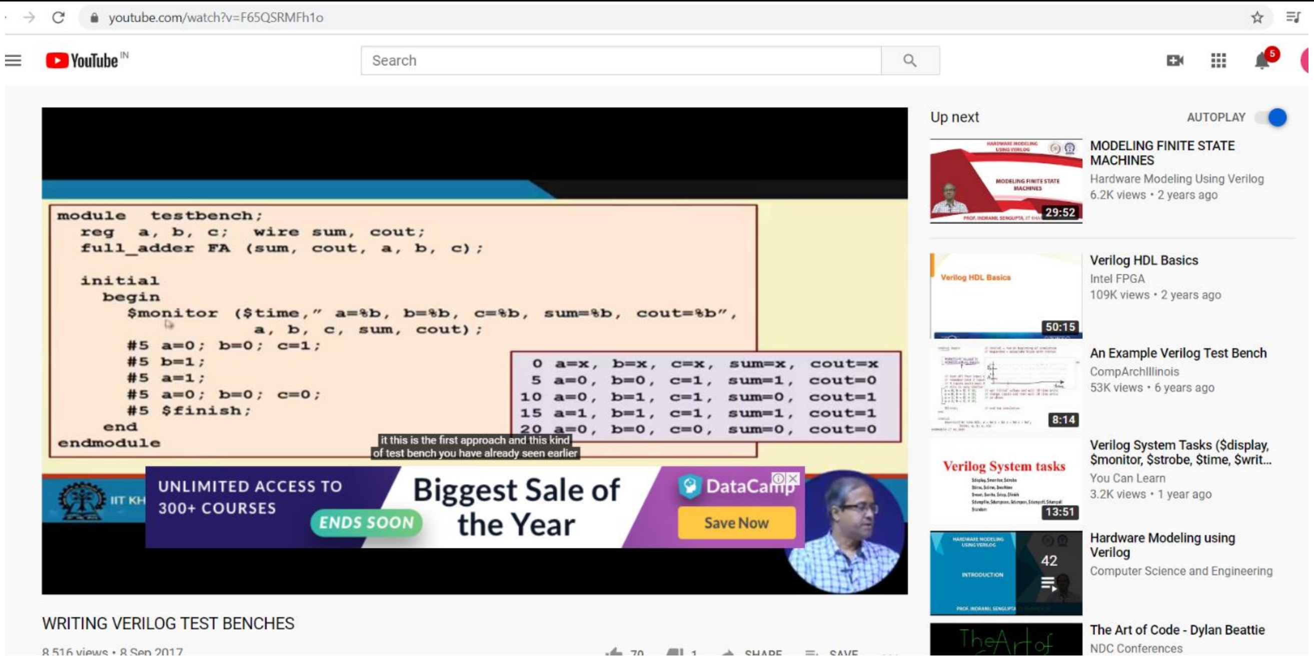
DAILY ASSESSMENT FORMAT

Date:	2-06-2020	Name:	BINDUSHRI
Course:	Digital design using hdl	USN:	4AL17EC011
Topic:	1.fpga basics architechture,application and uses. 2.verilog hdl basics 3.test bench waveform 4.task2	Semester & Section:	6th A
Github Repository:	Bindushri		

FORENOON SESSION DETAILS



```
module  testbench;
  reg  a, b, c;  wire sum, cout;
  full_adder FA (sum, cout, a, b, c);

  initial
    begin
      $monitor ($time," a=%b, b=%b, c=%b, sum=%b, cout=%b",
                      a, b, c, sum, cout);
      #5 a=0; b=0; c=1;
      #5 b=1;
      #5 a=1;
      #5 a=0; b=0; c=0;
      #5 $finish;
    end
endmodule
```

