



Introduction to Mapbox

Dr Michail Pavlis
Department of Geography and Planning



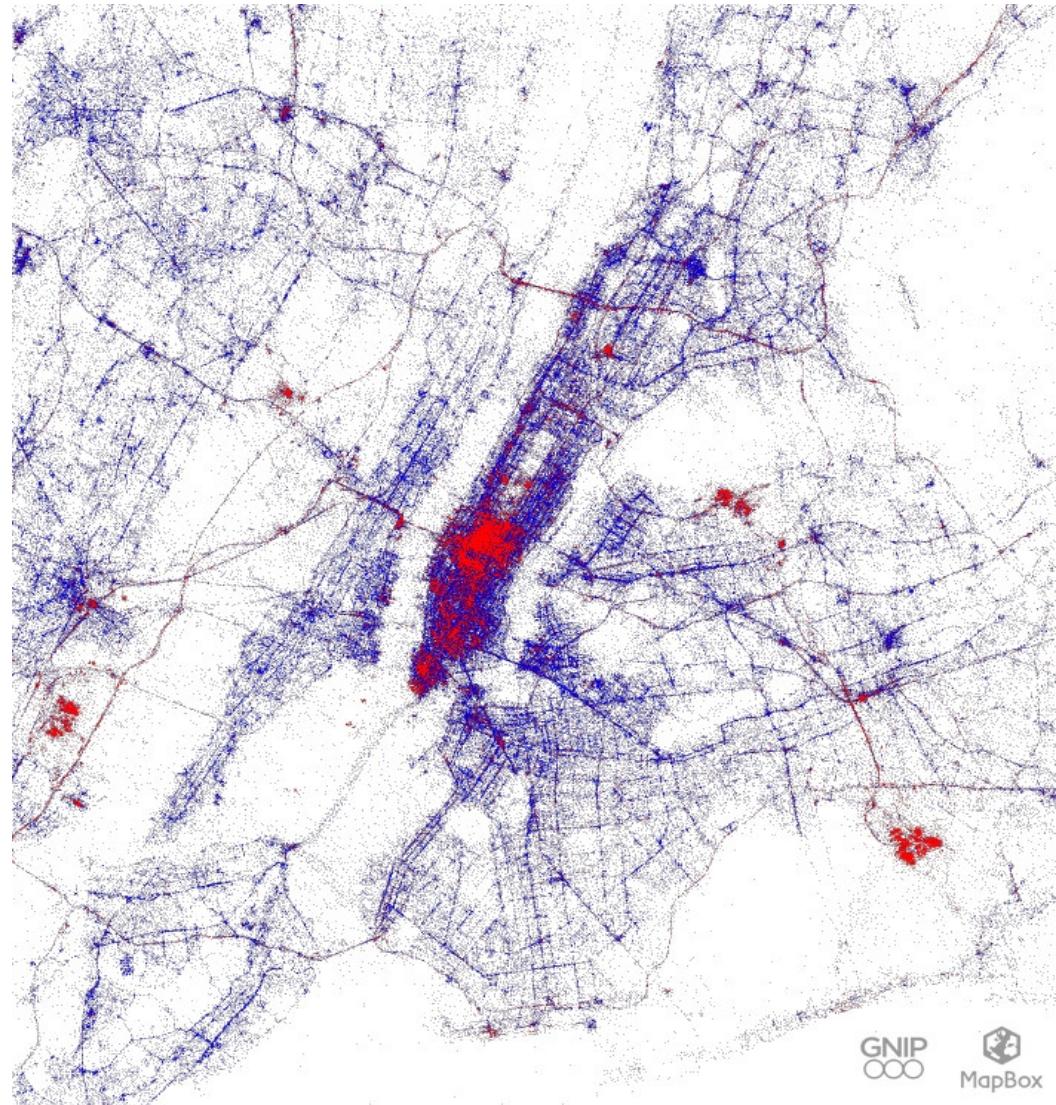
UNIVERSITY OF
LIVERPOOL

Structure

- Introduction to Mapbox Studio.
- Introduction to CartoCSS
 - Selectors
 - Properties
 - Variables
 - Attributes
- Style by attributes.
- Style by zoom level.
- Style labels.
- Style using custom selectors.

Mapbox

- Open source mapping platform.
- Can be used to convert your vector data to vector tiles.
- Advanced visualization using CartoCSS.
- Desktop and web-based interface (Mapbox Studio).



CartoCSS

- CartoCSS applies a style as:

```
#road {  
    property: value;  
}
```

- #road is a selector ('road' is a layer ID).
 - Styles are specified within curly brackets { }.
- The changes will only affect the properties of the selector.
 - Properties are defined per data type.
 - The data type of the #road layer is line.
 - line-color is a common property for all layers of data type line.

- Filters can be used to apply different styles to specific layer attributes.
 - Filters are declared in square brackets []
#road[class='motorway']
 - 'motorway' is a value of the attribute 'class' of the layer 'road'.
- Variables can be used to store information and start with the @ sign.
- A style will override any previously defined style that applies to the same selector.
- Define custom selectors using the double colon syntax '::', e.g.:
 - #road::new_style {
 property: value;
}

Mapbox Vector Tiles

- Widely supported
e.g. Mapbox Studio,
Leaflet and kosmtik
tile server.
- 3 Tilesets available.
 - Streets powered by OSM.
 - Terrain (elevation, hill shades, topography).
 - Satellite (remotely sensed data).

A screenshot of the Mapbox Tilesets page. At the top, there's a search bar and buttons for 'Sort by Name', 'Modified', 'Size', and a 'New tileset' button. Below this, there are two sections: '3 Mapbox tilesets' and '3 tilesets'.

- Mapbox tilesets:**
 - Mapbox Streets v7 (Default tileset)
 - Mapbox Terrain V2 (Default tileset)
 - Mapbox Satellite (Default tileset)
- tilesets:**
 - Untitled (2MB, Modified on Dec 10, 2015)
 - liverpool_wifi.shp (7KB, Modified on Dec 8, 2015)
 - liverpool_wifi.shp (7KB, Modified on Dec 8, 2015)

What is a tileset?

A tileset is a collection of raster or vector data broken up into a uniform grid of square tiles at 22 preset zoom levels. [Read more](#)

How to use tilesets

Upload: Click [New tileset](#) to upload your data. Mapbox renders [vector tiles](#) from your data so you can create styles from it.

Visualize: Once your vector tiles are ready, you can add it to a new or existing style.

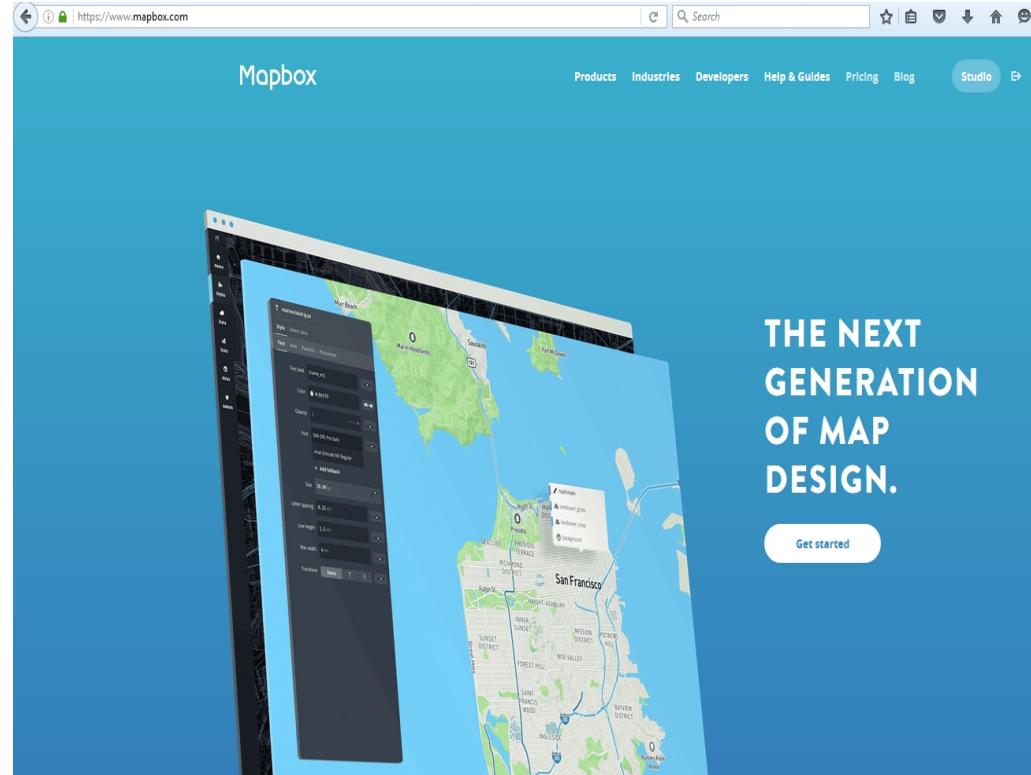
Default tilesets

Styling or querying default tilesets? Learn what data is available and how to reference it.

[View documentation →](#)

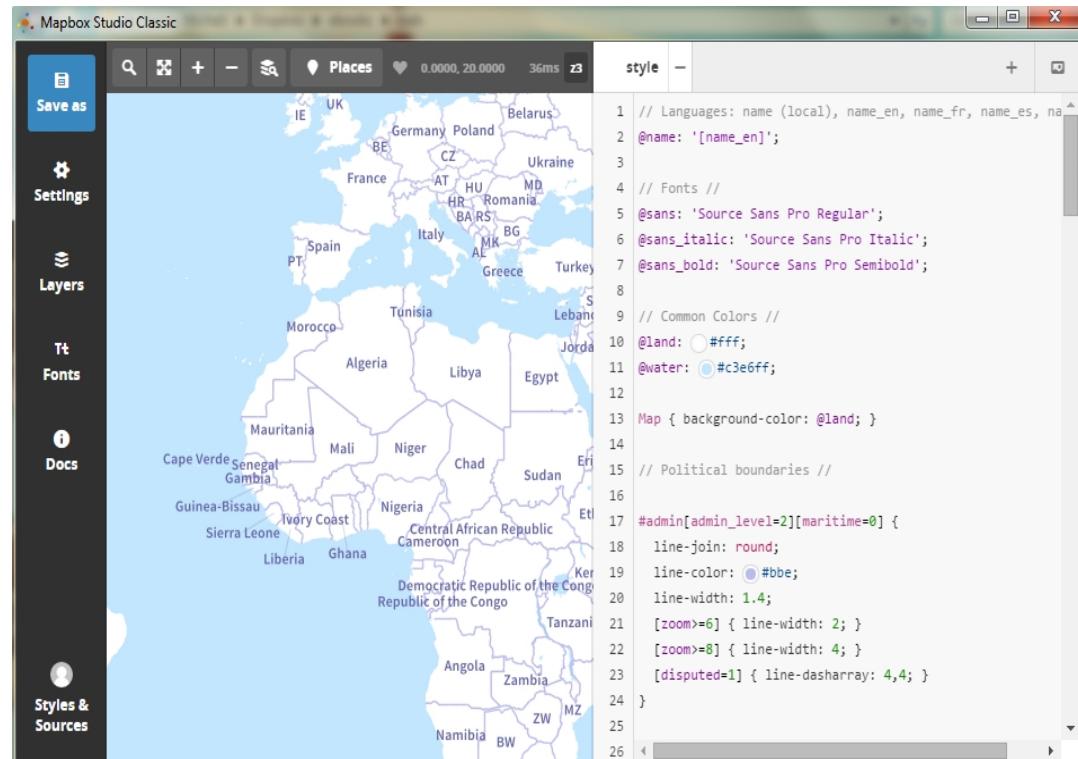
Sign Up

- Create an account:
<https://www.mapbox.com/>
- Download and install
the Mapbox Studio
desktop application.
- Run Mapbox Studio
and select Blank
Source.



