Lab_2_Andrade

January 30, 2024

24 January 2024 #

Lab 2 Assignment - CS 4315

Doug Andrade

1. A function that takes the variables a, b, and c as input and returns $a * b + c$, with c having a default value of 0.

[1]:
```
# function that takes in three values, one of which is set with a default of 0
def func(a, b, c = 0):
    # return the result of an arthimetic operation of the 3 (or 2) inputted␣
 ↪values
    return a * b + c
```

2. Write a text file named `laws.txt` that contains the following four lines:

```
Isaac Asimov's Three Laws of Robotics

1. A robot may not injure a human being or, through inaction, allow a human being to come to h

2. A robot must obey orders given it by human beings except where such orders would conflict w

3. A robot must protect its own existence as long as such protection does not conflict with th
```

[2]:
```
# create an object for the 'laws.txt' file to avoid hard coding
txt_name = 'laws.txt'

# Open (create) the 'laws.txt' file by setting the open function to 'write' mode
laws_file = open(txt_name, mode = 'w')

# Write each line of text to the 'laws.txt' file
laws_file.write("Isaac Asimov's Three Laws of Robotics\n")
laws_file.write("1. A robot may not injure a human being or, through inaction,␣
 ↪allow a human being to come to harm.\n")
laws_file.write("2. A robot must obey orders given it by human beings except␣
 ↪where such orders would conflict with the First Law.\n")
laws_file.write("3. A robot must protect its own existence as long as such␣
 ↪protection does not conflict with the First or Second Law.\n")
```

```
# Close the 'laws.txt' file to avoid unintended changes to the file, or losing/
 ↪corrupting the file
laws_file.close()
```

3. Opening `laws.txt` and printing every odd line.

[3]:
```
# Open the previously created 'laws.txt' file by setting the 'open()' function␣
 ↪to 'read' mode
laws_file = open(txt_name, mode = 'r')

# for loop to read (enumerate) each line of the txt, starting at the 0 (even)␣
 ↪index
for line_number, line in enumerate(laws_file, start = 0):
    # Print the line if the enumerated text's line is odd (not divisible by 2)
    if line_number % 2 != 0:
        # .strip() removes the default white space between each line printed
        print(line.strip())

# Close the 'laws.txt' file to avoid unintended changes to the file, or losing/
 ↪corrupting the file
laws_file.close()
```

```
1. A robot may not injure a human being or, through inaction, allow a human
being to come to harm.
3. A robot must protect its own existence as long as such protection does not
conflict with the First or Second Law.
```

4. Write a CSV file named `forwards_and_backwards.csv` that contains the following four rows of data:

1	2	3	4	5
5	4	3	2	1
6	7	8	9	10
10	9	8	7	6

[4]:
```
# Create a list-of-lists object with the specified values
tbl = [
    [1, 2, 3, 4, 5],
    [5, 4, 3, 2, 1],
    [6, 7, 8, 9, 10],
    [10, 9, 8, 7, 6]
]
```

[5]:
```
# import the comma separated value (csv) module to perform operation on csv␣
 ↪files
import csv
```

```
# create an object for the 'forwards_and_backwards.csv' file to avoid hard␣
 ↪coding
csv_file = 'forwards_and_backwards.csv'

# Open (create) the 'forward_and_backwards.csv' file by setting the open()␣
 ↪function to 'write' mode
# newline = '' is added as common practice to override the default control␣
 ↪character logic adds additional space between lines
# when reading csv file in Python enviroment
csv_tbl = open(file = csv_file, mode = 'w', newline = '')

# A writer object to convert an object into a delimited set of strings to the␣
 ↪'forwards_and_backwards.csv' file
tbl_writer = csv.writer(csv_tbl, delimiter = ',')

# Write the tbl list-of-lists object to the 'forwards_and_backwards.csv' file
tbl_writer.writerows(tbl)

# Close the 'forwards_and_backwards.csv' file to avoid unintended changes to␣
 ↪the file, or losing/corrupting the file
csv_tbl.close()
```

5. Open `forwards_and_backwards.csv` and printing the third number in the third row.

[6]:
```
# Open the 'forward_and_backwards.csv' file by setting the open() function to␣
 ↪'read' mode
with open(file = csv_file, mode = 'r', newline = '') as csv_tbl:

    # Reader object to iterate through each row of the 'forwards_and_backwards.
 ↪csv' file
    reader = csv.reader(csv_tbl)

    # Convert the reader object to a list for visualization
    my_csv = list(reader)

    # Iterate through each row of the 'forwards_and_backwards.csv' file and␣
 ↪print it on a new line
    for line in my_csv:
        print(line)

# Close the 'forwards_and_backwards.csv' file to avoid unintended changes to␣
 ↪the file, or losing/corrupting the file
csv_tbl.close()
```

```
['1', '2', '3', '4', '5']
['5', '4', '3', '2', '1']
```

```
['6', '7', '8', '9', '10']
['10', '9', '8', '7', '6']
```

[8]:
```
# Apply string formatting to print the specific index value from the my_csv␣
↪list-of-lists object
print('The 3rd element in the 3rd row is "%s".' % my_csv[2][2])
```

```
The 3rd element in the 3rd row is "8".
```