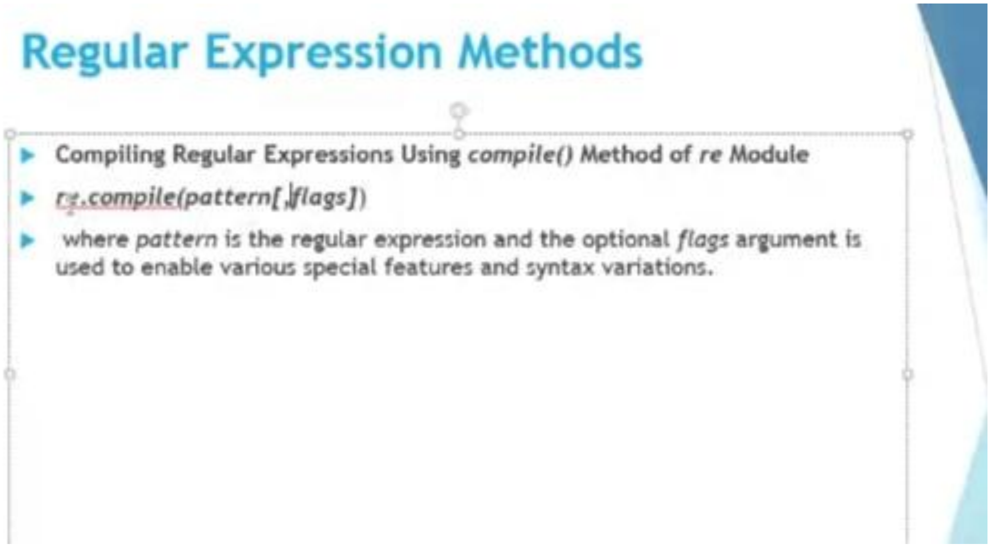
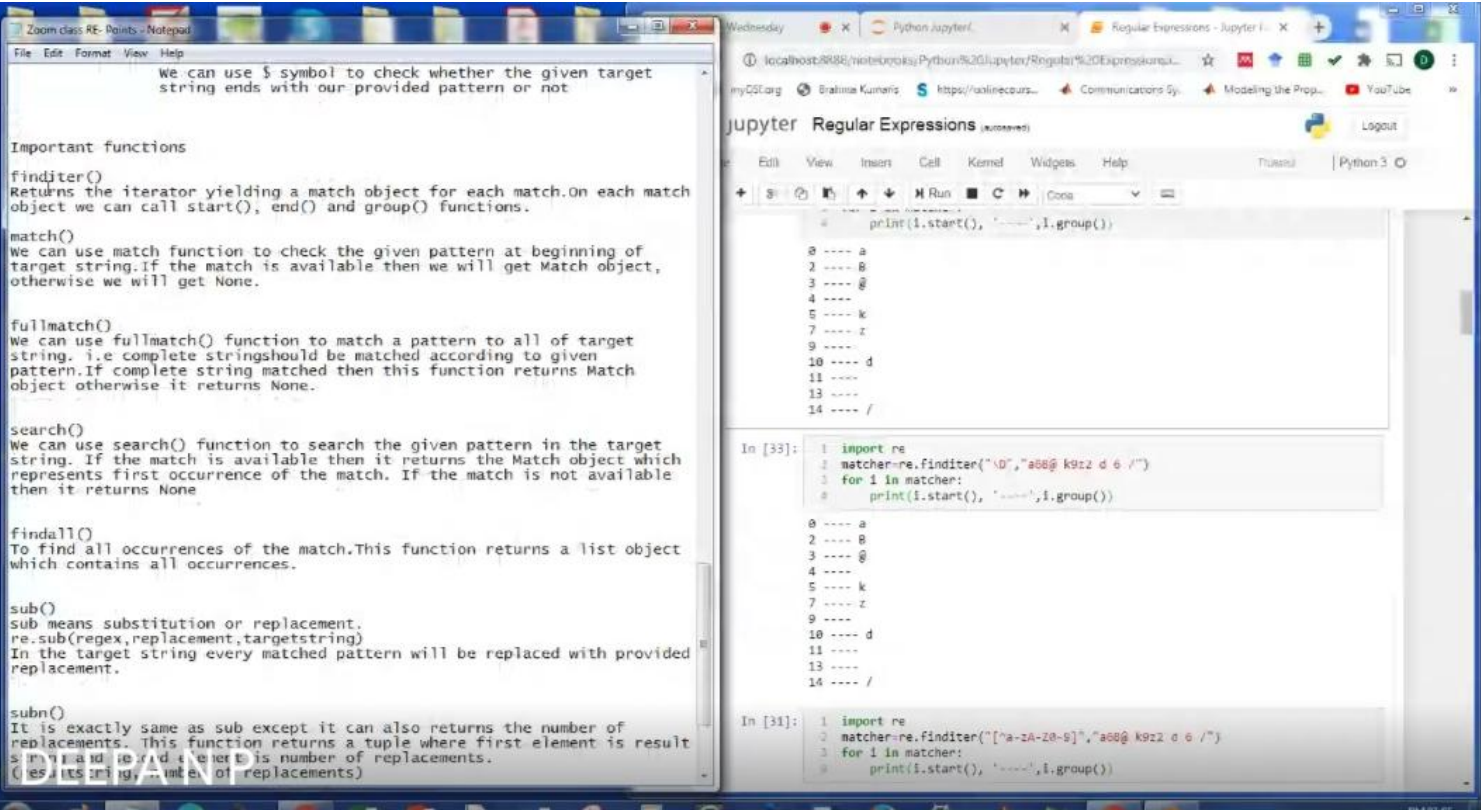


Regular Expression Methods

- **Compiling Regular Expressions Using *compile()* Method of *re* Module**
- *re.compile(pattern[,flags])*
- where *pattern* is the regular expression and the optional *flags* argument is used to enable various special features and syntax variations.

Steps- to build and use regular expressions

- In order to build and use regular expressions, perform the following steps:
- *Step 1*: Import *re* regular expression module.
- *Step 2*: Compile regular expression pattern using *re.compile()* method. This method returns the regular expression pattern as an object.
- *Step 3*: Invoke an appropriate method supported by the compiled regular expression object which returns a matched object instance containing information about matched strings.
- *Step 4*: Call methods (*group()* method is appropriate for most cases) associated with the matched object to display the results.



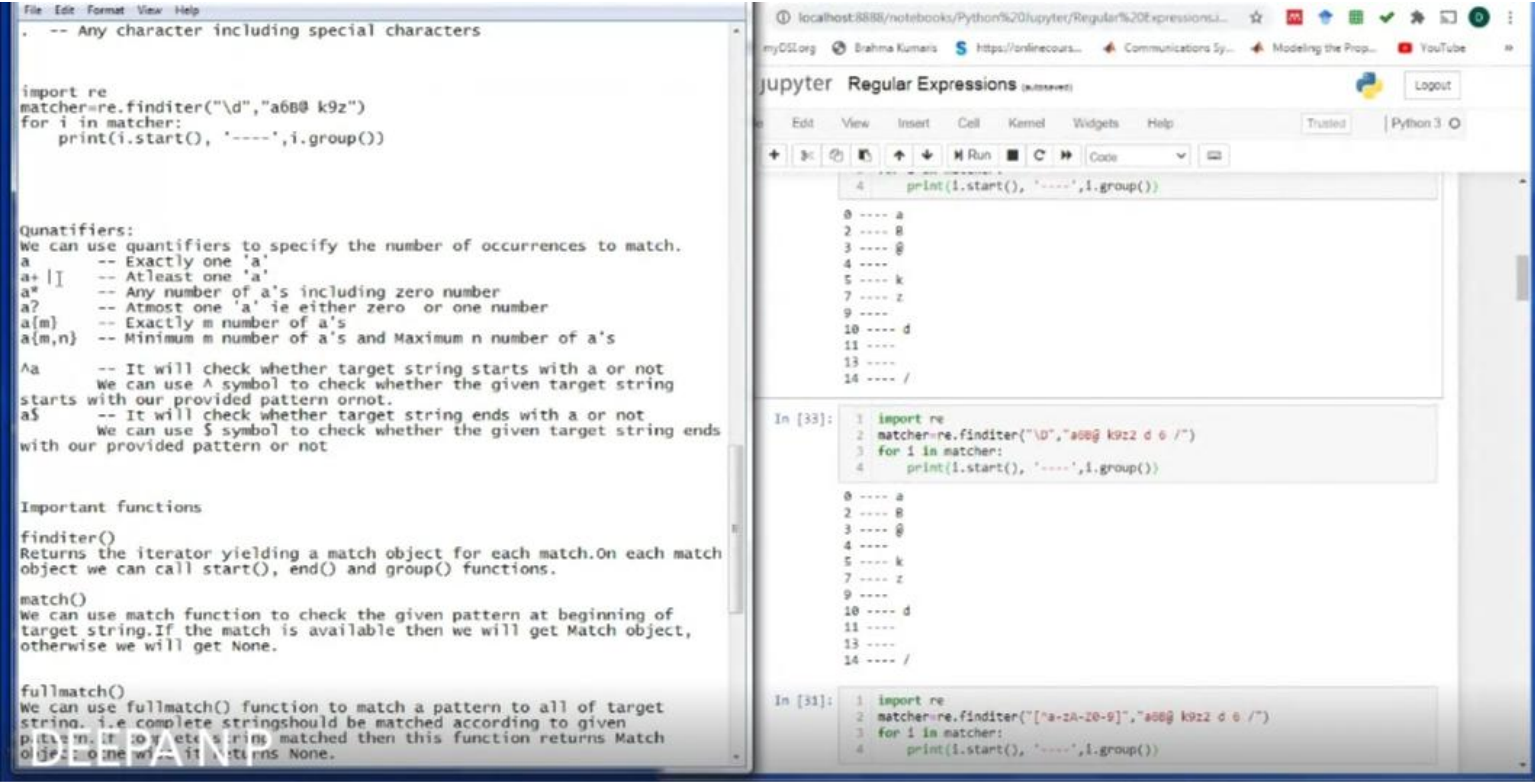
```
                 We can use $ symbol to check whether the given target
                 string ends with our provided pattern or not


Important functions

finditer()
Returns the iterator yielding a match object for each match.On each match
object we can call start(), end() and group() functions.

match()
We can use match function to check the given pattern at beginning of
target string.If the match is available then we will get Match object,
otherwise we will get None.


fullmatch()