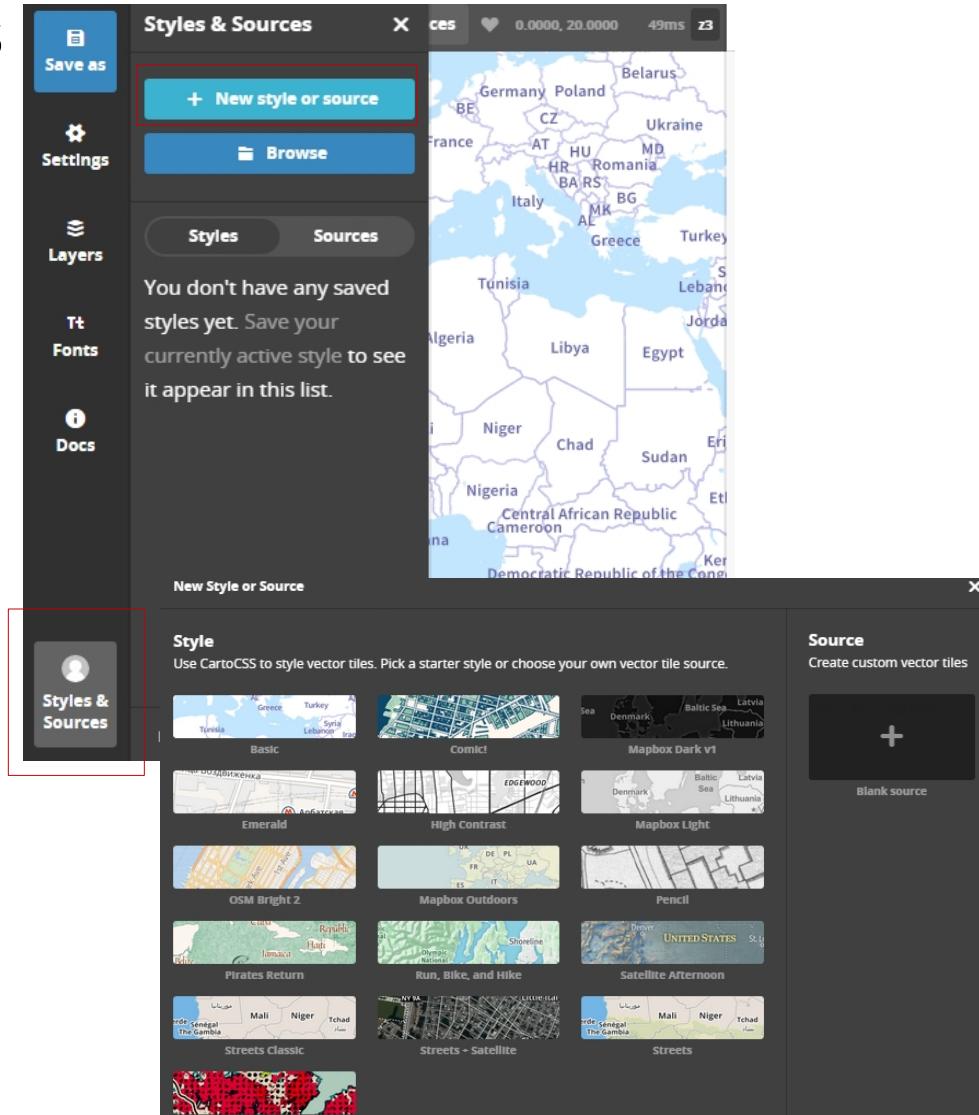
Mapbox Studio

- Desktop software.
- Uses vector tiles.
- 300+ professional fonts.
- Unlike the web-based application it can be used to edit CartoCSS.



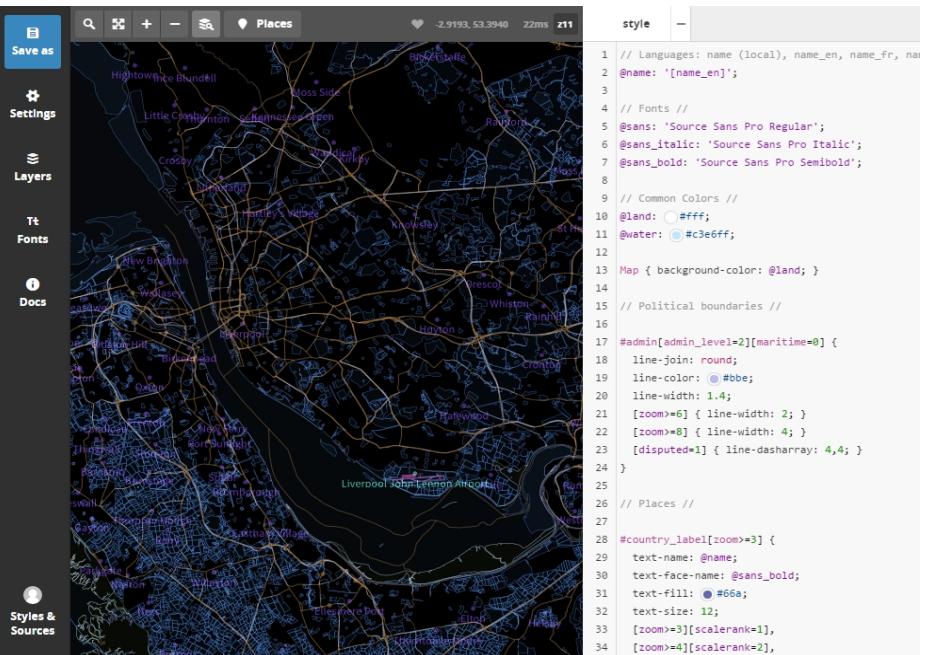
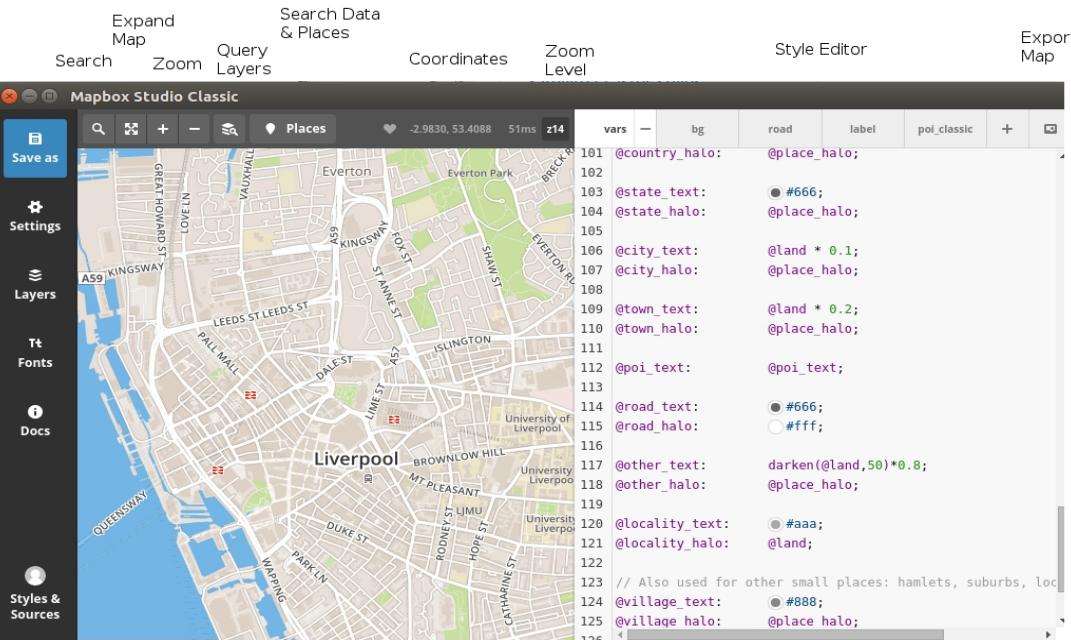
Select a new Style

- Click on Styles & Sources on the lower left hand side.
- Click on New Style or Source.
- Click on the Streets Classic style.
- The OSM road data were imported (note that an internet connection is required).



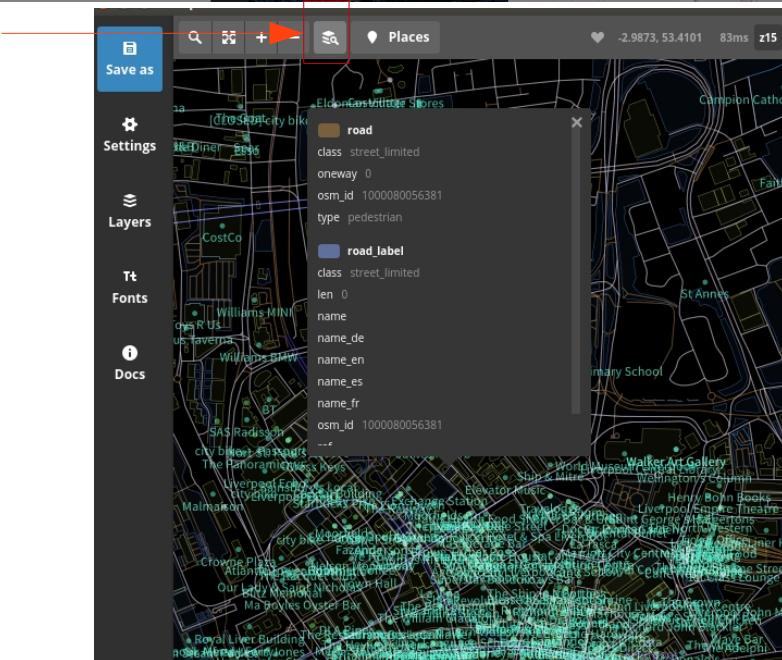
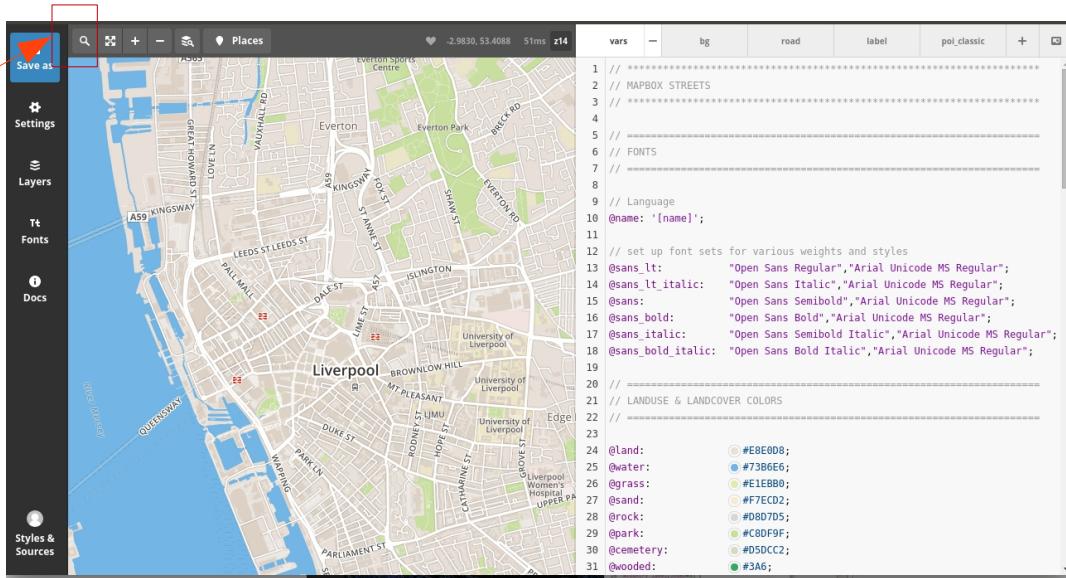
Streets Classic

- The Streets Classic style is optimized for transportation or navigation projects.
- It provides styling for all of the features available in the OSM streets dataset (compare its style editor against the Mapbox Dark style editor below).



