1. Create a function that takes the width, height and character and returns a picture frame as a 2D list.

**Examples**

get\_frame(4, 5, "#") ➞ [

["####"],

["# #"],

["# #"],

["# #"],

["####"]

]

# Frame is 4 characters wide and 5 characters tall.

get\_frame(10, 3, "\*") ➞ [

["\*\*\*\*\*\*\*\*\*\*"],

["\* \*"],

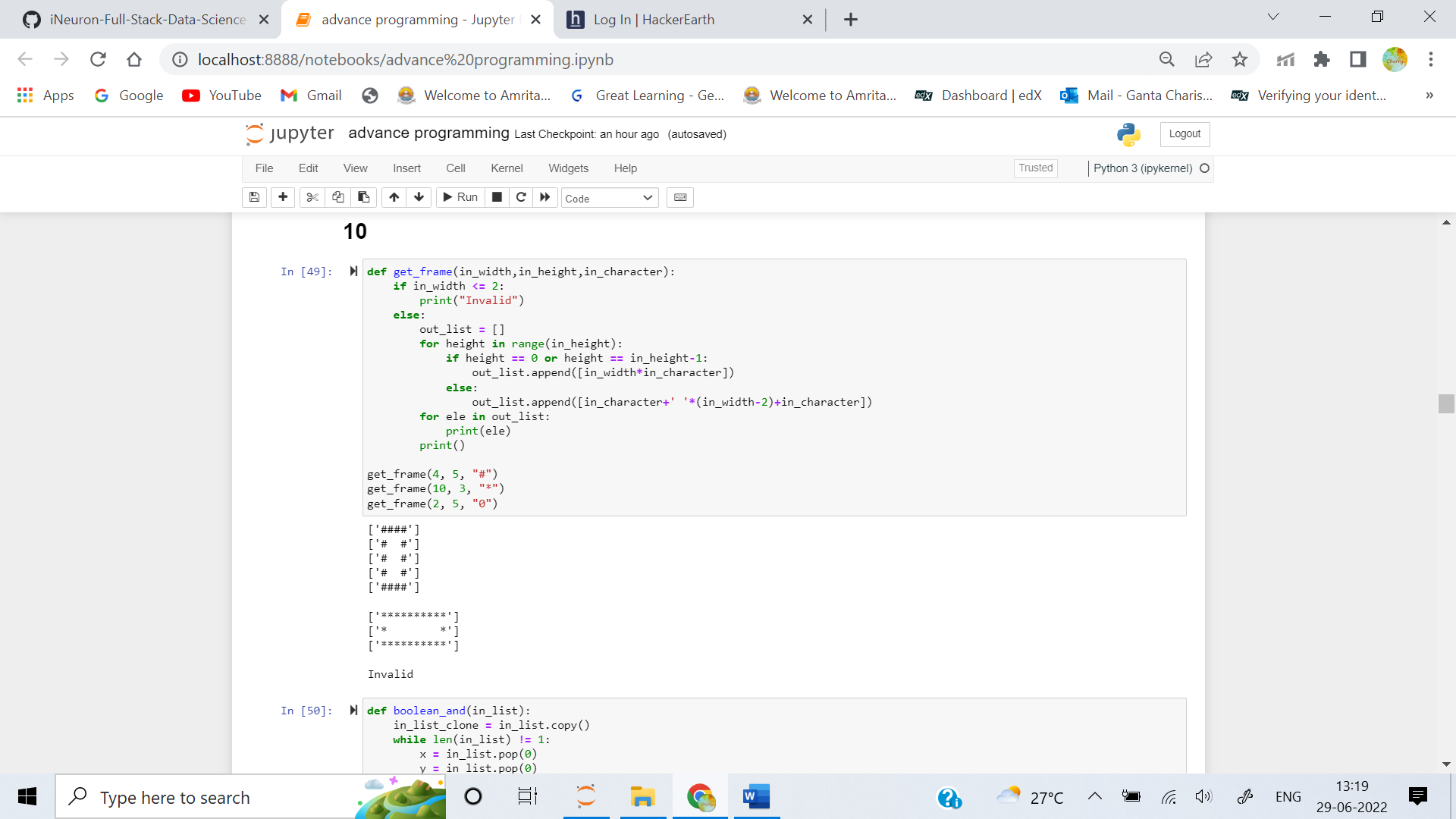
["\*\*\*\*\*\*\*\*\*\*"]

]

# Frame is 10 characters and wide and 3 characters tall.

get\_frame(2, 5, "0") ➞ "invalid"

# Frame's width is not more than 2.

**ANS:** 

2. Write three functions:

1. boolean\_and

2. boolean\_or

3. boolean\_xor

These functions should evaluate a list of True and False values, starting from the leftmost element and evaluating pairwise.

