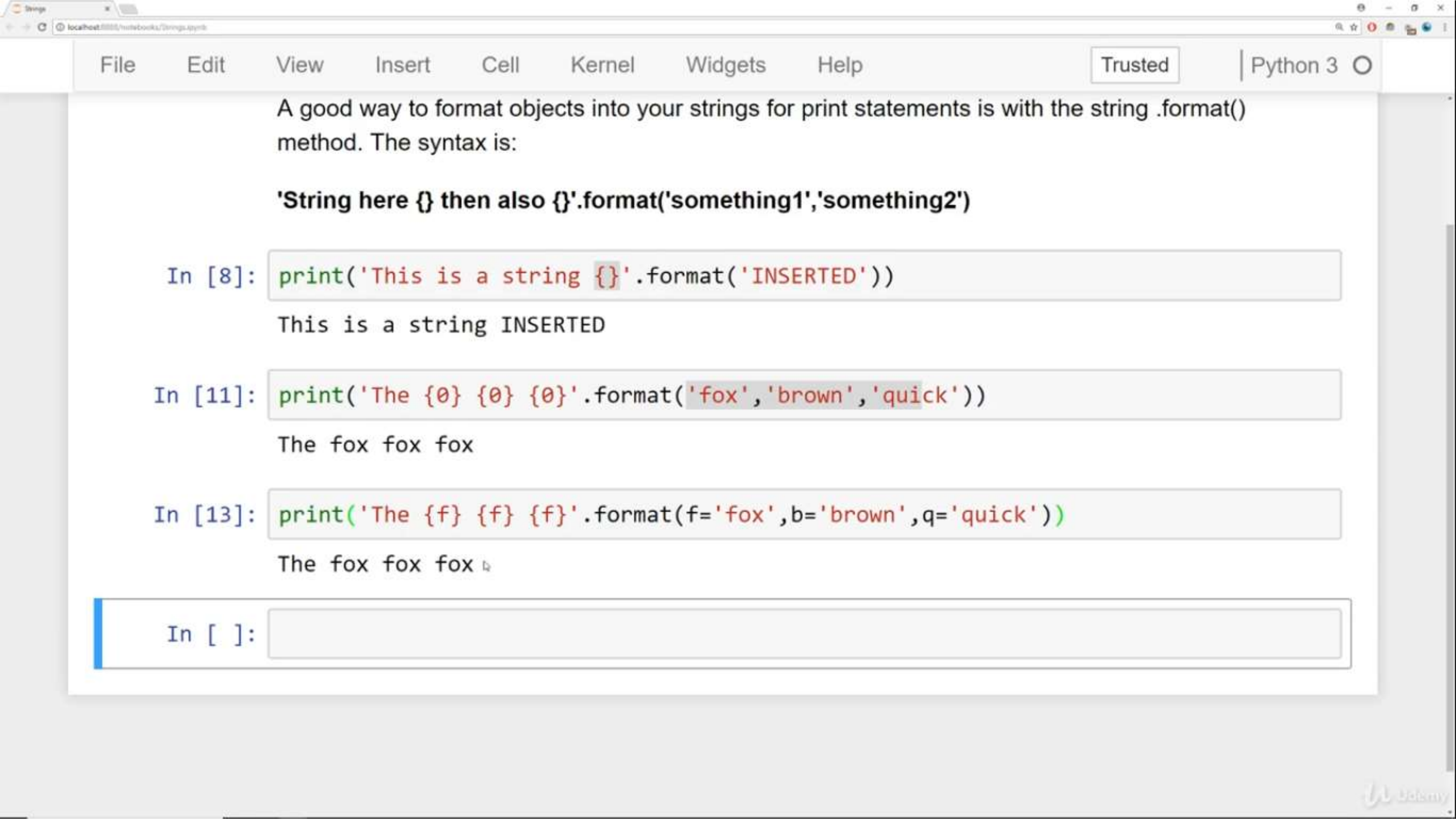
```
In [11]: print('The {0} {0} {0}'.format('fox','brown','quick'))
```

```
The fox fox fox
```

```
In [12]: print('The {q} {b} {f}'.format(f='fox',b='brown',q='quick'))
```

```
The quick brown fox
```

```
In [ ]:
```



A good way to format objects into your strings for print statements is with the string .format() method. The syntax is:

'String here {} then also {}'.format('something1','something2')

```
In [8]: print('This is a string {}'.format('INSERTED'))
```

```
This is a string INSERTED
```

