0:13
Hello and welcome to this short tutorial on web scraping. Where we are going to literally scrape
0:21
some information off the internet specifically from the BBC's weather website and then put it
0:29
into a nice panda’s data frame and save it into a CSV and Excel file for processing later. So,
0:38
let us dive into today's exercise. So, this is the BBC's weather website and the information that you
0:45
see here for the city of Mumbai and the data that we are interested in this far for this particular
0:53
tutorial is the daily high-temperature values, the daily low-temperature values, and the daily
1:01
weather summary for the next14 days for the city of Mumbai.
1:10
So, let us look into the piece of code or the notebook that does this for us. So,

1:16
before I begin let me put this caveat here that web scraping might not be always legal.
1:23
So, it is a good idea to go through the terms and conditions of the website they are planning
1:29
to scrape before you go and scrape to see if it is legal. So, let us look at the libraries that
1:38
we are importing for this exercise two libraries are very important, one is the request library and
1:45
the other is the Beautiful Soup. So, what does the request do?
1:51
It has functionalities that help you go fetch the HTML code from
1:59
any website that you are interested in scraping data. So,
2:07
HTML code is what your browser renders into this beautiful web page but then what you want
2:13
for web scraping is the HTML code ss from the web server. So, the request library has functions to
2:22
do that for you. If you pass the URL of the website, it will fetch your HTML code.

2:28
And then Beautiful Soup helps you go through these HTML codes line by line
2:34
and then you can fetch whatever information you want and then put it into a different
2:38
place and then process it right. So, these are two important libraries JSON and URL and code I mean
2:46
I have just included that for you know for instance if you see this URL here this last
2:55
piece of information. So, that is what tells the web server that I am looking for Mumbai city.
3:01
Now the simplest thing to do is just copy-paste this URL and pass it onto
3:13
the get function of the request library but then the cooler way to do that is to use
3:23
BBC's location service ape. So, just pass the name Mumbai to it, and then it will give you back
3:33
this stuff here. Did the last code for the city of Mumbai and then you can just append it to this
3:39
BBC.com weather slash that location id but that is not important.
3:48
So, let us get started. So, this piece of code is what does this cool stuff for me that is
3:55
just past the city of the name of whatever city you are interested in and it gets you back the
4:03
URL for that city. So, let us get started by importing the libraries um why is it taking time.
4:27
Now to get the URL, I have the URL here and then I use the get functionality from the link request
4:40
library to go fetch the HTML contents of this page and then store it in the response object.
4:49
After that, we are going to initiate an instance of Beautiful Soup
4:53
and we are going to tell Beautiful Soup that hey look I need an HTML
4:57
parser to pass through the contents of this particular web page that you have stored here,
5:03

you do that and we have done. We have initiated, we have got the HTML code and
5:09
all that we have to do is go through this HTML code and figure out where our information is
5:14
of interest resides and then extract that and put it into whatever format you want.
5:19
So, for that, we need to know or we need to tell beautiful soup where all this information resides.
5:27
So, I am on a chrome browser. So, as I said, if you are interested in the daily high values,
5:32
really low values in the summary. So, to know where in the HTML code this information resides
5:37
all I have to do is right-click on these 32 guys here which is 30-degree Celsius
5:42
high temperature for today. And click on inspect
5:45
which will show you the HTML code here and we can figure out that this information is stored in
5:58
the block with the name hyphen day hyphen temperature underscore high
6:06
right. So, similarly, if we go to the low temperature, it would be underscored by hyphen
6:14
day hyphen temperature and it is going to be low, that is where your information resides which is
6:22
like 25 degrees Celsius. So, what we have to do is
6:28
there is the find all functionality in Beautiful Soup which does you know it
6:38
you can pass this name here and what it does is go and it will go and figure out all the places
6:48
with a span block um and where this one and the span block has a name this temperature high value
7:01
and then it will return all the information for you let us look at what that looks like.
7:06
So, it returns a lot of junk along with some information that we are interested in
7:12
for the next 14 days. Like a list of 14 elements, this is a list of 14 elements. Each has the

7:21
daily high-temperature values in degrees Celsius and the daily look high-temperature
7:25
values in degrees Fahrenheit along with a lot of chunks. Similarly, we do that for the
7:32
low temperature and we get back a list of 14 elements along with a lot of junk
7:37
and the information that we are interested in. And the daily summary
7:46
if we look at how it is returned unfortunately it is not a list of 14 elements it is one big
7:54
string like just one big string. So, we will have to do some post-processing here

8:01
before putting it into our pre-processing. Before putting it into a CSC format
8:06
we will come to that in a while. So, as I said in both these high values and low values it returned
8:15
along with the information a lot of junk. So, how do we extract the information alone? So,
8:20
if you remember it was a list of elements. So, if I have to access the first element of the list, I
8:26
access I mean and I am only interested in the next part of it I strip off all the other information
8:34
I get this similarly for the sixth day I get this okay I am not interested in the degree Fahrenheit
8:42
I am only interested in degree Celsius. So, I split it and then take the
8:47
first element putting this all together in a list iterator.
8:53
I get again a list of 14 elements but now with no junk no degree Fahrenheit just degree Celsius
9:00
right. Similarly, I do this for the daily low values and I am here
9:07
now if you remember daily the daily summary was not a list of 14 elements it was just a big
9:15
string. But luckily for us, the summary for each day began with an uppercase letter.
9:23
So, sunny and doubles and the general breeze sunny and gentle under general breeze

oh it is also ah
9:30
yeah light cloud and the general breeze. So, the first letter of the first word in every summary
9:38
started with the uppercase letter. So, with that knowledge, we are going to exploit that
9:43
to use a regular expression to split this large string or this long string into 14 different
9:53
strings and put them into a list And the splitting is going to be done
9:58
on the uppercase letter and that is what this regular expression is for us. So, I do that and
10:08
I get and I get a list of 14 strings. Now all I have to do is put this summary of high temperature
10:15
and blue temperature into it when I expand pandas' data frame but as I said we have got this
10:20
information for the next 14 days. So, if somebody has to use this later, they should probably know
10:25
which date each row belongs to. So, I am just creating a
10:30
date column using the panda's date range function and stripping off all the junk here, and using
10:37
only the date. Zipping everything together, the date lists high-value low values in the summary
10:45
putting it into a nice data frame with the column names and this is what I get. So, if you notice
10:55
that there is this there is a degree symbol here which might be a hindrance
10:59
for you know math operations later on. So, I am just replacing that with nothing and
11:03
then converting it into a float column now it is clean and again I am going to save it into a CSV
11:15
or an xls. I will do both here and it is a good idea to save the file with the
11:22
location name which is Mumbai here. Again, um you get the location id from this Mumbai guy here. So,
11:34
it is stored here but it is kind of a dump here because at the beginning we passed

Mumbai.
11:43
So, I can just use the required city as well anyway. So, we do that
11:52
and it saves a CSV and an xls file in my system. So, that is the end of this tutorial.
12:04
Two key takeaways one is for scraping you need the request library which fetches the
12:13
information the HTML code from any web server and 2 the Beautiful Soup library which actually
12:22
helps you pass through this HTML code and then gather all the information for you.

12:29
Until we see it again with another tutorial take care, bye.