**Task 1:**

Q1. Write a Python Program to implement your own myreduce() function which works exactly like Python's built-in function reduce().

Ans:

def myreduce(func, seq):

result = seq[0]

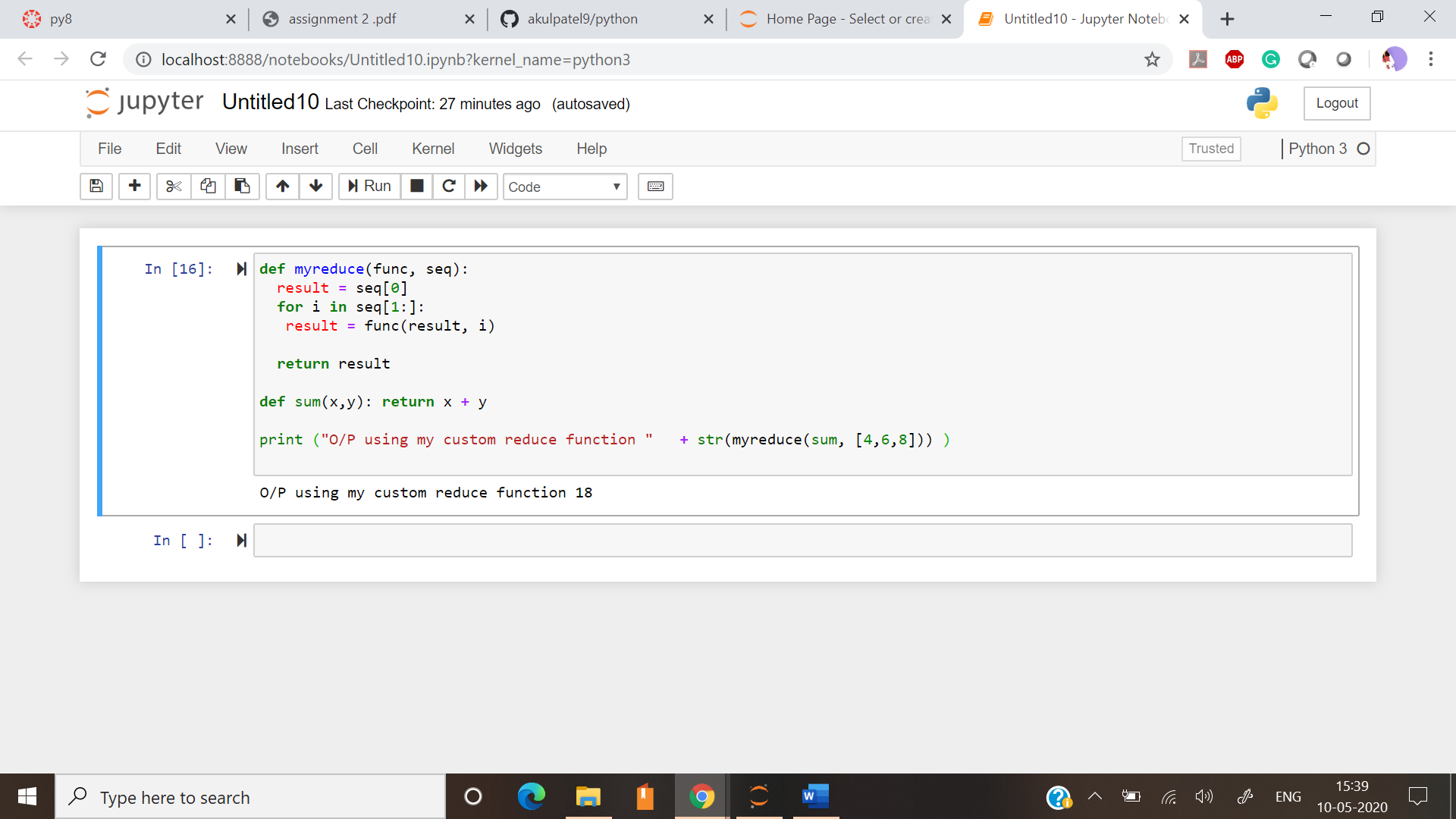
for i in seq[1:]:

result = func(result, i)

return result

def sum(x,y): return x + y

print ("O/P using my custom reduce function " + str(myreduce(sum, [4,6,8])) )