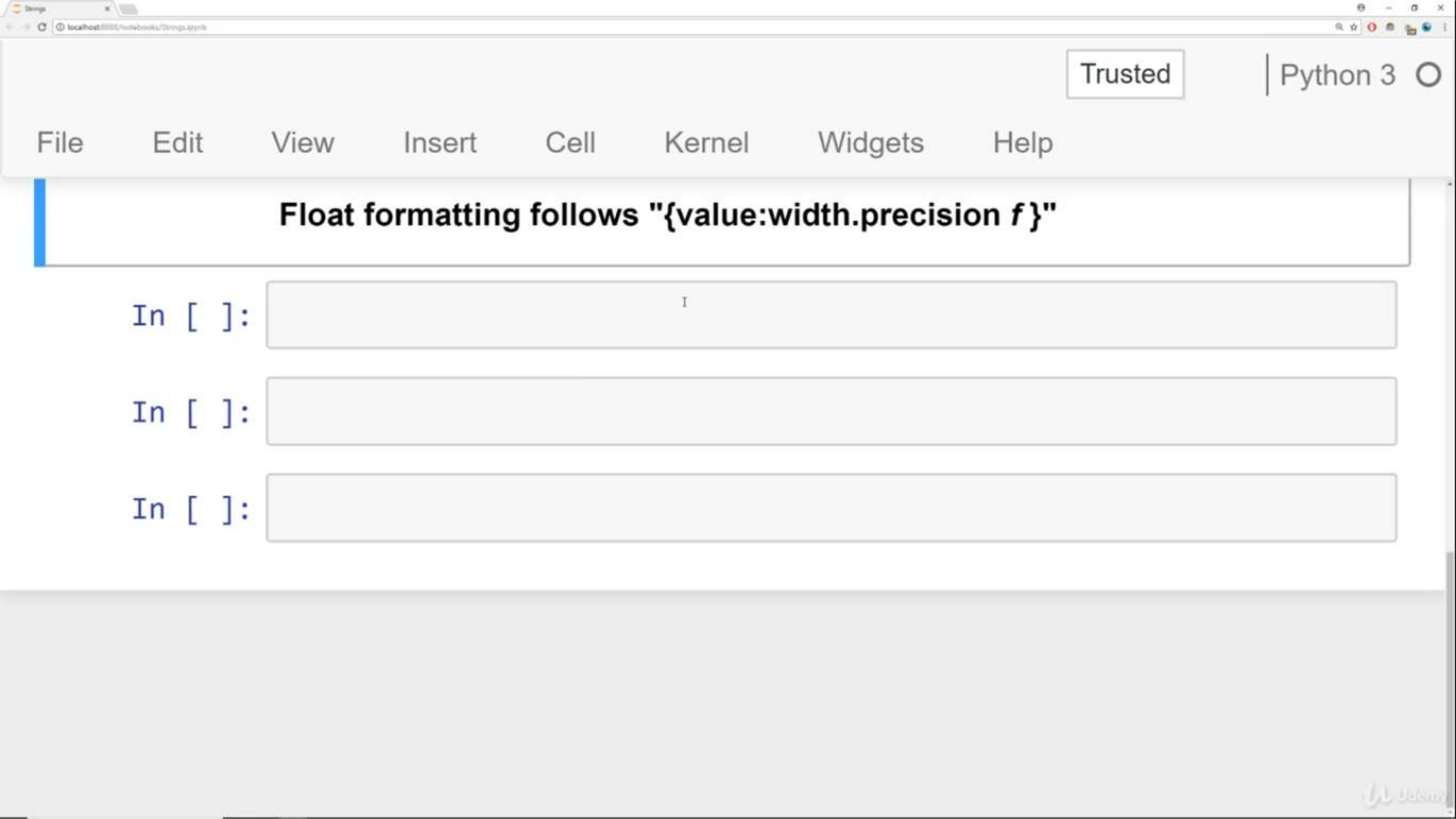
```
In [11]: print('The {0} {0} {0}'.format('fox','brown','quick'))
```

```
The fox fox fox
```

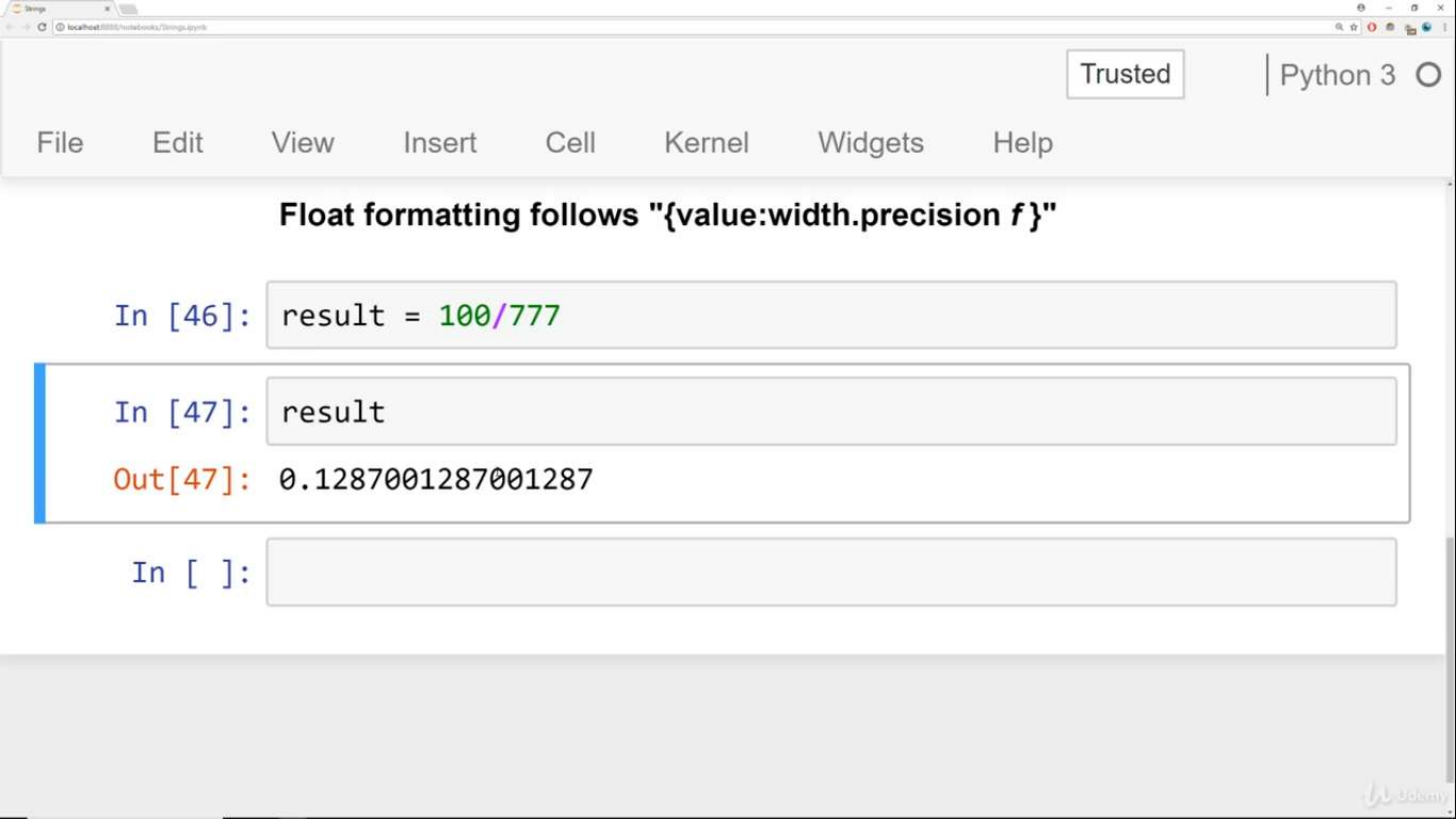
```
In [13]: print('The {f} {f} {f}'.format(f='fox',b='brown',q='quick'))
```

```
The fox fox fox
```

```
In [ ]:
```