**Examples**

boolean\_and([True, True, False, True]) ➞ False

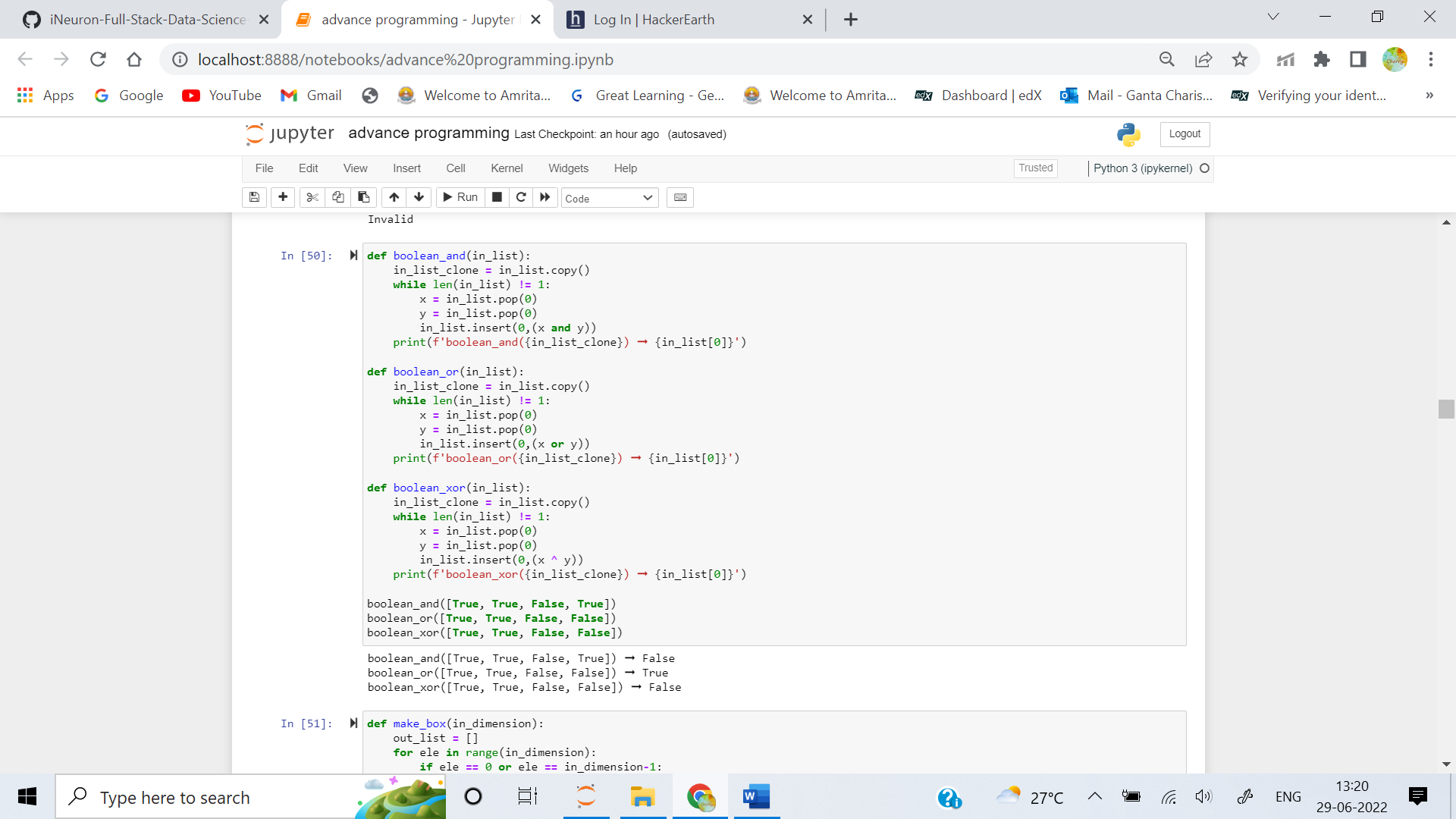
# [True, True, False, True] => [True, False, True] => [False, True] => False

boolean\_or([True, True, False, False]) ➞ True

# [True, True, False, True] => [True, False, False] => [True, False] => True

boolean\_xor([True, True, False, False]) ➞ False

# [True, True, False, False] => [False, False, False] => [False, False] => False

**ANS:** 

3. Create a function that creates a box based on dimension n.

**Examples**

make\_box(5) ➞ [

"#####",

"# #",

"# #",

"# #",

"#####"

