

# Where In The World

Jay Dance

5375110

## How It Works

After specifying your input file, run WhereTheHeck.java. This will produce a formattedInput file that is then converted into a .geojson file.

```
input.txt x ... formattedInput.txt x
input.txt
1 Coordinates of varying decimal places
2 -36.8, 174.7 Auckland
3 -41.29, 174.78 Wellington
4 -43.530, 172.628 Christchurch
5 -37.7876, 175.2789 Hamilton
6 -45.87876, 170.50285 Dunedin
7 -37.687623, 176.165983 Tauranga
8 Coordinates without comma
9 40.730610 -73.935242 New York City
10 34.052235 -118.243683 Los Angeles
11 48.856614 2.352222 Paris
12 51.507222 -0.127647 London
13 Standard form with N/S E/W. And in wrong order
14 40 N 40 E
15 5 S 50 W
16 S 10 W 80
17 N 80 E 10
18 DMS form, with and without decimal places or standard markers
19 47° 36' 21.6" N 122° 19' 55.6" W Seattle
20 29° 57' 0" N, 31° 7' 0" E
21 43 35 23.45 S 79 23 10.78 W
22 43 35 23.45 N 79 23 10.78 E
23 DM form
24 29° 57' 0" N, 31° 7' 0" E
25 51° 30.183' N, 0° 7.674' W
26
27 Ha, tricked you!
28
29 Other input
30 20 40 S 40 20 W
31
32 Error input
33 12 41 29 N 14 35 W
34 27 19 N 14 35 7 E
35 21 35 78
36 54612137845231

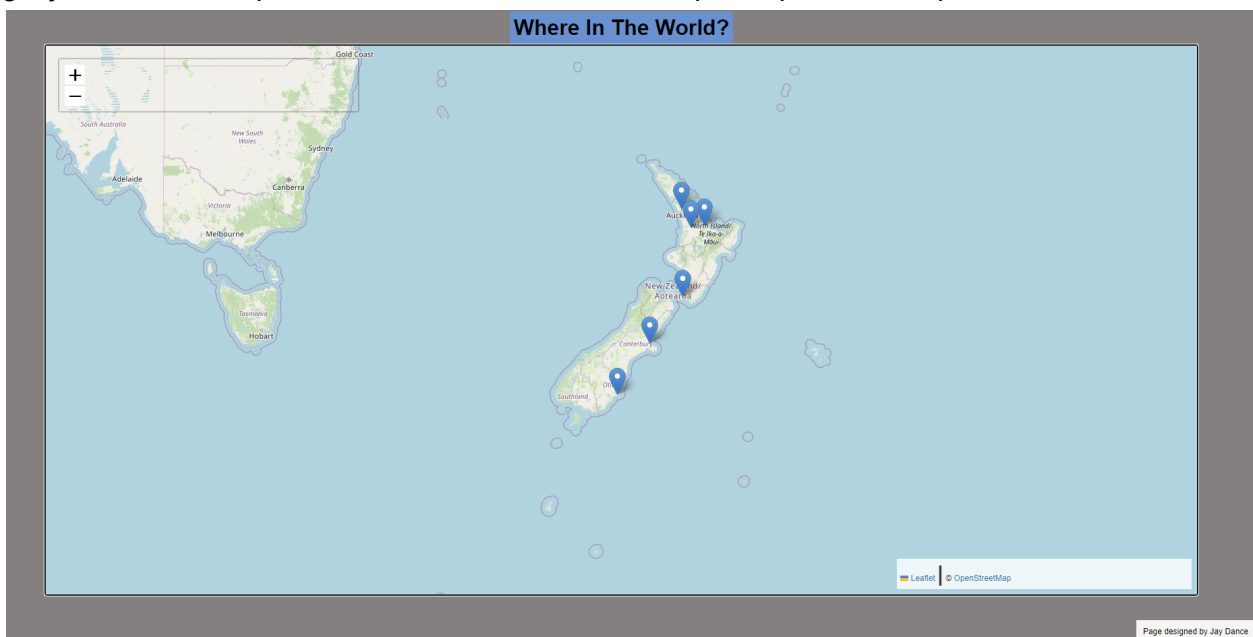
formattedInput.txt
1 -36.8 174.7 Auckland
2 -41.29 174.78 Wellington
3 -43.53 172.628 Christchurch
4 -37.7876 175.2789 Hamilton
5 -45.87876 170.50285 Dunedin
6 -37.687623 176.165983 Tauranga
7 40.73061 -73.935242 New York City
8 34.052235 -118.243683 Los Angeles
9 48.856614 2.352222 Paris
10 51.507222 -0.127647 London
11 40.0 40.0
12 -5.0 -50.0
13 -10.0 -80.0
14 80.0 10.0
15 47.606000 -122.332111 Seattle
16 29.950000 31.116667
17 -43.589847 -79.386328
18 43.589847 79.386328
19 29.950000 31.116667
20 51.503050 -0.127900
21 -20.666667 -40.333333
```

```

() output.geojson ×
() output.geojson > ...
1  { "type": "FeatureCollection", "features": [
2    { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ 174.700000, -36.800000 ] }, "properties": { "name": "Auckland" } },
3    { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ 174.780000, -41.290000 ] }, "properties": { "name": "Wellington" } },
4    { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ 172.628000, -43.530000 ] }, "properties": { "name": "Christchurch" } },
5    { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ 175.278900, -37.787600 ] }, "properties": { "name": "Hamilton" } },
6    { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ 170.502850, -45.878760 ] }, "properties": { "name": "Dunedin" } },
7    { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ 176.165983, -37.687623 ] }, "properties": { "name": "Tauranga" } },
8    { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ -73.935242, 40.730610 ] }, "properties": { "name": "New York City" } },
9    { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ -118.243683, 34.052235 ] }, "properties": { "name": "Los Angeles" } },
10   { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ 2.352222, 48.856614 ] }, "properties": { "name": "Paris" } },
11   { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ -0.127647, 51.507222 ] }, "properties": { "name": "London" } },
12   { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ 40.000000, 40.000000 ] }, "properties": { "name": "" } },
13   { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ -50.000000, -5.000000 ] }, "properties": { "name": "" } },
14   { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ -80.000000, -10.000000 ] }, "properties": { "name": "" } },
15   { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ 10.000000, 80.000000 ] }, "properties": { "name": "" } },
16   { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ -122.332111, 47.606000 ] }, "properties": { "name": "Seattle" } },
17   { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ 31.116667, 29.950000 ] }, "properties": { "name": "" } },
18   { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ -79.386328, -43.589847 ] }, "properties": { "name": "" } },
19   { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ 79.386328, 43.589847 ] }, "properties": { "name": "" } },
20   { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ 31.116667, 29.950000 ] }, "properties": { "name": "" } },
21   { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ -0.127900, 51.503050 ] }, "properties": { "name": "" } },
22   { "type": "Feature", "geometry": { "type": "Point", "coordinates": [ -40.333333, -20.666667 ] }, "properties": { "name": "" } } ] ]

```

Once the geojson file is created we can run our index.html file, which will read details from the geojson file and output those details onto a leaflet map with pins at the specified coordinates.



## Approach

I began by looking at how to display coordinates and after some research into my options I decided on using Leaflet. I constructed the .html and .css files for a very basic webpage that displayed a leaflet map and made sure everything with presenting the data was running smoothly. From there I began writing my .java file that would filter through many different forms of input and then output a .geojson file. I quickly realized that due to the varying forms of input that I would need multiple functions to handle the different forms, as I could not capture them all in a single regex expression. For simplicity I decided to standardize the input into Standard form coordinates before turning the formattedInput into a .geojson file.