Float formatting follows "{value:width.precision *f*}"



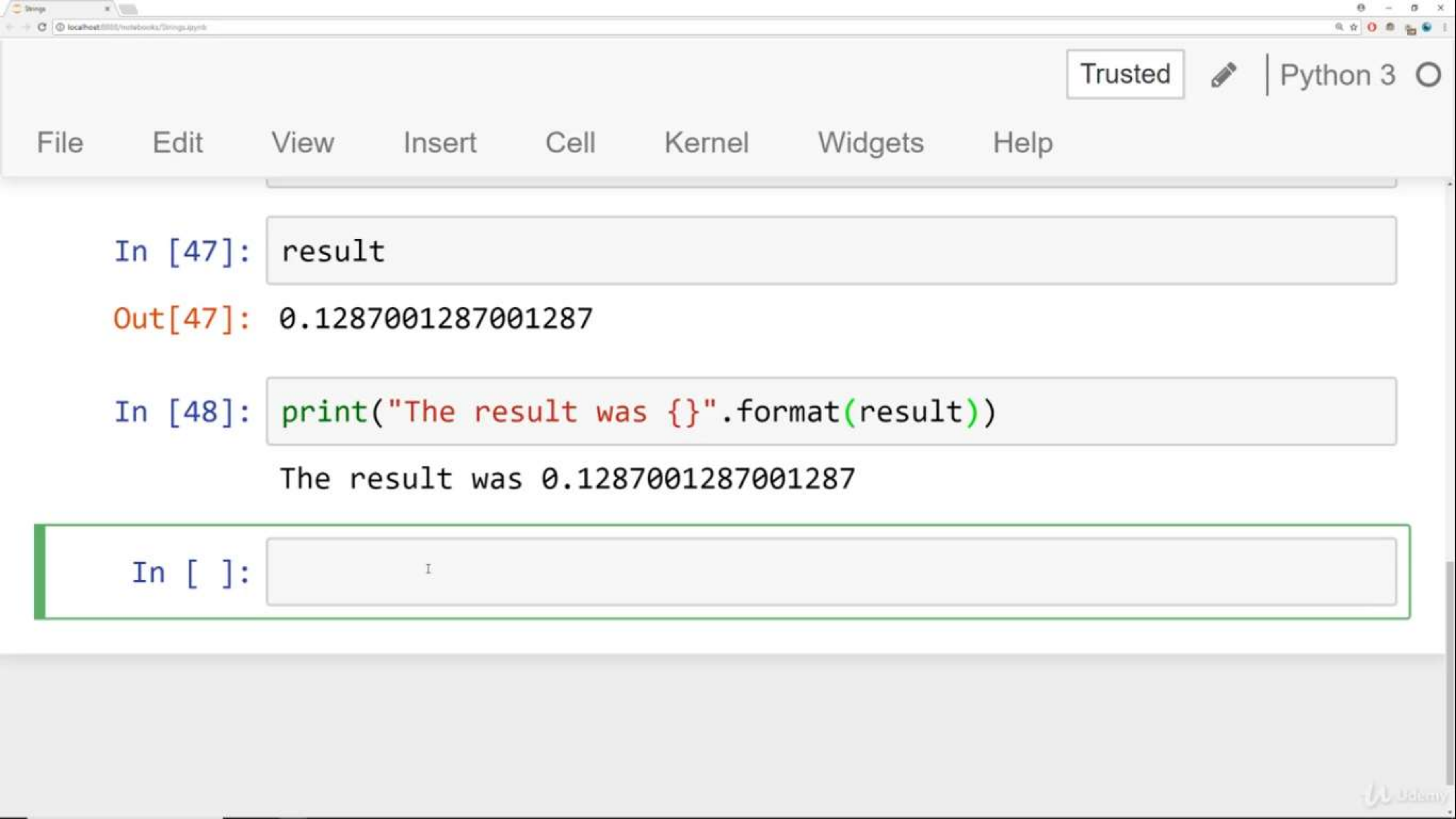
Float formatting follows "{value:width.precision *f* }"

```
In [46]: result = 100/777
```

```
In [47]: result
```

```
Out[47]: 0.1287001287001287
```

```
In [ ]:
```



