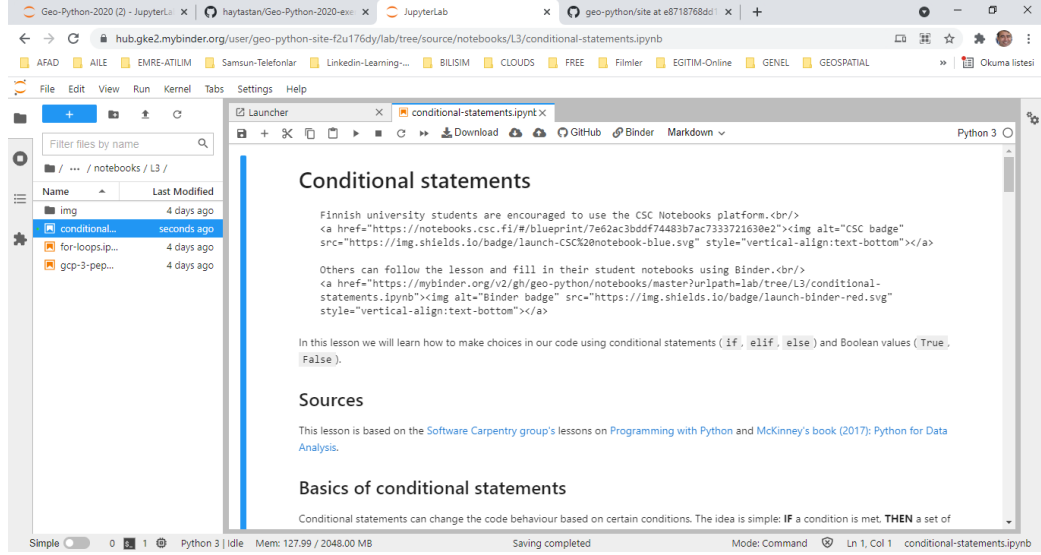


# NASIL YAPTIM

## Dr. Hayati TAŞTAN

### LAB 3 uygulamalarını on-line yapmak:

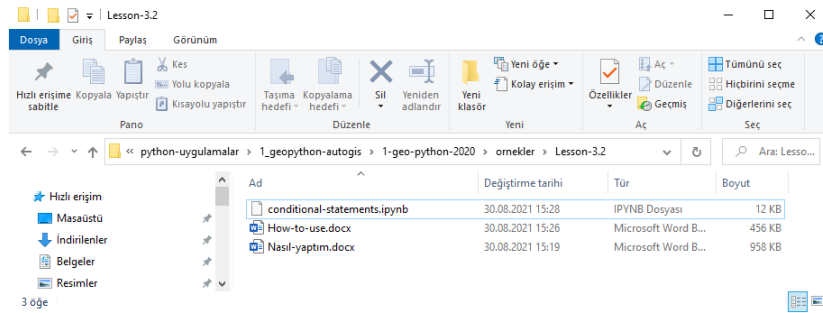
<https://hub.gke2.mybinder.org/user/geo-python-site-f2u176dy/lab/tree/source/notebooks/L3/conditional-statements.ipynb> adresine git. **conditional-statements.ipynb** dosyasına çift tıkla:



Sayfadaki kutular üzerine gelip SHIFT+ENTER basarak kutu içindeki python komutlarını çalıştır.

### LAB 3 uygulamalarını off-line yapmak:

Üstteki download ile **conditional-statements.ipynb** isimli jupyter projesini yerel diske indir ve Lesson 3.2 içine aktar:

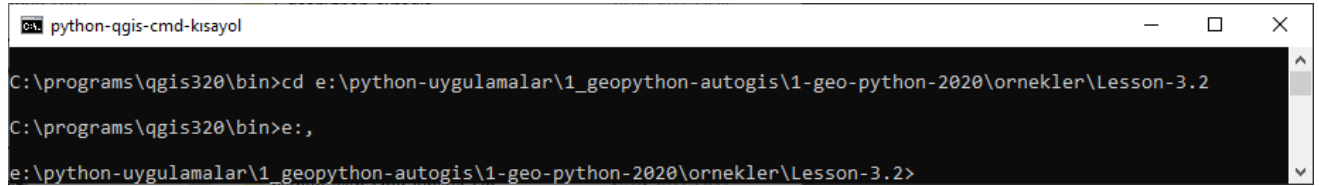


**conditional-statements.ipynb** isimli jupyter projesini, **jupyter lab** ortamında açalım:

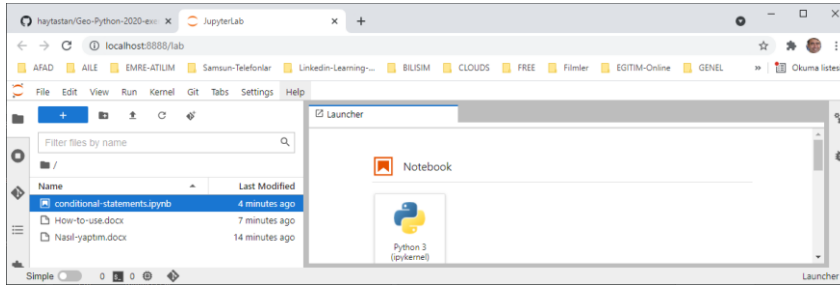
QGIS Shell

cd G:\python-uygulamalar\1\_geopython-autogis\1-geo-python-2020\ornekler\Lesson-3.2

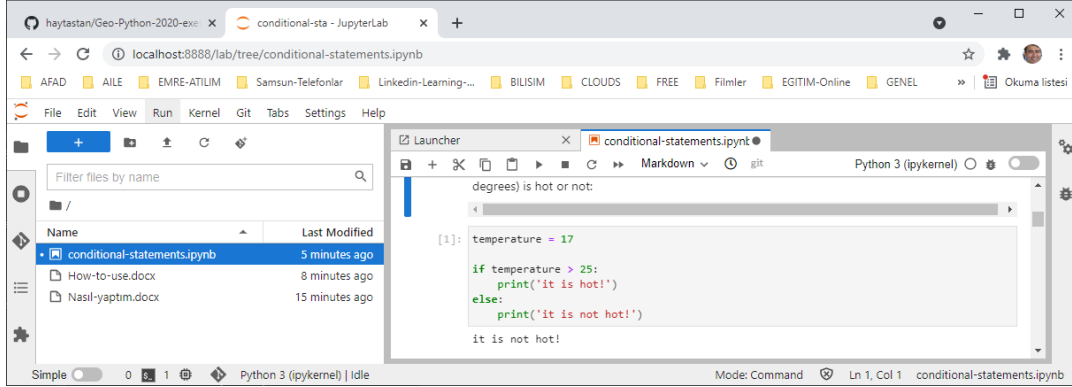
G:



Jupyter lab

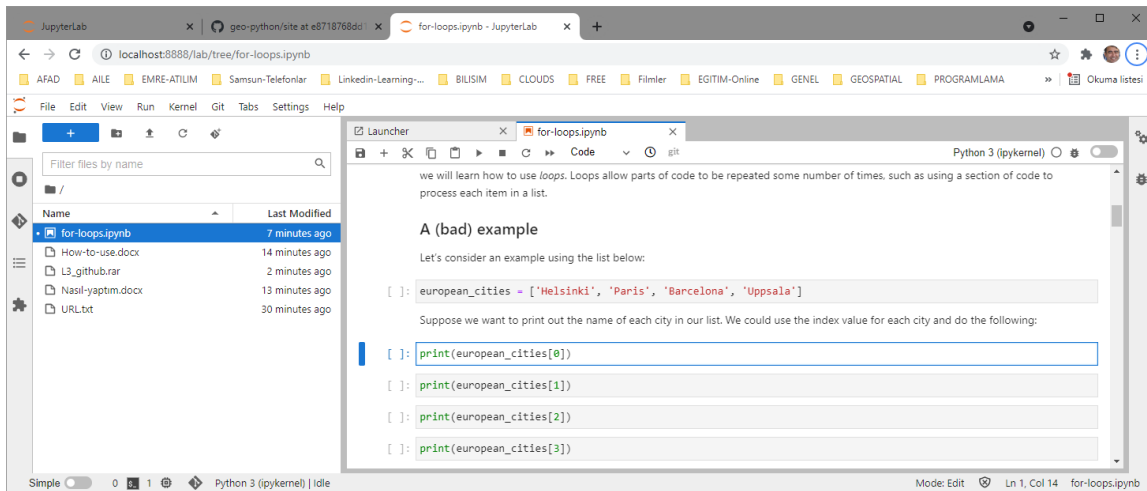


**conditional-statements.ipynb** isimli jupyter projesine çift tıklayalım:



Yukarıdaki sayfadaki kutular üzerine gelip, SHIFT+ENTER basarak kutu içindeki python komutlarını çalıştır (veya üstteki > düğmesi ile çalıştır):

**Geo-Python-2020-exercise-3- solutions** isiminde açılacak github reposuna konmak üzere **Örnek uygulama dosyaları :**



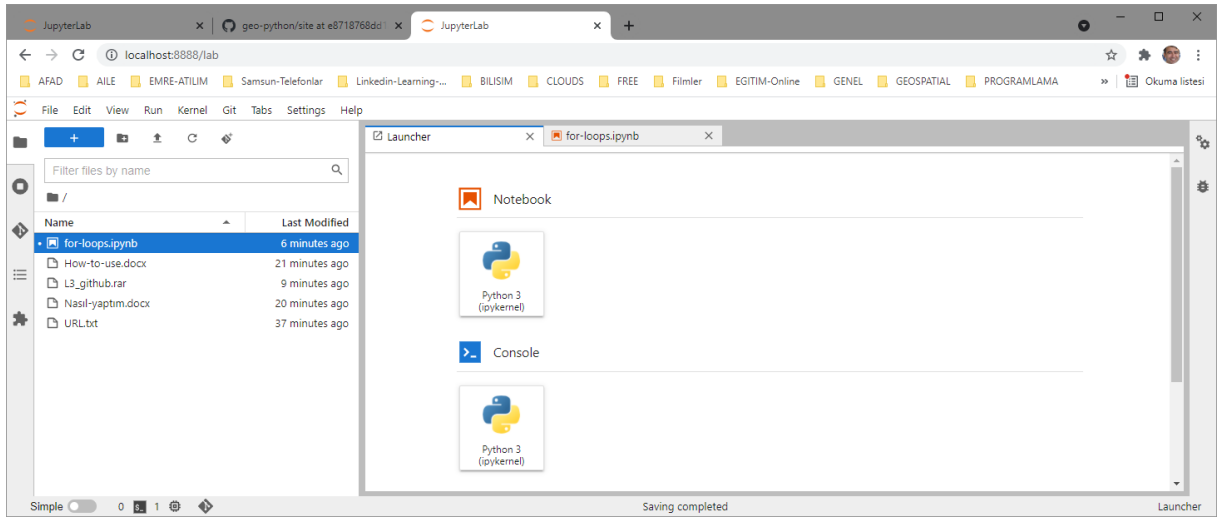
Yukarıdaki kodları **jupyter lab** ortamında çalıştıralım:

**QGIS Shell**

**cd G:\python-uygulamalar\1\_geopython-autogis\1-geo-python-2020\ornekler\Lesson-3.1**

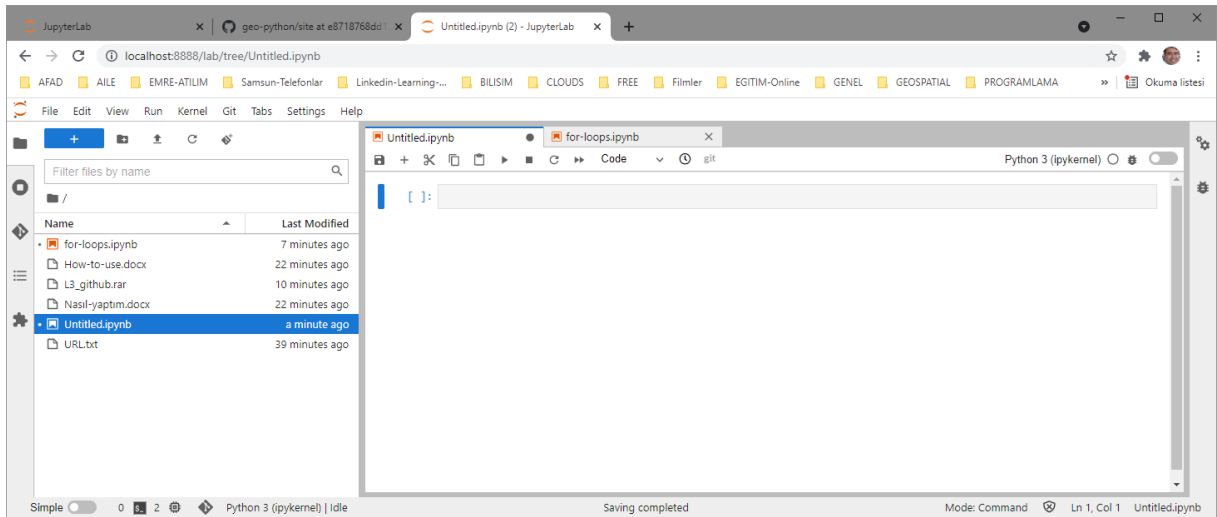
**G:**

**Jupyter lab**

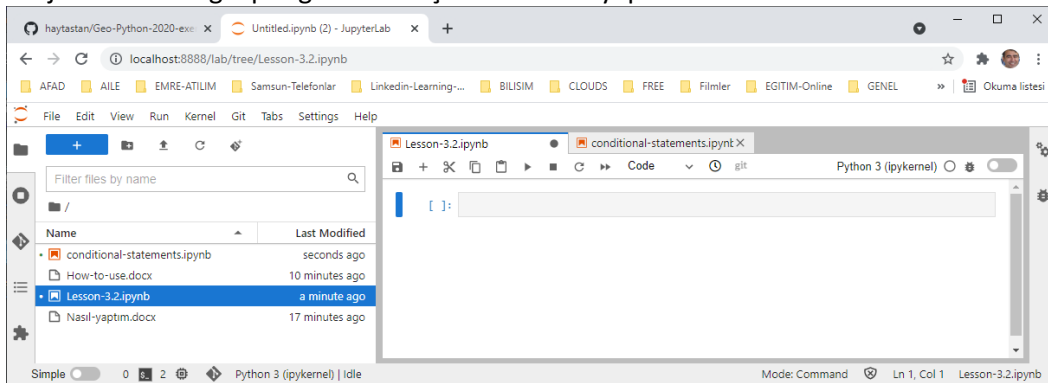


## Luncher / Python 3

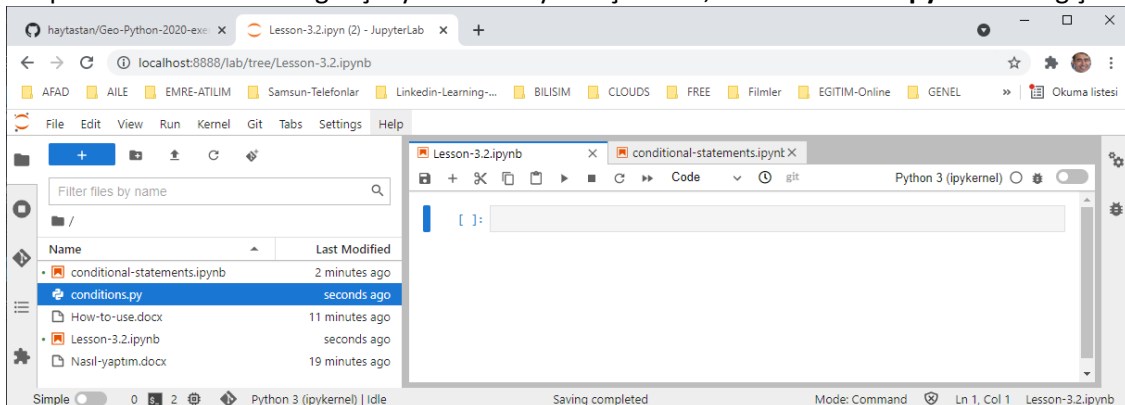
ile yeni bir jupyter projesi (**Untitled.ipynb**) oluşturulur:



Projenin üzerine gelip sağ mouse tuşu ile **rename** yapılır:



Sol panel üzerinde iken sağ tuşla yeni bir dosya oluşturulur, ismi **conditions.py** olarak değiştirilir:



**conditions.py** üzerine çift tıklayılıp aşağıdaki kod bu dosya içine yazılır ve kaydedilir (CTRL+S)  
(Aşağıdaki kodlar <http://localhost:8888/lab/tree/conditional-statements.ipynb> projesinden alınmıştır)

```
# if statement
temperature = 17
if temperature > 25:
    print('it is hot!')
else:
    print('it is not hot!')

weather = 'rain'
if weather == 'rain':
    print('Wear a raincoat!')
else:
    print('No raincoat needed.')

# if, elif and else
if temperature > 0:
    print(temperature, 'degrees celsius is above freezing')
elif temperature == 0:
    print(temperature, 'degrees celsius is at the freezing point')
else:
    print(temperature, 'degrees celsius is below freezing')
temperature = -3
if temperature > 0:
    print(temperature, 'degrees celsius is above freezing')
elif temperature == 0:
    print(temperature, 'degrees celsius is at the freezing point')
else:
    print(temperature, 'degrees celsius is below freezing')

yesterday = 14
today = 10
tomorrow = 13

if yesterday <= today:
    print('A')
elif today != tomorrow:
    print('B')
elif yesterday > tomorrow:
    print('C')
elif today == today:
    print('D')

#Combining conditions
if (1 > 0) and (-1 > 0):
    print('Both parts are true')
else:
    print('At least one part is not true')

if (1 < 0) or (-1 < 0):
    print('At least one test is true')

weather = 'rain'
wind_speed = 9

# If it is windy or raining, print "stay at home", else print "go out and enjoy the weather!"
if (weather == 'rain') or (wind_speed >= 8):
    print('Just stay at home')
else:
    print('Go out and enjoy the weather! :)')

#Combining for-loops and conditional statements
temperatures = [0, 12, 17, 28, 30]
# For each temperature, if the temperature is greater than 25, print "..is hot"
```

```

for temperature in temperatures:
    if temperature > 25:
        print(temperature, 'is hot')
    else:
        print(temperature, 'is not hot')

```

The screenshot shows the JupyterLab interface with a file browser on the left and a code editor on the right. The file browser shows a directory structure with files like 'conditional-statements.ipynb', 'conditions.py', 'How-to-use.docx', 'Lesson-3.2.ipynb', and 'Nasil-yapim.docx'. The code editor displays the following Python code:

```

1 # if statement
2 temperature = 17
3 if temperature > 25:
4     print('it is hot!')
5 else:
6     print('it is not hot!')
7
8 weather = 'rain'
9 if weather == 'rain':
10    print('wear a raincoat!')
11 else:
12    print('No raincoat needed.')
13
14 # if, elif and else
15 if temperature > 0:
16    print(temperature, 'degrees celsius is above freezing')
17 elif temperature == 0:
18    print(temperature, 'degrees celsius is at the freezing point')
19 else:
20    print(temperature, 'degrees celsius is below freezing')
21 temperature = -3
22 if temperature > 0:
23    print(temperature, 'degrees celsius is above freezing')
24 elif temperature == 0:
25    print(temperature, 'degrees celsius is at the freezing point')
26 else:
27    print(temperature, 'degrees celsius is below freezing')
28
29 yesterday = 14
30 today = 10
31

```

The screenshot shows the JupyterLab interface with a file browser on the left and a code editor on the right. The file browser shows a directory structure with files like 'conditional-statements.ipynb', 'conditions.py', 'How-to-use.docx', 'Lesson-3.2.ipynb', and 'Nasil-yapim.docx'. The code editor displays the following Python code:

```

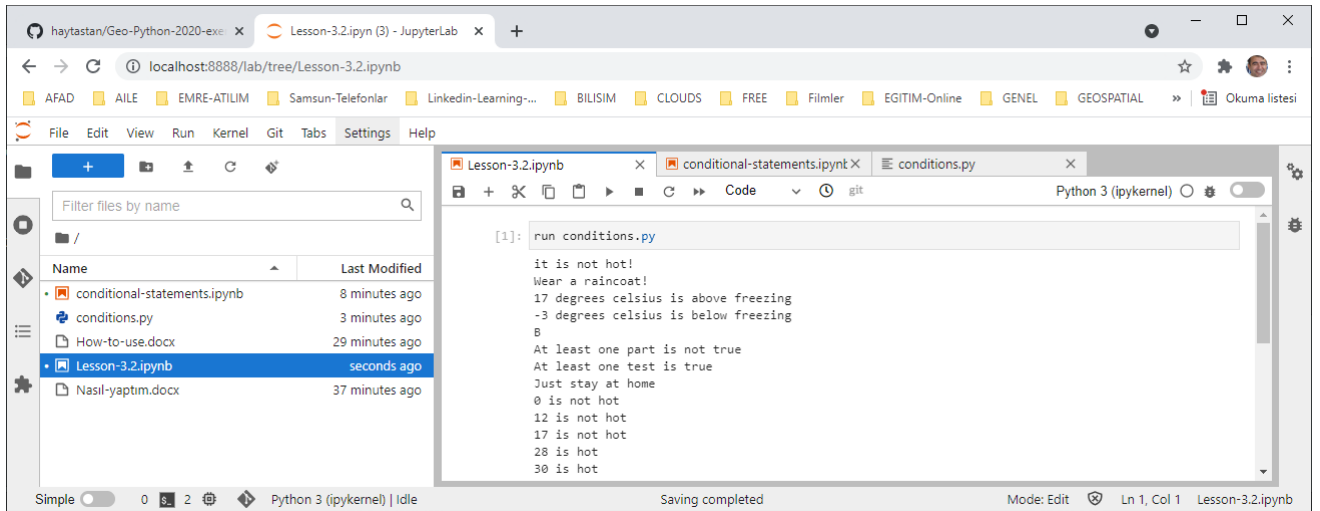
42 #Combining conditions
43 if (1 > 0) and (-1 > 0):
44     print('Both parts are true')
45 else:
46     print('At least one part is not true')
47
48 if (1 < 0) or (-1 < 0):
49     print('At least one test is true')
50
51 weather = 'rain'
52 wind_speed = 9
53
54 # If it is windy or raining, print "stay at home", else print "go out and enjoy the weather!"
55 if (weather == 'rain') or (wind_speed >= 0):
56     print('Just stay at home')
57 else:
58     print('Go out and enjoy the weather! :)')
59
60 #Combining for-Loops and conditional statements
61 temperatures = [0, 12, 17, 28, 30]
62 # For each temperature, if the temperature is greater than 25, print "..is hot"
63 for temperature in temperatures:
64     if temperature > 25:
65         print(temperature, 'is hot')
66     else:
67         print(temperature, 'is not hot')
68

```

**Lesson-3.2.ipynb** proje dosyasına çift tıklatılır ve açılan kutu içine aşağıdaki komut yazılır:

**run loop.py**

ve imleç kutu içinde iken **SHIFT+ENTER** tuşlarına basılara komut çalıştırılır:



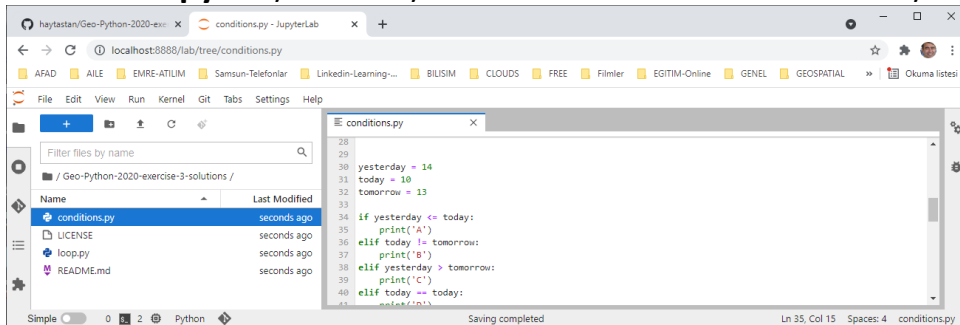
**conditions.py** dosyasını github'a aktarma:

Github ortamından **Geo-Python-2020-exercise-3-solutions** isimli repo clone edilir:  
(user: mygmail, pass: Çocuklar+1962)

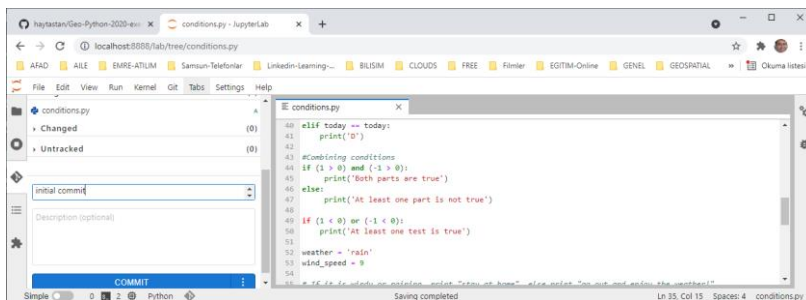
Repo clone edilir (Git Colne a repository)

Adres olarak: <https://github.com/haytastan/Geo-Python-2020-exercise-3-solutions>

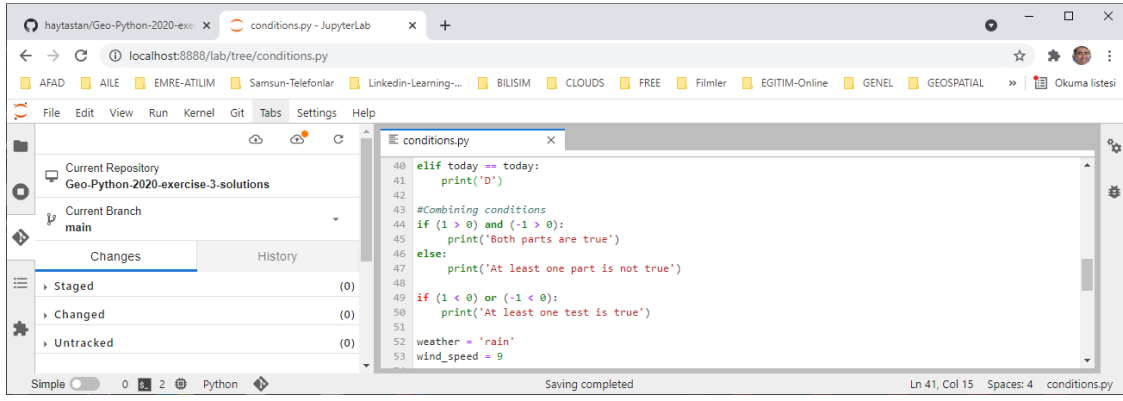
**conditions.py** dosyası Geo-Python-2020-exercise-3-solutions isimli yerel repo içine kopyalanır.



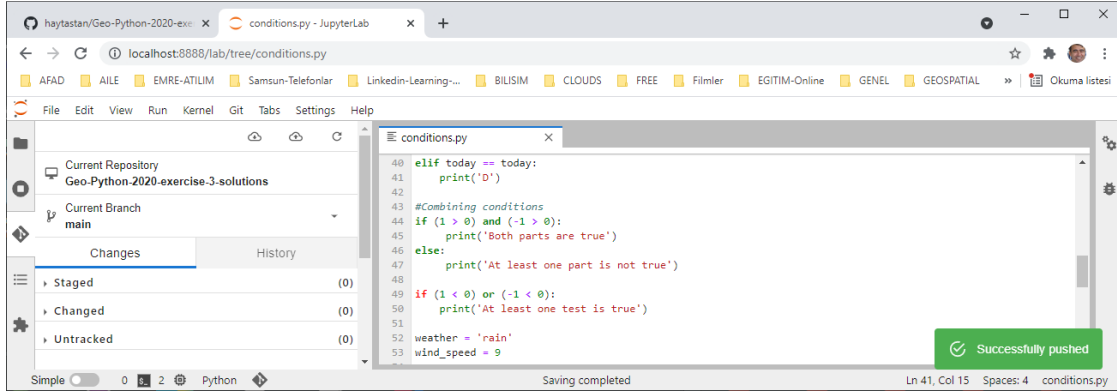
Sol panodaki **Git** ikonuna tıklanıp, **conditions.py** seçilip, dosyanın sağındaki **+** işareti (**Track this file**) tıklanır, dosya olduğu doğrudan "staged" area içine aktarılır.



Aşağıdaki **summary** kutusu içine initial commit yazılır. Ve en alttaki **COMMIT** düğmesine tıklanarak değişiklikler, dosya push için hazır hale getirilir:



Üstteki **Git/Push to remote** menüsü ile **stage area**'daki değişiklikler uzaktaki **github** reposuna gönderilir:



Github reposunda, **conditions.py** görülür:

