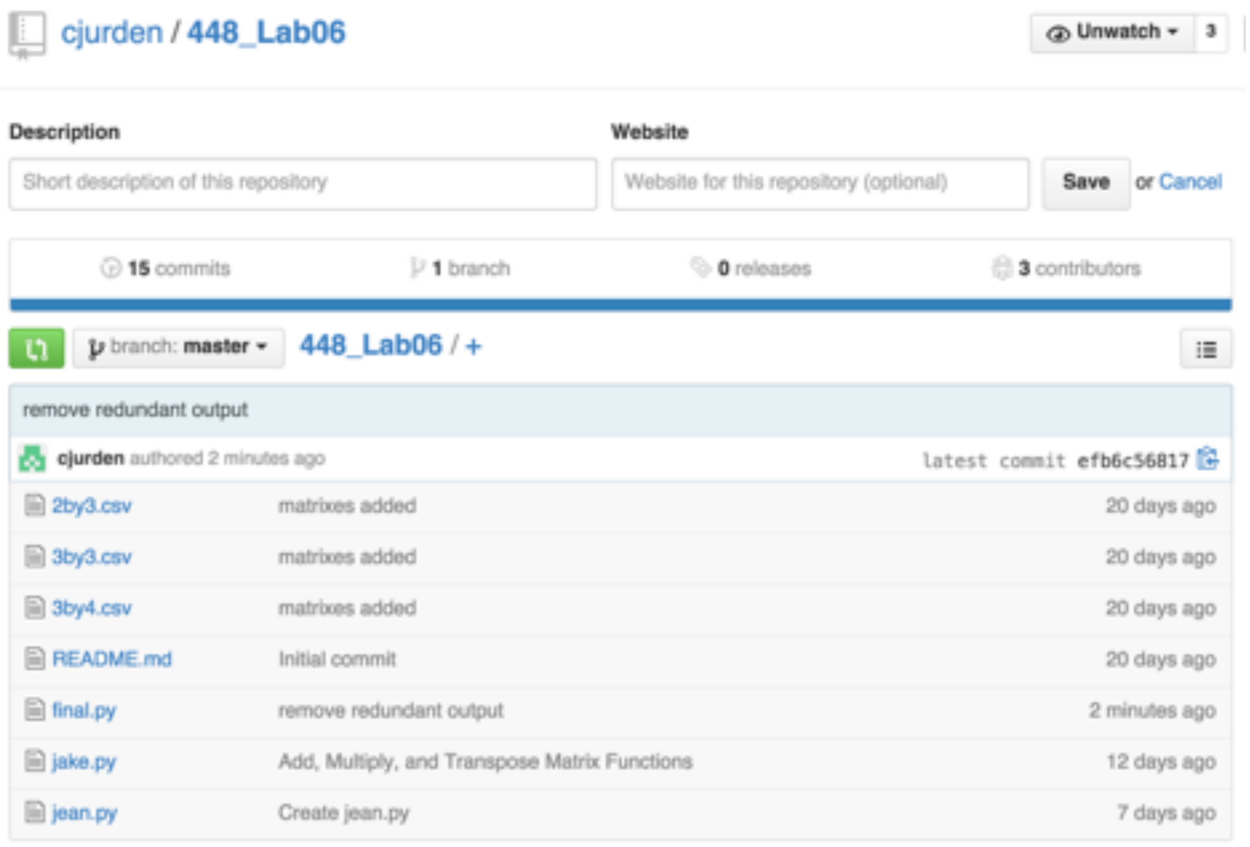
Cole Jurden, Brandon Teh, Jeanette Rusli, Jake Suddock