We can use fullmatch() function to match a pattern to all of target
string. i.e complete stringshould be matched according to given
pattern.If complete string matched then this function returns Match
object otherwise it returns None.


search()
We can use search() function to search the given pattern in the target
string. If the match is available then it returns the Match object which
represents first occurrence of the match. If the match is not available
then it returns None


findall()
To find all occurrences of the match.This function returns a list object
which contains all occurrences.


sub()
sub means substitution or replacement.
re.sub(regex,replacement,targetstring)
In the target string every matched pattern will be replaced with provided
replacement.


subn()
It is exactly same as sub except it can also returns the number of
replacements. This function returns a tuple where first element is result
string and second element is number of replacements.
(resultstring, number of replacements)
```

Regular Expressions

```
        print(i.start(), '----',i.group())

0 ---- a
2 ---- @
3 ---- @
4 ----
5 ---- k
7 ---- z
9 ----
10 ---- d
11 ----
13 ----
14 ---- /
```

```
In [33]: import re
         matcher=re.finditer("\D","a68@ k9z2 d 6 /")
         for i in matcher:
             print(i.start(), '----',i.group())

0 ---- a
2 ---- @
3 ---- @
4 ----
5 ---- k
7 ---- z
9 ----
10 ---- d
11 ----
13 ----
14 ---- /
```

```
In [31]: import re
         matcher=re.finditer("[^a-zA-Z0-9]","a68@ k9z2 d 6 /")
         for i in matcher:
             print(i.start(), '----',i.group())
```



```
.  -- Any character including special characters


import re
matcher=re.finditer("\d","a6B@ k9z")
for i in matcher:
    print(i.start(), '----',i.group())




Qunatifiers:
We can use quantifiers to specify the number of occurrences to match.
a       -- Exactly one 'a'
a+      -- Atleast one 'a'
a*      -- Any number of a's including zero number
a?      -- Atmost one 'a' ie either zero  or one number
a{m}    -- Exactly m number of a's
a{m,n}  -- Minimum m number of a's and Maximum n number of a's

^a      -- It will check whether target string starts with a or not
        We can use ^ symbol to check whether the given target string
starts with our provided pattern ornot.
a$      -- It will check whether target string ends with a or not
        We can use $ symbol to check whether the given target string ends
with our provided pattern or not



Important functions

finditer()
Returns the iterator yielding a match object for each match.On each match
object we can call start(), end() and group() functions.

match()
We can use match function to check the given pattern at beginning of
target string.If the match is available then we will get Match object,
otherwise we will get None.


fullmatch()
We can use fullmatch() function to match a pattern to all of target
string. i.e complete stringshould be matched according to given
pattern. If complete string matched then this function returns Match
object otherwise it returns None.
