

# CS 1301 Homework – Robot Web Writer

Due: Friday October 12th, before 11:55pm

## **This is a pair programming assignment!**

You are expected to work with the person you have been paired with in class, and you are **both** responsible for submitting the exact same code to T-Square. Follow good pair-programming practices by working together at a single computer and switching the driver/navigator role frequently.

Your pair may collaborate with other students in this class. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. You should not exchange code or write code for other pairs. Collaboration at a reasonable level will not result in substantially similar code.

For pair programming assignments, you and your partner should turn in identical assignments.

**Files to submit:**      **light.py**

For help:

-TA Helpdesk – Schedule posted on class website

-Email TAs

Notes:

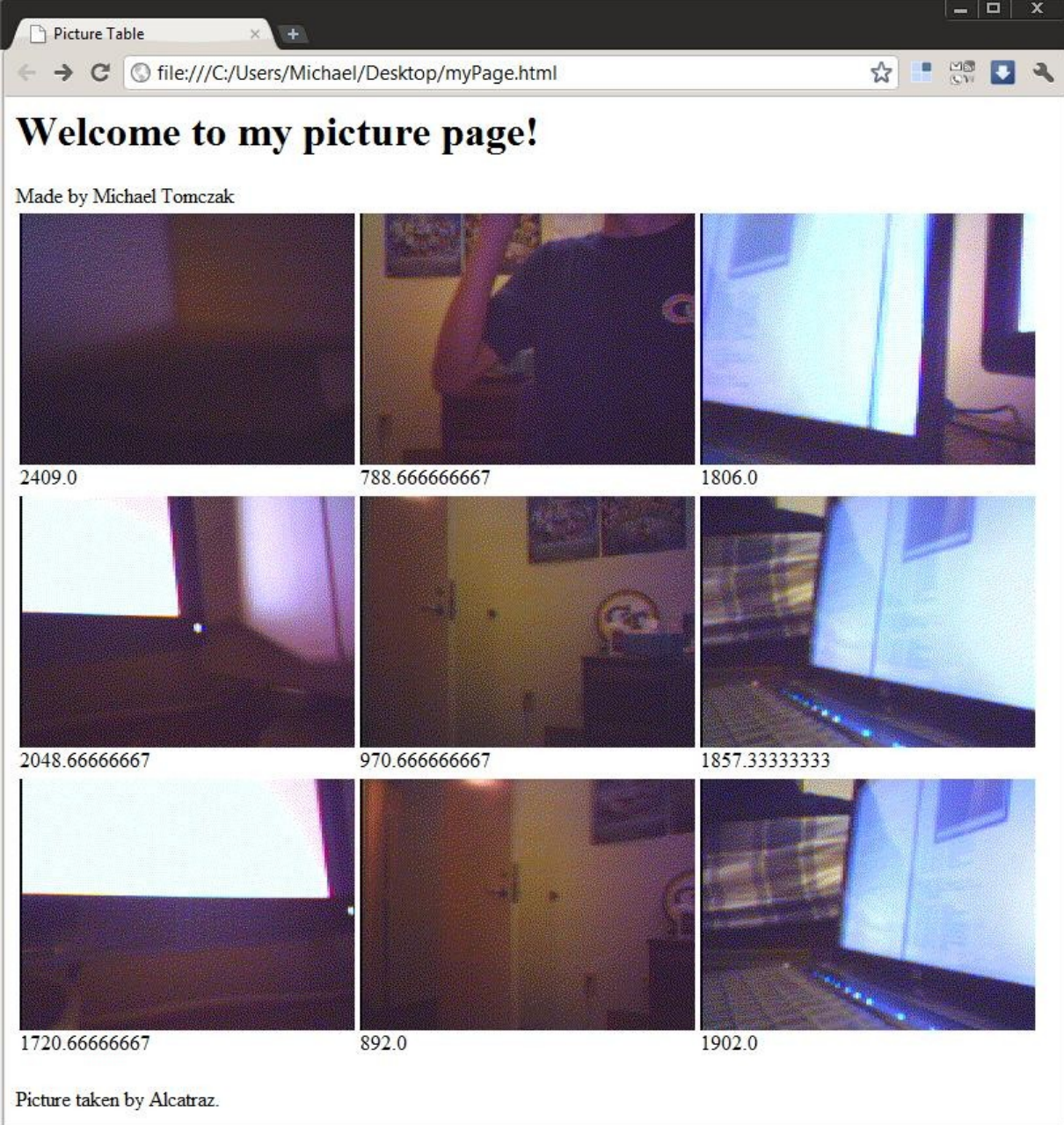
- **Don't forget to include the required comments and collaboration statement (as outlined on the course syllabus)**
- **Do not wait until the last minute** to do this assignment in case you run into problems.
- If you find a significant error in the homework assignment, please let a TA know immediately.

