

## DAILY ASSESSMENT FORMAT

Date:	03/06/2020	Name:	K B KUSHI
Course:	Python	USN:	4AL17EC107
Topic:	<ul style="list-style-type: none"><li>Application 7: Scrape Real Estate Property Data from the Web</li></ul>	Semester & Section:	6 B
Github Repository:	<a href="https://github.com/alvas-education-foundation/KUSHI-COURSES.git">https://github.com/alvas-education-foundation/KUSHI-COURSES.git</a>		

## FORENOON SESSION DETAILS

### Image of session

The screenshot displays a Jupyter Notebook interface with the following content:

```
In [55]: import requests
from bs4 import BeautifulSoup
import pandas

In [21]: r = requests.get("http://www.pyclass.com/real-estate/rock-springs-wy/LCWYROCKSPRINGS/", headers={'User-agent': 'Mozilla
c = r.content

In [22]: soup = BeautifulSoup(c,"html.parser")
print(soup.prettify())

<!DOCTYPE html>
<!-- saved from url=(0110)http://web.archive.org/web/20160127020422/http://www.century21.com/real-estate/rock-springs
-wy/LCWYROCKSPRINGS -->
<html lang="en" style="margin: 0px;overflow:hidden">
<script async="" src="/LCWYROCKSPRINGS1_files/beacon.js">
</script>
<script src="chrome-extension://pkljnnogdmlajgaoodihioopfkdpgjgg/Kernel.js?0.3685073930846756">
</script>
<head>
<meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
<script src="/LCWYROCKSPRINGS1_files/analytics.js" type="text/javascript">
</script>
<script type="text/javascript">
archive_analytics.values.server_name="wwwb-appl7.us.archive.org";archive_analytics.values.server_ms=227;
</script>
<link href="/LCWYROCKSPRINGS1_files/banner-styles.css" rel="stylesheet" type="text/css"/>
<title>
Rock Springs Real Estate | Find Houses &amp; Homes for Sale in Rock Springs, WY
</title>

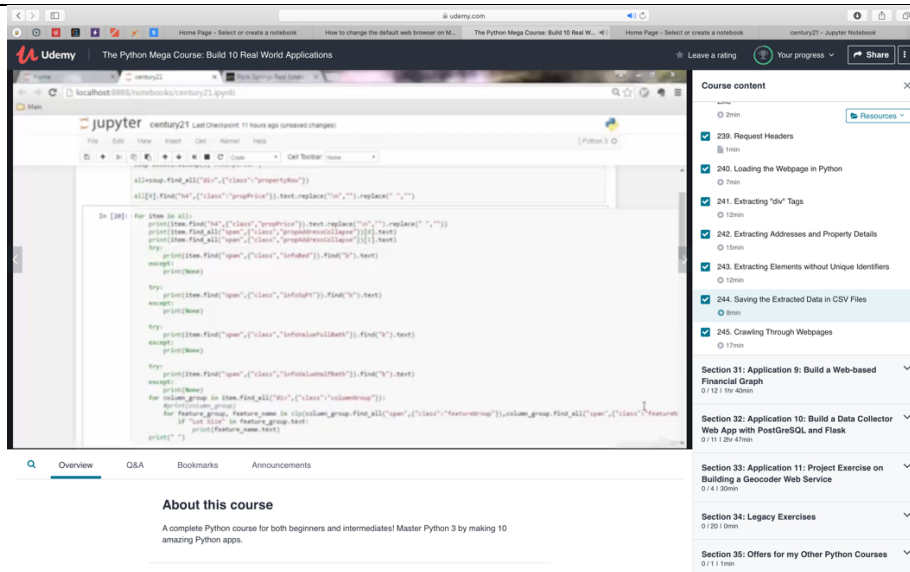
In [24]: all = soup.find_all("div",{"class":"propertyRow"})
all[0].find_all("h4",{"class":"propPrice"})

Out[24]: [<h4 class="propPrice">

$725,000

<span class="IconPropertyFavorite16"></span>
</h4>]

In [53]: l=[]
for item in all:
```



## Report:

In this section we were taught how to scrap data from a real estate website.

