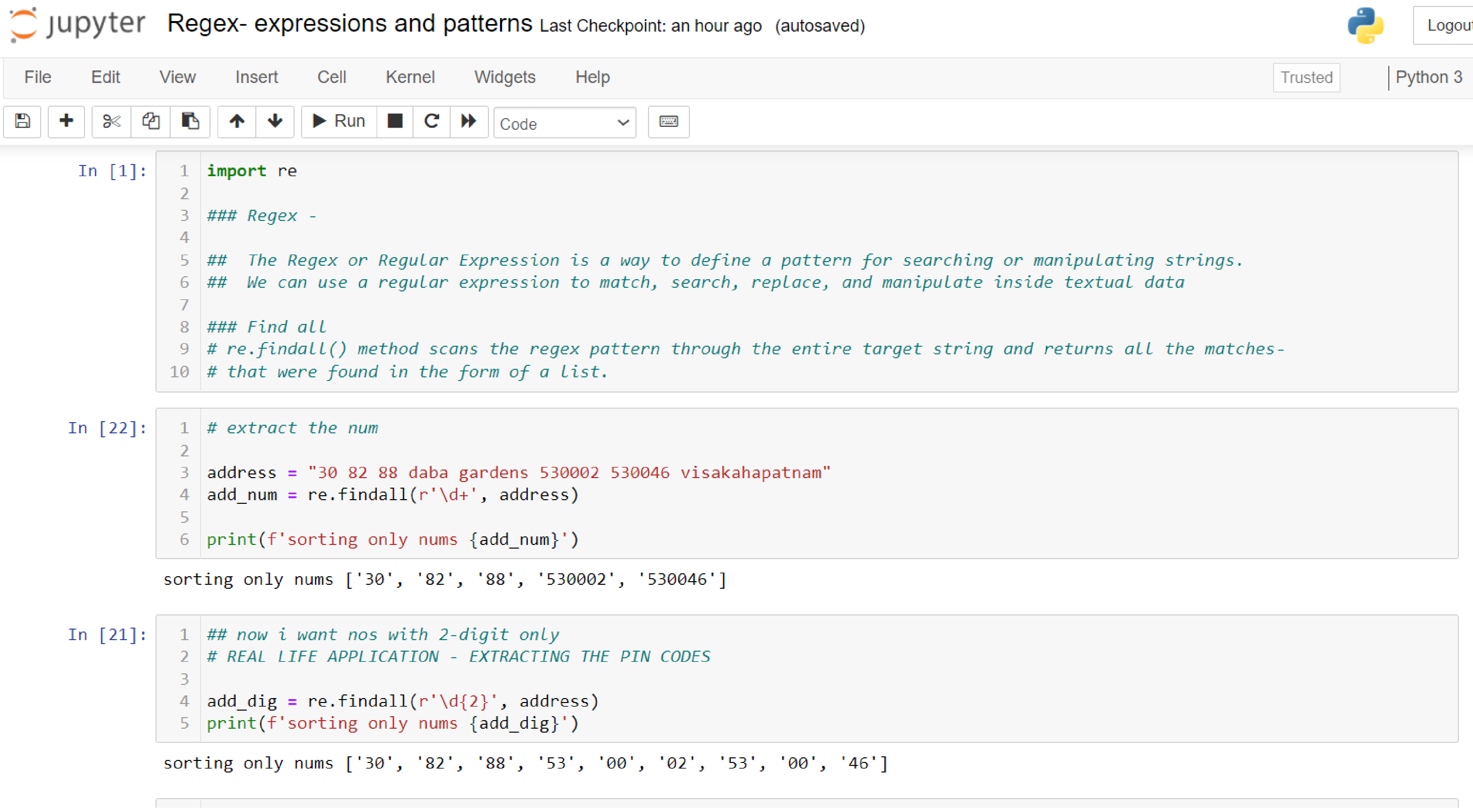

jupyter
Last Checkpoint: an hour ago (autosaved)
Logou
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

Regex- expressions and patterns

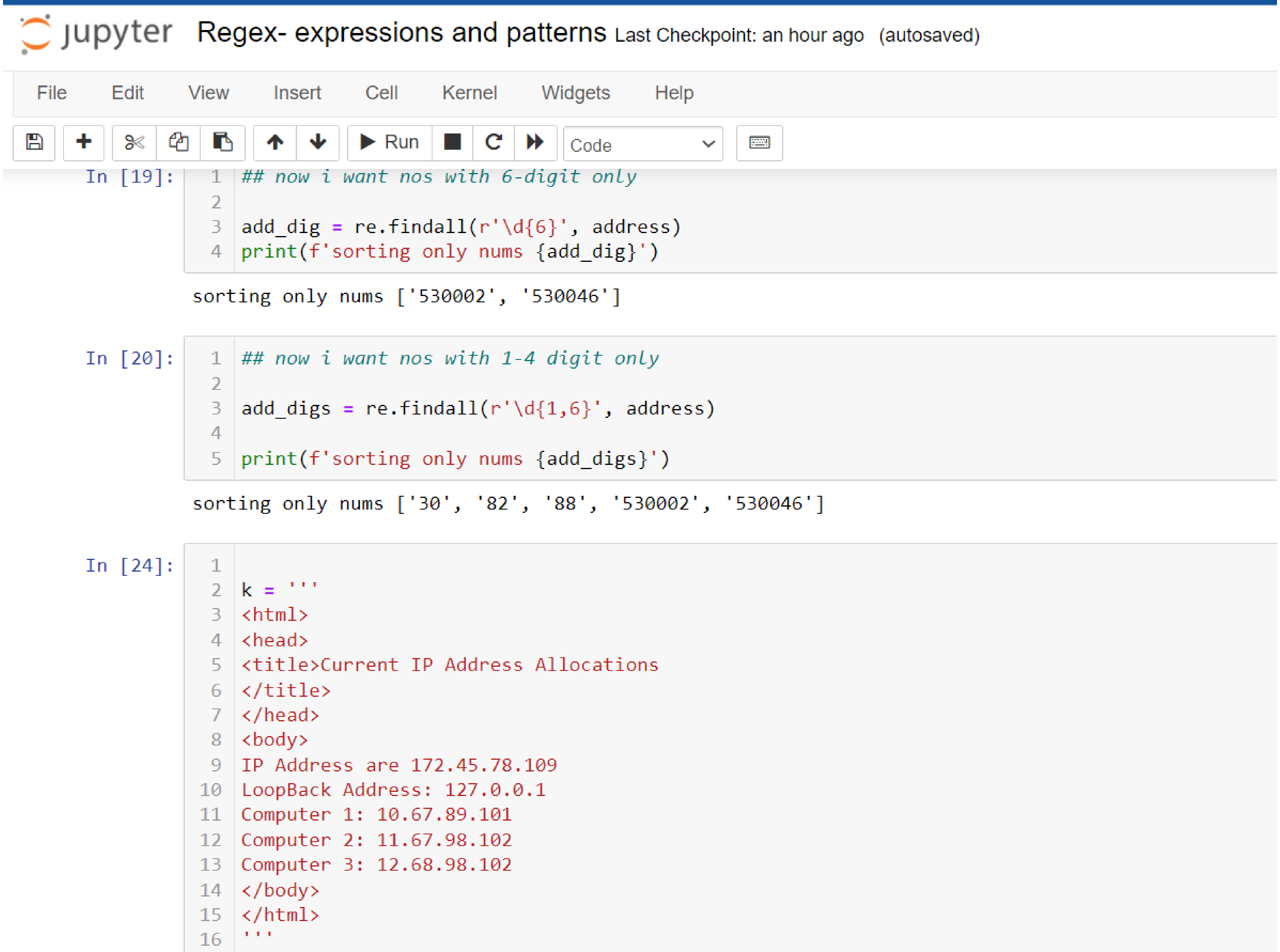
In [1]:
```python
import re

### Regex -

##  The Regex or Regular Expression is a way to define a pattern for searching or manipulating strings.
##  We can use a regular expression to match, search, replace, and manipulate inside textual data

### Find all
# re.findall() method scans the regex pattern through the entire target string and returns all the matches-
# that were found in the form of a list.
```