```

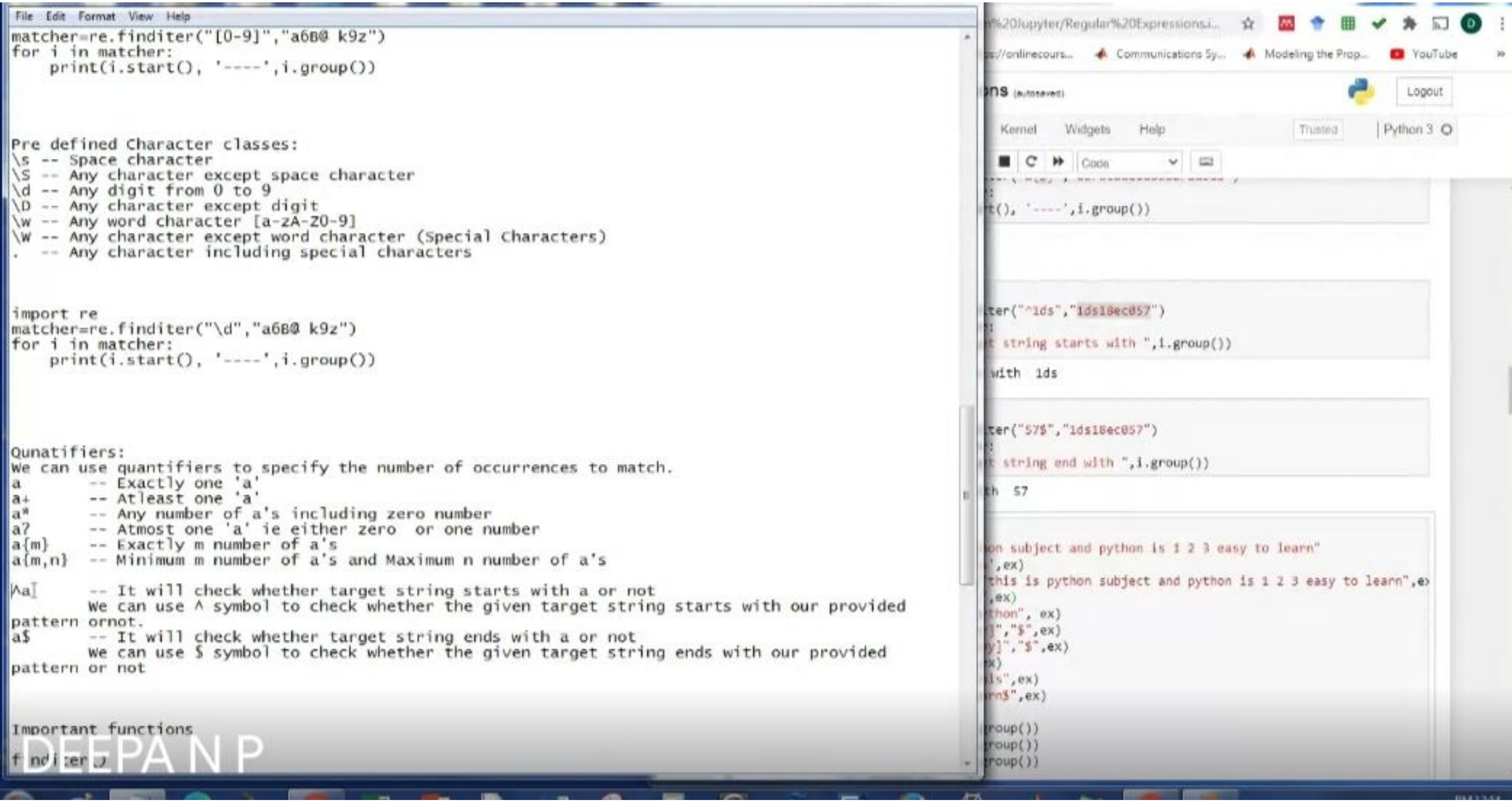
```
    print(i.start(), '----',i.group())
```

```
0 ---- a
2 ---- B
3 ---- @
4 ----  
5 ---- k
7 ---- z
9 ----  
10 ---- d
11 ----  
13 ----  
14 ---- /
```

```
In [33]:
import re
matcher=re.finditer("\D","a6B@ k9z2 d 6 /")
for i in matcher:
    print(i.start(), '----',i.group())
```

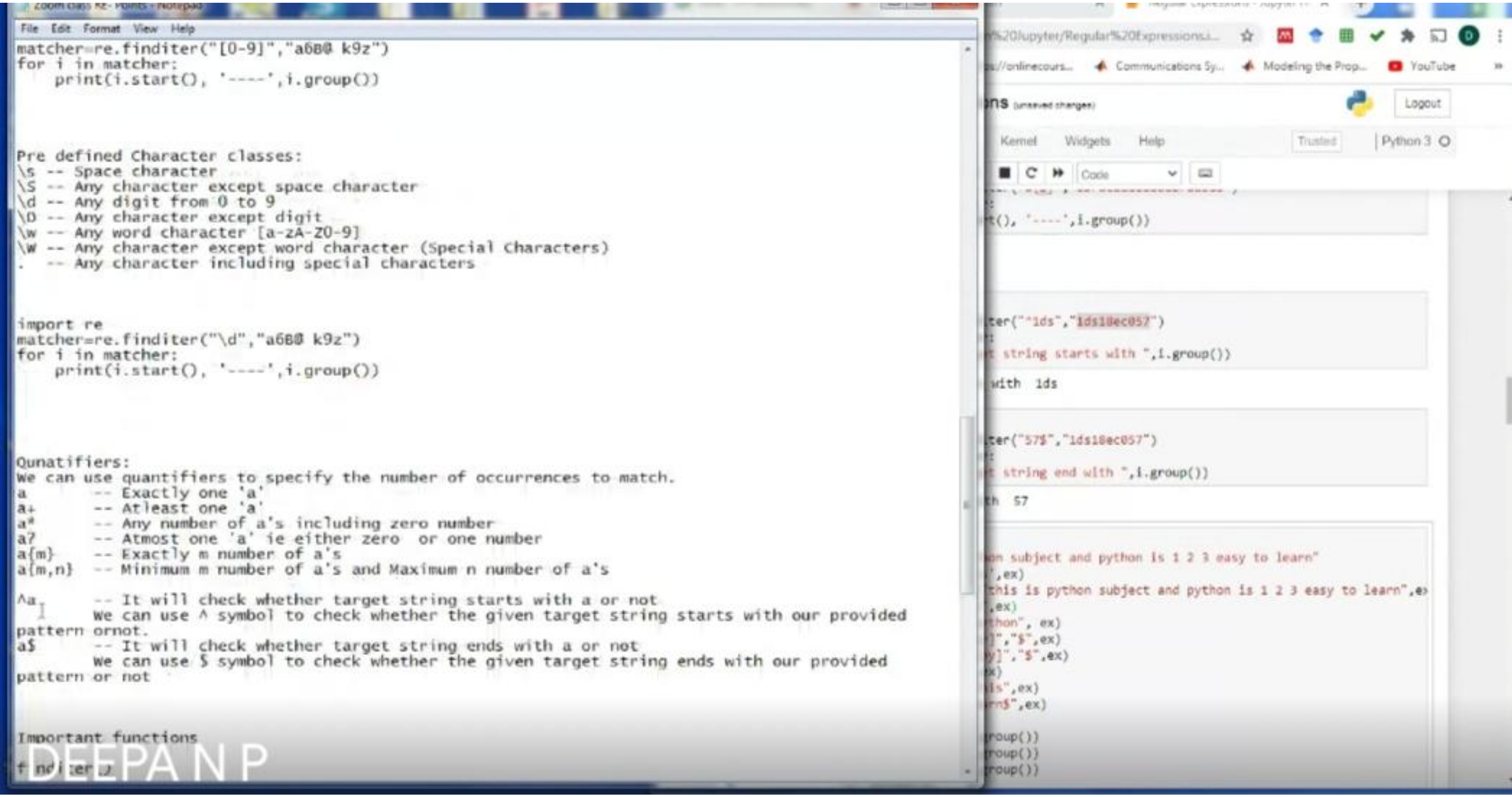
```
0 ---- a
2 ---- B
3 ---- @
4 ----  
5 ---- k
7 ---- z
9 ----  
10 ---- d
11 ----  
13 ----  
14 ---- /
```

```
In [31]:
import re
matcher=re.finditer("[^a-zA-Z0-9]","a6B@ k9z2 d 6 /")
for i in matcher:
    print(i.start(), '----',i.group())
```



```
matcher=re.finditer("[0-9]","a6B@ k9z")
for i in matcher:
    print(i.start(), '----',i.group())



Pre defined Character classes:
\s -- Space character
\S -- Any character except space character
\d -- Any digit from 0 to 9
\D -- Any character except digit
\w -- Any word character [a-zA-Z0-9]
\W -- Any character except word character (Special Characters)
.  -- Any character including special characters



import re
matcher=re.finditer("\d","a6B@ k9z")
for i in matcher:
    print(i.start(), '----',i.group())




Qunatifiers:
We can use quantifiers to specify the number of occurrences to match.