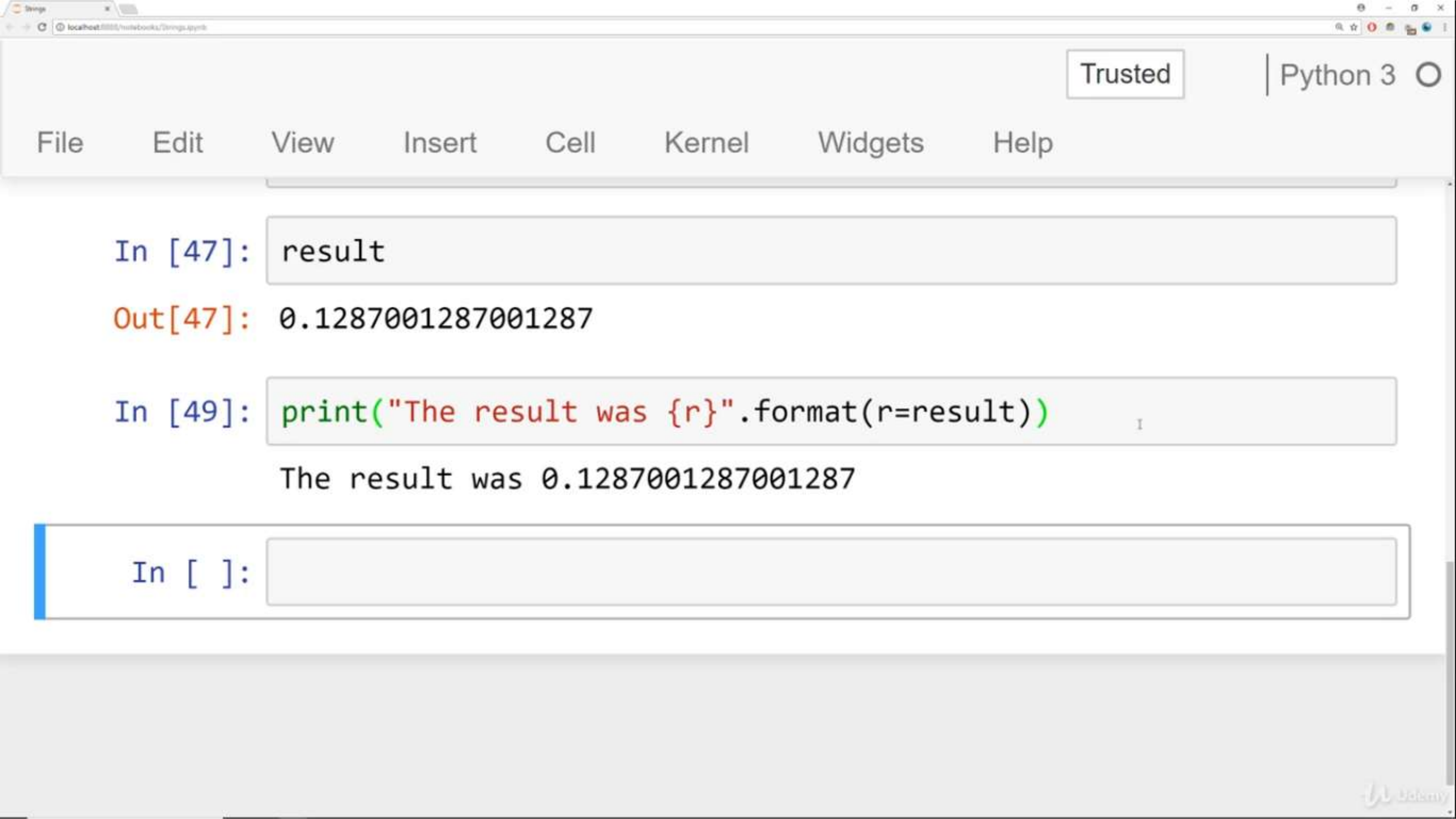
```
In [47]: result

Out[47]: 0.1287001287001287

In [48]: print("The result was {}".format(result))

         The result was 0.1287001287001287

In [ ]:
```



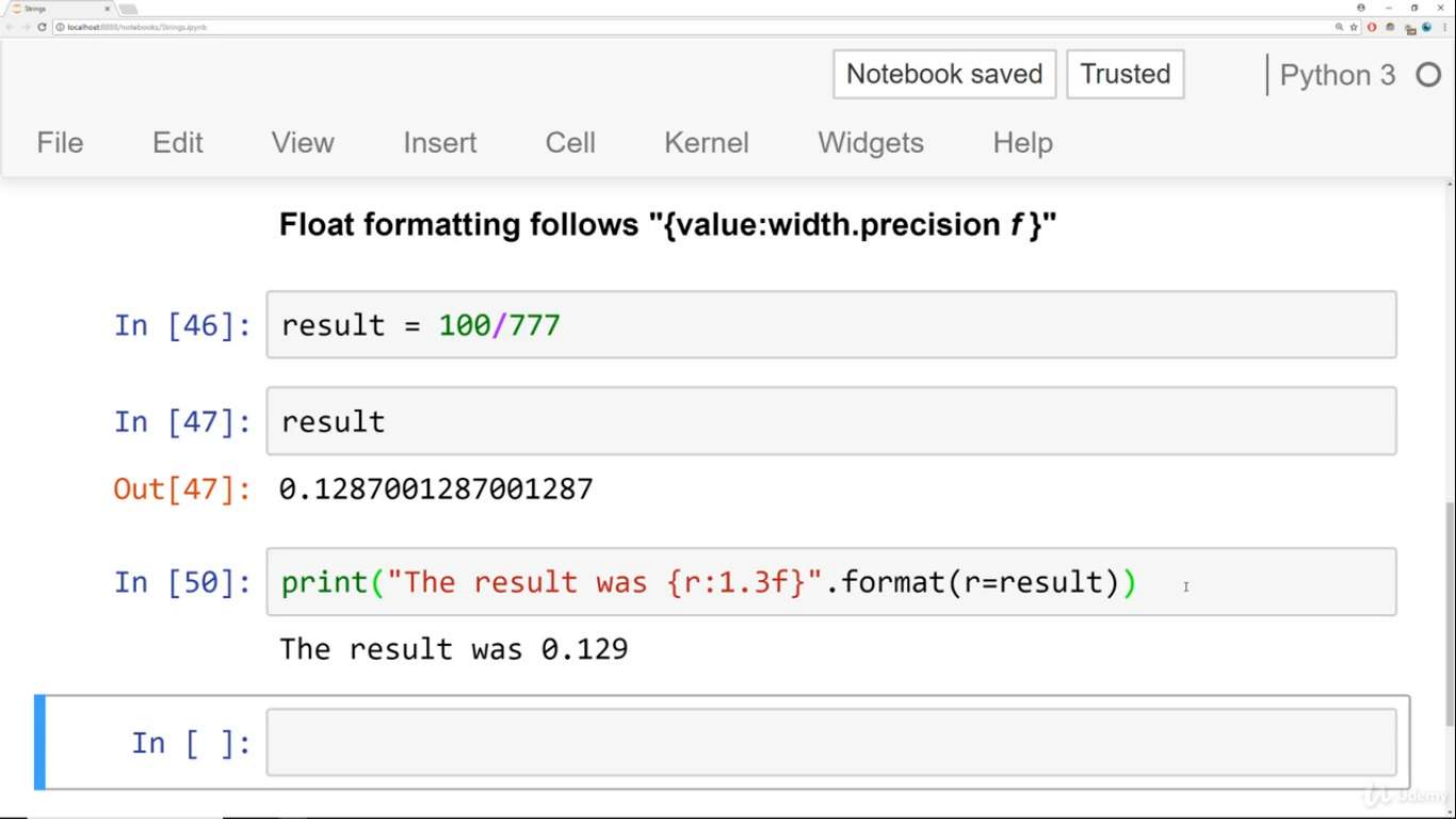
```
In [47]: result

Out[47]: 0.12870012870012870

In [49]: print("The result was {r}".format(r=result))

         The result was 0.12870012870012870

In [ ]:
```



Float formatting follows "{value:width.precision *f*}"

```
In [46]: result = 100/777
```

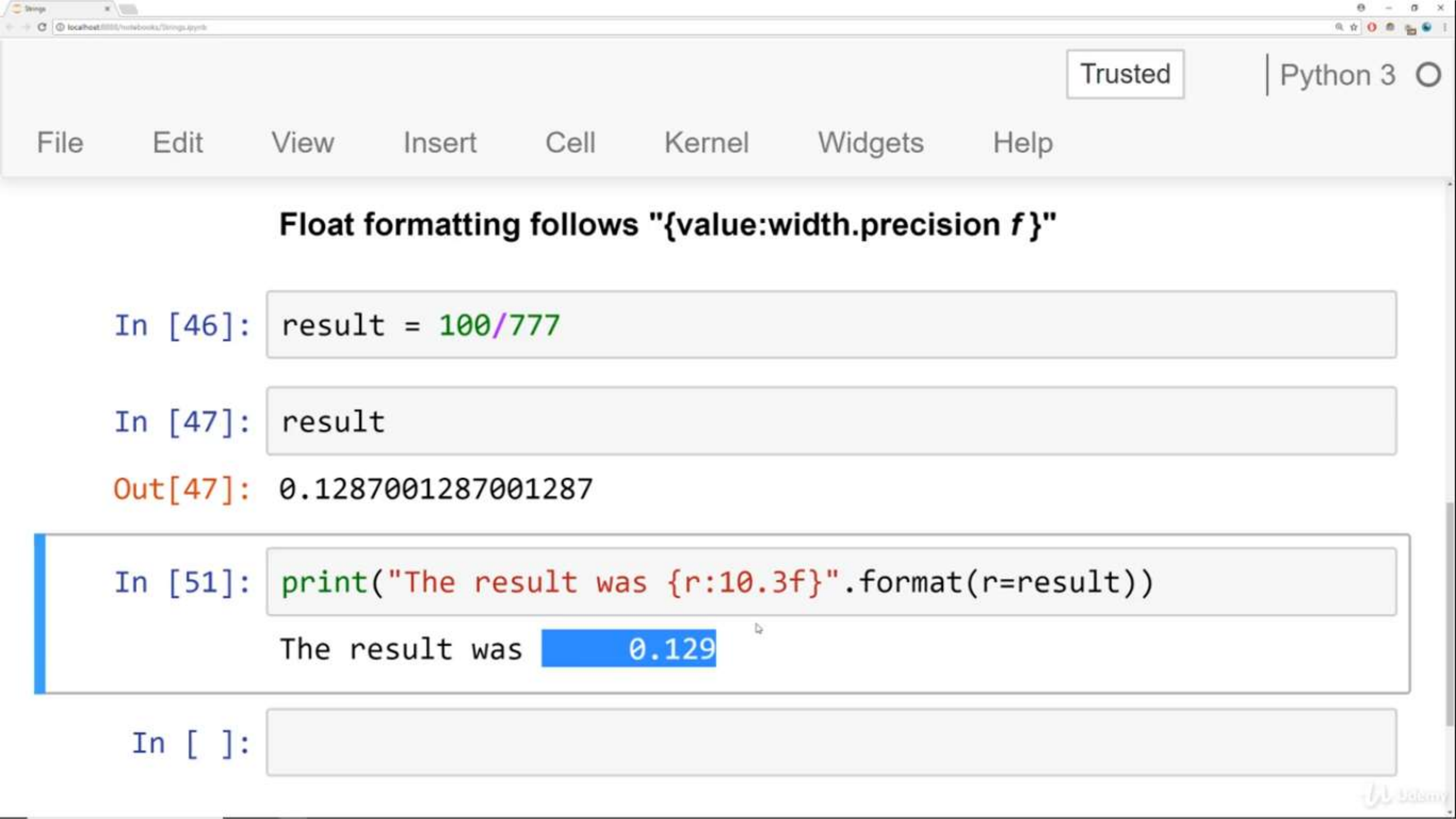
```
In [47]: result
```

```
Out[47]: 0.1287001287001287
```

```
In [50]: print("The result was {r:1.3f}".format(r=result))
```

```
The result was 0.129
```

```
In [ ]:
```



Float formatting follows "{value:width.precision *f* }"

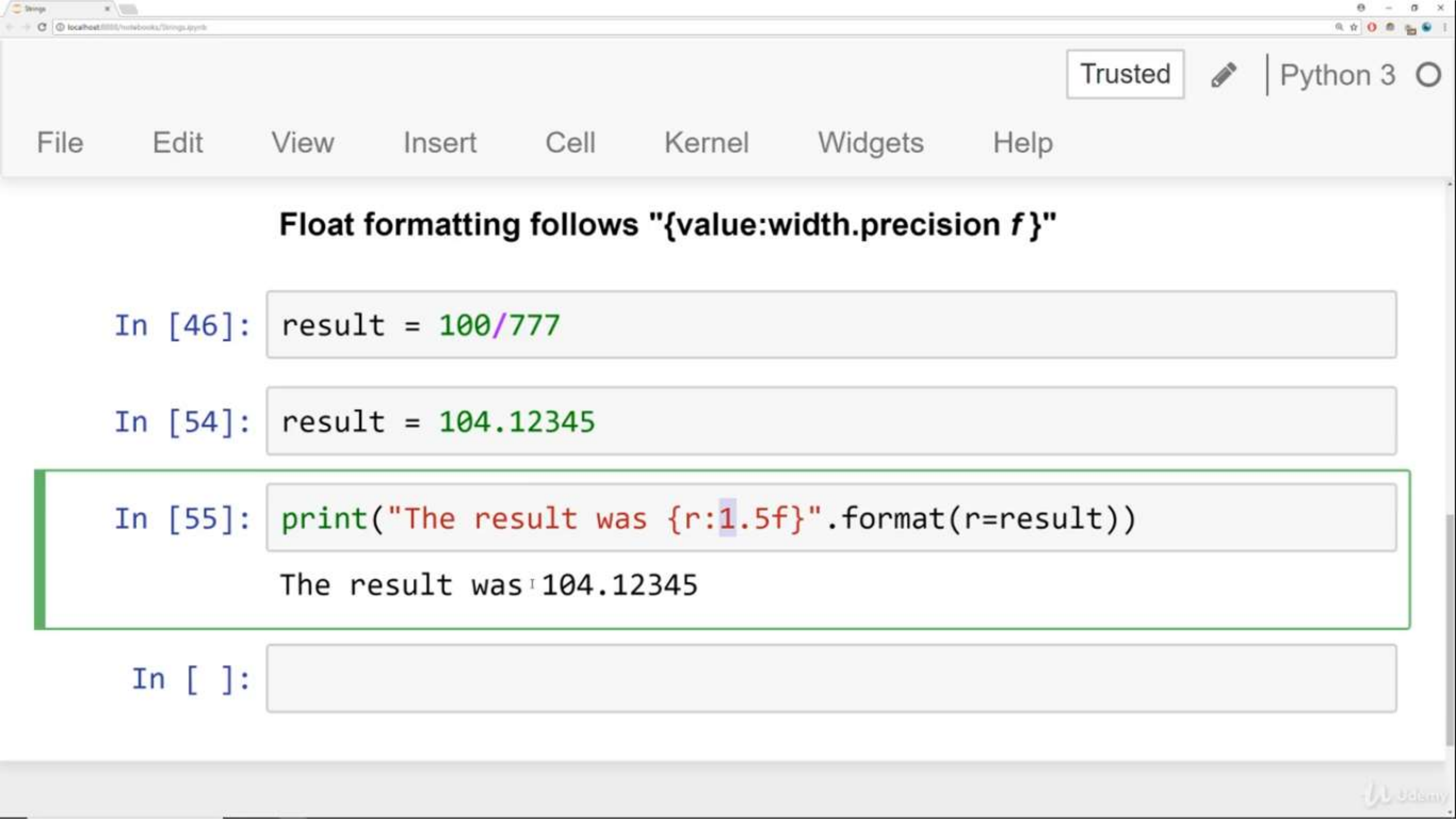
```
In [46]: result = 100/777
```

```
In [47]: result
Out[47]: 0.1287001287001287
```

```
In [51]: print("The result was {r:10.3f}".format(r=result))

The result was      0.129
```

```
In [ ]:
```



Float formatting follows "{value:width.precision *f* }"