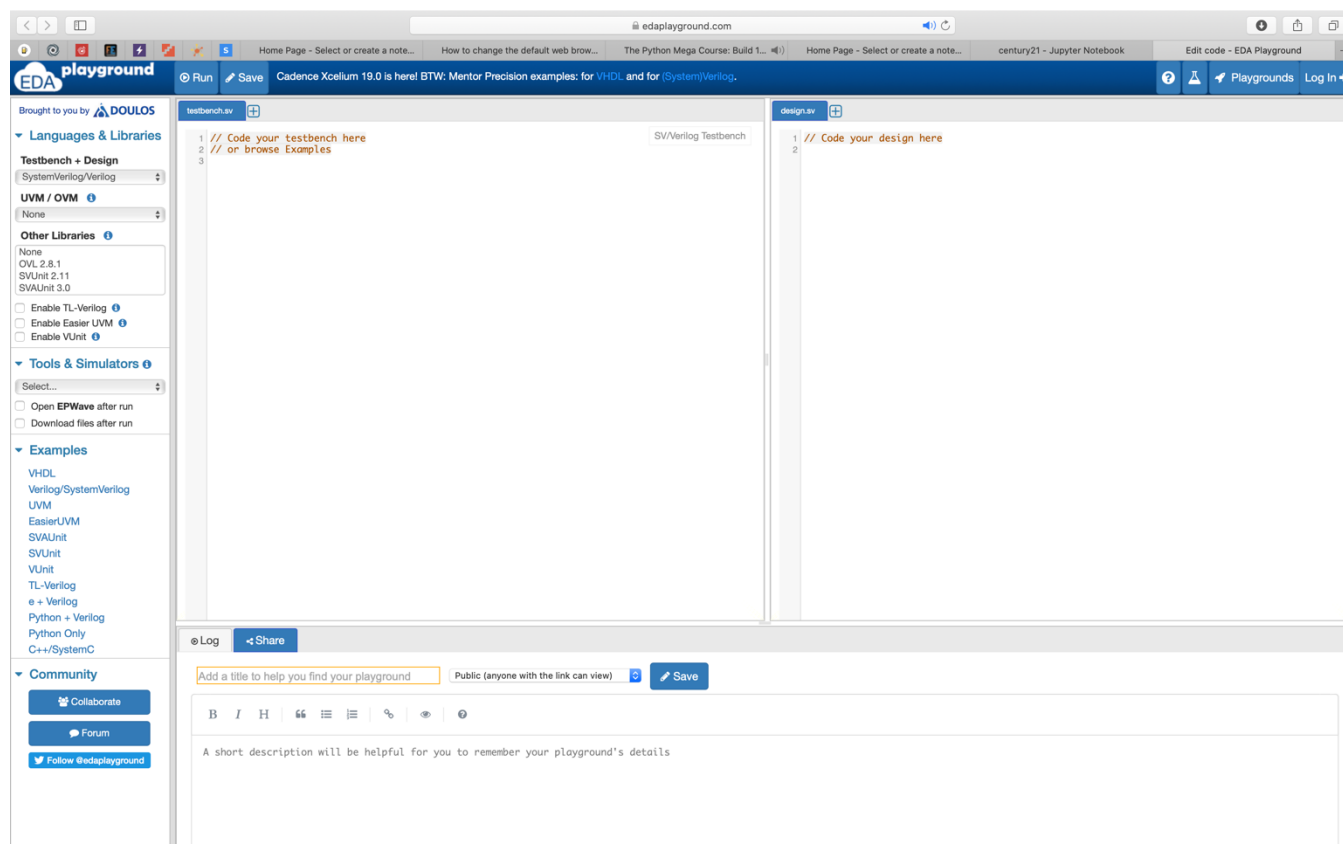
## Python for Web Scraping?