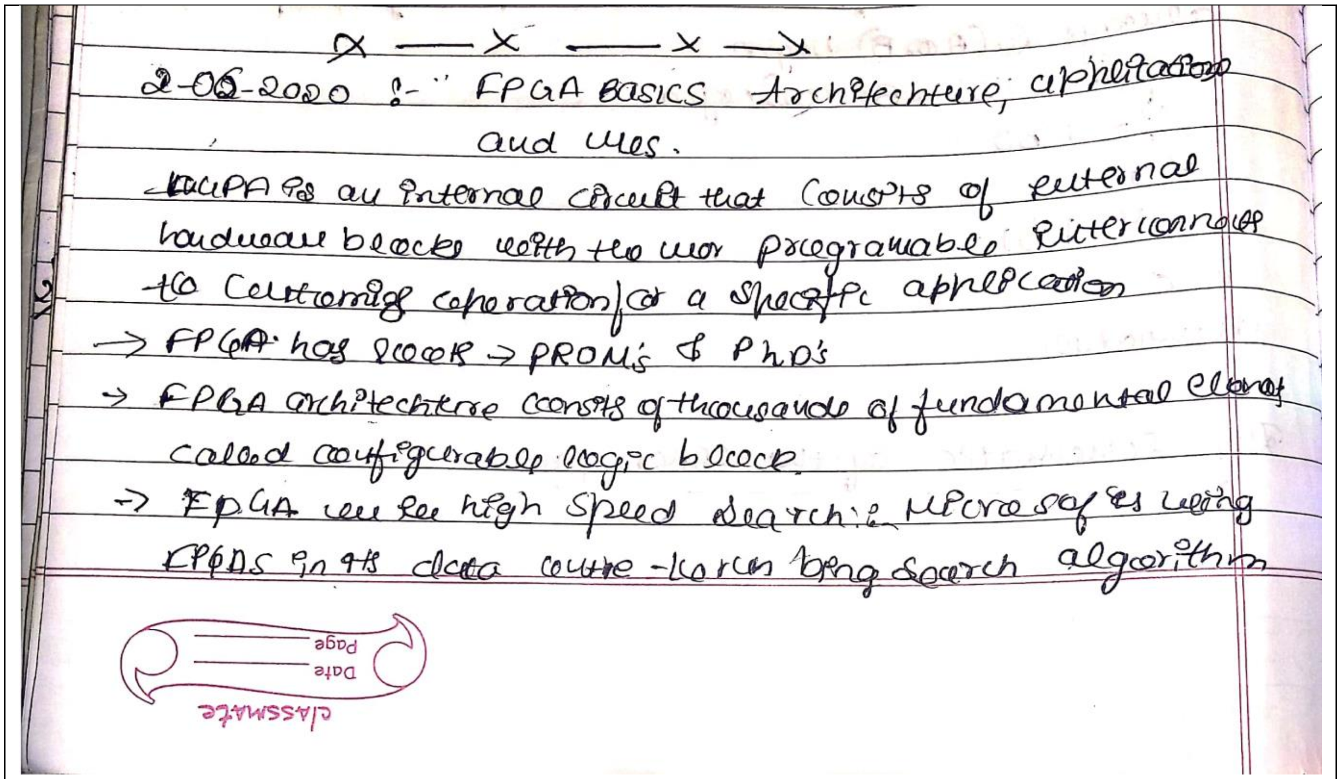
```
 0 a=x, b=x, c=x, sum=x, cout=x
 5 a=0, b=0, c=1, sum=1, cout=0
10 a=0, b=1, c=1, sum=0, cout=1
15 a=1, b=1, c=1, sum=1, cout=1
20 a=0, b=0, c=0, sum=0, cout=0
```

it this is the first approach and this kind of test bench you have already seen earlier

WRITING VERILOG TEST BENCHES

8,516 views • 8 Sep 2017

youtube.com/watch?v=F65QSRMFh1o
YouTube IN
Search
Up next
AUTOPLAY
MODELING FINITE STATE MACHINES
Hardware Modeling Using Verilog
6.2K views • 2 years ago
29:52
Verilog HDL Basics
Intel FPGA
109K views • 2 years ago
50:15
An Example Verilog Test Bench
CompArchIllinois
53K views • 6 years ago
8:14
Verilog System Tasks ($display, $monitor, $strobe, $time, $writ...
You Can Learn
3.2K views • 1 year ago
13:51
Hardware Modeling using Verilog
Computer Science and Engineering
42
The Art of Code - Dylan Beattie
NDC Conferences
UNLIMITED ACCESS TO 300+ COURSES
ENDS SOON
Biggest Sale of the Year
DataCamp
Save Now
70
1
SHARE
SAVE



x —— x —— x ——>

2-06-2020 :- FPGA BASICS Architechture, applications and uses.