Q2. Write a Python program to implement your own myfilter() function which works exactly like Python's built-in function filter()

Ans:

def myfilter(func, seq):

result = []

for item in seq:

if func(item):

result.append(item)

return result

def ispositive(x):

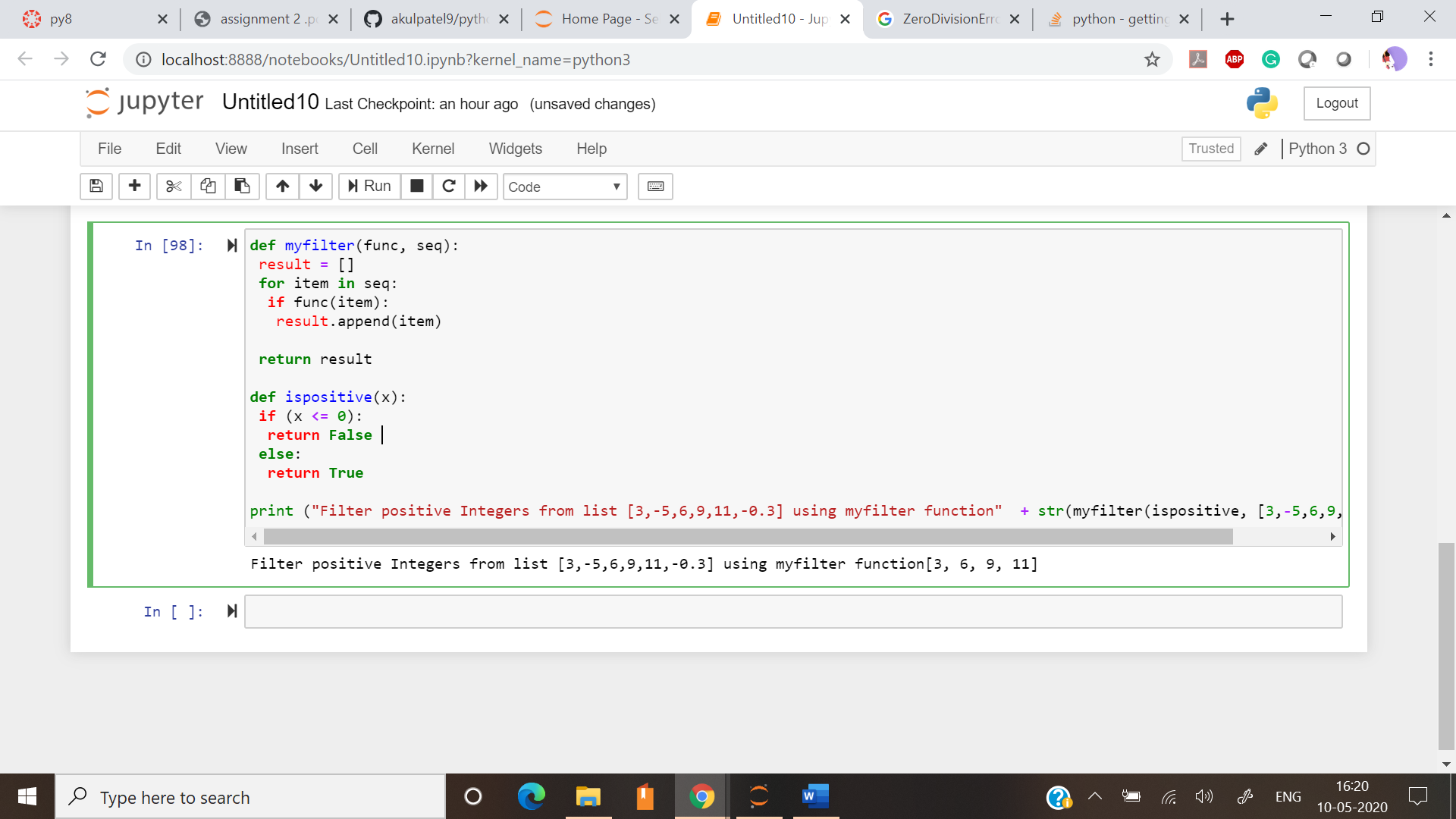
if (x <= 0):

return False

else:

return True

print ("Filter positive Integers from list [3,-5,6,9,11,-0.3] using myfilter function" + str(myfilter(ispositive, [3,-5,6,9,11,-0.3])))