]

make\_box(3) ➞ [

"###",

"# #",

"###"

]

make\_box(2) ➞ [

"##",

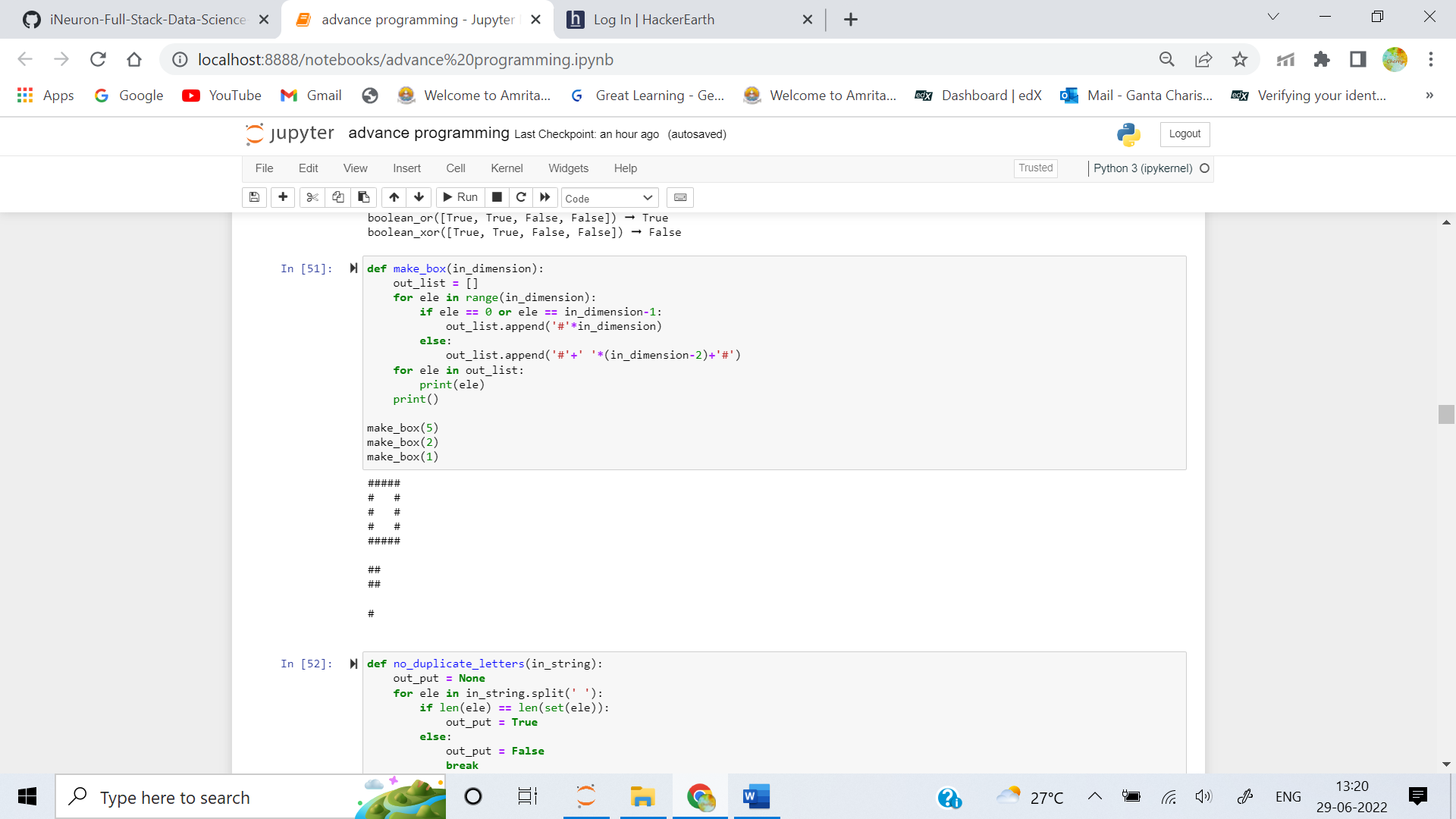
"##"

]

make\_box(1) ➞ [

"#"

]

**ANS:** 

4. Given a common phrase, return False if any individual word in the phrase contains duplicate letters. Return True otherwise.