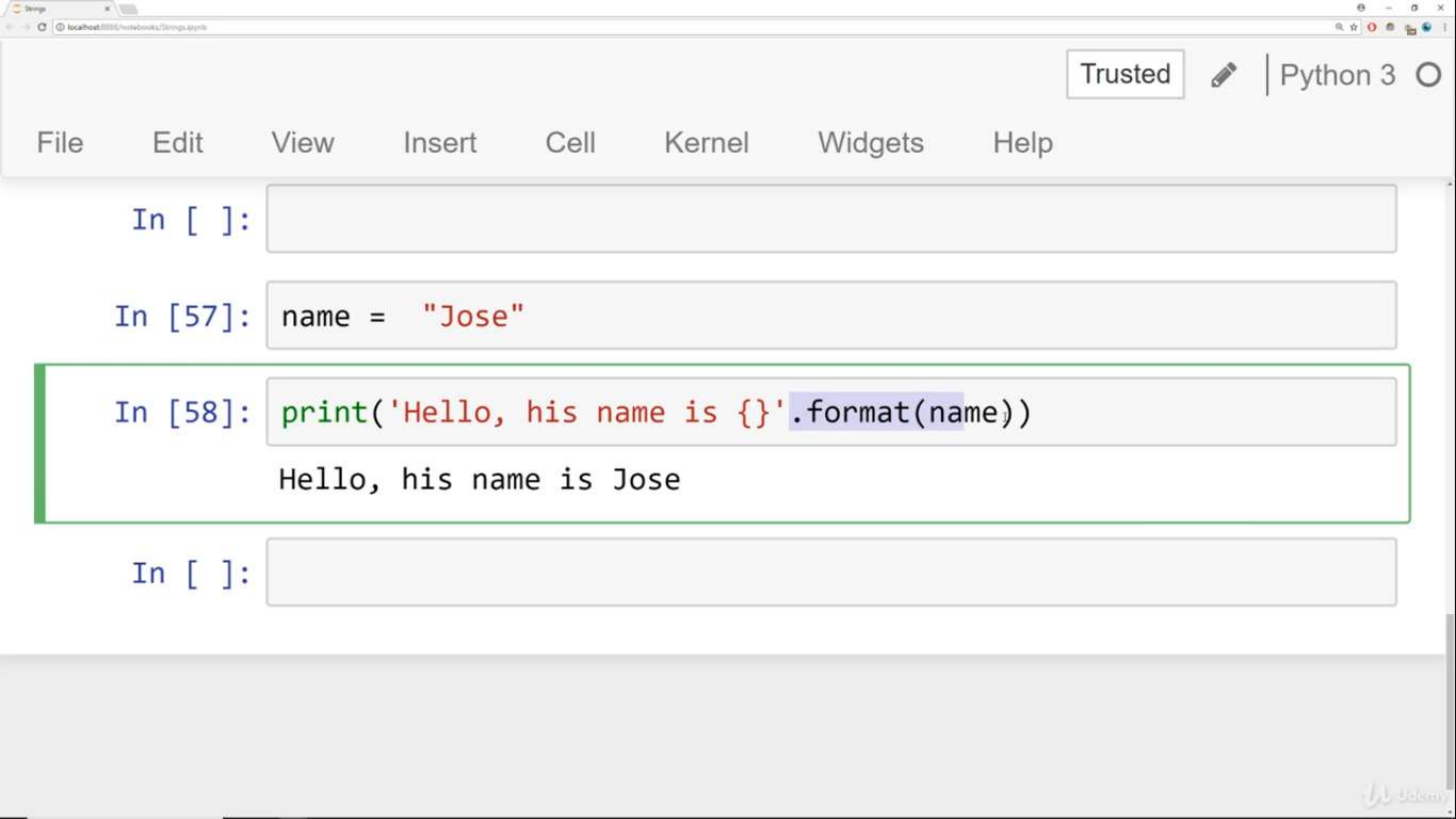
```
In [46]: result = 100/777
```

```
In [54]: result = 104.12345
```

```
In [55]: print("The result was {r:1.5f}".format(r=result))
```

```
The result was 104.12345
```

```
In [ ]:
```



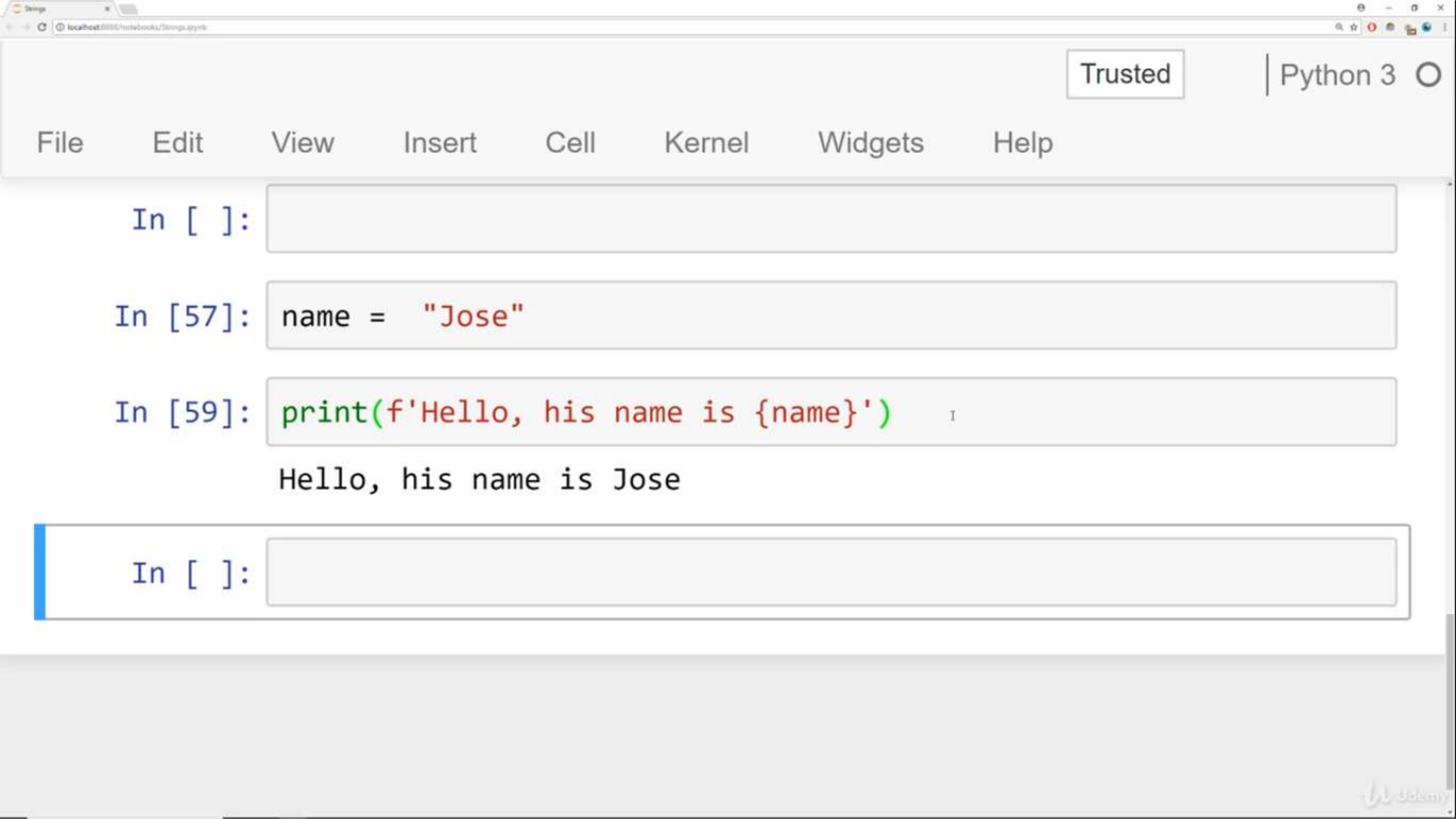
```
In [ ]:
```

```
In [57]: name =  "Jose"
```

```
In [58]: print('Hello, his name is {}'.format(name))
```

```
Hello, his name is Jose
```

```
In [ ]:
```