Q3. Implement List comprehensions to produce the following lists. Write List comprehensions to produce the following Lists

['A', 'C', 'A', 'D', 'G', 'I', ’L’, ‘ D’]

['x', 'xx', 'xxx', 'xxxx', 'y', 'yy', 'yyy', 'yyyy', 'z', 'zz', 'zzz', 'zzzz']

['x', 'y', 'z', 'xx', 'yy', 'zz', 'xx', 'yy', 'zz', 'xxxx', 'yyyy', 'zzzz']

[[2], [3], [4], [3], [4], [5], [4], [5], [6]]

[[2, 3, 4, 5], [3, 4, 5, 6], [4, 5, 6, 7], [5, 6, 7, 8]]

[(1, 1), (2, 1), (3, 1), (1, 2), (2, 2), (3, 2), (1, 3), (2, 3), (3, 3)]

Ans:

string = "ACADGILD"

string\_list = [ alphabet for alphabet in string]

print ("ACADGILD => " + str(string\_list))

ip\_1 = ['x','y','z']

result = [ item\*num for item in ip\_1 for num in range(1,5) ]

print("['x','y','z'] => " + str(result))

ip\_2 = ['x','y','z']

result = [ item\*num for num in range(1,5) for item in ip\_2 ]

print("['x','y','z'] => " + str(result))

ip\_3 = [2,3,4]

result = [ [item+num] for item in ip\_3 for num in range(0,3)]

print("[2,3,4] =>" + str(result))

ip\_4 = [2,3,4,5]

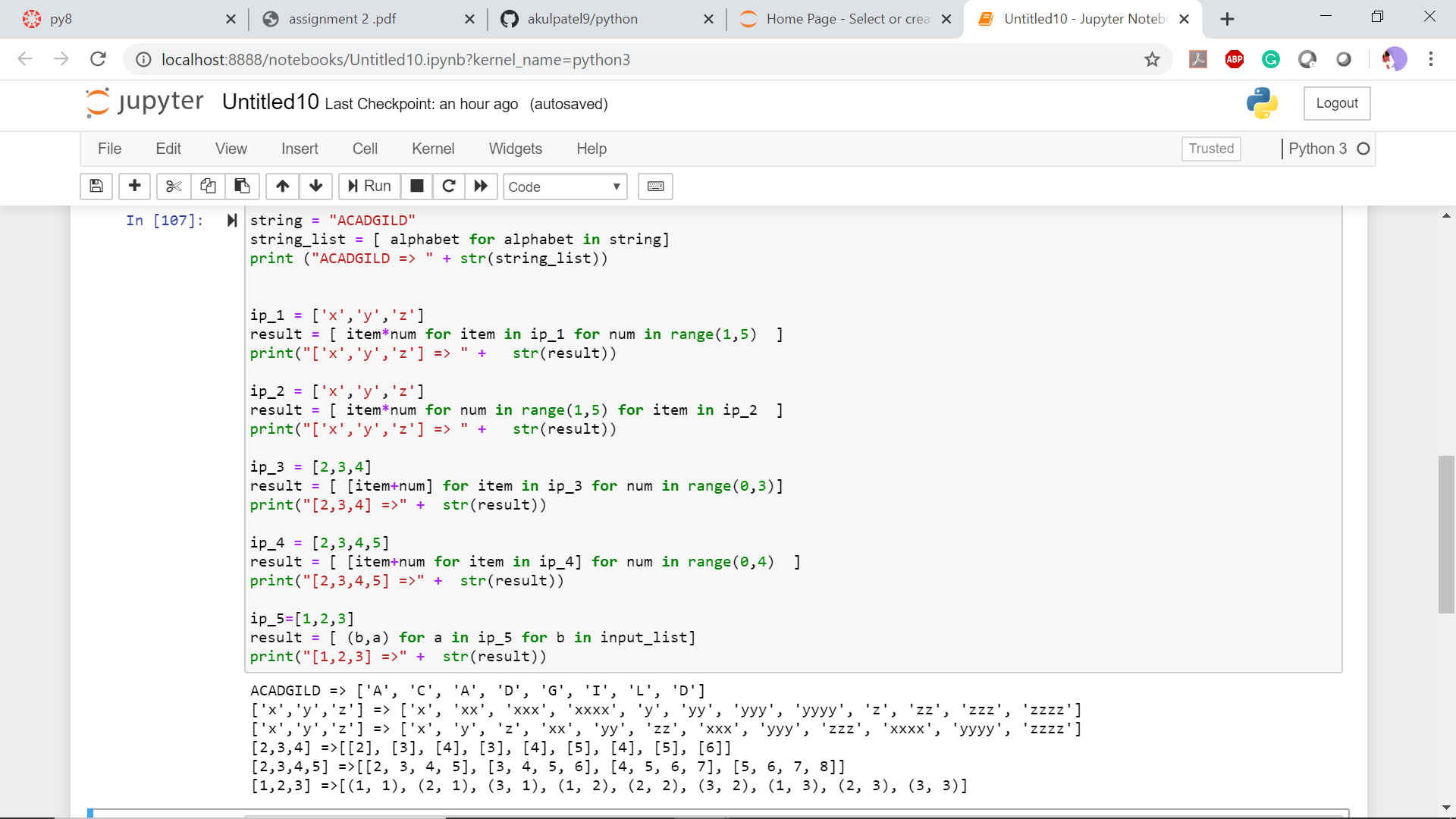
result = [ [item+num for item in ip\_4] for num in range(0,4) ]