FPGA is an internal circuit that consists of external hardware blocks with the user programable interconnects to customise operation for a specific application

→ FPGA has look → PROM's & PLD's

→ FPGA architechture consists of thousands of fundamental elements called configurable logic block.

→ FPGA uses high speed search: Microsoft is using FPGAs in its data centre - to run bing search algorithm

~~fri~~) verilog HDL Basics

RTL ⇒ type of behavioural modeling for the purpose of synthesis

```
module module_name(port_list);
Port declarations
data type declarations
Circuit functionality
timing specifications
endmodule
```

variable dat type ⇒ element to store data temporarily

net data type → physical interconnect b/w structure
↳ tri
↳ wire

ex:- module clk_gen

```
(
 output reg clk
);
 initial clk = 1'b0.
 always
   #(period/2)clk = ~clk;
   initial #100 $finish
endmodule
```

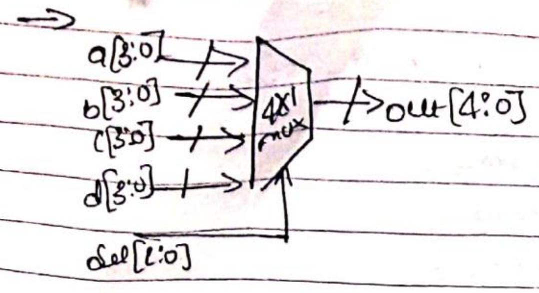
Procedural Assignments :- ① Blocking (=)
② non-blocking (<=)

→ verilog testbench code to verify the design under test

```
module full_adder(s, co, a, b, c);
  input a, b, c;
  output s, co;
  assign s = a^b^c;
  assign co = (a&b)|(b&c)|(c&a);
endmodule
```

```
module testbench
 reg a, b, c; wire sum, cout;
full_adder FA(sum, cout, a, b, c);
initial
 begin
 $monitor($time, "a=%b, b=%b,
 c=%b, sum=%b, cout=%b",
  a, b, c, sum, cout);
#5 a=0; b=0; c=1;
#5 b=1;
#5 a=1;
#5 a=0; b=0; c=0;
end $finish
endmodule
```

Task day2 Impleement a 4:1 mux and write the test bench code to verify the module

→


sel a

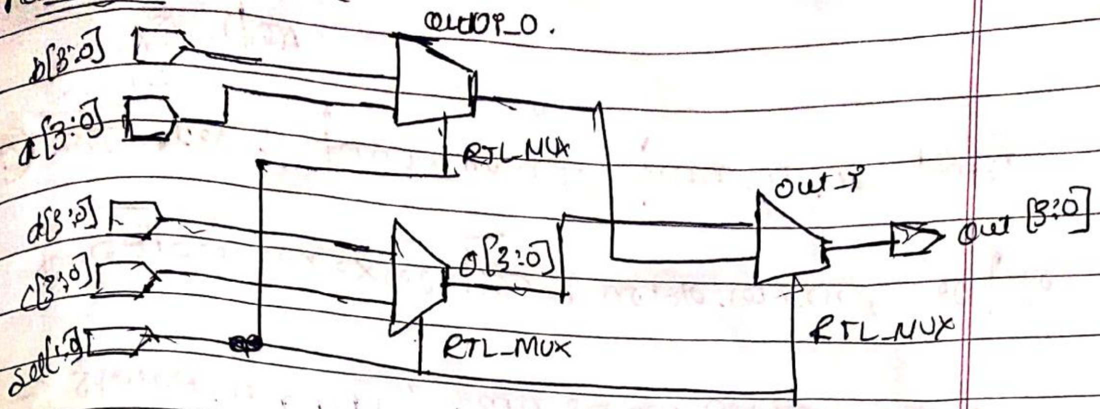
• using assign statement

```
module mux_4to1_assign(input [3:0] a,
                       input [3:0] b,
                       input [3:0] c,
                       input [3:0] d,
                       input [1:0] sel,
                       output [3:0] out);
  assign out = sel[1] ? (sel[0] ? a : c) : (sel[0] ? b : a);
endmodule.
```

using case statement

```
module ~~mux~~_4to1 case (input [3:0] a,
                        input [3:0] b,
                        input [3:0] c,
                        input [3:0] d,
                        input [1:0] sel,
                        output reg [3:0] out);
always @(a or b or c or d or sel) begin.
  case (sel)
       2'b00 : out <= a;
       2'b01 : out <= b;
       2'b10 : out <= c;
       2'b11 : out <= d;
    endcase
  end
endmodule.
```