a       -- Exactly one 'a'
a+      -- Atleast one 'a'
a*      -- Any number of a's including zero number
a?      -- Atmost one 'a' ie either zero  or one number
a{m}    -- Exactly m number of a's
a{m,n}  -- Minimum m number of a's and Maximum n number of a's

^a      -- It will check whether target string starts with a or not
        We can use ^ symbol to check whether the given target string starts with our provided
pattern ornot.
a$      -- It will check whether target string ends with a or not
        We can use $ symbol to check whether the given target string ends with our provided
pattern or not


Important functions

finditer()
```

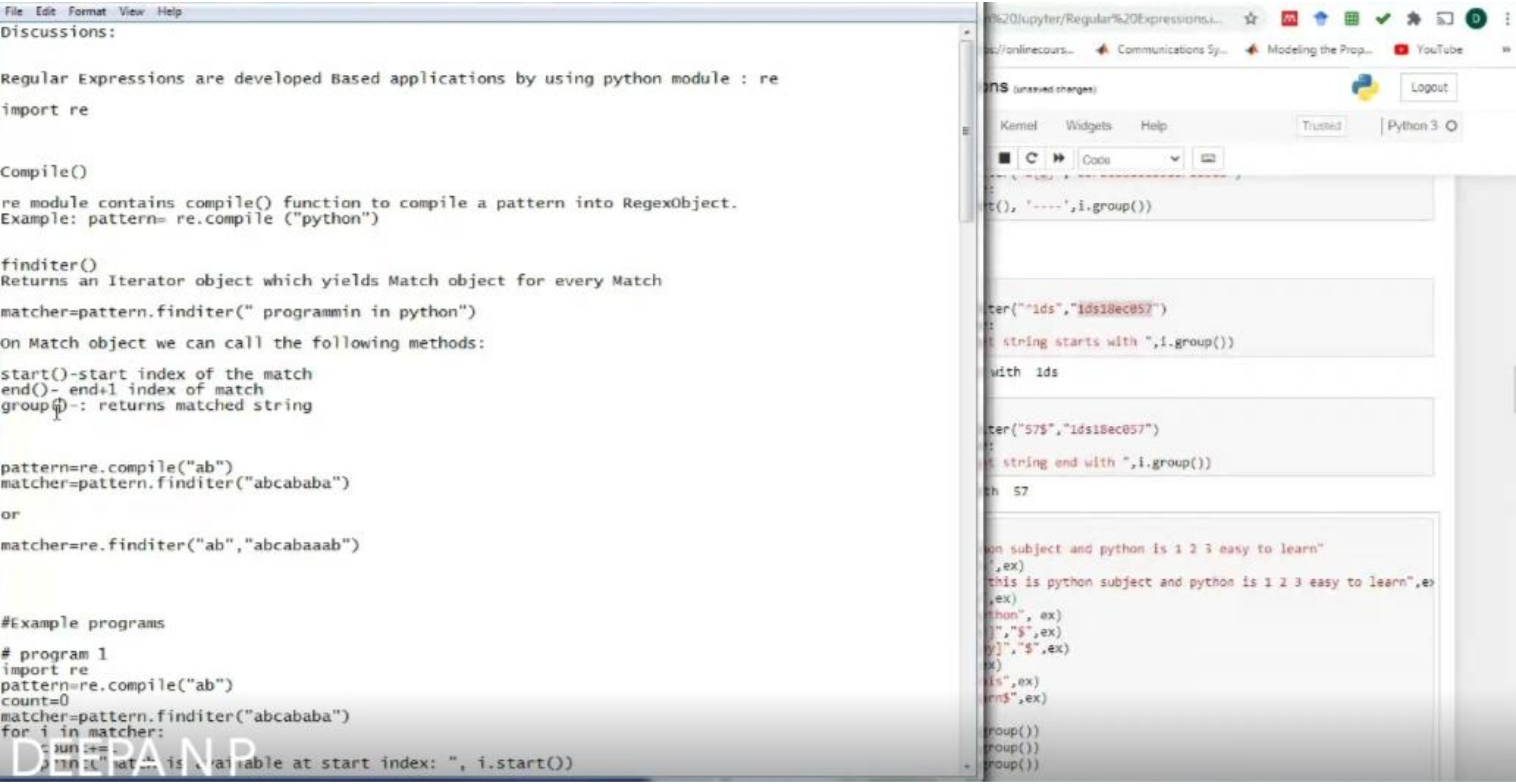
```
t(), '----',i.group())

ter("^1ds","1ds18ec057")
:
t string starts with ",i.group())
 with  1ds

ter("57$","1ds18ec057")
:
t string end with ",i.group())
h  57

on subject and python is 1 2 3 easy to learn"
',ex)
this is python subject and python is 1 2 3 easy to learn",e
,ex)
thon", ex)
]","$",ex)
y]","$",ex)
x)
is",ex)
rn$",ex)

roup())
roup())
roup())
```



```
matcher=re.finditer("[0-9]","a6B@ k9z")
for i in matcher:
    print(i.start(), '----',i.group())



Pre defined Character classes:
\s -- Space character
\S -- Any character except space character
\d -- Any digit from 0 to 9
\D -- Any character except digit
\w -- Any word character [a-zA-Z0-9]
\W -- Any character except word character (Special Characters)
.  -- Any character including special characters



import re
matcher=re.finditer("\d","a6B@ k9z")
for i in matcher:
    print(i.start(), '----',i.group())




Qunatifiers:
We can use quantifiers to specify the number of occurrences to match.
