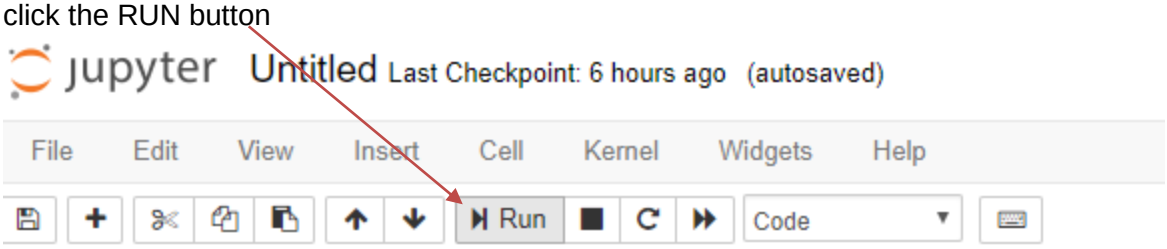
To run the python command in Jupyter:

- SHIFT + ENTER after entering each line to run the command. ENTER will add a new line below to continue entering

OR

- click the RUN button



TEST WITH SIMPLE HTML TAGS USING BEAUTIFULSOUP

1. Download simple.txt from i-learn
2. Copy the text and past it using notepad++, save it as html file on your desktop.
3. Open Anaconda Navigator
4. Launch Jupyter Notebook
5. New → python 3
6. Enter the following codes:
    a. from bs4 import BeautifulSoup as bs SHIFT + ENTER
    b. test_url = "C:\\Users\\User\\Desktop\\simple.html" SHIFT + ENTER
    c. soup = bs(open(test_url), 'html.parser') SHIFT + ENTER
    d. print (soup) SHIFT + ENTER

```
<html>
<head>
<title>
     Simple Web Page
         </title>
</head>
<body>
<p align="center" id="First content">
      This is the first content
           <b> Hello students </b>
</p>
<p align="center" id="Second content">
     This is the second content
          <b> This is the basic HTML tags <b>
</b></b></p>
</body>
</html>
```

    e. print (soup.prettify()) SHIFT + ENTER

```
<html>
 <head>
  <title>
   Simple Web Page
  </title>
 </head>
 <body>
  <p align="center" id="First content">
   This is the first content
   <b>
    Hello students
   </b>
  </p>
  <p align="center" id="Second content">
   This is the second content
   <b>
    This is the basic HTML tags
    <b>
     </b>
    </b>
   </p>
 </body>
</html>
```

f. soup.title SHIFT + ENTER

```
<title>
     Simple Web Page
         </title>
```

g. soup.body SHIFT + ENTER

```
<body>
<p align="center" id="First content">
      This is the first content
          <b> Hello students </b>
</p>
<p align="center" id="Second content">
     This is the second content
         <b> This is the basic HTML tags <b>
</b></b></p>
</body>
```

h. soup.body.contents[1] SHIFT + ENTER

```
<p align="center" id="First content">
      This is the first content
          <b> Hello students </b>
</p>
```

i. soup.get_text()

```
'\n\n\n    Simple Web Page\n\t \n\n\n\n      This is the first content\n\t   Hello students \n\n\n     This is the second cont
ent\n\t  This is the basic HTML tags \n\n\n'
```

j. print (soup.get_text())

```
Simple Web Page


 This is the first content
      Hello students


This is the second content
     This is the basic HTML tags
```

k. print (soup.get_text(strip=True))

```
Simple Web PageThis is the first contentHello studentsThis is the second contentThis is the basic HTML tags
```

l. print (soup.get_text(‘ ’, strip=True))

```
Simple Web Page This is the first content Hello students This is the second content This is the basic HTML tags
```

m. soup.findAll(‘p’)

```
[<p align="center" id="First content">
       This is the first content
          <b> Hello students </b>
 </p>, <p align="center" id="Second content">
      This is the second content
         <b> This is the basic HTML tags <b>
 </b></b></p>]
```

n. soup.findAll('p',{'id':'First content'})

```
[<p align="center" id="First content">
       This is the first content
          <b> Hello students </b>
 </p>, <p align="center" id="Second content">
      This is the second content
         <b> This is the basic HTML tags <b>
 </b></b></p>]
```

How to read and write data from/to a file using python

Create file for writing

The code below creates a new file (or overwrites) with the data.

```
#!/usr/bin/env python

# Filename to write
filename = "newfile.txt"

# Open the file with writing permission
myfile = open(filename, 'w')

# Write a line to the file
myfile.write('Written with Python\n')

# Close the file
myfile.close()
```

The 'w' flag makes Python truncate the file if it already exists. That is to say, if the file contents exists it will be replaced.

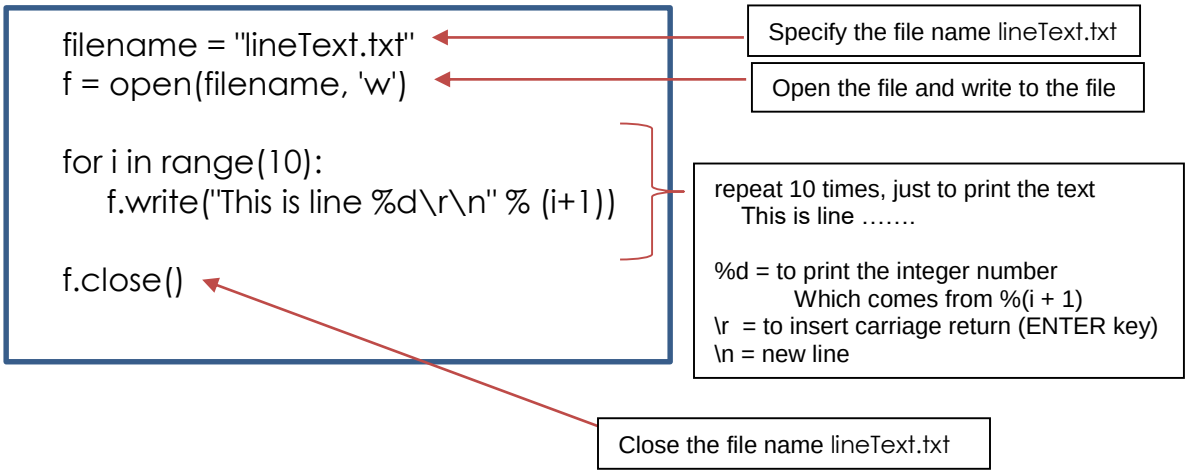
open (filename, file mode)

File Modes in Python

Mode	Description
'r'	This is the default mode. It Opens file for reading.
'w'	This Mode Opens file for writing. If file does not exist, it creates a new file. If file exists it truncates the file.
'x'	Creates a new file. If file already exists, the operation fails.
'a'	Open file in append mode. If file does not exist, it creates a new file.
't'	This is the default mode. It opens in text mode.
'b'	This opens in binary mode.
'+'	This will open a file for reading and writing (updating)

1. How to write data into a file. (if the file exists, then the content will be overwritten)

Example : writing a text into file named 'lineText.txt'



2. How to append to the existing file. (if the file exists, the new content will be appended, the existing content still intact)

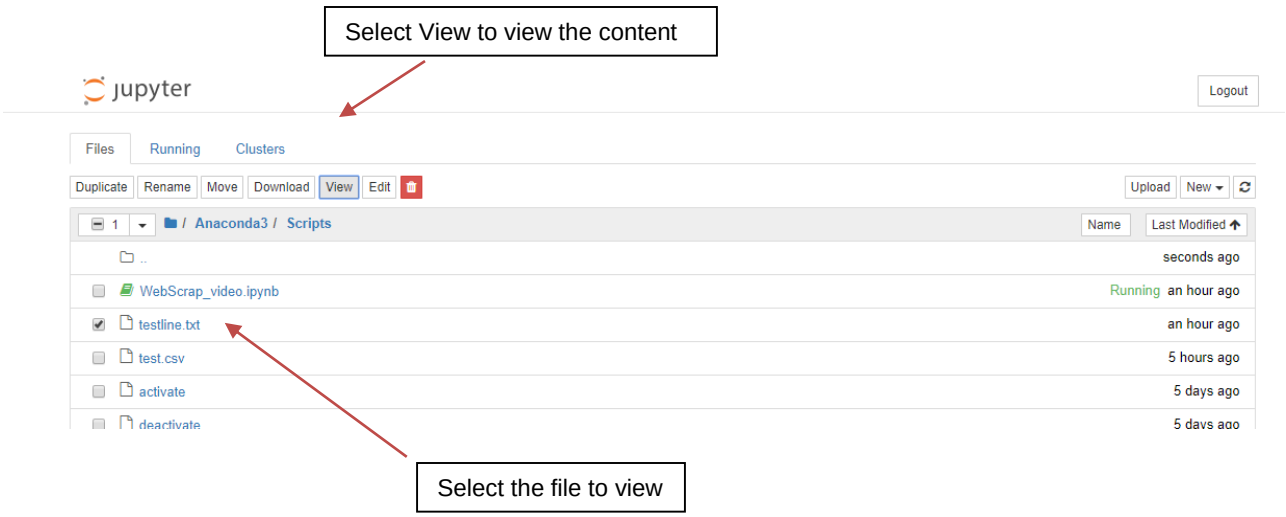
Example : append the text to file named 'lineText.txt'

```
filename = "lineText.txt"
f = open(filename, 'a+')

for i in range(5):
    f.write("Appended line %d\r\n" % (i+1))

f.close()
```