Hardware Schematic

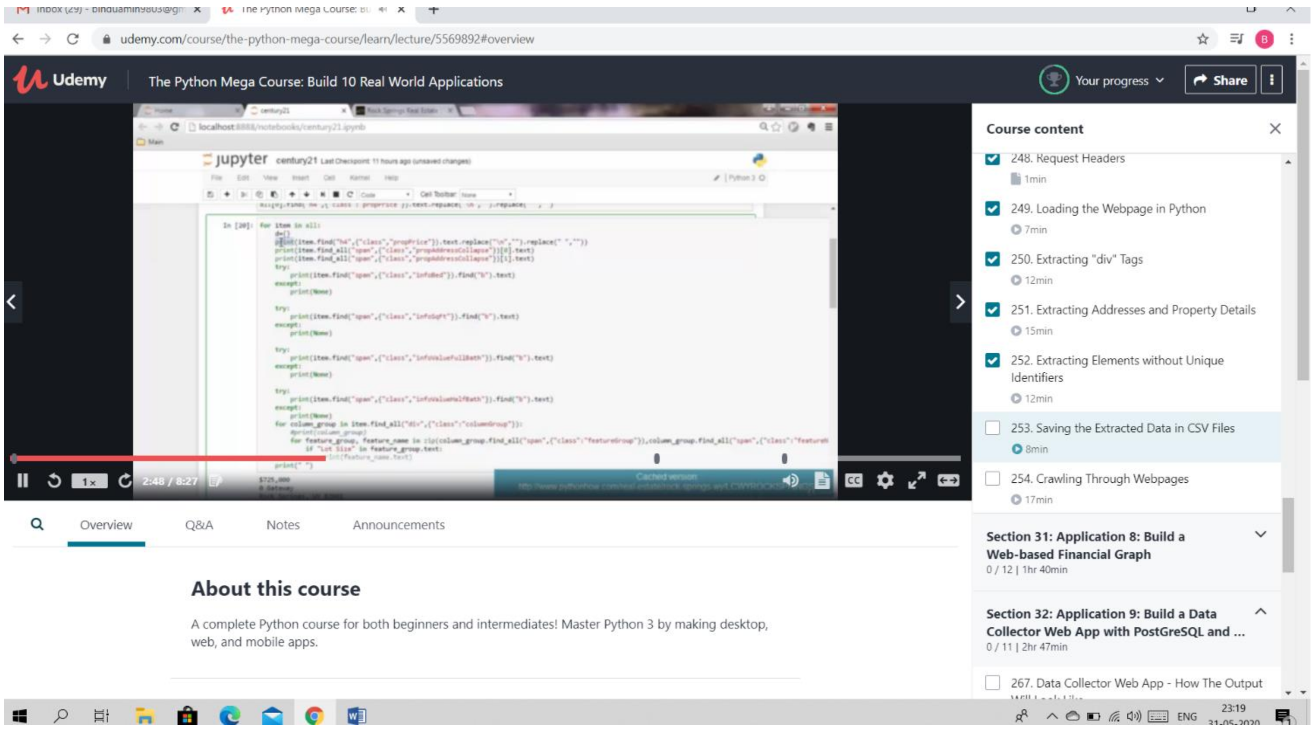


```
module tb_4to1_mux;
 reg [3:0] a;
 reg [3:0] b;
 reg [3:0] c;
 reg [3:0] d;
 wire [3:0] out;
 reg [1:0] sel;
 integer i;
 mux_4to1_case mux0 ( .a(a)
                      .b(b),
                      .c(c),
                      .d(d),
                      .sel(sel),
                      .out(out));
initial begin
      $monitor ("[%0t] sel =0x%0h a=0x%0h b=0x%0h c=0x%0h
                 d=0x%0h out= 0x%0h, $time, sel, a,
                sel <=0;
                a <= $random;
                b <= $random;
                c <= $random;
                d <= $random;
          for(i=1; i<4 ; i=i+1) begin.
       end
       end #5 finish;
   endmodule
```

Date:2june2020 Name:Bindushri
Course: python USN:4AL17EC011
Topic: sec30 Sem&Sec:6th A

AFTERNOON SESSION DETAILS

Image of session



udemy.com/course/the-python-mega-course/learn/lecture/5569892#overview

Udemy The Python Mega Course: Build 10 Real World Applications

jupyter century21 Last Checkpoint: 11 hours ago (unsaved changes)

```
In [20]: for item in all:
             d={}
             print(item.find("h4",{"class","propPrice"}).text.replace("\n","").replace(" ",""))
             print(item.find_all("span",{"class","propAddressCollapse"})[0].text)
             print(item.find_all("span",{"class","propAddressCollapse"})[1].text)
             try:
                 print(item.find("span",{"class","infoBed"}).find("b").text)
             except:
                 print(None)

             try:
                 print(item.find("span",{"class","infoSqFt"}).find("b").text)
             except:
                 print(None)

             try:
                 print(item.find("span",{"class","infoValueFullBath"}).find("b").text)
             except:
                 print(None)

             try:
                 print(item.find("span",{"class","infoValueHalfBath"}).find("b").text)
             except:
                 print(None)
             for column_group in item.find_all("div",{"class":"columnGroup"}):
                 #print(column_group)
                 for feature_group, feature_name in zip(column_group.find_all("span",{"class":"featureGroup"}),column_group.find_all("span",{"class":"featureN
                     if "Lot Size" in feature_group.text:
                         print(feature_name.text)
             print(" ")
```

2:48 / 8:27

Course content

- ✓ 248. Request Headers 1min
- ✓ 249. Loading the Webpage in Python 7min
- ✓ 250. Extracting "div" Tags 12min
- ✓ 251. Extracting Addresses and Property Details 15min
- ✓ 252. Extracting Elements without Unique Identifiers 12min
- ☐ 253. Saving the Extracted Data in CSV Files 8min
- ☐ 254. Crawling Through Webpages 17min

Section 31: Application 8: Build a Web-based Financial Graph
0 / 12 | 1hr 40min

Section 32: Application 9: Build a Data Collector Web App with PostGreSQL and ...
0 / 11 | 2hr 47min

- ☐ 267. Data Collector Web App - How The Output Will Look Like

Overview Q&A Notes Announcements

About this course

A complete Python course for both beginners and intermediates! Master Python 3 by making desktop, web, and mobile apps.