print("[2,3,4,5] =>" + str(result))

ip\_5=[1,2,3]

result = [ (b,a) for a in ip\_5 for b in input\_list]

print("[1,2,3] =>" + str(result))



Q3. Implement a function longestWord() that takes a list of words and returns the longest one.

Ans:

def get\_longest\_word(w\_list):

word\_len = []

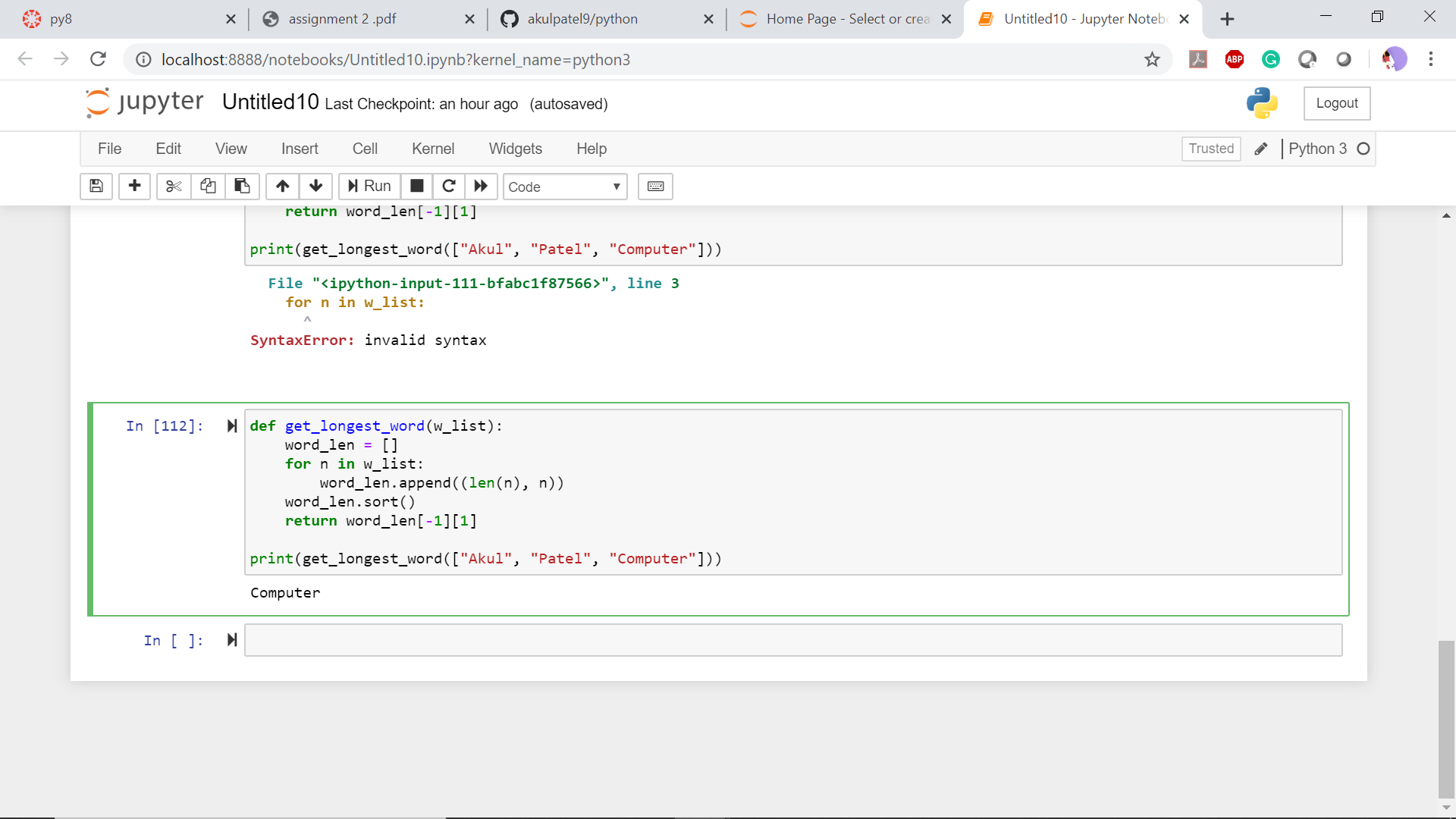
for n in w\_list:

word\_len.append((len(n), n))

word\_len.sort()

return word\_len[-1][1]

print(get\_longest\_word(["Akul", "Patel", "Computer"]))



**Task 2:**

Q1: Write a Python Program(with class concepts) to find the area of the triangle using the below formula. area = (s\*(s-a)\*(s-b)\*(s-c)) \*\* 0.5

Function to take the length of the sides of triangle from user should be defined in the parent class and function to calculate the area should be defined in subclass.

Ans:

class Triangle:

def \_\_init\_\_(user,a,b,c):

user.a = float(a)

user.b = float(b)

user.c = float(c)

def area(user):

s=(user.a + user.b + user.c)/2

return((s\*(s-user.a)\*(s-user.b)\*(s-user.c))\*\*0.5)