- **Ease of Use:** Python is simple to code. You do not have to add semi-colons “;” or curly-braces “{}” anywhere. This makes it less messy and easy to use.
- **Large Collection of Libraries:** Python has a huge collection of libraries such as Numpy, Matplotlib, Pandas etc., which provides methods and services for various purposes. Hence, it is suitable for web scraping and for further manipulation of extracted data.
- **Dynamically typed:** In Python, you don’t have to define datatypes for variables, you can directly use the variables wherever required. This saves time and makes your job faster.
- **Easily Understandable Syntax:** Python syntax is easily understandable mainly because reading a Python code is very similar to reading a statement in English. It is expressive and easily readable, and the indentation used in Python also helps the user to differentiate between different scope/blocks in the code.
- **Small code, large task:** Web scraping is used to save time. But what’s the use if you spend more time writing the code? Well, you don’t have to. In Python, you can write small codes to do large tasks. Hence, you save time even while writing the code.
- **Community:** What if you get stuck while writing the code? You don’t have to worry. Python community has one of the biggest and most active communities, where you can seek help from.

<b>Date:</b>	<b>03/06/2020</b>	<b>Name:</b>	<b>K B KUSHI</b>
<b>Course:</b>	<b>DIGITAL DESIGN USING HDL</b>	<b>USN:</b>	<b>4AL17EC107</b>
<b>Topic:</b>	<ol style="list-style-type: none"> <li>1. EDA Playground Online compiler</li> <li>2. EDA Playground Tutorial Demo Video</li> <li>3. How to Download And Install Xilinx Vivado Design Suite</li> <li>4. Vivado Design Suite for implementation of HDL code</li> </ol>	<b>Semester &amp; Section:</b>	<b>6 B</b>
<b>Github Repository:</b>	<a href="https://github.com/alvas-education-foundation/KUSHI-COURSES.git">https://github.com/alvas-education-foundation/KUSHI-COURSES.git</a>		

## AFTERNOON SESSION DETAILS

### Image of session



**Verilog Tutorial 1 -- Ripple Carry Counter**  
58,978 views • 11 Nov 2013

EDA Playground  
5.12K subscribers

In this Verilog tutorial, we implement a basic Ripple Carry Counter design and test using Verilog.

Complete Ripple Carry Counter from the Verilog tutorial:

**Report – Report can be typed or hand written for up to two pages.**

- In today's session we noted how to use a EDA Playground Online complier
- EDA Playground Tutorial Demo Video helped us in getting familiarized with the tool.
- How to Download And Install Xilinx Vivado Design Suite and Vivado Design Suite for implementation of HDL code video taught us how to download and use the Xilinx to implement Verilog code.

**Implement 4 to 1 MUX using two 2 to 1 MUX using structural modelling style and test the module in online/offline compiler.**

**Code:**

**library IEEE;**

**use IEEE.STD\_LOGIC\_1164.ALL;**

**entity mux2\_1 is**

**port(A,B : in STD\_LOGIC;**

**S: in STD\_LOGIC;**

**Z: out STD\_LOGIC);**

**end mux2\_1;**

**architecture Behavioral of mux2\_1 is**

**begin**

**process (A,B,S) is**

**begin**

**if (S ='0') then**

**Z <= A;**

**else**

**Z <= B;**

**end if;**

**end process;**

**end Behavioral;**

**library IEEE;**

**use IEEE.STD\_LOGIC\_1164.ALL;**

**entity mux4\_1 is**

**port(**

**A,B,C,D : in STD\_LOGIC;**

**S0,S1: in STD\_LOGIC;**

**Z: out STD\_LOGIC**

**);**

**end mux4\_1;**

**architecture Behavioral of mux4\_1 is**

```

component mux2_1

port( A,B : in STD_LOGIC;

S: in STD_LOGIC;

Z: out STD_LOGIC);

end component;

signal temp1, temp2: std_logic;

begin

m1: mux2_1 port map(A,B,S0,temp1);

m2: mux2_1 port map(C,D,S0,temp2);

m3: mux2_1 port map(temp1,temp2,S1,Z);


end Behavioral;

```

### Output:

