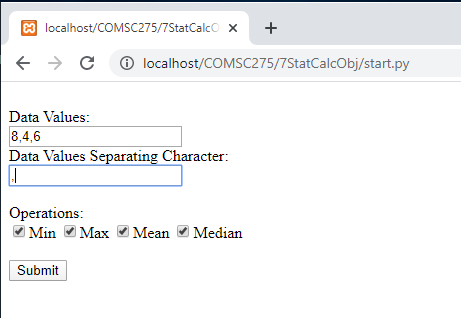
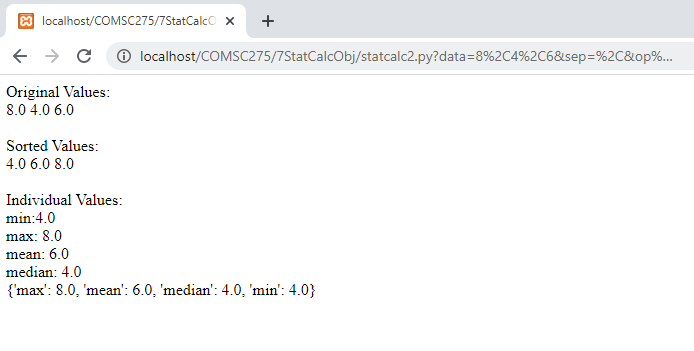
Python

Python Input:



Python Output:



headTail.py

#!/Python27/python  
  
def headHtml ():  
  
 print ("Content\_type: 'html/text' \n\n")  
  
 print ('''  
  
 <html>  
  
 <head>  
  
 </head>  
  
 <body>  
  
 ''')  
  
  
def tailHtml ():  
  
 print('''  
  
 </body>  
  
 </html>  
  
 ''')

Forms.py

#!/Python27/python  
  
import cgi  
  
def dispForm():  
 print('''  
<form action='statcalc2.py' method='get'>  
<br>Data Values:  
<br><input type='text' name='data' />  
<br>Data Values Separating Character:  
<br><input type="text" name='sep' />  
<br><br>Operations:  
<br><input type='checkbox' name='op[]' value='min' checked />Min  
<input type='checkbox' name='op[]' value='max' checked />Max  
<input type='checkbox' name='op[]' value='mean' checked />Mean  
<input type='checkbox' name='op[]' value='median' checked />Median  
<br><br><input type='submit' name='submit' value='Submit' /><br><br>  
</form>  
''')  
  
  
def getInput():  
 f = cgi.FieldStorage()  
 data = f.getvalue('data')  
 sep = f.getvalue('sep')  
 opList = f.getvalue('op[]')  
 dataList = data.split(sep)  
  
 i = 0  
 for item in dataList:  
 dataList[i] = float(item)  
 i = i + 1  
 return dataList, opList

Start.py

#! /Python27/python  
  
import cgi  
  
from headTail import headHtml  
  
from headTail import tailHtml  
  
from forms import dispForm  
  
  
def main():  
 headHtml()  
 dispForm()  
  
 tailHtml()  
 return  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
  
 try:  
  
 main()  
  
 except:  
  
 cgi.print\_exception()

Statcalc.py

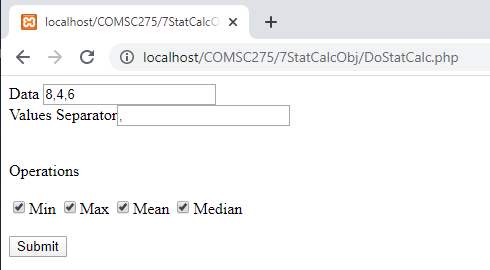
class StatCalc:  
 def \_\_init\_\_(self,dataList):  
 self.dataList=dataList  
 self.sdataList=[]+ self.dataList  
 self.sdataList.sort()  
  
 def showResults(self):  
 print("Original Values: <br>")  
 for item in self.dataList:  
 print(item)  
 print(" ")  
  
 print("<br><br>")  
 print("Sorted Values: <br>")  
 for item in self.sdataList:  
 print(item)  
 print("<br>")  
  
 def findMin(self):  
 return self.sdataList[0]  
  
 def findMax(self):  
 return self.sdataList \  
 [len(self.sdataList)-1]  
  
 def findMean(self):  
 mean = sum(self.sdataList) / \  
 len(self.sdataList)  
 return mean  
  
 def findMedian(self):  
 mid = len(self.sdataList) / 2  
 if len(self.sdataList) % 2 == 0: # if even set  
 result = (self.sdataList[int(mid)] + self.sdataList[int(mid - 1)]) / 2  
 else:  
 result = self.sdataList[int(mid - 0.5)]  
 return result  
  
 def calcValues(self,opList):  
 dictOpRes = {}  
 for op in opList:  
 if op == "min":  
 dictOpRes[op] = self.findMin()  
 if op == "max":  
 dictOpRes[op] = self.findMax()  
 if op == "mean":  
 dictOpRes[op] = self.findMean()  
 if op == "median":  
 dictOpRes[op] = self.findMedian()  
  