23:19 31-05-2020

2-06-2020 Sec: 30. Scrape Real Estate property Data from the web

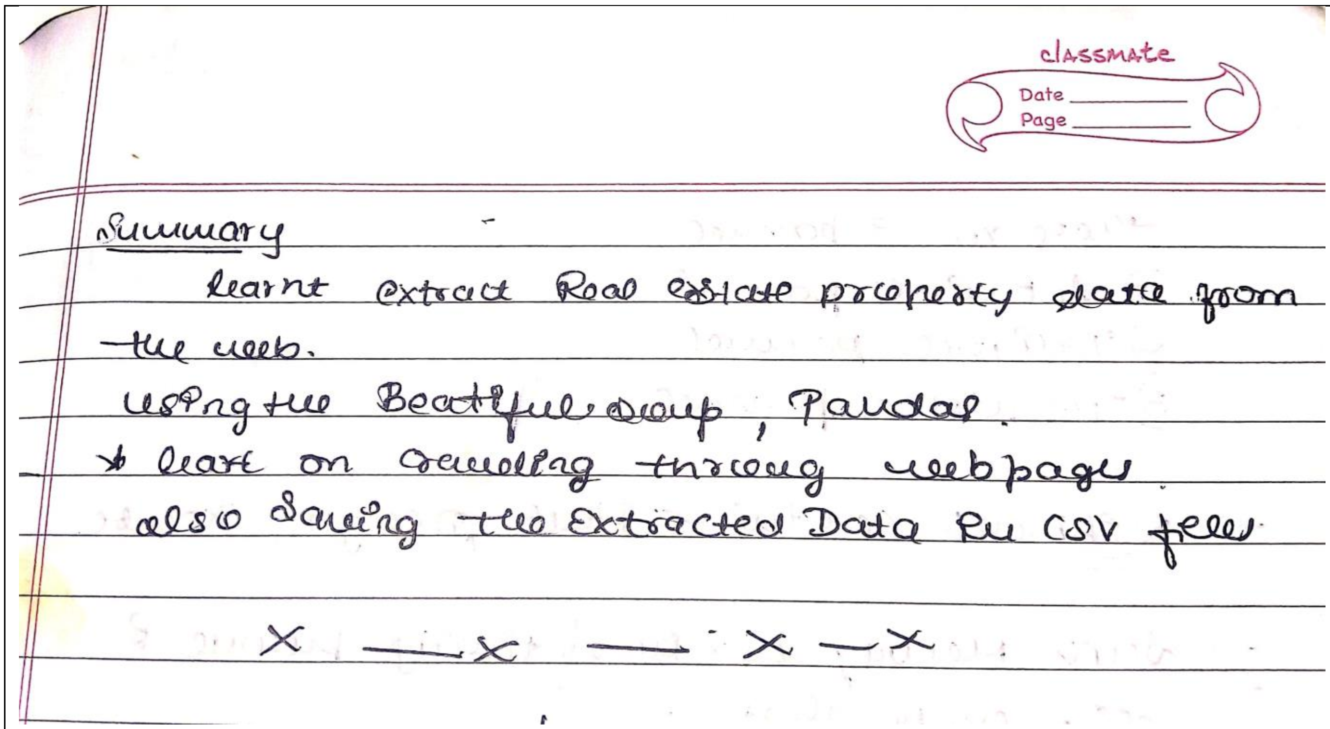
→ Jupyter notebook

```
In[] 1. import request
     2. from bs4 import Beautifulsoup

In[] r= requests.get ("http://www.century21.com/real-estate/
                       rock-springs-wy/LCWYROCKSPRINGS/
     C= r.content

In[] Soup= Beautifulsoup (C, "html. parser")
     Print

In[] all = Soup.find_all ("div", {"class": "propertyRow"})

In[] all [0]. find ("h4", {"class": "propPrice"}). text.replace
                       ("\n", ""). replace (" ", " ")
```

o/p $725,000

Date
Page

```
In[]: for item in all:
        print (item.find ("h4", {"class" "propPrice"}).
                    text.replace("\n", "").replace(" ", "")

        ~~print~~(item.find_all("span", {"class", "propAddressCollapse"})
                                        [0].text)

        print (item.find_all ("span", {"class" "propAddressCollapse"})
                                        [1].text)

        try:
        ~~be~~ print (item.find ("span", {"class", "infoBed"}).
                            find("b").text}
        except:
            ~~print ("")~~ print(none)



        print(" ")

Extracting Elements without unique.

for column_group in item.find_all("div",{"class":
    "columngroup"}):
    print(column_group)
    for feature_group, feature_name in zip (
    column_group.find_all ("span", {"class":
    "featureGroup"}), column_group.find_all
    ("span",{"class":" featureName"})):
    print (feature_group.text, feature_name.text)
    if "lot size" in feature_group.text:
        print (feature_name.text)

print(" ")
```