In [22]:
```python
# extract the num

address = "30 82 88 daba gardens 530002 530046 visakahapatnam"
add_num = re.findall(r'\d+', address)

print(f'sorting only nums {add_num}')
```

```
sorting only nums ['30', '82', '88', '530002', '530046']
```

In [21]:
```python
## now i want nos with 2-digit only
# REAL LIFE APPLICATION - EXTRACTING THE PIN CODES

add_dig = re.findall(r'\d{2}', address)
print(f'sorting only nums {add_dig}')
```

```
sorting only nums ['30', '82', '88', '53', '00', '02', '53', '00', '46']
```


jupyter
Regex- expressions and patterns Last Checkpoint: an hour ago (autosaved)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Code

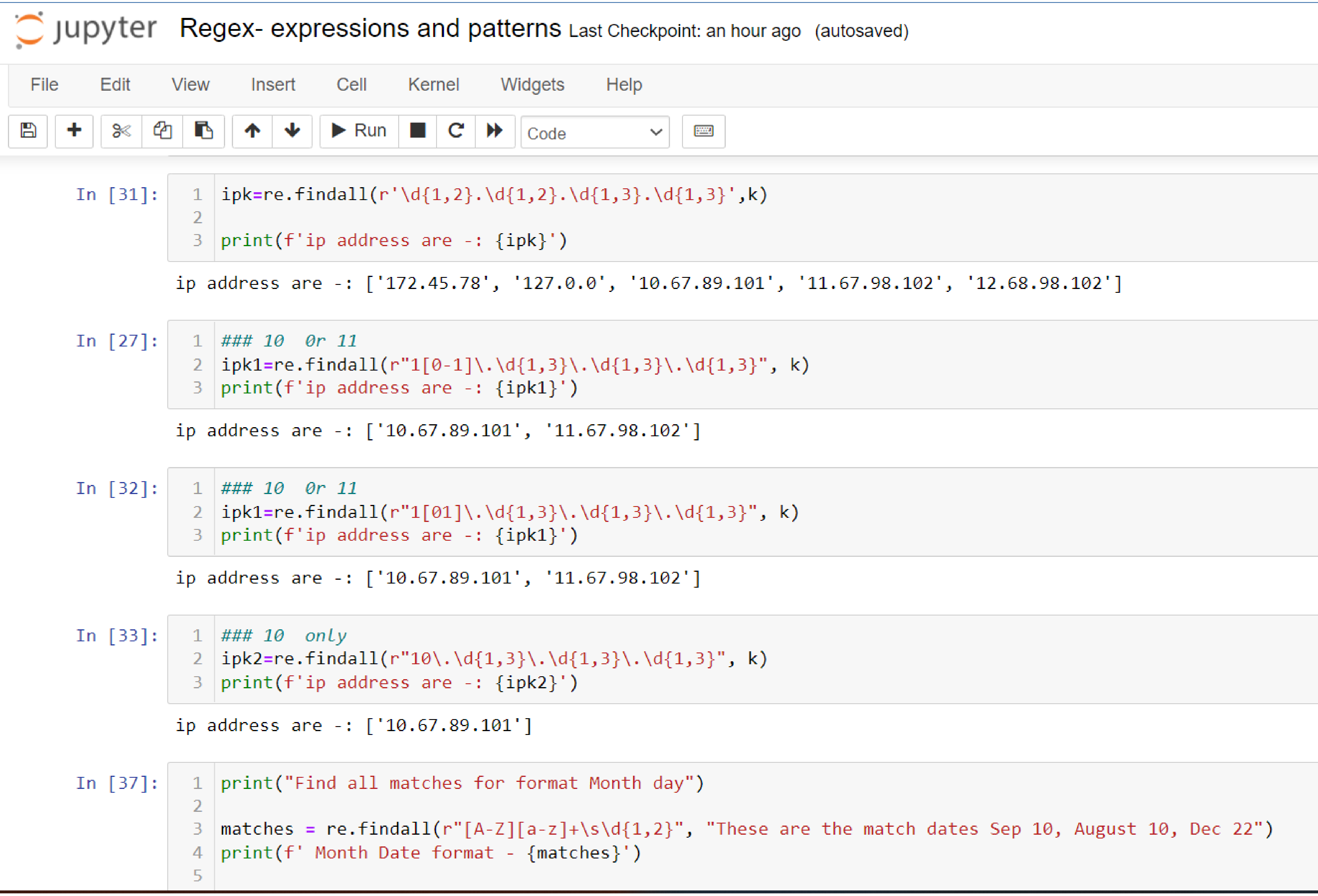
In [19]:
```python
## now i want nos with 6-digit only

add_dig = re.findall(r'\d{6}', address)
print(f'sorting only nums {add_dig}')
```

```
sorting only nums ['530002', '530046']
```

In [20]:
```python
## now i want nos with 1-4 digit only

add_digs = re.findall(r'\d{1,6}', address)

print(f'sorting only nums {add_digs}')
```

```
sorting only nums ['30', '82', '88', '530002', '530046']
```

In [24]:
```python

k = '''
<html>
<head>
<title>Current IP Address Allocations
</title>
</head>
<body>
IP Address are 172.45.78.109
LoopBack Address: 127.0.0.1
Computer 1: 10.67.89.101
Computer 2: 11.67.98.102
Computer 3: 12.68.98.102
</body>
</html>
'''
```


jupyter
Regex- expressions and patterns Last Checkpoint: an hour ago (autosaved)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Code

In [31]:
```
ipk=re.findall(r'\d{1,2}.\d{1,2}.\d{1,3}.\d{1,3}',k)

print(f'ip address are -: {ipk}')
```

```
ip address are -: ['172.45.78', '127.0.0', '10.67.89.101', '11.67.98.102', '12.68.98.102']
```

In [27]:
```
### 10  Or 11
ipk1=re.findall(r"1[0-1]\.\d{1,3}\.\d{1,3}\.\d{1,3}", k)
print(f'ip address are -: {ipk1}')
```

```
ip address are -: ['10.67.89.101', '11.67.98.102']
```