*******Note : you can search the file in folder anaconda3/script, since it is a text file, you can view it using notepad. You can view the file in Jupyter Notebook explorer***



3. How to read all contents in the file.

Example : read all the contents in a file named 'lineText.txt'

```
filename = "lineText.txt"
f = open (filename, 'r')

f1 = f.read()
print (f1)

f.close()
```

4. How to read content in a file line by line.

Example : read the content in a file named 'lineText.txt' line by line

```
filename = "lineText.txt"
f = open(filename, 'r')

f1 = f.readlines()

for x in f1:
  print (x)

f.close()
```

How to start web scraping in Jupyter Notebook

1. Import BeautifulSoup from package bs4

```
In [37]: from bs4 import BeautifulSoup as bs
```

2. Import the URL package, to read the URL address in the website

```
In [39]: from urllib.request import urlopen as ureq
```

3. Copy the URL address selected from the website

```
In [40]: my_url = 'https://www.history.com/this-day-in-history'
```

4. Request to open the connection, read the webpage and download to our machine

```
In [41]: uclient = ureq(my_url)
```

5. Read the HTML tags from the webpage (scraped contents)

```
In [42]: page_html = uclient.read()
```

6. Close the connection to the webpage

```
In [43]: uclient.close()
```

7. To parse the contents (synthesize the webpage contents)

```
In [45]: soup = bs(page_html,'html.parser')
```

How to write the scraping data into a file (csv file – excel format)

Example :

Scrap data from webpage –
https://www.newegg.com/Video-Cards-Video-Devices/Category/ID-38?Tpk=graphics%20CARD

save the data into a file named test.csv (excel format delimited), it will be saved in the anaconda3/script folder

```
from bs4 import BeautifulSoup as soup
from urllib.request import urlopen as uReq

my_url = 'https://www.newegg.com/Video-Cards-Video-Devices/Category/ID-38?Tpk=graphics%20CARD'
uClient = uReq(my_url)
page_html = uClient.read()
uClient.close()

page_soup = soup(page_html, 'html.parser')
containers = page_soup.findAll('div', {'class':'item-container'})

filename = 'test.csv'
f = open(filename, 'w')

for container in containers:
    brand = container.div.div.a.img['title']
    title_container = container.findAll('a', {'class':'item-title'})
    product_name = title_container[0].text
    shipping_container = container.findAll('li',{'class':'price-ship'})
    shipping = shipping_container[0].text.strip()
    print('brand : ' + brand)
    print('productname : ' + product_name)
    print('shippping : ' + shipping)
    f.write(brand + ', ' + product_name.replace(',','|') + ', ' + shipping + '\n')

f.close()
```

Example :
To scrap
https://www.newegg.com/Laptops-Notebooks/Category/ID-223?Tid=17489

```
from bs4 import BeautifulSoup as soup
from urllib.request import urlopen as uReq

my_url = "https://www.newegg.com/Laptops-Notebooks/Category/ID-223?Tid=17489"

uClient = uReq(my_url)            #to request the connection to URL specified

my_page = uClient.read()          #read the webpage connected

page_soup = soup(my_page, "html.parser")     #to parse the webpage content


#to select all tags <div class = item-container>
my_content = page_soup.findAll("div", {"class":"item-container"})


print (my_content)            #to display what is in my_content


for x in my_content:                    #looping through all the contents in the item-container
    model = x.div.div.a.img['title']     #scrap the title and pun in array – tree navigation
    print (model)


for x in my_content:
    model = x.div.div.a.img['title']                    #different title name for each image
    item_desc = x.findAll('a',{'class':'item-title'})   #find all the <a href .....class= item-title
    print(len(item_desc))                               #how many contents are there?
    print(item_desc[0])                                 # Array index always starts with 0


for x in my_content:
    model = x.div.div.a.img['title']
    item_desc = x.findAll('a',{'class':'item-title'})
    print(len(item_desc))
    print(item_desc[0].text)                            # display only text
```

```
for x in my_content:
    model = x.div.div.a.img['title']
    item_desc = x.findAll('a',{'class':'item-title'})
    print ('Model : ' + brand)
    print ('Product Name : ' + item_desc[0].text + '\n')
```

```
for x in my_content:
    model = x.div.div.a.img['title']
    item_desc = x.findAll('a',{'class':'item-title'})
    shipping = x.findAll('li',{'class':'price-ship'})          #shipping information
    print(shipping[0].text.strip())
```

```
for x in my_content:
    model = x.div.div.a.img['title']
    item_desc = x.findAll('a',{'class':'item-title'})
    shipping = x.findAll('li',{'class':'price-ship'})
    print('Model : ' + model)
    print('Product Description : ' + item_desc[0].text)
    print('Shipping : ' + shipping[0].text.strip() + '\n')
```

To print those data into file csv (excel format delimited)