## Part 1 – Don't Follow the Lights...







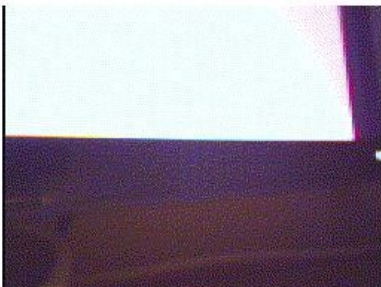


For this assignment, you will need to know how your robot's light sensors work. They are found in the three holes built into the back of the Scribbler, and are accessed with the `getLight()`

function. This function will return to you a list of three values, ranging from zero to upwards of 30,000, with average values ranging from 500 to 4000. The more light hitting the sensors, the lower the values will be. If you give the getLight() function a single parameter, (such as 0, 1, or 2 or the strings “left”, “center”, or “right”.) it will return a single value from a single sensor.

Your job will be to take nine different pictures using your Scribbler, with the robot moving in between each picture (such as turning or scooting forward), and save the pictures as pic0.gif, pic1.gif...pic8.gif. For each picture you should also record the corresponding average of the 3 light values from your Scribbler’s sensors. (avg value = (Left + Center + Right) / 3.0)



The screenshot shows a web browser window titled "Picture Table" with the address bar displaying "file:///C:/Users/Michael/Desktop/myPage.html". The page content includes a heading "Welcome to my picture page!" and a sub-heading "Made by Michael Tomczak". Below this is a 3x3 grid of images, each with a numerical value underneath it. The values are: 2409.0, 788.666666667, 1806.0, 2048.666666667, 970.666666667, 1857.333333333, 1720.666666667, 892.0, and 1902.0. At the bottom of the page, it says "Picture taken by Alcatraz."

Image	Average Value
	2409.0
	788.666666667
	1806.0
	2048.666666667
	970.666666667
	1857.333333333
	1720.666666667
	892.0
	1902.0

Picture taken by Alcatraz.

## Part 2 – Quick, to the Internet!

For Part 2, your program will create an HTML file using Python file operations named “myPage.html.” On the page, have a descriptive title, a header welcoming the user to the page, your name(s) in the body of the document, a table showing your nine pictures and their corresponding light values below them in a 3x3 grid, and a photo credit, giving credit to the robot who took the picture, for example “photos taken by [name],” where [name] is the name of the robot. Your code should correctly generate the name of whatever robot is connected to the computer, and not just print your own robot's name! Be sure to use proper html tags in your file. The above picture is an example page to give you an idea of what yours should look like (though yours doesn't have to look exactly the same as the example, it must have all the same pieces of data.)

Your webpage must consist of valid HTML as tested by <http://validator.w3.org/> (You may use "detect automatically" for the document type.)

Note that you do not have to upload the page to a web server, just generate it on the local computer.

### Grading:

Robot moves between taking each picture	5
Robot collects light values with each picture	5
Program saves the 9 pictures correctly	10
Program averages 3 light values correctly	10
Program opens correct file for writing	5
Program correctly closes the file.	5
Program writes required elements of the HTML page: (60 total)	
– Descriptive Title	10
– Welcome header text	10
– Name of student(s) in body	10
– Table showing the 9 photos and average light values	10
– Photo credit giving name of the robot. ( using getName())	10
– HTML Validates with no errors	10