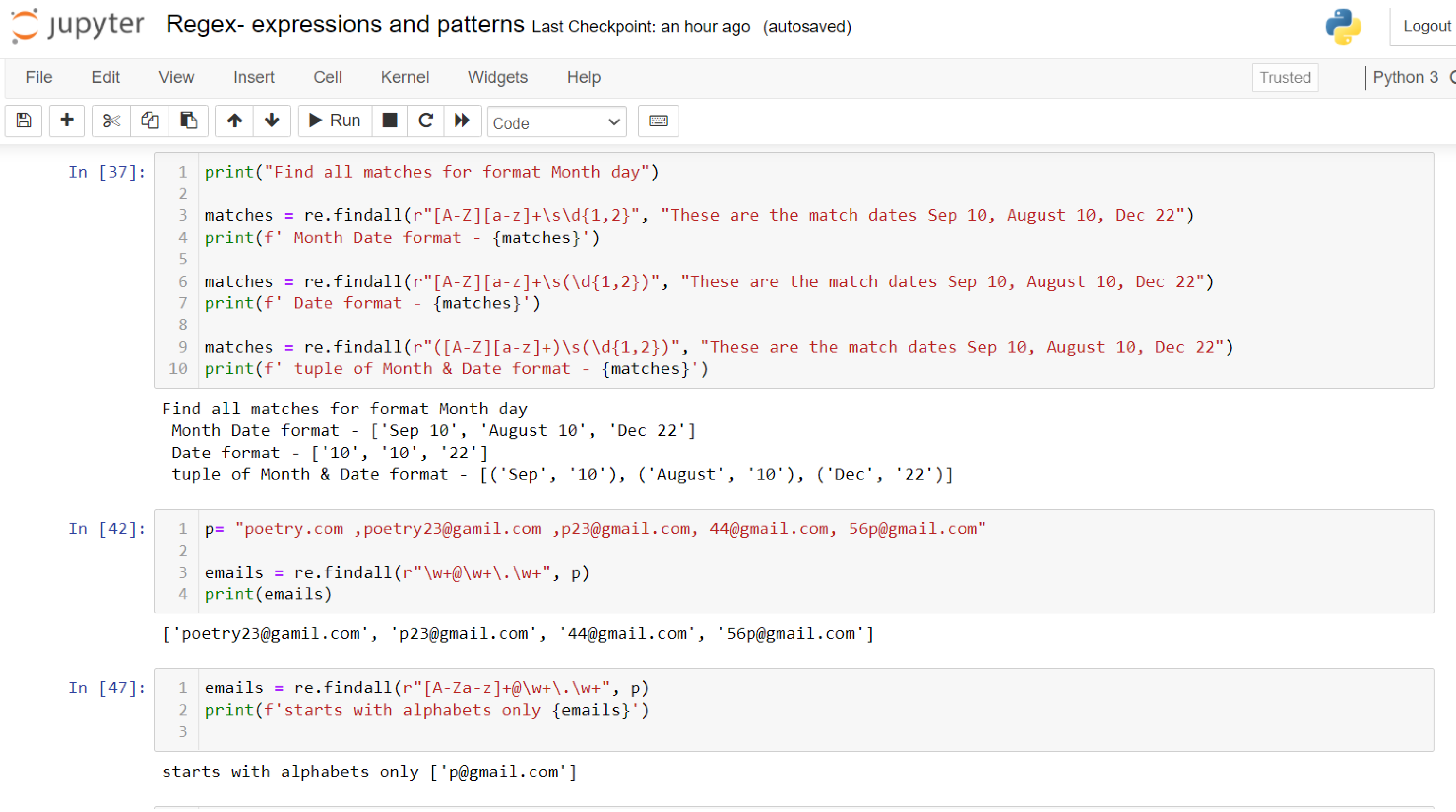
In [32]:
```
### 10  Or 11
ipk1=re.findall(r"1[01]\.\d{1,3}\.\d{1,3}\.\d{1,3}", k)
print(f'ip address are -: {ipk1}')
```

```
ip address are -: ['10.67.89.101', '11.67.98.102']
```

In [33]:
```
### 10  only
ipk2=re.findall(r"10\.\d{1,3}\.\d{1,3}\.\d{1,3}", k)
print(f'ip address are -: {ipk2}')
```

```
ip address are -: ['10.67.89.101']
```

In [37]:
```
print("Find all matches for format Month day")

matches = re.findall(r"[A-Z][a-z]+\s\d{1,2}", "These are the match dates Sep 10, August 10, Dec 22")
print(f' Month Date format - {matches}')

```


jupyter
Regex- expressions and patterns Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

In [37]:
```python
print("Find all matches for format Month day")

matches = re.findall(r"[A-Z][a-z]+\s\d{1,2}", "These are the match dates Sep 10, August 10, Dec 22")
print(f' Month Date format - {matches}')

matches = re.findall(r"[A-Z][a-z]+\s(\d{1,2})", "These are the match dates Sep 10, August 10, Dec 22")
print(f' Date format - {matches}')

matches = re.findall(r"([A-Z][a-z]+)\s(\d{1,2})", "These are the match dates Sep 10, August 10, Dec 22")
print(f' tuple of Month & Date format - {matches}')
```

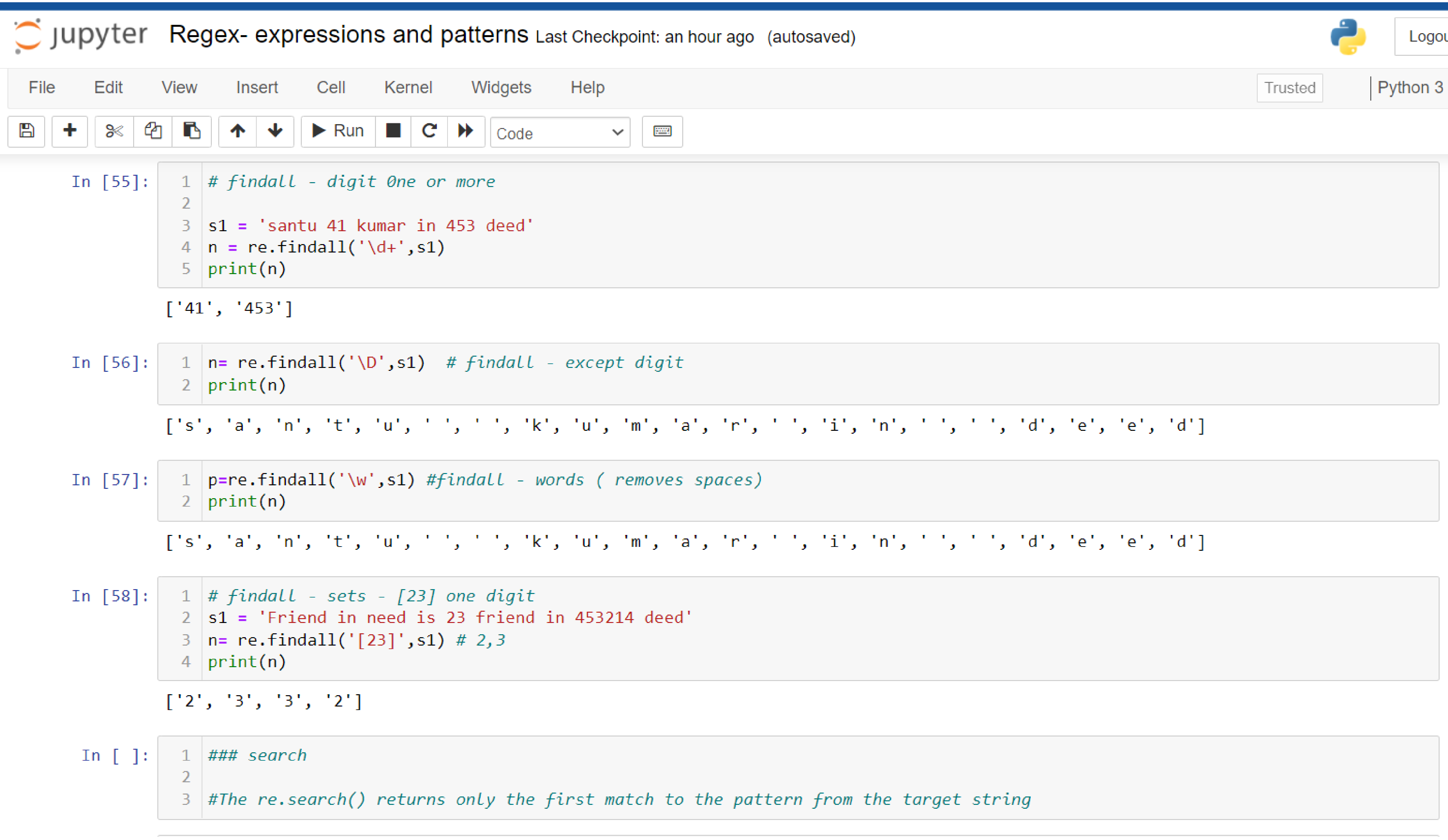
```
Find all matches for format Month day
 Month Date format - ['Sep 10', 'August 10', 'Dec 22']
 Date format - ['10', '10', '22']
 tuple of Month & Date format - [('Sep', '10'), ('August', '10'), ('Dec', '22')]
```

In [42]:
```python
p= "poetry.com ,poetry23@gamil.com ,p23@gmail.com, 44@gmail.com, 56p@gmail.com"

emails = re.findall(r"\w+@\w+\.\w+", p)
print(emails)
```

```
['poetry23@gamil.com', 'p23@gmail.com', '44@gmail.com', '56p@gmail.com']
```

In [47]:
```python
emails = re.findall(r"[A-Za-z]+@\w+\.\w+", p)
print(f'starts with alphabets only {emails}')

```

```
starts with alphabets only ['p@gmail.com']
```


jupyter
Regex- expressions and patterns Last Checkpoint: an hour ago (autosaved)
Logou
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

In [55]:
```
# findall - digit One or more

s1 = 'santu 41 kumar in 453 deed'
n = re.findall('\d+',s1)
print(n)
```

```
['41', '453']
```

In [56]:
```
n= re.findall('\D',s1)  # findall - except digit
print(n)
```

```
['s', 'a', 'n', 't', 'u', ' ', ' ', 'k', 'u', 'm', 'a', 'r', ' ', 'i', 'n', ' ', ' ', 'd', 'e', 'e', 'd']
```

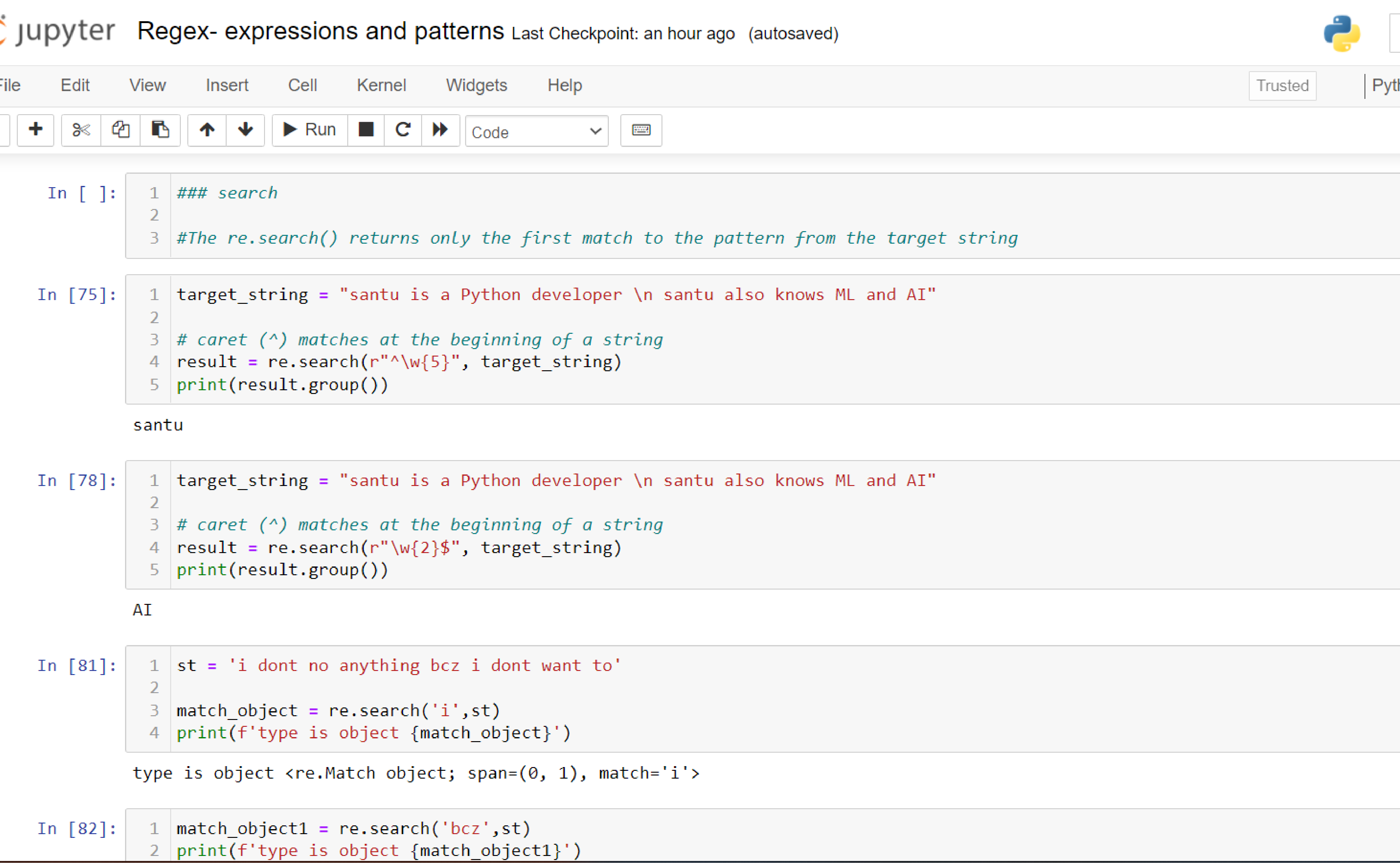
In [57]:
```
p=re.findall('\w',s1) #findall - words ( removes spaces)
print(n)
```

```
['s', 'a', 'n', 't', 'u', ' ', ' ', 'k', 'u', 'm', 'a', 'r', ' ', 'i', 'n', ' ', ' ', 'd', 'e', 'e', 'd']
```

In [58]:
```
# findall - sets - [23] one digit
s1 = 'Friend in need is 23 friend in 453214 deed'
n= re.findall('[23]',s1) # 2,3
print(n)
```

```
['2', '3', '3', '2']
```

In [ ]:
```
### search

#The re.search() returns only the first match to the pattern from the target string
```


jupyter
Regex- expressions and patterns Last Checkpoint: an hour ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Run
Code

In [ ]:
```
### search

#The re.search() returns only the first match to the pattern from the target string
```

In [75]:
```
target_string = "santu is a Python developer \n santu also knows ML and AI"

# caret (^) matches at the beginning of a string
result = re.search(r"^\w{5}", target_string)
print(result.group())
```

```
santu
```

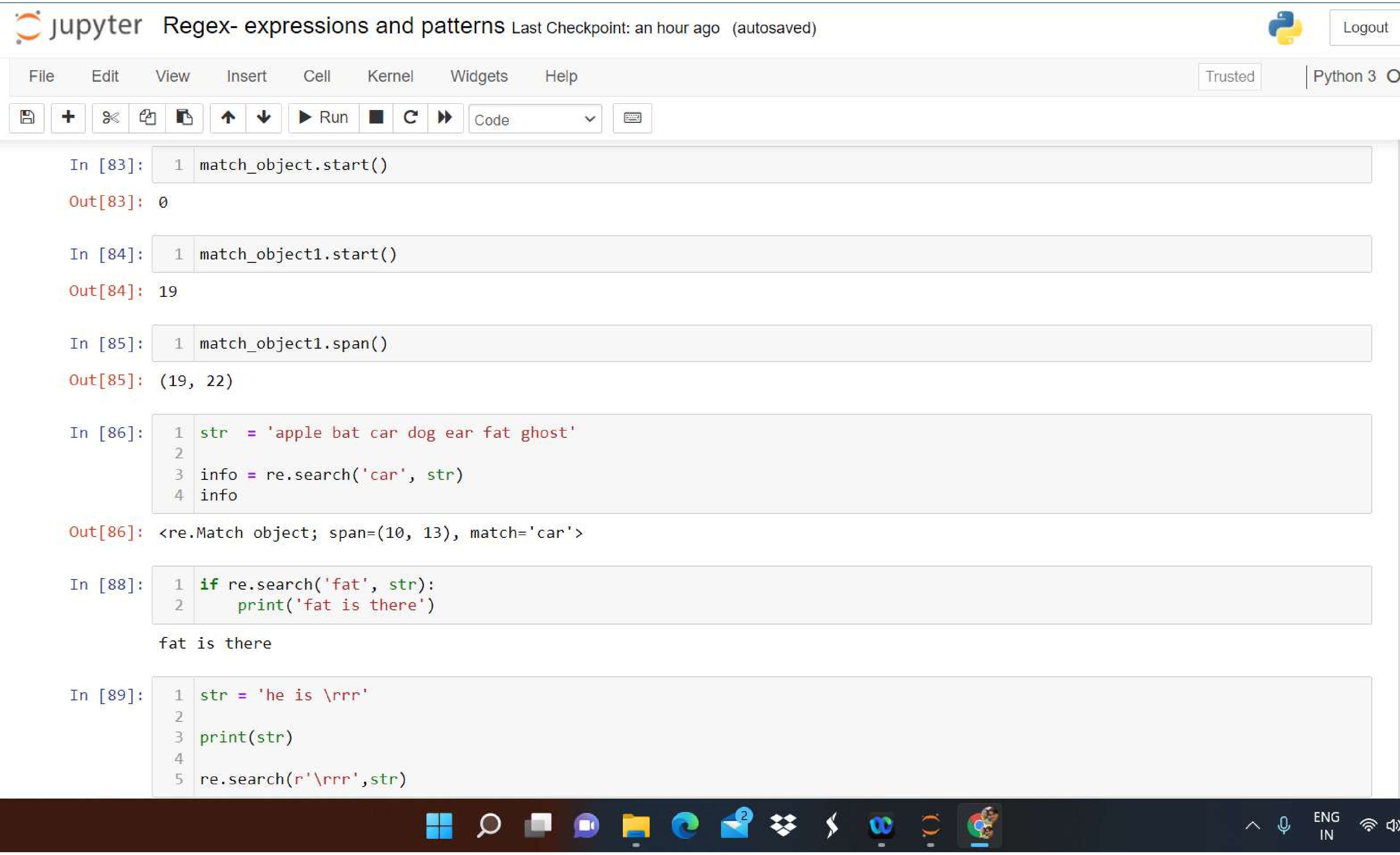
In [78]:
```
target_string = "santu is a Python developer \n santu also knows ML and AI"

# caret (^) matches at the beginning of a string
result = re.search(r"\w{2}$", target_string)
print(result.group())
```

```
AI
```

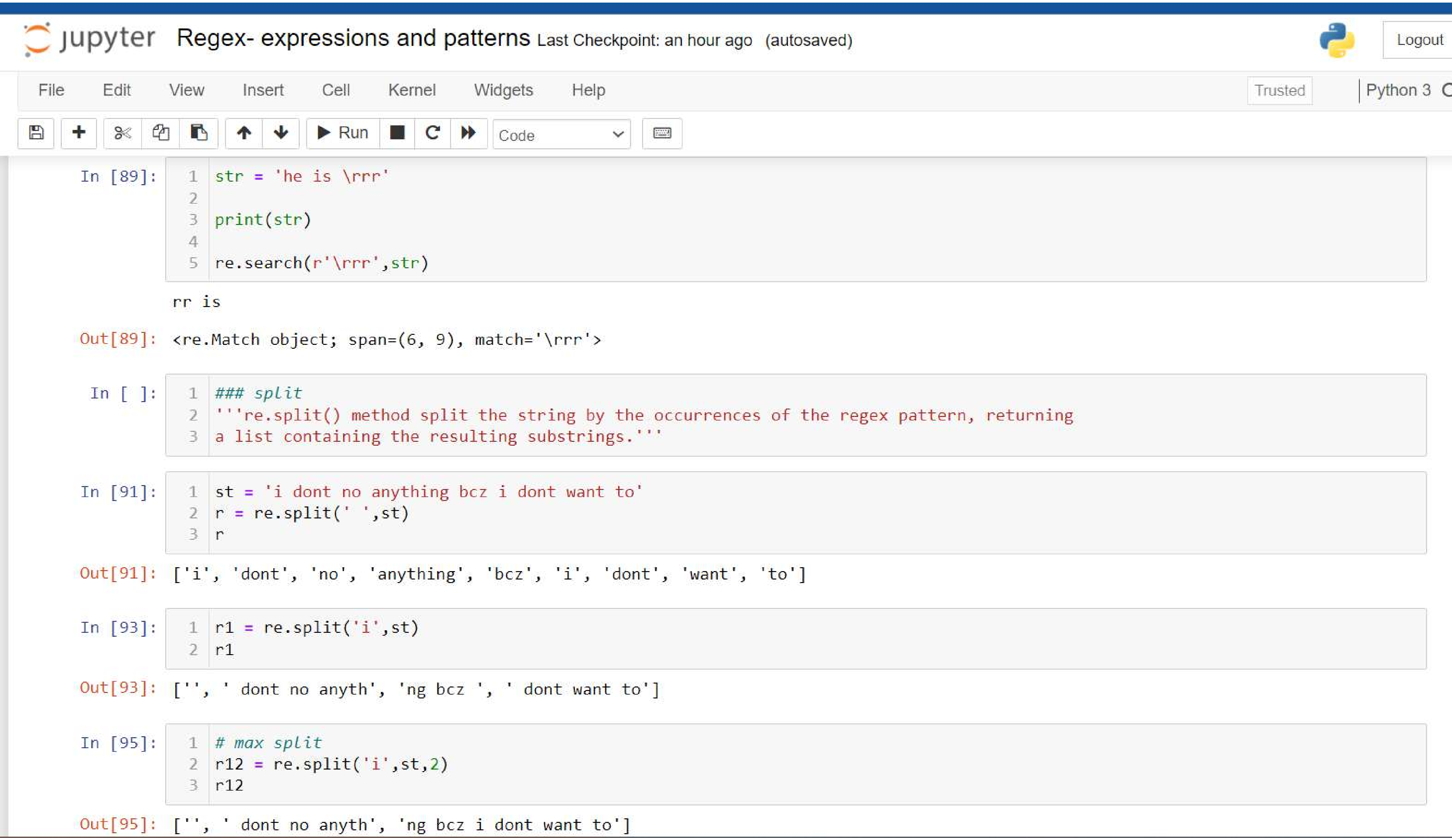
In [81]:
```
st = 'i dont no anything bcz i dont want to'

match_object = re.search('i',st)
print(f'type is object {match_object}')
```

```
type is object <re.Match object; span=(0, 1), match='i'>
```

In [82]:
```
match_object1 = re.search('bcz',st)
print(f'type is object {match_object1}')
```


jupyter
Regex- expressions and patterns Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [83]: match_object.start()

Out[83]: 0

In [84]: match_object1.start()

Out[84]: 19

In [85]: match_object1.span()

Out[85]: (19, 22)

