On Tuseday, 6/2/15, I was looking on Stack Overflow to find ways to execute Python from the browser, not specifically JavaScript. I found a post that included PHP: <http://stackoverflow.com/questions/19735250/running-a-python-script-from-php>.

I looked up examples of PHP and stumbled across popen, which I have used in Python before to get a command line output. I ultimately found this PHP documentation, which was ultimately what was helpful: <http://php.net/manual/en/function.popen.php>.

Finally, I needed to find a way to input arguments to the Python script via the command line. This I found: <http://www.tutorialspoint.com/python/python_command_line_arguments.htm>.

The final PHP code (saved as shell.php):

<html>

<head>

<title>PHP Test</title>

</head>

<body>

<?php

$handle = popen('python test.py hello lorem world ipsum', 'r');

$read = fread($handle, 2096);

echo $read;

pclose($handle);

?>

<div style="height:400px; width:700px; background-color:#ffff00">

Extra Content

</div>

</body>

</html>

The final Python code (saved as test.py):

import sys

variables = sys.argv

for i in variables:

print i

This output exactly what was expected: test.py hello lorem world ipsum. PHP could not be run from my laptop, but it was run from devtano.com after I uploaded the needed files.

I next experimented with adding time.sleep functions to the Python. It did not do what I was expecting, which was to gradually display the outputs. Instead, the page “loaded” until the program finished executing. The idea I had to work around this would be to write the console outputs to a file, and have an Ajax request being constantly fired and constantly updating the screen that way; regardless of extra time to execute file writes, it is ultimately important that the user know the progress rather than get impatient. The final output will be printed to the screen.

On 6/3/2015, I made an output stream to an HTML file from the Python, and integrated it into the script. This included an HTML template for the output.

On 6/4/2015, I started integrating existing documentation into the input-help.html. I plan to make a single documentation page. I also began work on the HTML shell for the application. I asked a question on Stack Overflow about changing the titles on markers. Built-in to the shell is a geocoder that updates a marker position on the map for start and end points, as well as the map orientation. I also have geocoding exception catching. I still need to work on the layout of the page. I plan to have the input box docked on the right side, along with some content beneath that, and have the map be the entire page.