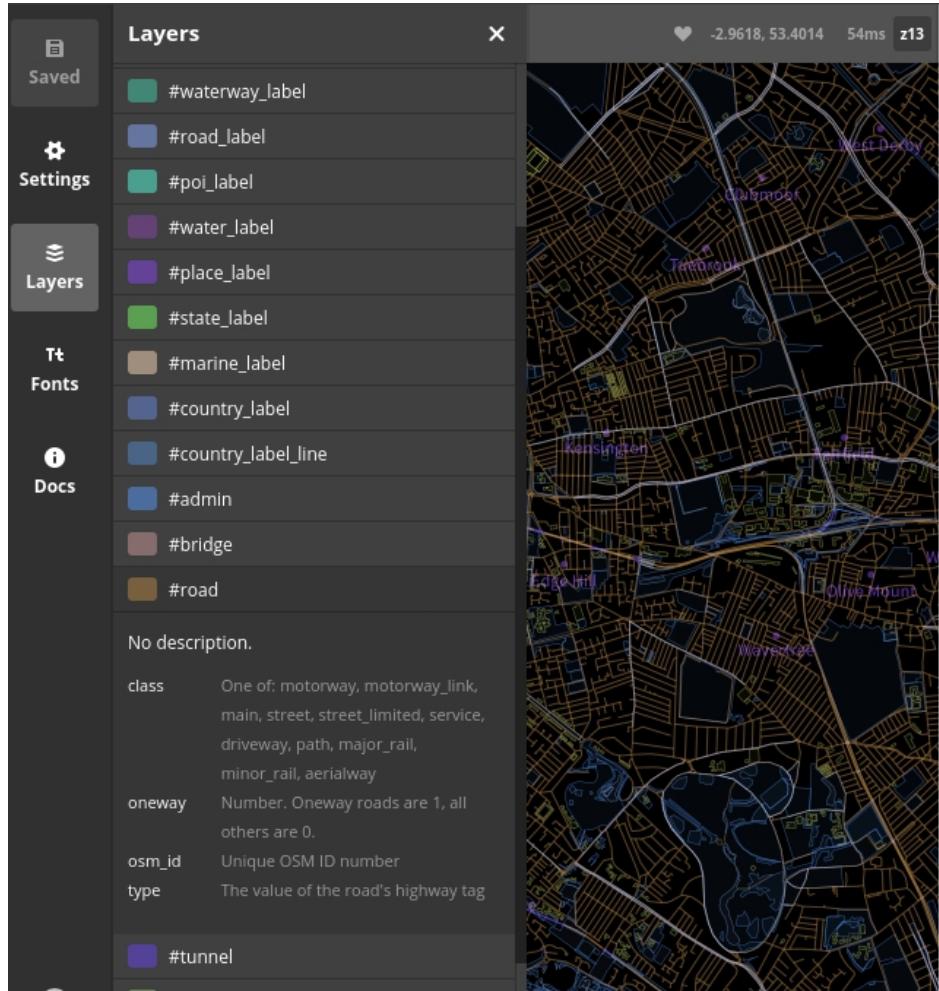
Search places and Query Layers

- In Mapbox Studio search for Liverpool.
- Click on the Query Layers icon to view all layers.
- Click anywhere on the map to see the attributes of the selected layers.



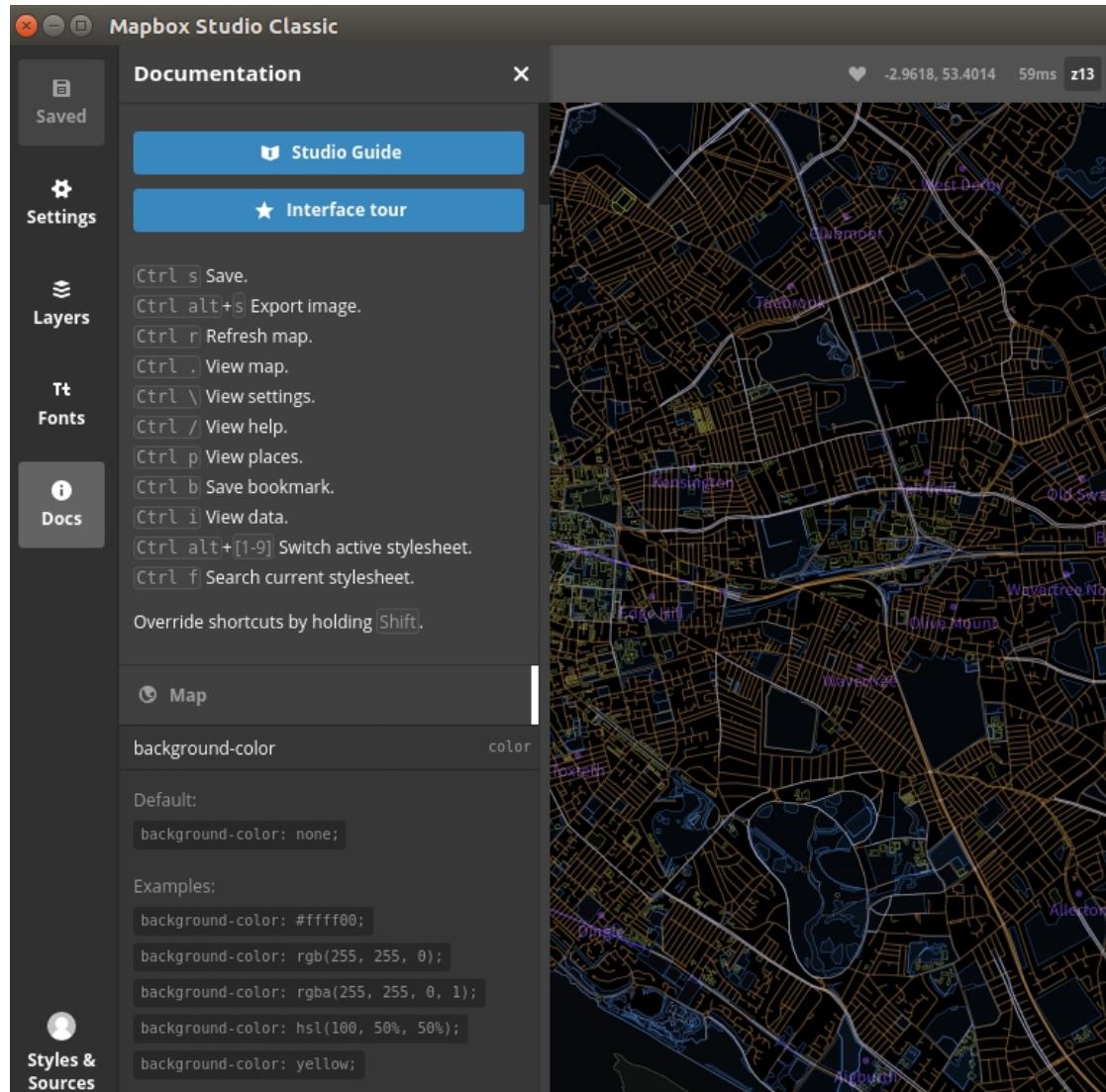
Layer Attributes

- Click on Layers on the left hand side panel.
- Click on a layer (e.g. #road).
 - The ‘class’ attribute has values ‘motorway’, ‘main_street’ etc.
 - Other attributes: ‘oneway’ (0, 1), ‘osm_id’, ‘type’.



Map Element Properties

- Click on Docs on the left hand side panel.
- Select a property (e.g. background-color) to read about the values it can take.



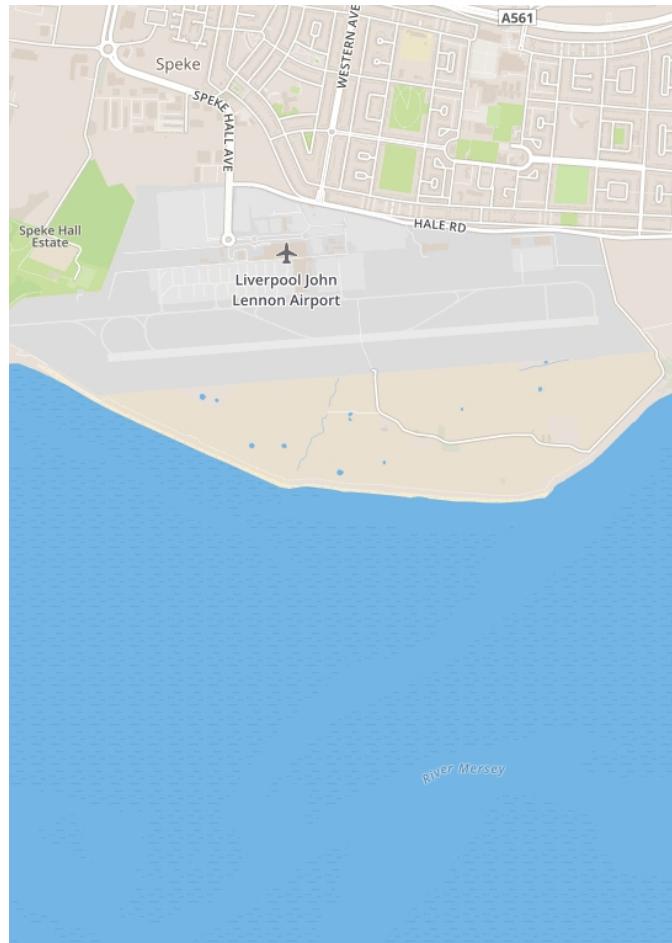
Style by Zoom Level

- You can apply zoom filters using square brackets:

```
#layer[zoom=12] {}
```

```
#layer[zoom>=5][zoom<=9] {}
```

- For example airports (tab bg, line 169) are visible only at zoom level > 9.
 - The line-width property also varies based on the zoom level.



```
164 // =====
165 // AEROWAYS
166 //
167
168 // lines
169 #aeroway['mapnik::geometry_type'=2][zoom>9] {
170   line-color:@aeroway;
171   line-cap:butt;
172   line-join:miter;
173   [type='runway'] {
174     [zoom=10]{ line-width:1; }
175     [zoom=11]{ line-width:2; }
176     [zoom=12]{ line-width:3; }
177     [zoom=13]{ line-width:5; }
178     [zoom=14]{ line-width:7; }
179     [zoom=15]{ line-width:11; }
180     [zoom=16]{ line-width:15; }
181     [zoom=17]{ line-width:19; }
182     [zoom>17]{ line-width:23; }
183   }
184   [type='taxiway'] {
185     [zoom<13]{ line-width:0.2; }
186     [zoom=13]{ line-width:1; }
187     [zoom=14]{ line-width:1.5; }
188     [zoom=15]{ line-width:2; }
189     [zoom=16]{ line-width:3; }
190     [zoom=17]{ line-width:4; }
191     [zoom>17]{ line-width:5; }
192   }
193 }
194
195 // polygons
196 #aeroway['mapnik::geometry_type'=3][zoom>=13] {
197   polygon-fill: @aeroway;
198   [type='apron'] {
199     polygon-fill: @parking;
200   }
201 }
```

Variables

- Look at the vars tab in the CartoCSS editor.
- It lists all the variables used in the project.
- Variables are declared with @.
- They can be used to store and reuse information.

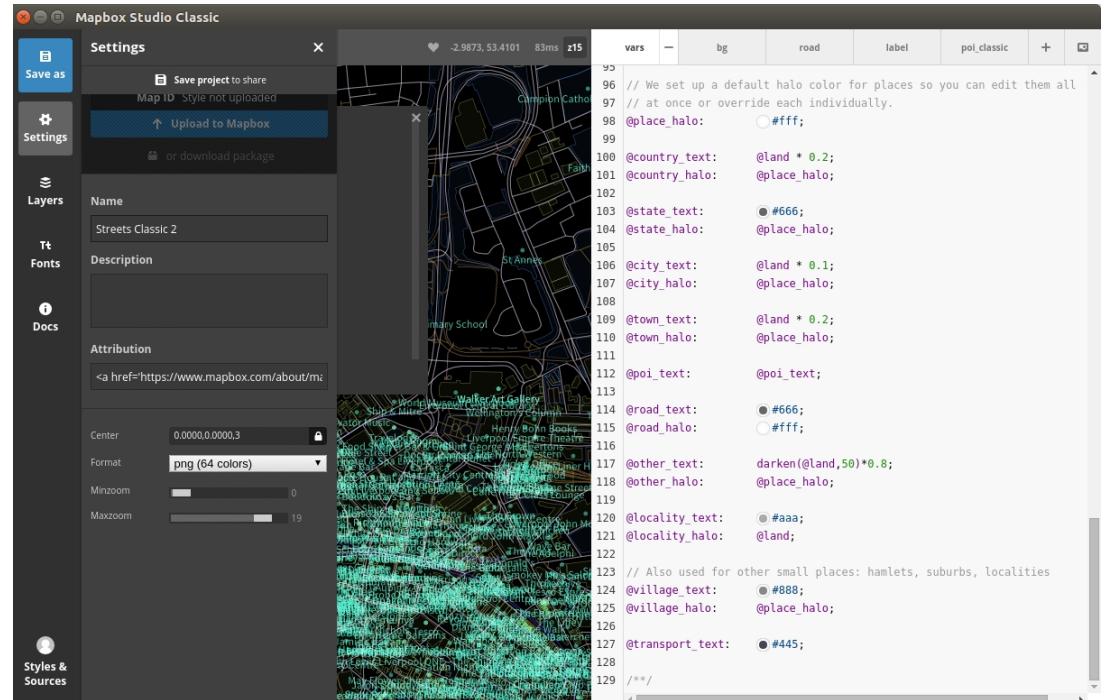
```
vars - bg road label poi_classic + 🎞
3
4
5 // =====
6 // FONTS
7 // =====
8
9 // Language
10 @name: '[name]';
11
12 // set up font sets for various weights and styles
13 @sans_lt: "Open Sans Regular", "Arial Unicode MS Regular";
14 @sans_lt_italic: "Open Sans Italic", "Arial Unicode MS Regular";
15 @sans: "Open Sans Semibold", "Arial Unicode MS Regular";
16 @sans_bold: "Open Sans Bold", "Arial Unicode MS Regular";
17 @sans_italic: "Open Sans Semibold Italic", "Arial Unicode MS Regular";
18 @sans_bold_italic: "Open Sans Bold Italic", "Arial Unicode MS Regular";
19
20 // =====
21 // LANDUSE & LANDCOVER COLORS
22 // =====
23
24 @land: #E8E0D8;
25 @water: #73B6E6;
26 @grass: #E1EBB0;
27 @sand: #F7ECD2;
28 @rock: #D8D7D5;
29 @park: #C8DF9F;
30 @cemetery: #D5DCC2;
31 @wooded: #3A6;
32 @industrial: #DDDCDC;
33 @agriculture: #EAE0D0;
34 @snow: #EDE5DD;
```

Recap

- 6 map elements
 - Map, Points, Lines, Polygons, Raster, Text
- 4 Data types
 - point, line, polygon, raster
- Every map element has unique properties (e.g. ‘line-color’ for Lines). Each data type can inherit properties from different map elements (e.g. line-color applies to lines but also to polygon boundaries).
- Each layer (e.g. road) is member of a single data type and inherits its properties.
- Many layers have attributes (e.g. the attribute ‘class’ for layer road), the values of which can be used to apply different styles to a subset of the data.
- Zoom level can also be used to filter your layers and apply different styles.
- Variables start with @ and can be used to save and reuse information.

Create Custom Style

- Click on Settings on the left hand side panel.
- Give a name to the style (e.g. Streets Classic 2).
- Click on “Save as”.
- A Mapbox Studio style project was created in a directory with .tm2 file extension.



- Main components of project directory:
 - project.yml contains the core metadata and a reference to its vector tile source and the CartoCSS stylesheets it uses.
 - project.xml is a compiled mapnik XML file that can be used directly with tilelive-vector to render the style project as a map.
 - *.mss are CartoCSS files used by the project.

Using the Style Editor

- Zoom in on Liverpool.
- In the vars tab comment out the @land variable (line 24) and then click Save (or Ctrl + S).

```
/* @land: #E8E0D8; */
```

or

```
// @land: #E8E0D8;
```

- Mapbox indicates where the error might have occurred in the style editor tabs.
- Uncomment @land and save changes.

The screenshot shows a Mapbox-style interface. On the left is a map of Liverpool with various neighborhoods labeled: MELR, Kirkdale, WALTON, Everton, KINGSWAL, ST ANNE ST, LIVERPOOL, DUKE ST, and HAMPST. On the right is a code editor with line numbers from 22 to 42. A red box highlights line 24, which contains the commented-out code /* @land: #E8E0D8; */. Above the code editor, three tabs are visible: 'Error: Line 37' (highlighted in orange), 'Error: Line 260', and 'Error: Line 480'. A red arrow points from the bottom right towards the error tab. The code editor also includes a legend for different land types with their corresponding hex codes.

```
22 // =====
23
24 /* @land: #E8E0D8; */
25 @water: #151717;
26 @outlines: #efe8a1;
27 @grass: #E1EBB0;
28 @sand: #F7ECD2;
29 @rock: #D8D7D5;
30 @park: #C8DF9F;
31 @cemetery: #D5DCC2;
32 @wooded: #3A6;
33 @industrial: #DDCDC;
34 @agriculture: #EAE0D0;
35 @snow: #EDE5DD;
36
37 @building: darken(@land, 8);
38 @hospital: #F2E3E1;
39 @school: #F2EAB8;
40 @pitch: #CAE6A9;
41 @sports: @park;
```

Style Background

- Change background polygons colour:

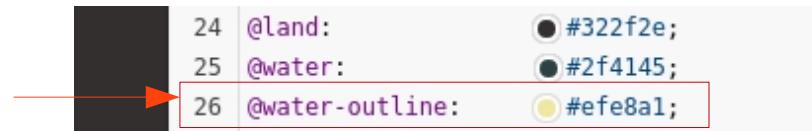
- @land: #322f2e;
- @water: #2f4145;
- Save changes.
- Which properties of the bg style use the @water and @land variables?



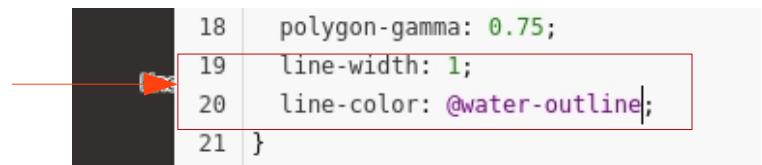
```
19 // =====
20 // =====
21 // LANDUSE & LANDCOVER COLORS
22 // =====
23
24 @land: #322f2e;
25 @water: #2f4145;
26 @grass: #E1EBB0;
27 }
```

- Style polygon boundary:

- Zoom in first in an area where you can see the boundary between water and the coastline.
- In tab vars after @water add:
@water-outline: #efe8a1;
- In tab bg after line 18 add:
line-width: 1;
line-color: @water-outline;
- Save changes.



```
24 @land: #322f2e;
25 @water: #2f4145;
26 @water-outline: #efe8a1;
```



```
18 polygon-gamma: 0.75;
19 line-width: 1;
20 line-color: @water-outline;
21 }
```

Create New Style Tab

- Remove all style tabs by clicking on “-” of each tab.
- Create a new style tab and give it a name (click on “+”).
- In the new style tab add the following variables:

@water: #2f4145;

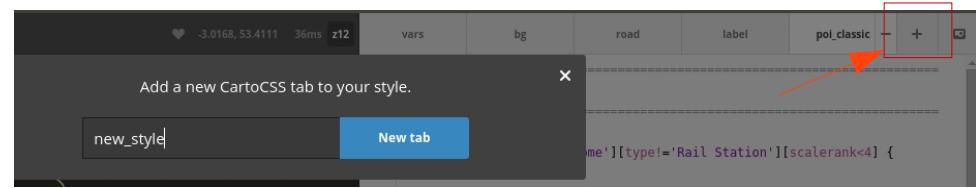
@land: #322f2e;

@streets: #D7AE26;

@streets-dark: #AC8812;

@buildings: #6D6765;

@text: #392C00;