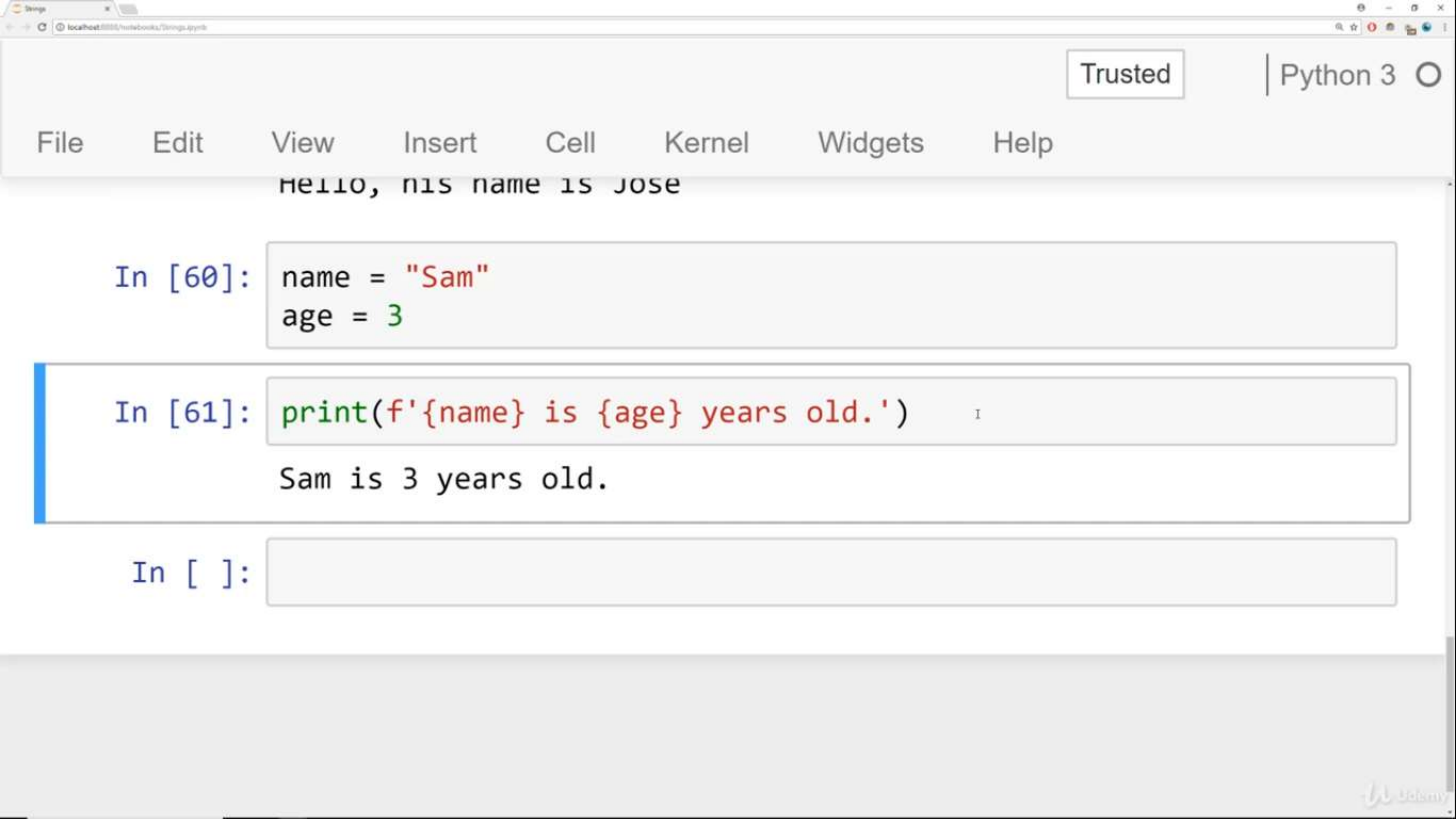
```
In [ ]:

In [57]: name =  "Jose"

In [59]: print(f'Hello, his name is {name}')

         Hello, his name is Jose

In [ ]:
```



```
Hello, his name is Jose
```

```
In [60]: name = "Sam"
         age = 3
```

```
In [61]: print(f'{name} is {age} years old.')
```

```
Sam is 3 years old.
```

```
In [ ]:
```