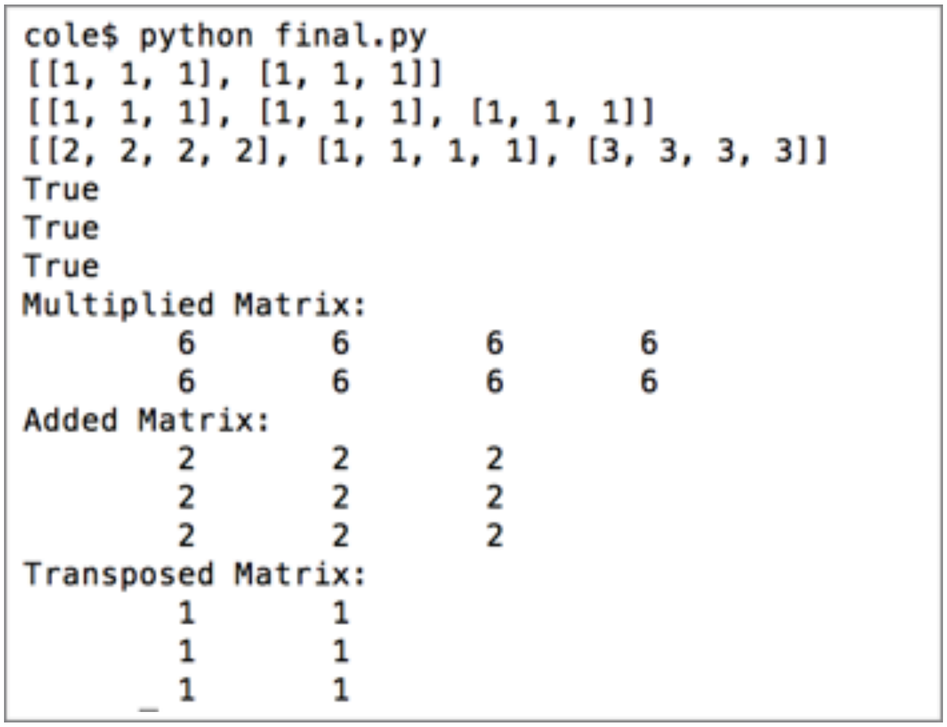
448 Lab 06
4-01-15



remove redundant output

cjurden authored 2 minutes ago — latest commit efb6c56817

2by3.csv	matrixes added	20 days ago
3by3.csv	matrixes added	20 days ago
3by4.csv	matrixes added	20 days ago
README.md	Initial commit	20 days ago
final.py	remove redundant output	2 minutes ago
jake.py	Add, Multiply, and Transpose Matrix Functions	12 days ago
jean.py	Create jean.py	7 days ago



```
cole$ python final.py
[[1, 1, 1], [1, 1, 1]]
[[1, 1, 1], [1, 1, 1], [1, 1, 1]]
[[2, 2, 2, 2], [1, 1, 1, 1], [3, 3, 3, 3]]
True
True
True
Multiplied Matrix:
        6       6       6       6
        6       6       6       6
Added Matrix:
        2       2       2
        2       2       2
        2       2       2
Transposed Matrix:
        1       1
        1       1
      _ 1       1
```

```
# Declare the 3 matrices.
m1,m2,m3 = [],[],[]

# Reading all 3 CSV files.
import csv
with open('2by3.csv', 'r') as f1:
    reader = csv.reader(f1)
    m1 = [[int(e) for e in r]for r in reader]
    print(m1)

with open('3by3.csv', 'r') as f2:
    reader = csv.reader(f2)
    m2 = [[int(e) for e in r]for r in reader]
    print(m2)

with open('3by4.csv', 'r') as f3:
    reader = csv.reader(f3)
    m3 = [[int(e) for e in r]for r in reader]
    print(m3)



# Function to check whether the matrix is valid.
def checkMat(m):

        l = []

  # Get the lengths of all rows.
        for x in range(0, len(m)):
                l.append(len(m[x]))

  # Compare all lengths.
        for x in range(1, len(l)):
                if l[x-1] == l[x]:
                        mat = True  # Valid matrix.
                else:
                        mat = False # Invalid matrix.

        return mat

#print result of matrix check
print(checkMat(m1))
print(checkMat(m2))
print(checkMat(m3))

#############################################################################################
#Transposition

def TransposeMatrix(matrix):
        transposed = [[0 for x in range(len(matrix))] for x in range(len(matrix[0]))]
        for i in range(0, len(matrix)):
                for j in range(0, len(matrix[i])):
                        transposed[j][i] = matrix[i][j]
        print 'Transposed Matrix:'
        printMatrix(transposed)

#############################################################################################
#Multiplication

def MultiplyMatrices(matrix1,matrix2):
        if len(matrix1[0]) == len(matrix2):
                multiplied = [[0 for x in range(len(matrix2[0]))] for x in range(len(matrix1))]
                for i in range(0, len(multiplied)):
                        for j in range(0, len(multiplied[i])):
                                temp = 0
                                for k in range(0, len(matrix1[0])):
                                        temp = temp + (matrix1[i][k] * matrix2[k][j])
                                multiplied[i][j] = temp
                print 'Multiplied Matrix:'
                printMatrix(multiplied)
        else:
                print 'Cannot multiply matrices'

#############################################################################################
#Addition

def AddMatrices(matrix1,matrix2):
        if len(matrix1)==len(matrix2) and len(matrix1[0])==len(matrix2[0]):
                added = [[0 for x in range(len(matrix1[0]))] for x in range(len(matrix1))]
                for i in range(0, len(added)):
                        for j in range(0, len(added[0])):
                                added[i][j] = matrix1[i][j] + matrix2[i][j]
                print 'Added Matrix:'
                printMatrix(added)
        else:
                print 'Cannot add matrices'


###############################################################################################
#Print Matrix

def printMatrix(matrix):
        for i in range(0, len(matrix)):
                string = ''
                for j in range(0, len(matrix[0])):
                        string = string + '\t' + str(matrix[i][j])
                print string

###############################################################################################
#program


if ((checkMat(m1) and checkMat(m3))==True):
        MultiplyMatrices(m1, m3)
else:
        print 'Invalid matrices'

if ((checkMat(m1) and checkMat(m2))==True):
        AddMatrices(m2, m2)
else:
        print 'Invalid matrices'

if (checkMat(m1)==True):
        TransposeMatrix(m1)
else:
        print 'Invalid matrices'
```