 return dictOpRes

statcalc2.py

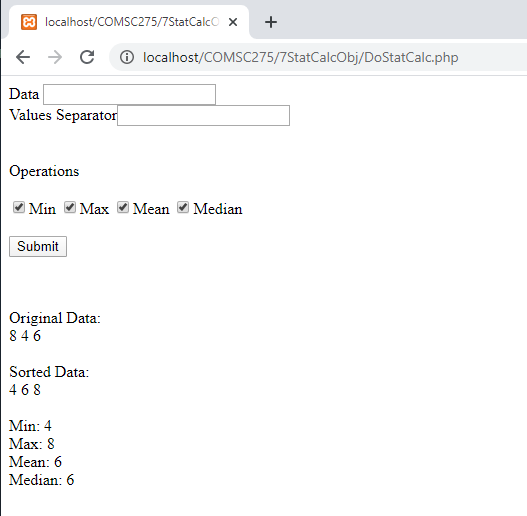
#!/Python27/python  
  
import cgi  
  
from headTail import headHtml  
from headTail import tailHtml  
from forms import getInput  
  
import statcalc  
  
  
def main():  
 headHtml()  
  
 dataList, opList = getInput()  
  
 # create object  
  
 statCalc = statcalc.StatCalc(dataList)  
  
 statCalc.showResults()  
  
 print('<br>Individual Values:<br>')  
 for op in opList:  
  
 if op == 'min':  
  
 min = statCalc.findMin()  
  
 print('min:' + str(min) + '<br>')  
  
 elif op == 'max':  
  
 max = statCalc.findMax()  
  
 print('max: ' + str(max) + '<br>')  
  
 elif op == 'mean':  
  
 mean = statCalc.findMean()  
  
 print('mean: ' + str(mean) + '<br>')  
  
 elif op == 'median':  
 median = statCalc.findMedian()  
  
 print('median: ' + str(median) + '<br>')  
  
 dictOpRes = statCalc.calcValues(opList)  
  
 print(dictOpRes)  
 tailHtml()  
 return  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
  
 try:  
 main()  
  
 except:  
 cgi.print\_exception()

PHP

PHP Input



PHP Output



StatCalc.php

<?php

class StatCalc {

private $data;

private $sdata;

public function \_\_construct ($d){

$this->data = $d;

$this->sdata = $this->data;

sort ($this->sdata);

}

public function findMin(){

return $this->sdata[0];

}

public function findMax (){

return $this->sdata[count($this->sdata)-1];

}

public function findMean (){

$sum = array\_sum($this->sdata);

$mean = $sum / count ($this->sdata);

return $mean;

}

public function findMedian (){

//Determine if the length is odd or even.

if ( (count($this->sdata) % 2) != 0 ){

$index = count ($this->sdata) / 2;

$median = $this->sdata [$index];

}

else{

$indexHi = count ($this->sdata) / 2;

$indexLo = $indexHi - 1;

$median = ($this->sdata[$indexLo] +

$this->sdata [$indexHi]) / 2;

}

return $median;

}

public function getOrigData(){

return $this->data;

}

public function getSdataData(){

return $this->sdata;

}

}

?>

DoStatCalc.php

<form action='DoStatCalc.php' method='post'>

Data <input type='text' name='data' /><br>

Values Separator<input type="text" name='separator'><br><br><br>

Operations<br><br>

<input type='checkbox' name='op[]' value='min' checked />Min

<input type='checkbox' name='op[]' value='max' checked />Max

<input type='checkbox' name='op[]' value='mean' checked />Mean

<input type='checkbox' name='op[]' value='median' checked />Median<br><br>

<input type='submit' name='submit' value='Submit' /><br><br>

</form>

<?php

require "StatCalc.php";

//if it is the first invocation of the script, exit the script.

if (!isset($\_REQUEST['submit']))

exit();

//It is not the first invocation of the script. So process input.

//Validate input.

$str = $\_REQUEST ['data'];

$separator = $\_REQUEST ['separator'];

$op = $\_REQUEST ['op'];

if (empty($str) || empty($separator)){

print "Enter Data and Values Separator";

exit (0);

}

if (empty($op)){

print ("Check one or more operations");

exit(0);

}

//Input validated.

//Convert data to an array

$ar = explode($separator, $str);

//create a statCalc object

$statCalc = new StatCalc ($ar);

//Utilize the above object to do the rest of the assignment

print('<br>Original Data:<br>');

$list = $statCalc->getOrigData();

foreach($list as $value){

echo $value . " ";

}

print('<br><br>');

print('Sorted Data:<br>');

$listS = $statCalc->getSdataData();

foreach($listS as $value){

echo $value . " ";

}

print('<br><br>');

if(in\_array("min", $op)){

print('Min: ');

print($statCalc->findMin());

print('<br>');

}

if(in\_array("max", $op)){

print('Max: ');

print($statCalc->findMax());

print('<br>');

}

if(in\_array("mean", $op)){

print('Mean: ');

print($statCalc->findMean());

print('<br>');

}

if(in\_array("median", $op)){

print('Median: ');

print($statCalc->findMedian());

}

?>