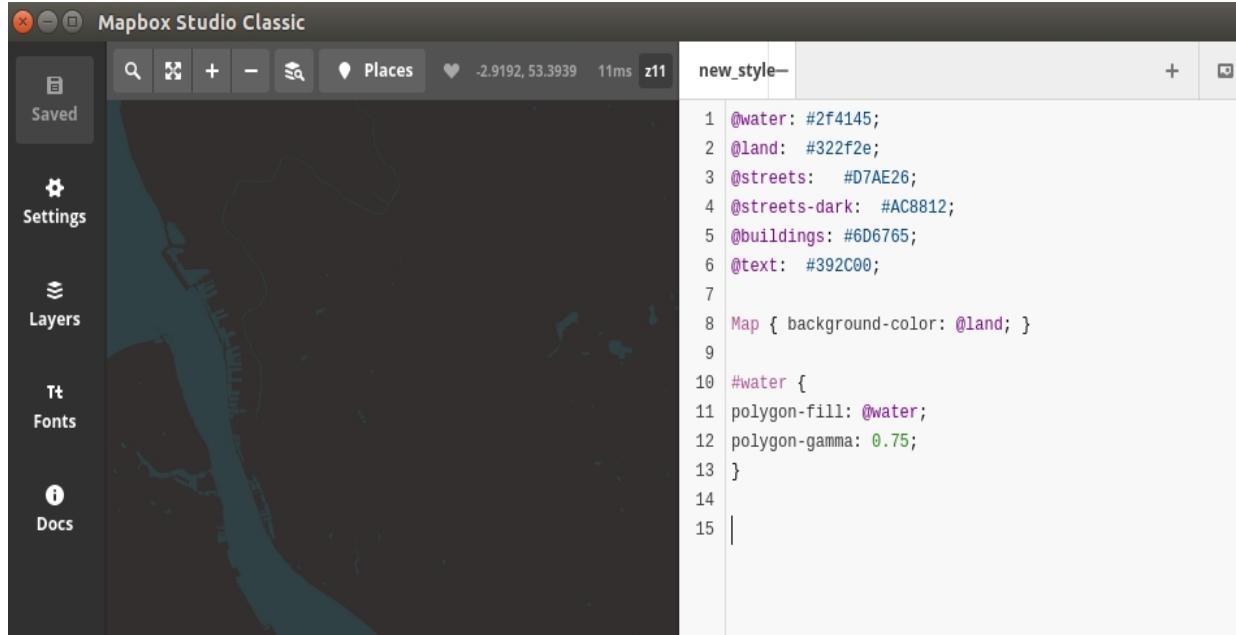


- Then add the following selectors:
Map { background-color: @land; }

#water {
polygon-fill: @water;
polygon-gamma: 0.75;
}

Create New Style Tab

- The new style should like the figure on the right.



The screenshot shows the Mapbox Studio Classic interface. On the left is a sidebar with options: Saved, Settings, Layers, Fonts, and Docs. The main area displays a map of Liverpool with various layers like water bodies, land, streets, and buildings. The top bar includes standard map controls (zoom, search, etc.) and a location indicator (-2.9192, 53.3939). The bottom right corner shows a code editor tab labeled "new_style" containing the following CSS-like styling:

```
1 @water: #2f4145;
2 @land: #322f2e;
3 @streets: #D7AE26;
4 @streets-dark: #AC8812;
5 @buildings: #6D6765;
6 @text: #392C00;
7
8 Map { background-color: @land; }
9
10 #water {
11 polygon-fill: @water;
12 polygon-gamma: 0.75;
13 }
14
15 |
```

Style Roads

- In the new style tab set the color of roads and save:

```
#road{  
    line-color: @streets;  
}
```

- Set the line width for motorways and save:

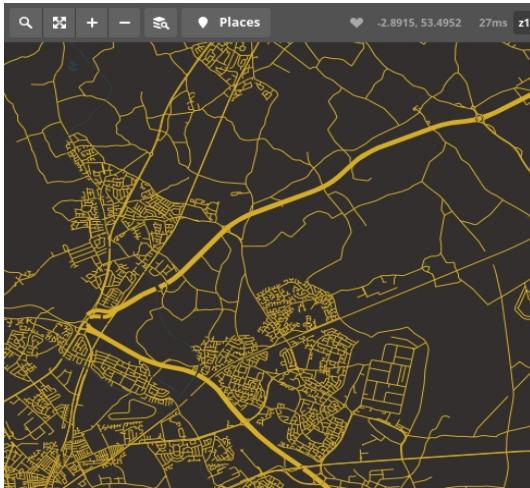
```
#road {  
  
[class='motorway'] {  
    line-width: 4; }  
  
}
```

- Set the same value for a common style property (line-width) using a comma as follows:

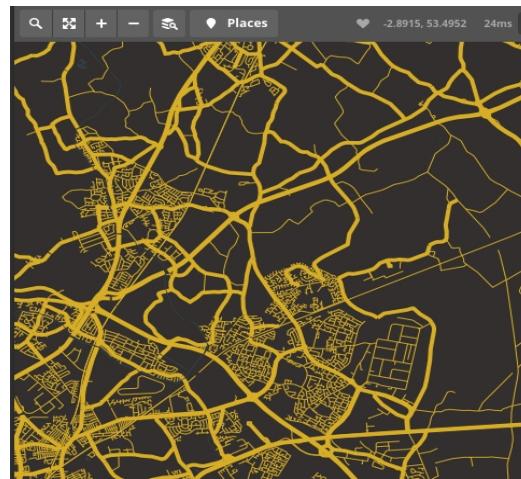
```
#road {  
  
[class='motorway'],  
[class='main'] {  
    line-width: 4; }  
  
}
```



```
1 @water: #2f4145;  
2 @land: #322f2e;  
3 @streets: #D7AE26;  
4 @streets-dark: #AC8812;  
5 @buildings: #6D6765;  
6 @text: #392C00;  
7  
8 Map { background-color: @land; }  
9  
10 #water {  
11 polygon-fill: @water;  
12 polygon-gamma: 0.75;  
13 }  
14  
15 #road{  
    line-color: @streets;  
16 }
```



```
1 @water: #2f4145;  
2 @land: #322f2e;  
3 @streets: #D7AE26;  
4 @streets-dark: #AC8812;  
5 @buildings: #6D6765;  
6 @text: #392C00;  
7  
8 Map { background-color: @land; }  
9  
10 #water {  
11 polygon-fill: @water;  
12 polygon-gamma: 0.75;  
13 }  
14  
15 #road{  
    line-color: @streets;  
16 }  
17  
18 #road {  
20 [class='motorway'] {  
21 line-width: 4; }  
22 }
```

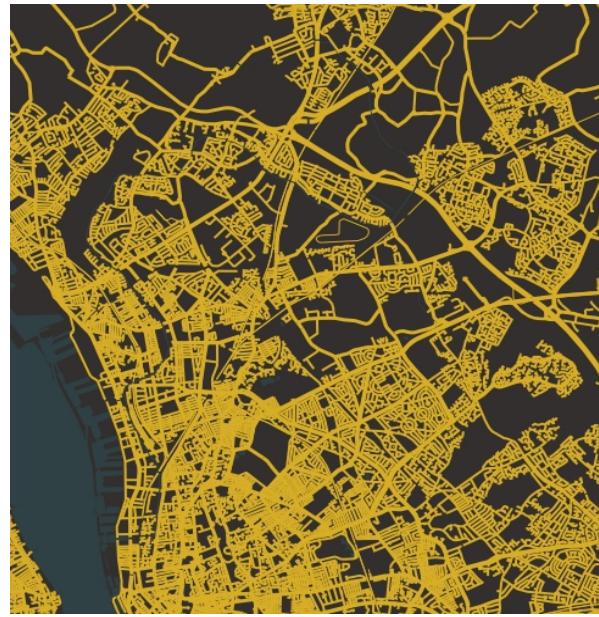


```
1 @water: #2f4145;  
2 @land: #322f2e;  
3 @streets: #D7AE26;  
4 @streets-dark: #AC8812;  
5 @buildings: #6D6765;  
6 @text: #392C00;  
7  
8 Map { background-color: @land; }  
9  
10 #water {  
11 polygon-fill: @water;  
12 polygon-gamma: 0.75;  
13 }  
14  
15 #road{  
    line-color: @streets;  
16 }  
17  
18 #road {  
20 [class='motorway'],  
21 [class='main'] {  
22 line-width: 4; }  
23 }
```

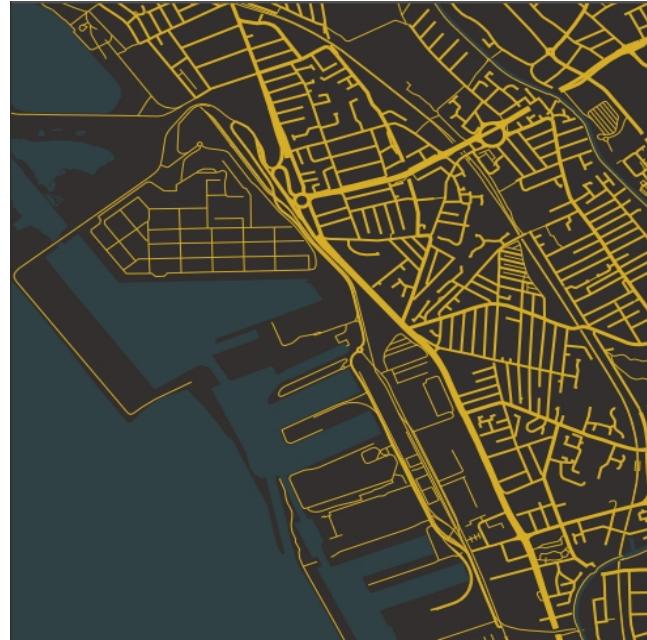
Style Roads: Line Width

Change the line width based on the class attribute as follows:

```
#road {  
  
    // For motorways set line width to 4  
  
    [class='motorway'] {  
        line-width: 4; }  
  
    // For main roads set line width to 3  
  
    [class='main'] {  
        line-width: 3; }  
  
    // For streets set line width to 2  
  
    [class='street'] {  
        line-width: 2; }  
  
    // For streets set line width to 1  
  
    [class='path'] {  
        line-width: 1; }  
  
}
```



```
7  Map { background-color: @land; }  
8  
9  #water {  
10     polygon-fill: @water;  
11     polygon-gamma: 0.75;  
12 }  
13  
14  #road{  
15     line-color: @streets;  
16 }  
17  
18  #road {  
19     // For motorways set line width to 4  
20     [class='motorway'] {  
21         line-width: 4; }  
22     // For main roads set line width to 3  
23     [class='main'] {  
24         line-width: 3; }  
25     // For streets set line width to 2  
26     [class='street'] {  
27         line-width: 2; }  
28     // For paths set line width to 1  
29     [class='path'] {  
30         line-width: 1; }  
31 }  
32 }
```



```
7  Map { background-color: @land; }  
8  
9  #water {  
10     polygon-fill: @water;  
11     polygon-gamma: 0.75;  
12 }  
13  
14  #road{  
15     line-color: @streets;  
16 }  
17  
18  #road {  
19     // For motorways set line width to 4  
20     [class='motorway'] {  
21         line-width: 4; }  
22     // For main roads set line width to 3  
23     [class='main'] {  
24         line-width: 3; }  
25     // For streets set line width to 2  
26     [class='street'] {  
27         line-width: 2; }  
28     // For paths set line width to 1  
29     [class='path'] {  
30         line-width: 1; }  
31 }  
32 }
```

Style Roads: Zoom Level and Opacity

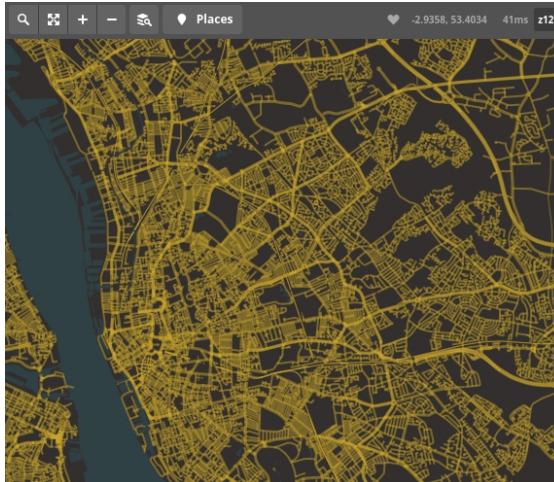
- If you zoom out the streets are very dense with a lot of detail.