a       -- Exactly one 'a'
a+      -- Atleast one 'a'
a*      -- Any number of a's including zero number
a?      -- Atmost one 'a' ie either zero  or one number
a{m}    -- Exactly m number of a's
a{m,n}  -- Minimum m number of a's and Maximum n number of a's

^a      -- It will check whether target string starts with a or not
        We can use ^ symbol to check whether the given target string starts with our provided
pattern ornot.
a$      -- It will check whether target string ends with a or not
        We can use $ symbol to check whether the given target string ends with our provided
pattern or not


Important functions

finditer()
```

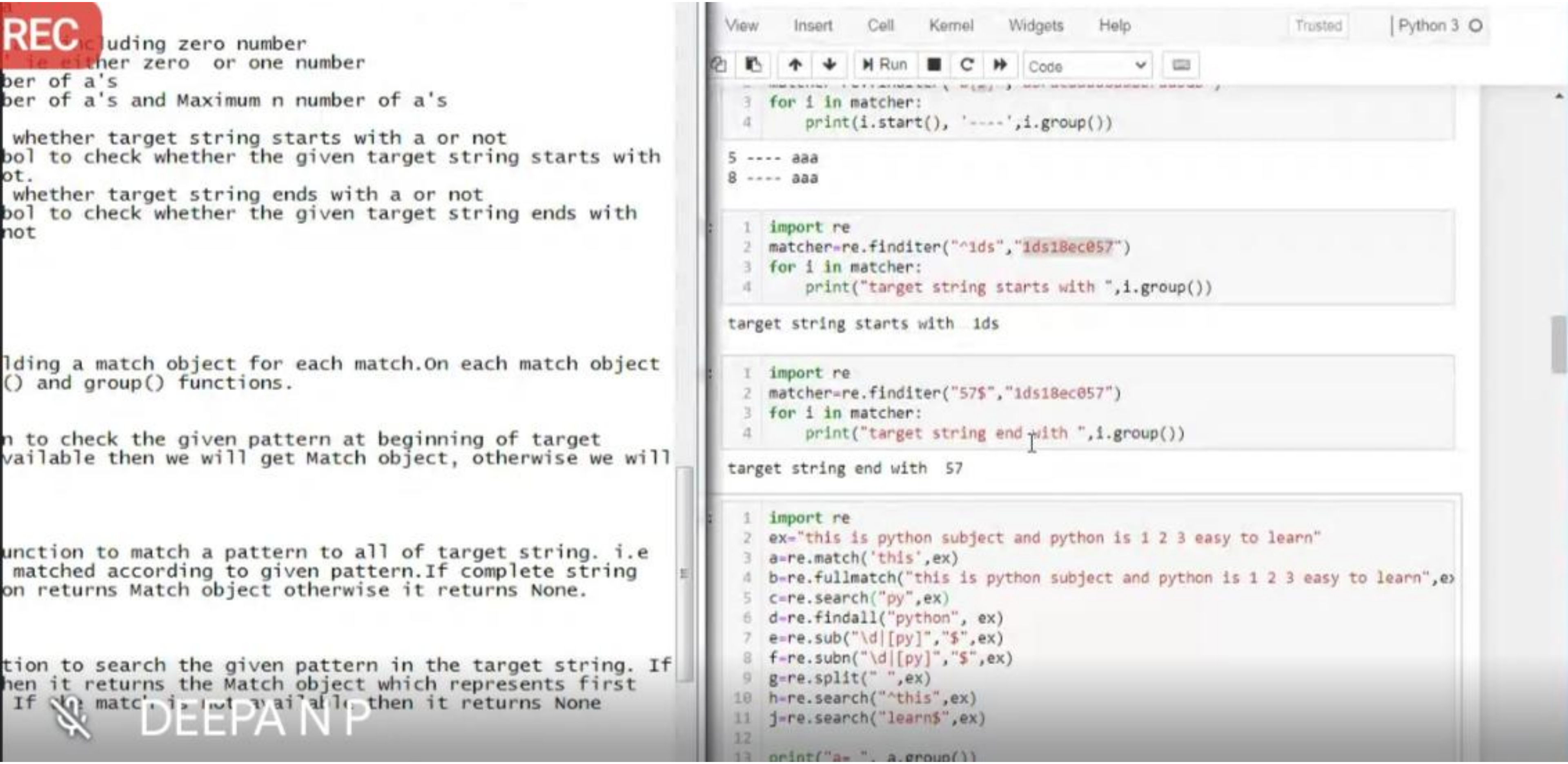
```
t(), '----',i.group())

ter("^1ds","1ds18ec057")
t string starts with ",i.group())
with  1ds

ter("57$","1ds18ec057")
t string end with ",i.group())
th  57

on subject and python is 1 2 3 easy to learn"
',ex)
this is python subject and python is 1 2 3 easy to learn",e
,ex)
thon", ex)
]","$",ex)
y]","$",ex)
x)
is",ex)
rn$",ex)

roup())
roup())
roup())
```

DEEPA N P



```
Discussions:

Regular Expressions are developed Based applications by using python module : re

import re


Compile()

re module contains compile() function to compile a pattern into RegexObject.
Example: pattern= re.compile ("python")


finditer()
Returns an Iterator object which yields Match object for every Match

matcher=pattern.finditer(" programmin in python")

On Match object we can call the following methods:

start()-start index of the match
end()- end+1 index of match
group()-: returns matched string


pattern=re.compile("ab")
matcher=pattern.finditer("abcababa")

or

matcher=re.finditer("ab","abcabaaab")



#Example programs

# program 1
import re
pattern=re.compile("ab")
count=0
matcher=pattern.finditer("abcababa")
for i in matcher:
    count+=1
    print("match is available at start index: ", i.start())
```

```
t(), '----',i.group())

ter("^1ds","1ds18ec057")
:
 string starts with ",i.group())
with  1ds

ter("57$","1ds18ec057")
:
 string end with ",i.group())
th  57

on subject and python is 1 2 3 easy to learn"
",ex)
this is python subject and python is 1 2 3 easy to learn",e
,ex)
thon", ex)
]","$",ex)
y]","$",ex)
x)
is",ex)
rn$",ex)

roup())
roup())
roup())
```



```
uding zero number
her zero  or one number
ber of a's
ber of a's and Maximum n number of a's

 whether target string starts with a or not
bol to check whether the given target string starts with
ot.
 whether target string ends with a or not
bol to check whether the given target string ends with
not




lding a match object for each match.On each match object
() and group() functions.

n to check the given pattern at beginning of target
vailable then we will get Match object, otherwise we will



unction to match a pattern to all of target string. i.e
 matched according to given pattern.If complete string
on returns Match object otherwise it returns None.



tion to search the given pattern in the target string. If
hen it returns the Match object which represents first
 If the match is not available then it returns None
```

```python
for i in matcher:
    print(i.start(), '----',i.group())
```

```
5 ---- aaa
8 ---- aaa
```

```python
import re
matcher=re.finditer("^1ds","1ds18ec057")
for i in matcher:
    print("target string starts with ",i.group())
```

```
target string starts with  1ds
```

```python
import re
matcher=re.finditer("57$","1ds18ec057")
for i in matcher:
    print("target string end with ",i.group())
```

```
target string end with  57
```

```python
import re
ex="this is python subject and python is 1 2 3 easy to learn"
a=re.match('this',ex)
b=re.fullmatch("this is python subject and python is 1 2 3 easy to learn",ex
c=re.search("py",ex)
d=re.findall("python", ex)
e=re.sub("\d|[py]","$",ex)
f=re.subn("\d|[py]","$",ex)
g=re.split(" ",ex)
h=re.search("^this",ex)
j=re.search("learn$",ex)

print("a= ", a.group())
```