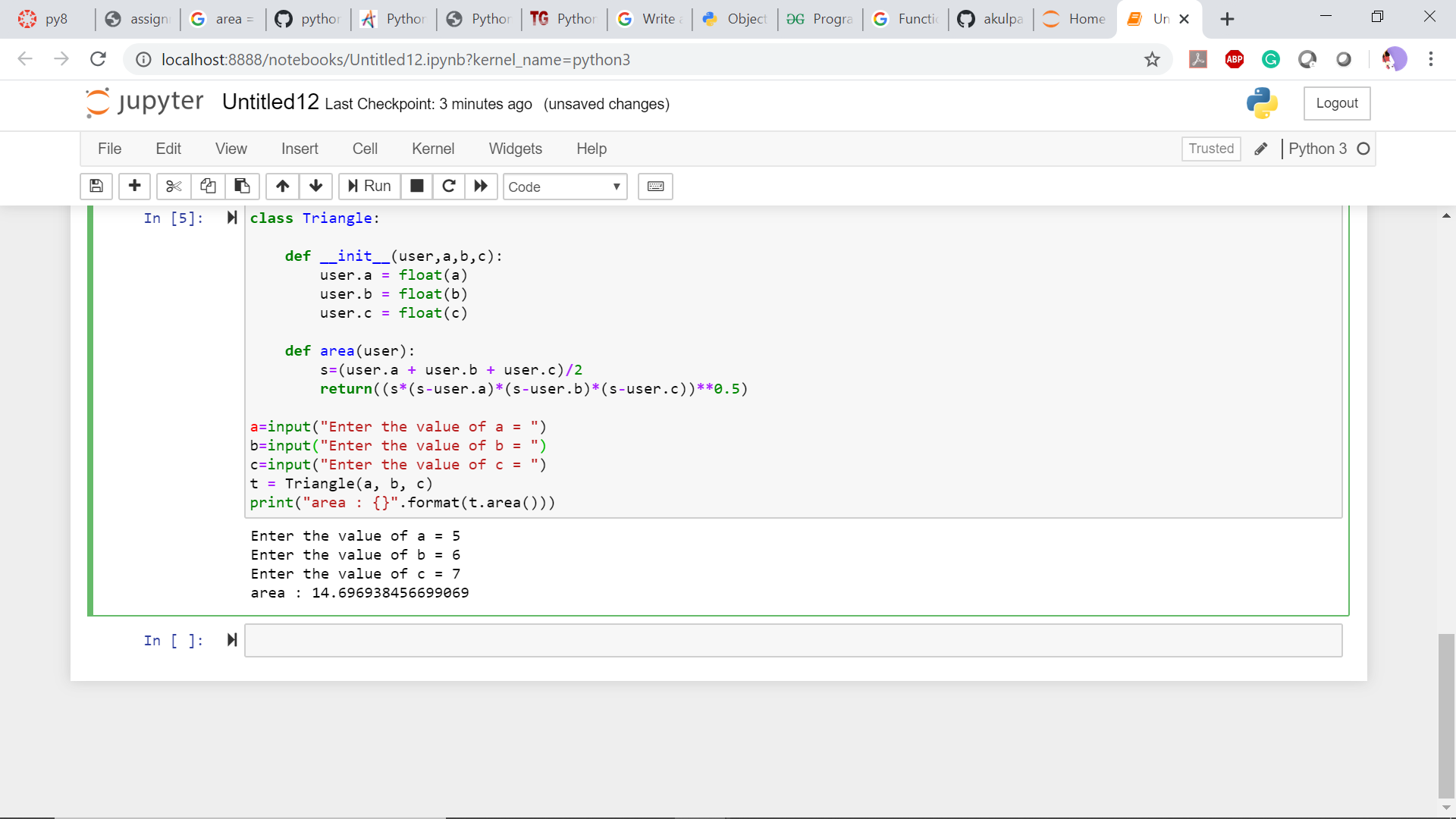
a=input("Enter the value of a = ")

b=input("Enter the value of b = ")

c=input("Enter the value of c = ")

t = Triangle(a, b, c)

print("area : {}".format(t.area()))



Q2. Write a function filter\_long\_words() that takes a list of words and an integer n and returns the list of words that are longer than n.

Ans: Tried using lambda function but couldn’t get desired output, hence used split function.

def filter\_long\_words(l,a):

words=[]

for j in l:

if(len(j)>=a):

words.append(j)

return words

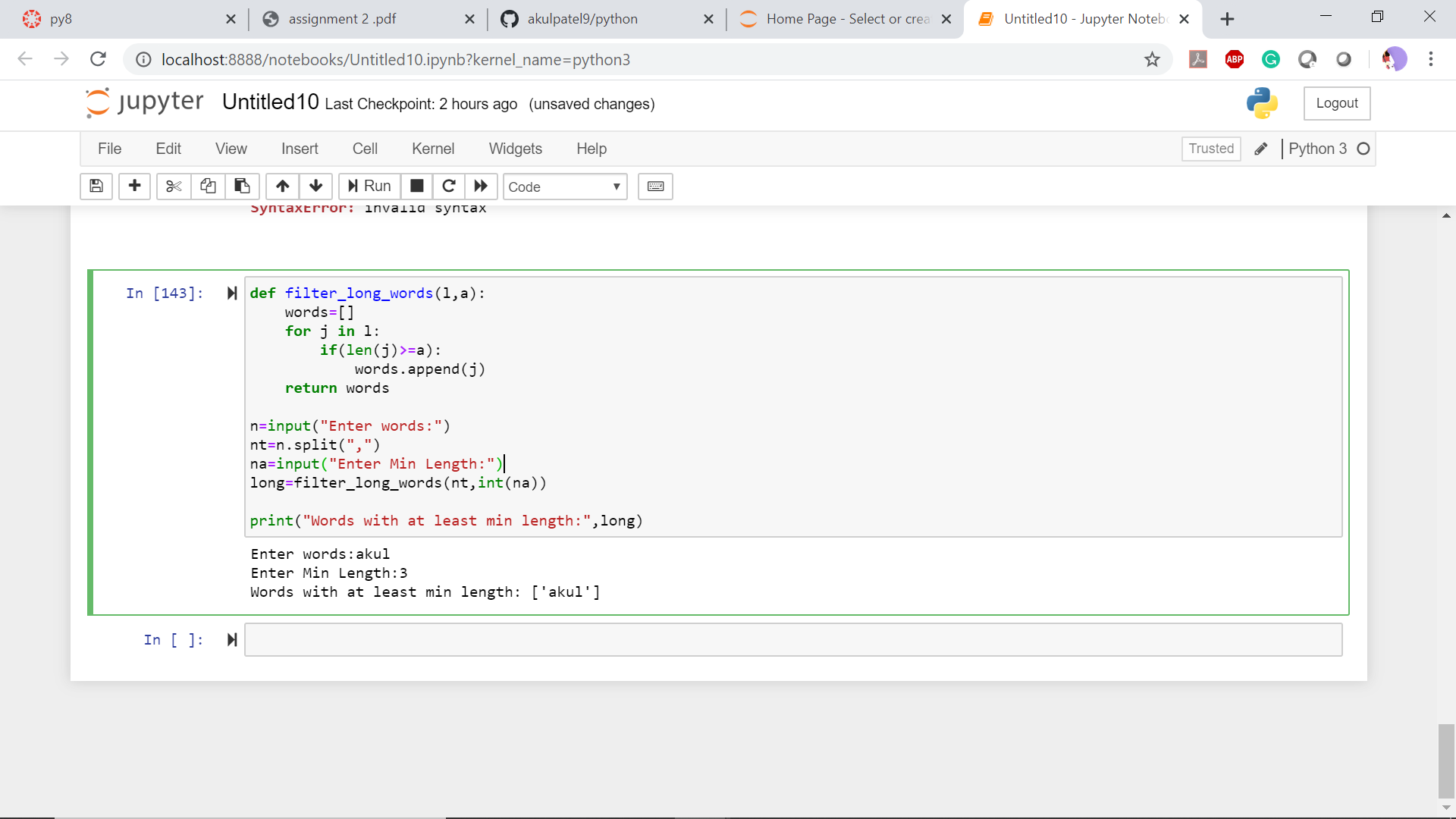
n=input("Enter words:")

nt=n.split(",")

na=input("Enter Min Length:")

long=filter\_long\_words(nt,int(na))

print("Words with at least min length:",long)



Q3. Write a Python program using function concept that maps list of words into a list of integers representing the lengths of the corresponding words. Hint: If a list [ ab,cde,erty] is passed on to the python function output should come as [2,3,4] Here 2,3 and 4 are the lengths of the words in the list.

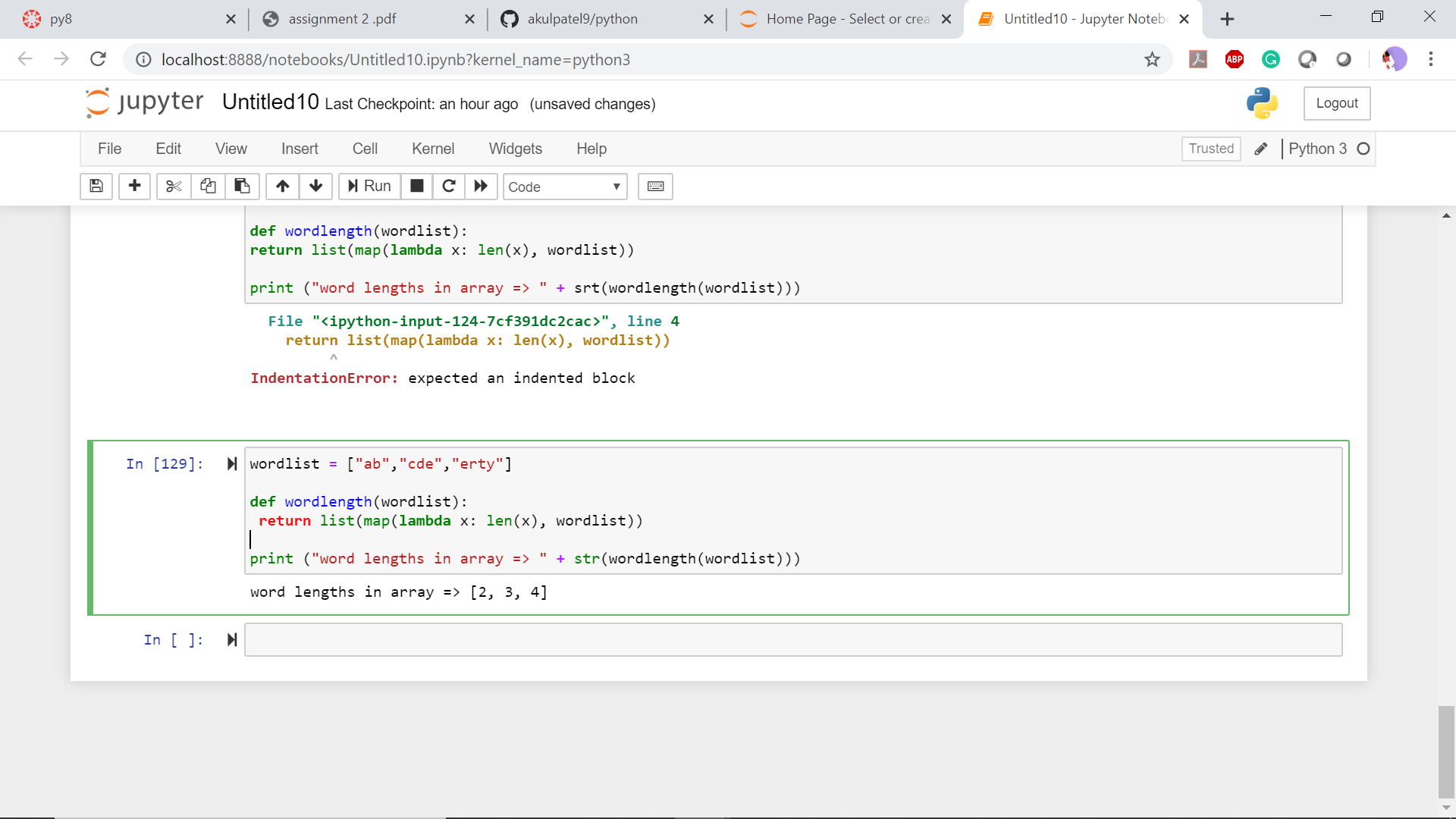
Ans:

wordlist = ["ab","cde","erty"]

def wordlength(wordlist):

return list(map(lambda x: len(x), wordlist))

print ("word lengths in array => " + str(wordlength(wordlist)))



Q4. Write a Python function which takes a character (i.e. a string of length 1) and returns True if it is a vowel, False otherwise

Ans:

def is\_vowel(char):

all\_vowels = 'aeiou'

return char in all\_vowels

print(is\_vowel('a'))

print(is\_vowel('k'))

print(is\_vowel('u'))

print(is\_vowel('l'))