- Reduce the opacity of all roads to 0.45 when the zoom level is lower than 16:

```
[zoom < 16] {  
  #road{ line-opacity: 0.45; }  
}
```

- Reduce the opacity of all streets to 0.25 if the zoom level is lower than 14:

```
[zoom < 14] {  
  #road{ line-opacity: 0.25; }  
}
```



```
new_style —  
32 }  
33 // For streets set line width to 2  
34 [class='street'] {  
35   line-width: 2;  
36 }  
37 // For paths set line width to 1  
38 [class='path'] {  
39   line-width: 1;  
40 }  
41 }  
42 [zoom < 16] {  
43   #road{  
44     line-opacity: 0.45;  
45   }  
46 }  
47 }  
48 }  
49 }  
50 }  
51 }  
52 }  
53 }  
54 }
```

```
32 }  
33 // For streets set line width to 2  
34 [class='street'] {  
35   line-width: 2;  
36 }  
37 // For paths set line width to 1  
38 [class='path'] {  
39   line-width: 1;  
40 }  
41 }  
42 [zoom < 16] {  
43   #road{  
44     line-opacity: 0.45;  
45   }  
46 }  
47 }  
48 [zoom < 14] {  
49   #road{  
50     line-opacity: 0.25;  
51   }  
52 }  
53 }  
54 }
```

Style Roads: Zoom Level and Opacity

- Add to the style:

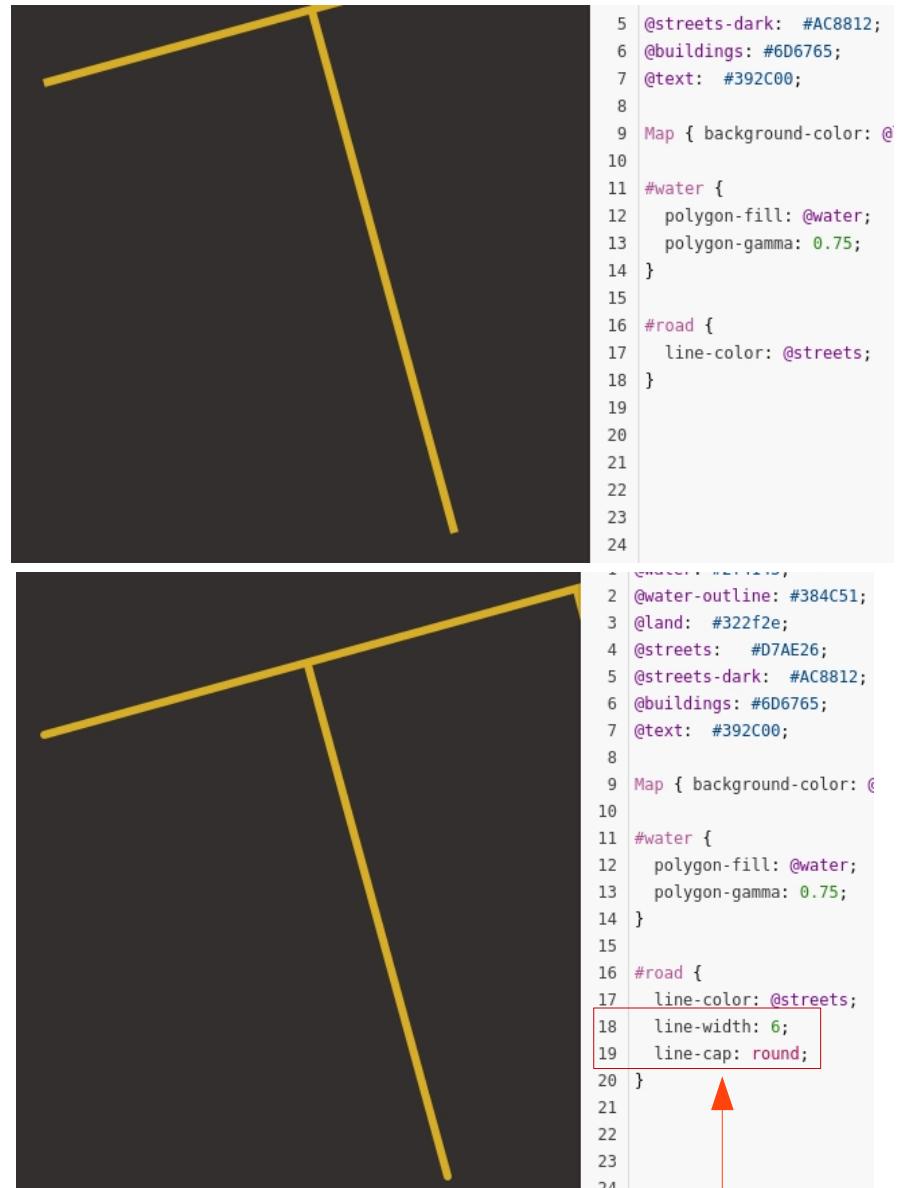
- Hide the streets when the zoom level is less than 12:
[zoom < 12] {
#road{ line-opacity: 0; }
}
- Zoom out at level 11.
- Styles at the bottom of the script overwrite the ones above them.
- Add the following lines at the bottom that set opacity equal to 1 when zoom level less than 16:
[zoom < 16] {
#road{ line-opacity: 1; }
}

```
new_style
40 }
41 }
42 [zoom < 16] {
43   #road{
44     line-opacity: 0.45;
45   }
46 }
47 }
48 [zoom < 14] {
49   #road{
50     line-opacity: 0.25;
51   }
52 }
53 }
54 [zoom < 12] {
55   #road{
56     line-opacity: 0;
57   }
58 }
59 }
```

```
new_style
40 }
41 }
42 [zoom < 16] {
43   #road{
44     line-opacity: 0.45;
45   }
46 }
47 }
48 [zoom < 14] {
49   #road{
50     line-opacity: 0.25;
51   }
52 }
53 }
54 [zoom < 12] {
55   #road{
56     line-opacity: 0;
57   }
58 }
59 }
60 [zoom < 16] {
61   #road{
62     line-opacity: 1;
63   }
64 }
```

Style Roads: line-cap

- Delete the lines below the line-color property of the layer road.
- Increase maximum zoom.
 - “Settings” → Drag “maxzoom” bar to the right up to zoom 22.
- Use line-cap to set how the end of the lines look.
 - Zoom in an area where you can clearly see the end of a road line.
 - Add from line 17:
line-width: 6;
line-cap: round;



```
5 @streets-dark: #AC8812;
6 @buildings: #D6D765;
7 @text: #392C00;
8
9 Map { background-color: @streets-dark; }
10
11 #water {
12   polygon-fill: @water;
13   polygon-gamma: 0.75;
14 }
15
16 #road {
17   line-color: @streets;
18 }
19
20
21
22
23
24
```

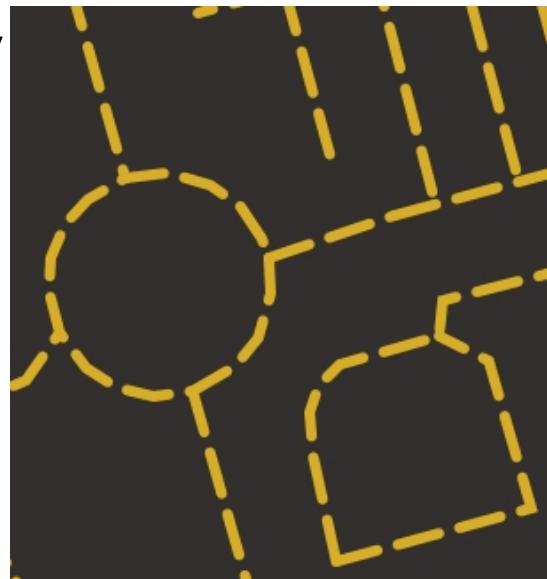


```
1 @water: #384C51;
2 @water-outline: #384C51;
3 @land: #322f2e;
4 @streets: #D7AE26;
5 @streets-dark: #AC8812;
6 @buildings: #D6D765;
7 @text: #392C00;
8
9 Map { background-color: @streets-dark; }
10
11 #water {
12   polygon-fill: @water;
13   polygon-gamma: 0.75;
14 }
15
16 #road {
17   line-color: @streets;
18   line-width: 6;
19   line-cap: round;
20 }
21
22
23
24
```



Style Roads: line-dasharray

- The line-dasharray property takes a list of comma separated pixel-widths.
 - Each value represents a dash and a space.
- Set pixel width for dash to 24, pixel width for space to 12:
 - Line 19:
line-dasharray: 24, 12;
- More complex:
 - Line 20:
line-dasharray: 24, 12, 15, 7, 12, 3;



```
6 @buildings: #6D6765;
7 @text: #392C00;
8
9 Map { background-color: @land; }
10
11 #water {
12   polygon-fill: @water;
13   polygon-gamma: 0.75;
14 }
15
16 #road {
17   line-color: @streets;
18   line-width: 6;
19   line-cap: round;
20   line-dasharray: 24, 12;
21 }
```

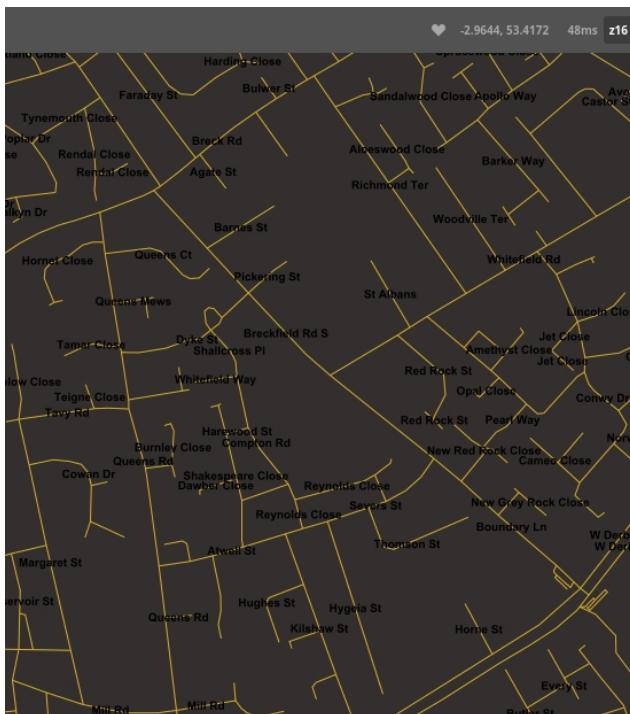


```
7 @text: #392C00;
8
9 Map { background-color: @land; }
10
11 #water {
12   polygon-fill: @water;
13   polygon-gamma: 0.75;
14 }
15
16 #road {
17   line-color: @streets;
18   line-width: 6;
19   line-cap: round;
20   line-dasharray: 24, 12, 15, 7, 12, 3;
21 }
```