In [86]: str  = 'apple bat car dog ear fat ghost'

         info = re.search('car', str)
         info

Out[86]: <re.Match object; span=(10, 13), match='car'>

In [88]: if re.search('fat', str):
             print('fat is there')

         fat is there

In [89]: str = 'he is \rrr'

         print(str)

         re.search(r'\rrr',str)
```


jupyter
Regex- expressions and patterns Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

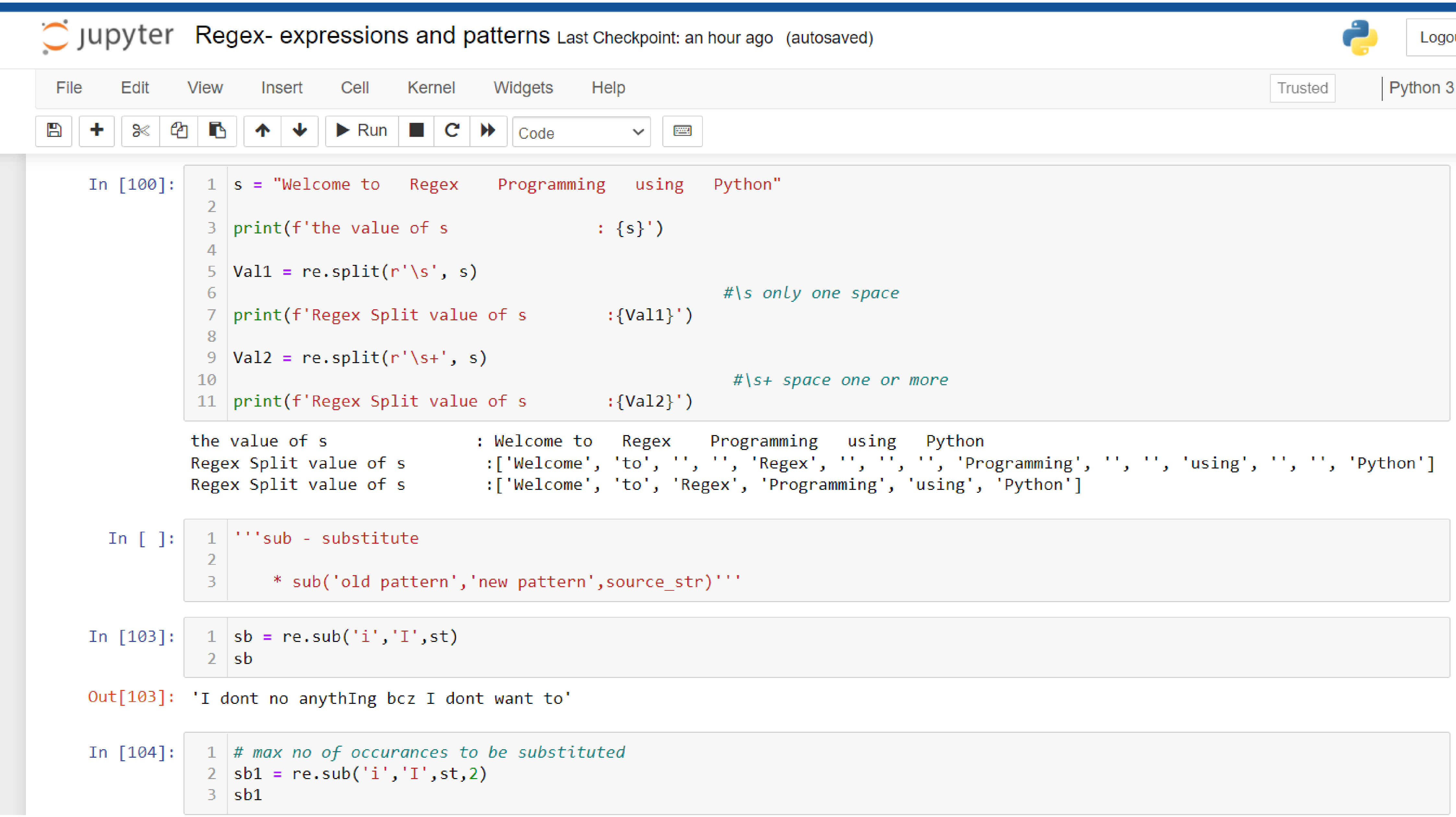
```
In [89]:  str = 'he is \rrr'

          print(str)

          re.search(r'\rrr',str)

rr is

Out[89]: <re.Match object; span=(6, 9), match='\rrr'>
```

```
In [ ]:   ### split
          '''re.split() method split the string by the occurrences of the regex pattern, returning
          a list containing the resulting substrings.'''
```

```
In [91]:  st = 'i dont no anything bcz i dont want to'
          r = re.split(' ',st)
          r

Out[91]: ['i', 'dont', 'no', 'anything', 'bcz', 'i', 'dont', 'want', 'to']
```

```
In [93]:  r1 = re.split('i',st)
          r1

Out[93]: ['', ' dont no anyth', 'ng bcz ', ' dont want to']
```

```
In [95]:  # max split
          r12 = re.split('i',st,2)
          r12

Out[95]: ['', ' dont no anyth', 'ng bcz i dont want to']
```


jupyter Regex- expressions and patterns Last Checkpoint: an hour ago (autosaved)
Logo
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

In [100]:
```
s = "Welcome to   Regex    Programming   using   Python"

print(f'the value of s                    : {s}')

Val1 = re.split(r'\s', s)
                                                  #\s only one space
print(f'Regex Split value of s        :{Val1}')

Val2 = re.split(r'\s+', s)
                                                   #\s+ space one or more
print(f'Regex Split value of s        :{Val2}')
```

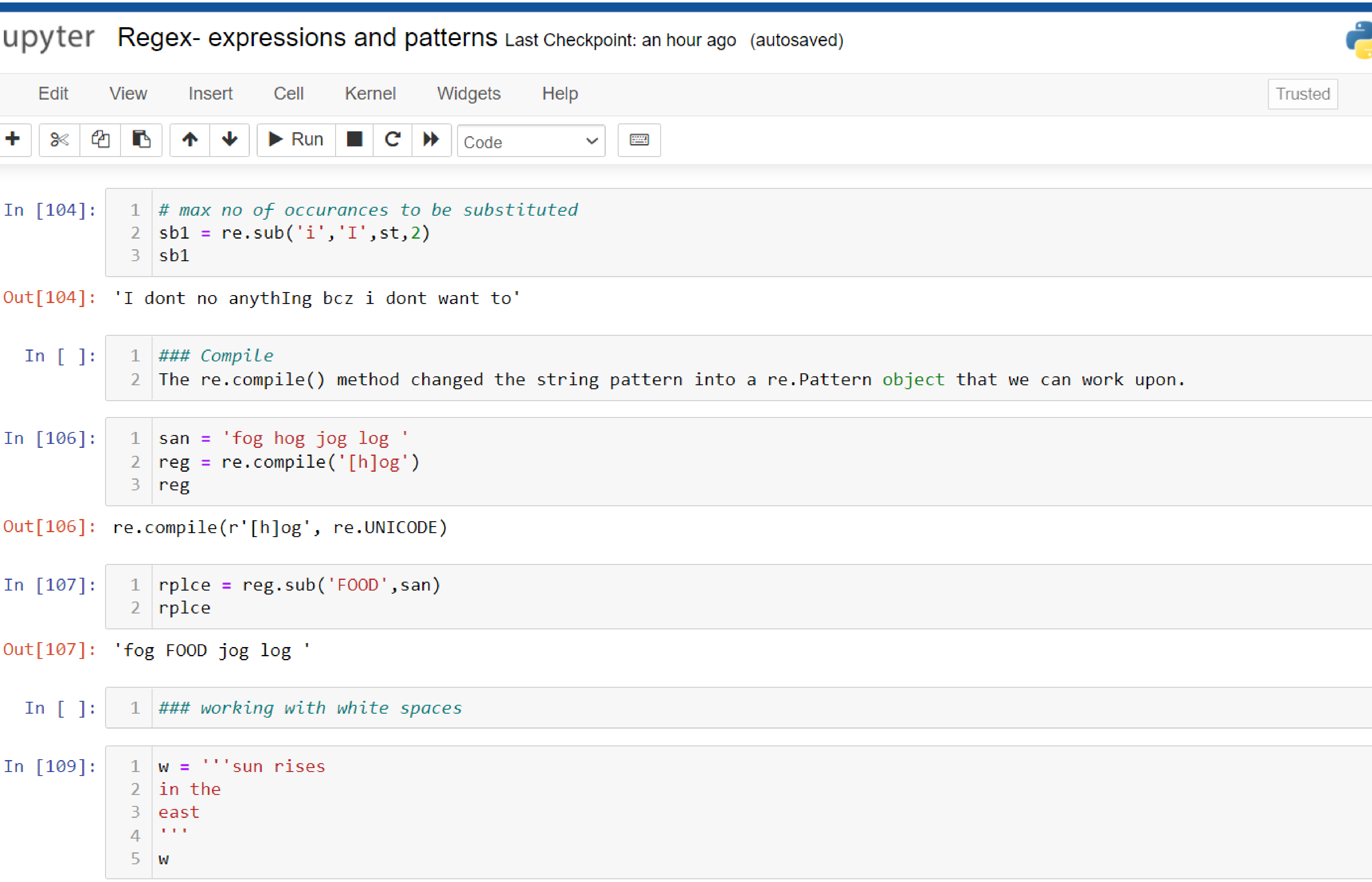
```
the value of s                    : Welcome to   Regex    Programming   using   Python
Regex Split value of s         :['Welcome', 'to', '', '', 'Regex', '', '', '', 'Programming', '', '', 'using', '', '', 'Python']
Regex Split value of s         :['Welcome', 'to', 'Regex', 'Programming', 'using', 'Python']
```

In [ ]:
```
'''sub - substitute

    * sub('old pattern','new pattern',source_str)'''
```

In [103]:
```
sb = re.sub('i','I',st)
sb
```

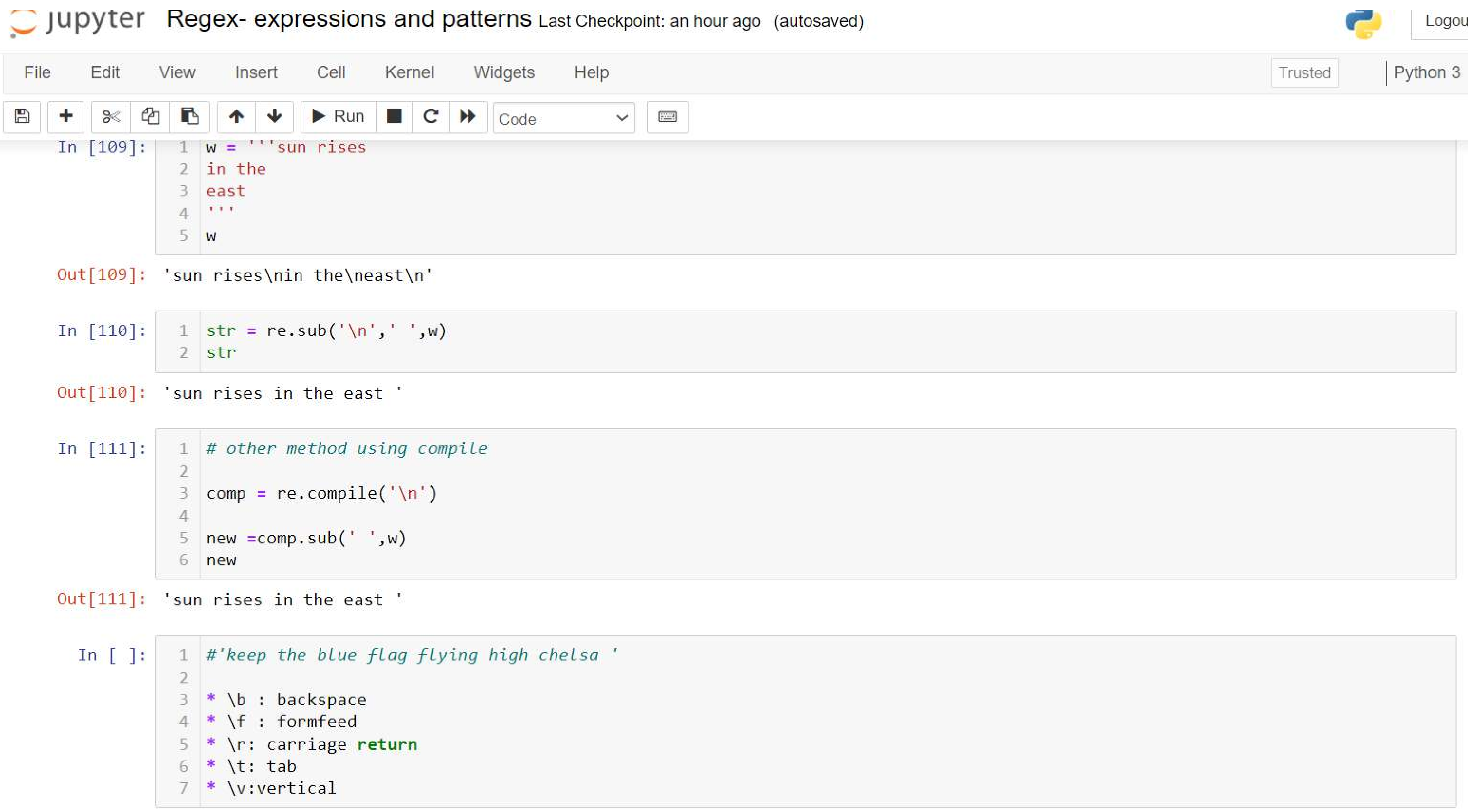
Out[103]:
```
'I dont no anythIng bcz I dont want to'
```

In [104]:
```
# max no of occurances to be substituted
sb1 = re.sub('i','I',st,2)
sb1
```

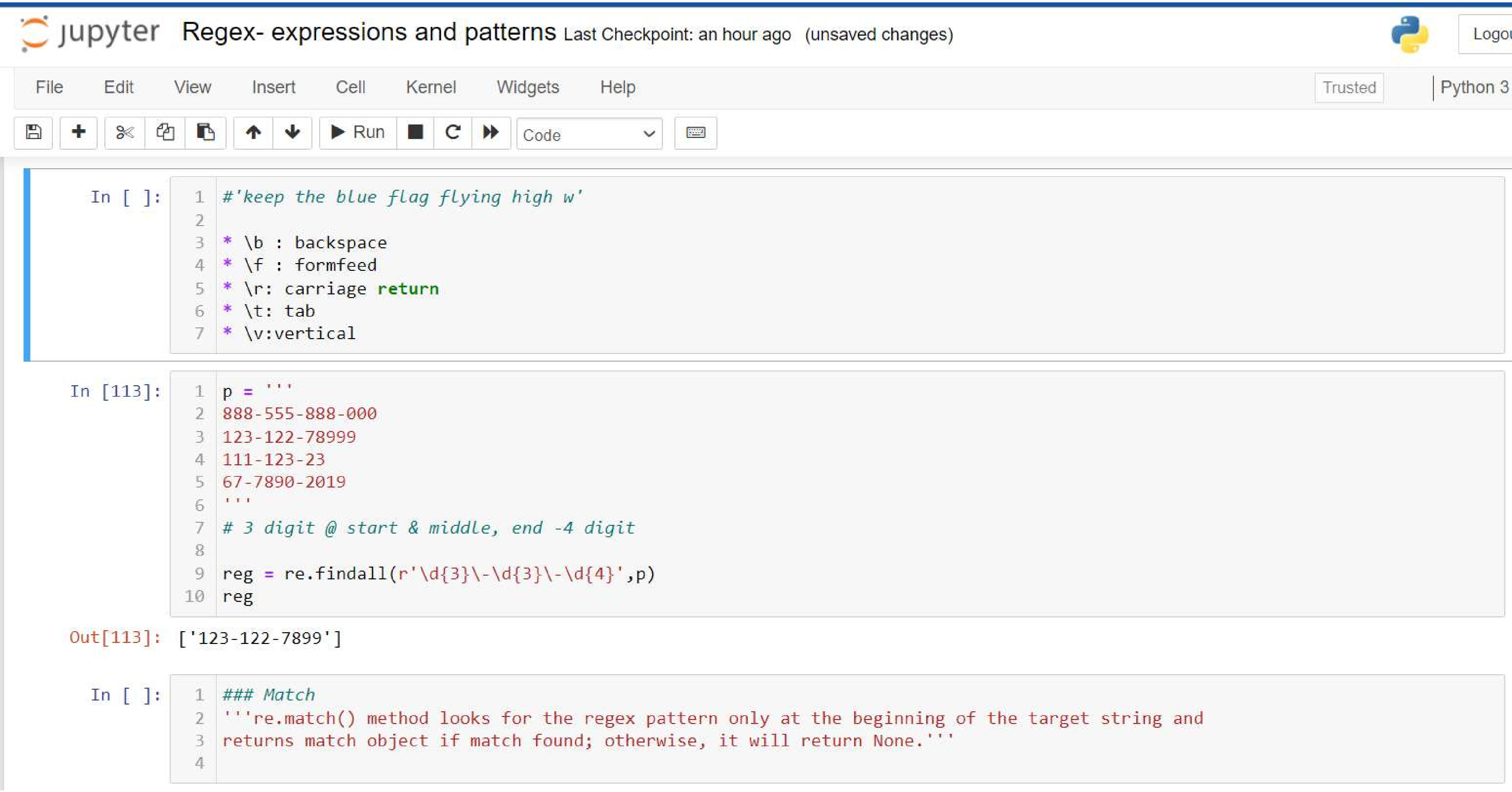


Regex- expressions and patterns Last Checkpoint: an hour ago (autosaved)

In [104]:
```
# max no of occurances to be substituted
sb1 = re.sub('i','I',st,2)
sb1
```
Out[104]:
```
'I dont no anythIng bcz i dont want to'
```
In [ ]:
```
### Compile
The re.compile() method changed the string pattern into a re.Pattern object that we can work upon.
```
In [106]:
```
san = 'fog hog jog log '
reg = re.compile('[h]og')
reg
```
Out[106]:
```
re.compile(r'[h]og', re.UNICODE)
```
In [107]:
```
rplce = reg.sub('FOOD',san)
rplce
```
Out[107]:
```
'fog FOOD jog log '
```
In [ ]:
```
### working with white spaces
```
In [109]:
```
w = '''sun rises
in the
east
'''
w
```


jupyter
Regex- expressions and patterns Last Checkpoint: an hour ago (autosaved)
Logou
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [109]: w = '''sun rises
          in the
          east
          '''
          w

Out[109]: 'sun rises\nin the\neast\n'

In [110]: str = re.sub('\n',' ',w)
          str

Out[110]: 'sun rises in the east '

In [111]: # other method using compile

          comp = re.compile('\n')

          new =comp.sub(' ',w)
          new

Out[111]: 'sun rises in the east '

In [ ]:   #'keep the blue flag flying high chelsa '

          * \b : backspace
          * \f : formfeed
          * \r: carriage return
          * \t: tab
          * \v:vertical
```

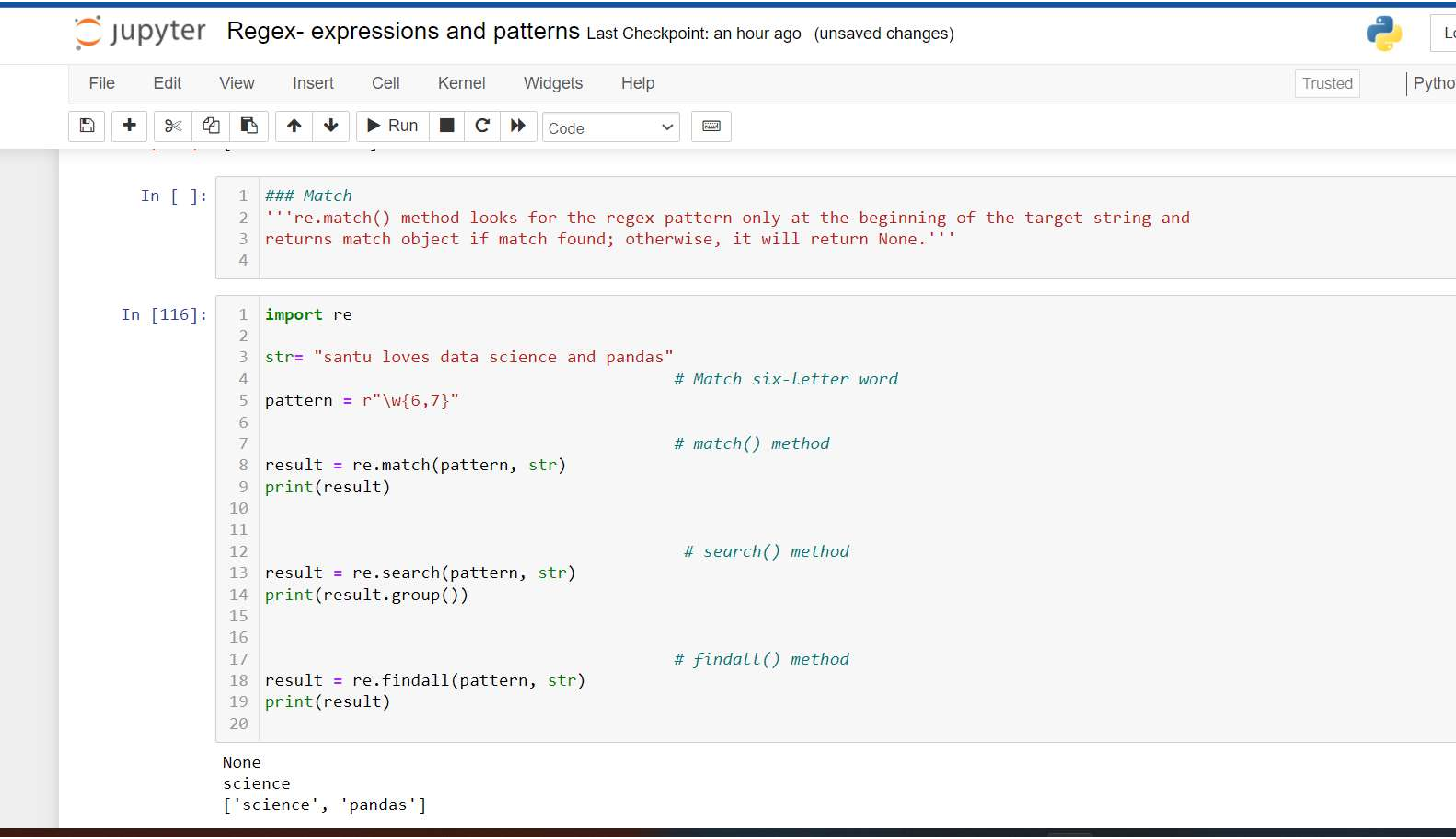

jupyter
Regex- expressions and patterns Last Checkpoint: an hour ago (unsaved changes)
Logou
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

In [ ]:
```
#'keep the blue flag flying high w'

* \b : backspace
* \f : formfeed
* \r: carriage return
* \t: tab
* \v:vertical
```

In [113]:
```
p = '''
888-555-888-000
123-122-78999
111-123-23
67-7890-2019
'''
# 3 digit @ start & middle, end -4 digit

reg = re.findall(r'\d{3}\-\d{3}\-\d{4}',p)
reg
```

Out[113]:
```
['123-122-7899']
```

In [ ]:
```
### Match
'''re.match() method looks for the regex pattern only at the beginning of the target string and
returns match object if match found; otherwise, it will return None.'''

```


jupyter
Regex- expressions and patterns Last Checkpoint: an hour ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Run
Code

In [ ]:
```
### Match
'''re.match() method looks for the regex pattern only at the beginning of the target string and
returns match object if match found; otherwise, it will return None.'''

```

In [116]:
```
import re

str= "santu loves data science and pandas"
                                          # Match six-letter word
pattern = r"\w{6,7}"

                                          # match() method
result = re.match(pattern, str)
print(result)


                                           # search() method
result = re.search(pattern, str)
print(result.group())


                                          # findall() method
result = re.findall(pattern, str)
print(result)

```

```
None
science
['science', 'pandas']
```