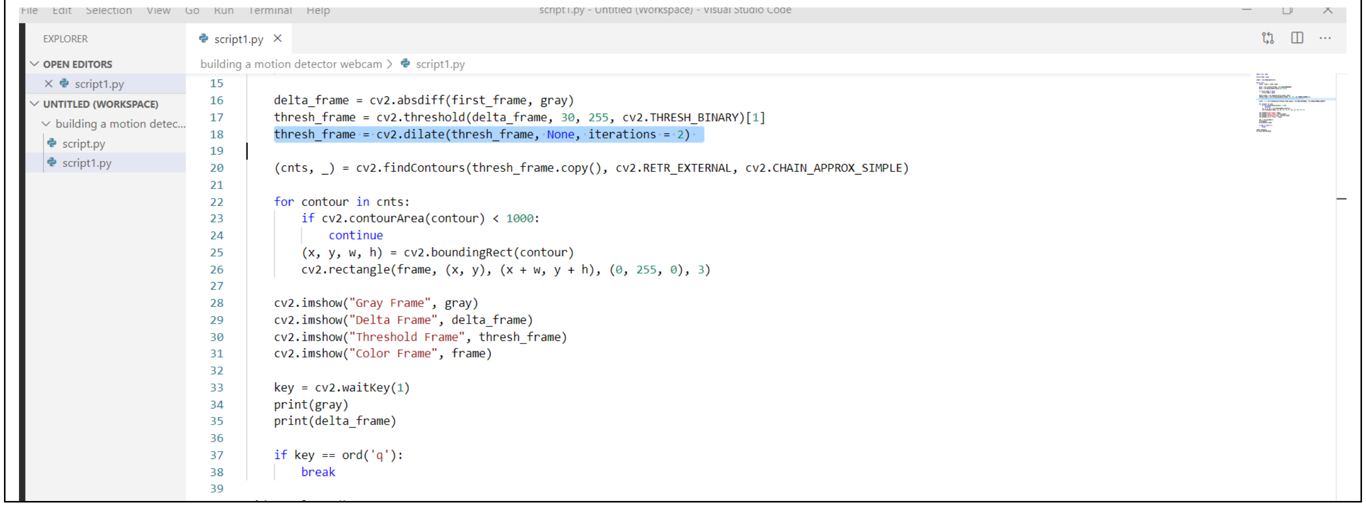
classmate
Date
Page

Summary
learnt extract Real estate property data from the web.
using the Beatiful soup, Pandas.
* leart on crawling throug webpage.
also saving the Extracted Data in csv file.

x — x — x — x

Python excersise program:

```
import cv2, time

first_frame = None

video = cv2.VideoCapture(0)

while True:
    check, frame = video.read()

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    gray = cv2.GaussianBlur(gray,(21,21),0)

    if first_frame is None:
        first_frame = gray
    delta_frame = cv2.absdiff(first_frame, gray)
    thresh_frame = cv2.threshold(delta_frame, 30, 255, cv2.THRESH_BINARY)[1]
    thresh_frame = cv2.dilate(thresh_frame, None, iterations = 2)

    (cnts, _) = cv2.findContours(thresh_frame.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

    for contour in cnts:
        if cv2.contourArea(contour) < 1000:
            continue
        (x, y, w, h) = cv2.boundingRect(contour)
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 3)

    cv2.imshow("Gray Frame", gray)
    cv2.imshow("Delta Frame", delta_frame)
    cv2.imshow("Threshold Frame", thresh_frame)
    cv2.imshow("Color Frame", frame)

    key = cv2.waitKey(1)
    print(gray)
    print(delta_frame)

    if key == ord('q'):
        break

video.release()
cv2.destroyAllWindows
```


File Edit Selection View Go Run Terminal Help
script1.py - Untitled (Workspace) - Visual Studio Code
EXPLORER
OPEN EDITORS
script1.py
UNTITLED (WORKSPACE)
building a motion detec...
script.py
script1.py
building a motion detector webcam > script1.py

```
    delta_frame = cv2.absdiff(first_frame, gray)
    thresh_frame = cv2.threshold(delta_frame, 30, 255, cv2.THRESH_BINARY)[1]
    thresh_frame = cv2.dilate(thresh_frame, None, iterations = 2)

    (cnts, _) = cv2.findContours(thresh_frame.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    for contour in cnts:
        if cv2.contourArea(contour) < 1000:
            continue
        (x, y, w, h) = cv2.boundingRect(contour)
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 3)

    cv2.imshow("Gray Frame", gray)
    cv2.imshow("Delta Frame", delta_frame)
    cv2.imshow("Threshold Frame", thresh_frame)
    cv2.imshow("Color Frame", frame)

    key = cv2.waitKey(1)
    print(gray)
    print(delta_frame)

    if key == ord('q'):
        break

```