Style Roads: Labels

- Delete the lines below the line-color property of the layer road.
- Add labels to the roads if the zoom level is higher than 15:

```
#road_label[zoom > 15] {  
    text-name: [name];  
    text-face-name: "Arial Unicode  
    MS Bold";  
}  
  
– You have to zoom in at least to level 16  
to see the road name.
```
- The text-name property can pull
text from your layer's attributes:
– name is an attribute of the #road layer.



```
new_style —  
4 @streets-dark: #AC8812;  
5 @buildings: #6D6765;  
6 @text: #392C00;  
7  
8 Map { background-color: @land; }  
9  
10 #water {  
11 polygon-fill: @water;  
12 polygon-gamma: 0.75;  
13 }  
14  
15 #road{  
16 line-color: @streets;  
17 }  
18  
19 #road_label[zoom > 15] {  
20    text-name: [name];  
21    text-face-name: "Arial Unicode MS Bold";  
22 }
```

Style Roads: Labels

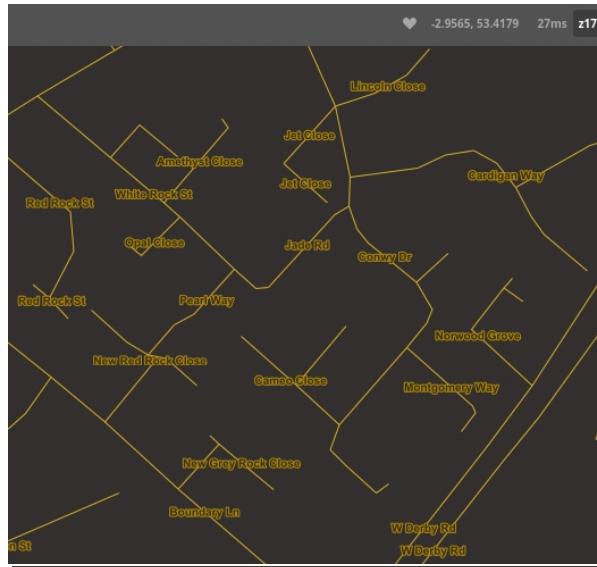
- Make labels more legible by changing text colour and adjusting the halo colour and radius.

- From line 22

```
text-fill: @text;  
text-halo-radius: 1;  
text-halo-fill: #AC8812;
```

- Align labels to streets to make it easier to identify which label belongs to which street.

- Use text-placement property:
text-placement: line;



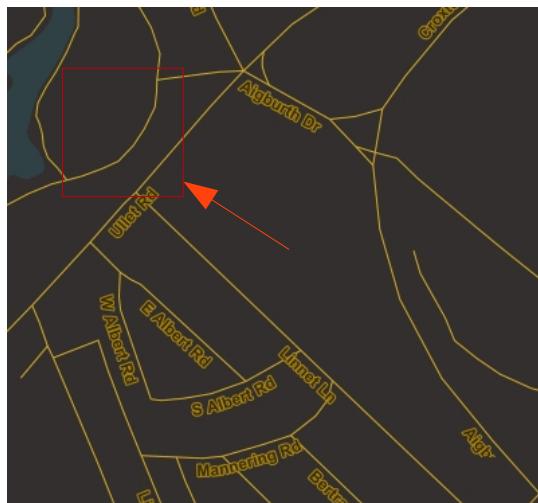
```
new_style —  
4 @streets-dark: #AC8812;  
5 @buildings: #6D6765;  
6 @text: #392C00;  
7  
8 Map { background-color: @land; }  
9  
10 #water {  
11 polygon-fill: @water;  
12 polygon-gamma: 0.75;  
13 }  
14  
15 #road{  
16 line-color: @streets;  
17 }  
18  
19 #road_label[zoom > 15] {  
20 text-name: [name];  
21 text-face-name: "Arial Unicode MS Bold";  
22 text-fill: @text;  
23 text-halo-radius: 1;  
24 text-halo-fill: ● #AC8812; ----->  
25 }  
26 }  
  
new_style —  
4 @streets-dark: #AC8812;  
5 @buildings: #6D6765;  
6 @text: #392C00;  
7  
8 Map { background-color: @land; }  
9  
10 #water {  
11 polygon-fill: @water;  
12 polygon-gamma: 0.75;  
13 }  
14  
15 #road{  
16 line-color: @streets;  
17 }  
18  
19 #road_label[zoom > 15] {  
20 text-name: [name];  
21 text-face-name: "Arial Unicode MS Bold";  
22 text-fill: @text;  
23 text-halo-radius: 1;  
24 text-halo-fill: ● #AC8812; ----->  
25 text-placement: line; ----->  
26 }
```

Style Roads: Labels

- Offset the road labels and increase text size.
 - Use the text-dy property to offset labels by 5 pixels, use text-size to increase label size to 12 pixels.
`text-size: 12;`
`text-dy: 5;`
- Specify maximum angle that the text will try to wrap around.
 - Use text-max-char-angle-delta property:
`text-max-char-angle-delta: 10;`



```
14 #road {  
15   line-color: @streets;  
16 }  
17  
18 #road_label[zoom > 15] {  
19   text-name: [name];  
20   text-face-name: "Arial Unicode MS Bold";  
21   text-fill: @text;  
22   text-halo-radius: 1;  
23   text-halo-fill: #AC8812;  
24   text-placement: line;  
25   text-size: 12;  
26   text-dy: 5;  
27 }  
28  
29 }
```



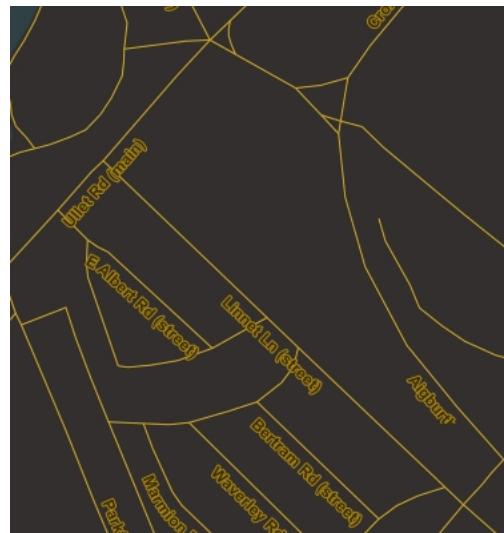
```
12 polygon-gamma: 0.75;  
13 }  
14  
15 #road {  
16   line-color: @streets;  
17 }  
18  
19 #road_label[zoom > 15] {  
20   text-name: [name];  
21   text-face-name: "Arial Unicode MS Bold";  
22   text-fill: @text;  
23   text-halo-radius: 1;  
24   text-halo-fill: #AC8812;  
25   text-placement: line;  
26   text-size: 12;  
27   text-dy: 5;  
28   text-max-char-angle-delta: 10;  
29 }
```

Style Roads: Custom Labels

- You can pull text from multiple fields.
 - Add to the 'text-name' property the road field 'class'.

```
text-name: [name] + '  
( ' + [class] + ' )';
```
- Wrap long labels to multiple lines.
 - Combine 'text-wrap-with' and 'text-wrap-before' properties:

```
text-wrap-width: 100;  
text-wrap-before:  
true;
```



```
13 }  
14  
15 #road {  
16   line-color: @streets;  
17 }  
18  
19 #road_label[zoom > 15] {  
20   text-name: [name] + ' (' + [class] + ')';  
21   text-face-name: "Arial Unicode MS Bold";  
22   text-fill: @text;  
23   text-halo-radius: 1;  
24   text-halo-fill: ● #AC8812;  
25   text-placement: line;  
26   text-size: 12;  
27   text-dy: 5;  
28   text-max-char-angle-delta: 10;  
29 }  
30  
31  
32 }
```



```
18  
19 #road_label[zoom > 15] {  
20   text-name: [name] + ' (' + [class] + ')';  
21   text-face-name: "Arial Unicode MS Bold";  
22   text-fill: @text;  
23   text-halo-radius: 1;  
24   text-halo-fill: ● #AC8812;  
25   text-placement: line;  
26   text-size: 12;  
27   text-dy: 5;  
28   text-max-char-angle-delta: 10;  
29   text-wrap-width: 100;  
30   text-wrap-before: true;  
31 }  
32 }
```

Style Roads: Custom Selector

- Specify a custom selector with '...'. That way you can define a style for the same layer without overriding any previous style.
- Add a road style with blue road line:
 - Line 19:

```
#road::glow {  
    line-color: #0AF;  
    line-opacity: 0.2;  
    line-width: 10  
}
```
- The new style adds a blue glow on top of the road network instead of replacing the previous style.



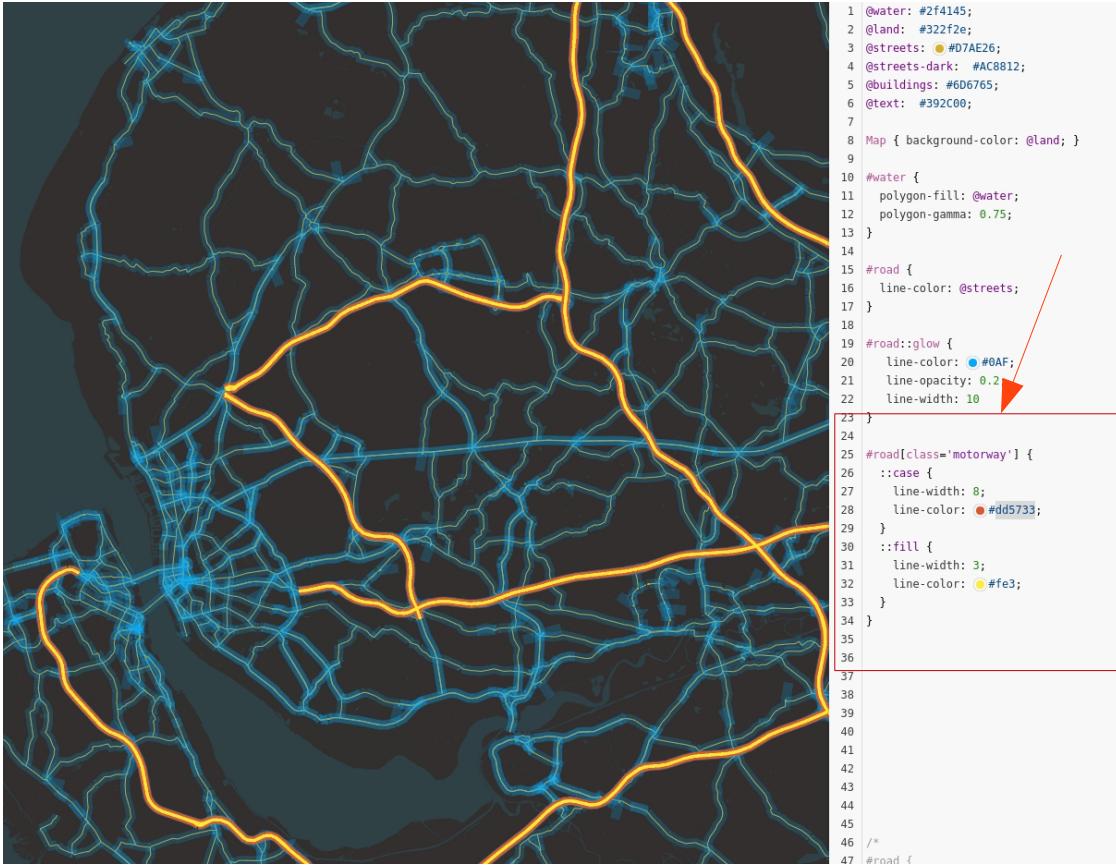
- Note that 'glow' is an arbitrary name not a special keyword.