**Examples**

no\_duplicate\_letters("Fortune favours the bold.") ➞ True

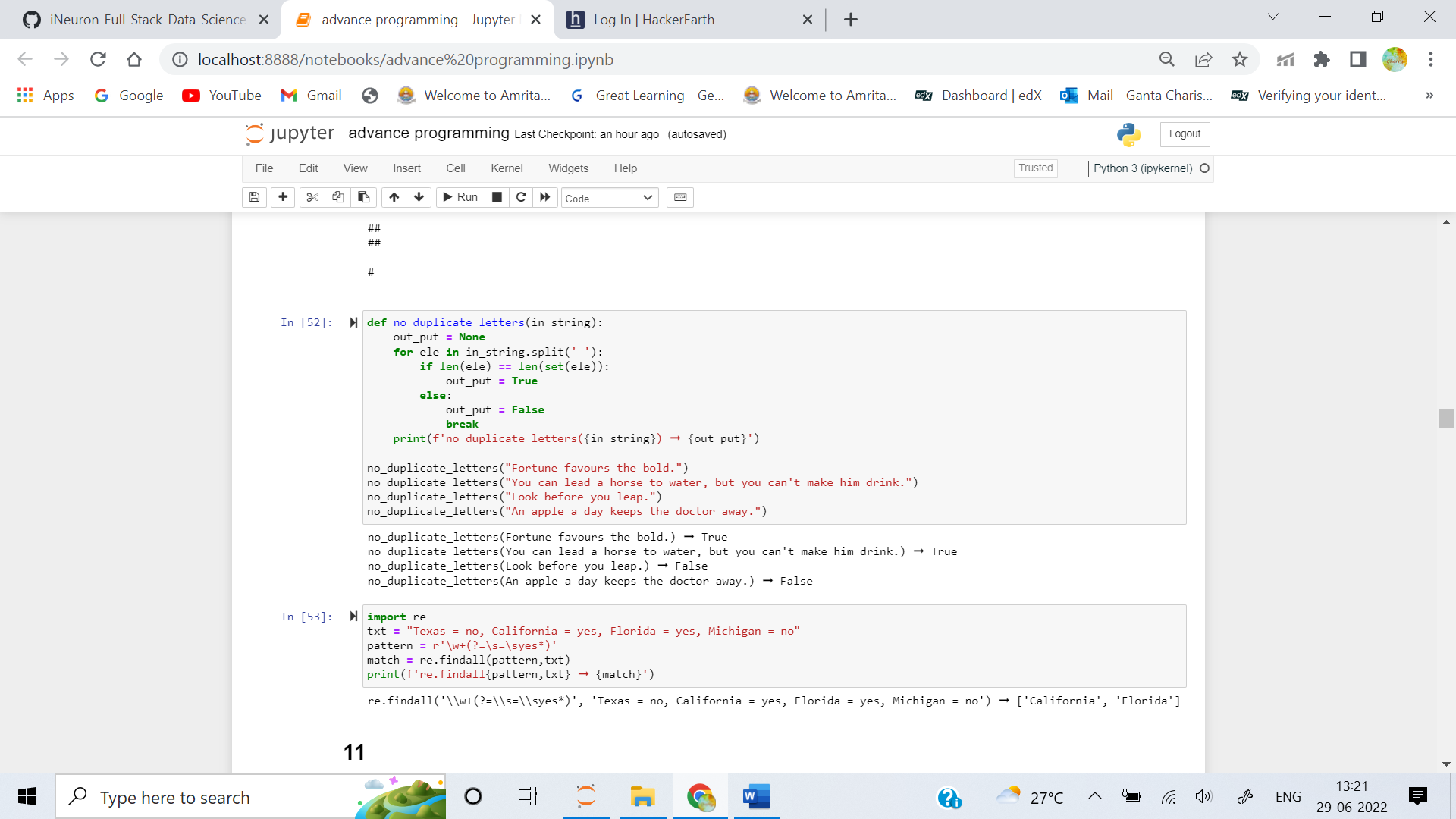
no\_duplicate\_letters("You can lead a horse to water, but you can't make him drink.") ➞ True

no\_duplicate\_letters("Look before you leap.") ➞ False

# Duplicate letters in "Look" and "before".

no\_duplicate\_letters("An apple a day keeps the doctor away.") ➞ False

# Duplicate letters in "apple", "keeps", "doctor", and "away".

**ANS:** 

5. Write a regular expression that will match the states that voted yes to President Trump's impeachment. You must use RegEx positive lookahead.

Example

txt = "Texas = no, California = yes, Florida = yes, Michigan = no"

pattern = "yourregularexpressionhere"

re.findall(pattern, txt) ➞ ["California", "Florida"]

**ANS:** 