Style Roads: Custom Selector

- You can use a custom selector within the curly brackets:

- Line 25:

```
#road[class='motorway'] {  
  ::case {  
    line-width: 8;  
    line-color:  
    #dd5733;  
  }  
  ::fill {  
    line-width: 3;  
    line-color: #fe3;  
  }  
}
```



- The new style will override the existing road style (line 15) but only for motorways.
- Within the new style we define two styles for motorway lines.
 - ::case is 8 pixels wide, uses bolder colour and defines the boundary of the motorway.
 - ::fill is 3 pixels wide it is drawn after the ::case style but does not override it.

Style Roads: Custom Selector

- To apply different styles for multiple types of road you need to group your styles by attributes:

- Line 25:

```
#road {  
  [class='motorway'] {  
    ::case {  
      line-width: 8;  
      line-color:  
    #dd5733; }  
    ::fill {  
      line-width: 3;  
      line-color: #fe3; } }  
  [class='main'] {  
    ::case {  
      line-width: 6;  
      line-color: #ca8; }  
    ::fill {  
      line-width: 2.5;  
      line-color: #ffa; } } }
```



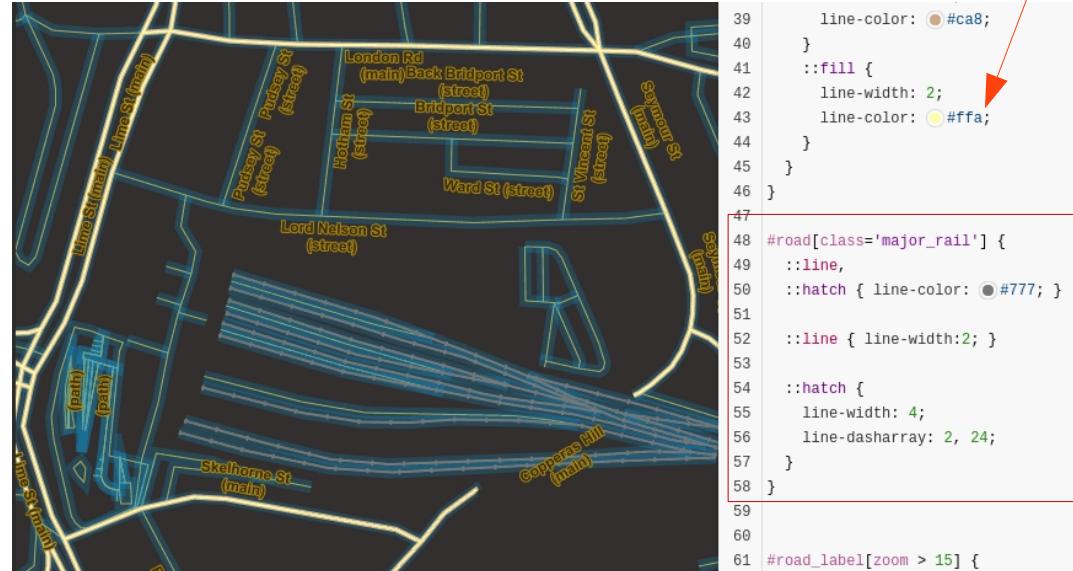
- In this case we grouped our styles by the attribute ‘class’.
- The names do not have to be unique (::case and ::fill are used for both styles).

Style Railroads

- Style railroads using regular hatches on a thin line.
- Two styles one thin and solid line the other thick and dashed.
- The dash should be short with wide spacing.
- Add

– Line 48:

```
#road[class='major_rail'] {  
  ::line, ::hatch { line-  
    color: #777; }  
  ::line { line-width: 2; }  
  ::hatch {  
    line-width: 4;  
    line-dasharray: 2, 24;  
  }  
}
```



- The value for the line-color property is common so we specify it for both ::line and ::hatch styles separating them with a comma.
- Given that the dashes are wider than the ::line styling they create the hatch pattern for railroads.
- We can call multiple times each custom style to add more properties.

Style Tunnels

- Create a tunnel style with longer dash and shorter spacing compared to the previous example.

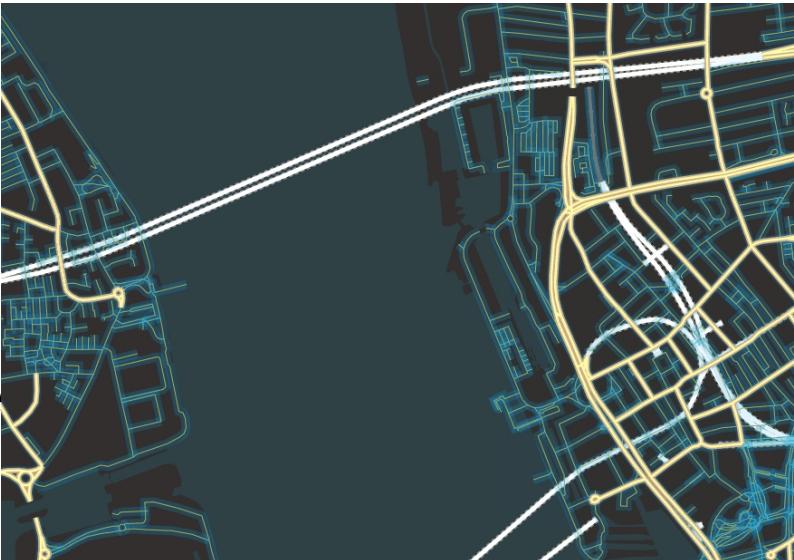
- Line 59:

```
#tunnel {  
  ::case {  
    line-width: 8;  
    line-color: #888;  
    line-dasharray: 4, 3;  
  }  
  ::fill {  
    line-width: 5;  
    line-color: #fff;  
  }  
}
```

- Move the case style after the fill, what's the difference?



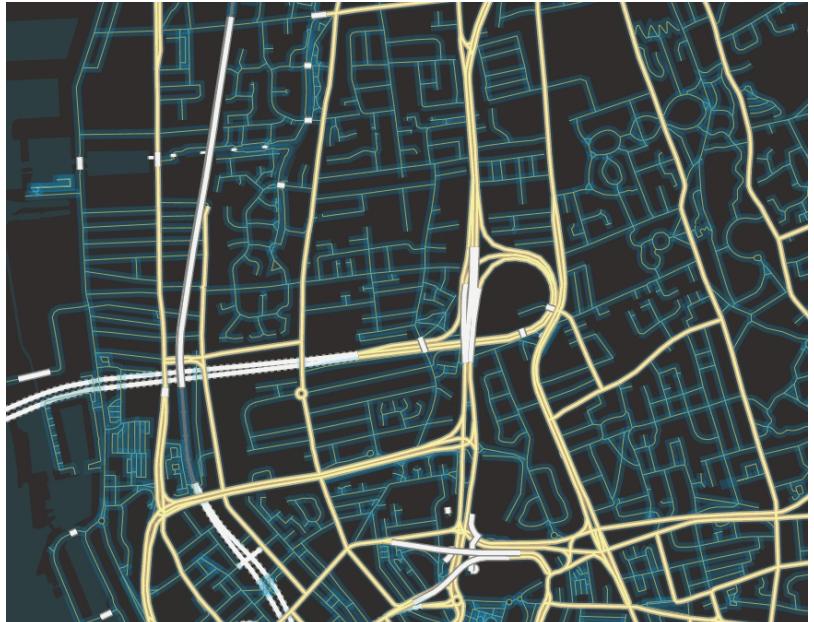
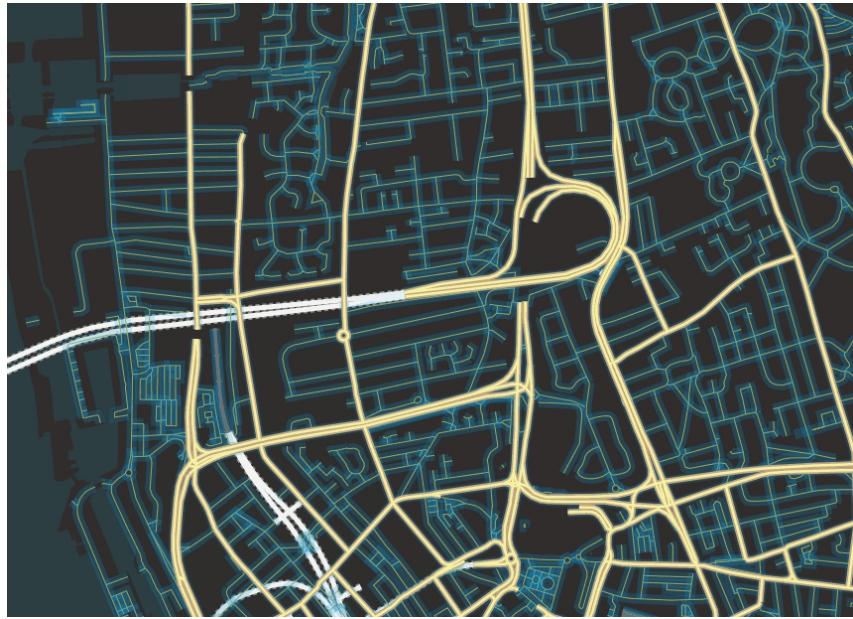
```
51  ::line { line-width: 2; }  
52  ::hatch {  
53    line-width: 4;  
54    line-dasharray: 2, 24;  
55  }  
56  ...  
57 }  
58 ...  
59 #road_label [zoom > 15] {  
60   text-name: [name] + ' (' + [class] + ')';  
61   text-face-name: "Arial Unicode MS Bold";  
62   text-fill: @text;  
63   text-halo-radius: 1;  
64   text-halo-fill: ●#AC8812;  
65   text-placement: line;  
66   text-size: 12;  
67   text-dy: 5;  
68   text-max-char-angle-delta: 10;  
69   text-wrap-width: 100;  
70   text-wrap-before: true;  
71 }  
72 ...  
73 ...  
74 ...  
75 ...  
76 ...  
77 ...  
78 ...
```



```
48 #road [class = 'major_rail'] {  
49   ::line, ::hatch { line-color: ●#777; }  
50   ::line { line-width: 2; }  
51   ::hatch {  
52     line-width: 4;  
53     line-dasharray: 2, 24;  
54   }  
55 }  
56 ...  
57 ...  
58 ...  
59 #tunnel {  
60   ::case {  
61     line-width: 8;  
62     line-color: ●#888;  
63     line-dasharray: 4, 3;  
64   }  
65   ::fill {  
66     line-width: 5;  
67     line-color: ○#fff;  
68   }  
69 }  
70 ...  
71 ...  
72 ...  
73 ...  
74 ...  
75 ...  
76 ...  
77 ...  
78 ...
```

Style Bridges

- Create a style for bridges, there should be a case style and a fill style for greater effect.
- Create a style to label bridges using the attribute ‘class’ when the zoom level is greater than 15.



Many thanks...

