You can use tags and CSS classes together to style categories of points the same way, making items of like qualities easier to identify on the map. To do so, you'll use the relevant tag for that item and assign it a CSS class of the same name. For example, let's say that I want every item tagged with "electricity" to show up as a yellow dot on the map. I would first create a CSS class called "`electricity`" and give it the following attributes:

```
.electricity {
  fill-color: yellow;
  fill-color-select: yellow;
  fill-opacity: .75;
  fill-opacity-select: .75;
  stroke-color: black;
  min-zoom: 11;
  max-zoom: 20;
}
```

Each point tagged with "electricity" would now be represented by a dot that looks like this:

A quick rundown of the attributes listed here…

`fill-color` refers to the fill color of the dot or shape. You can set this with web colors, or by using a six-character hex code. In this case, `yellow` is also `#ffcc00`.

`fill-color-select` is the fill color of the dot or shape when the user has clicked on it. This can remain the same or it can change when selected.

`fill-opacity` sets the opacity of the fill color. Here it is set to 100%. If you want the dot or shape to be somewhat transparent, you can set it at `.75` or `.8` (anything lower than that will result in a dot or shape that is very difficult to see on the map).

`fill-opacity-select`, like `fill-color-select`, this sets the opacity for the selected dot or shape. Usually you would want this to be the same as `fill-opacity`.

`stroke-color` is the color of the outline of the dot or shape. Generally, this is set to `black`, `white`, or the `fill-color`. You could also add a `stroke-color-select` attribute and change the color to `gray` or `white` when the dot or shape is selected.

`min-zoom` and `max-zoom` set the zoom levels for which this point will be visible. By altering these values, you can control what the user sees at various zoom levels.

Here are some sample classes to get you started, along with a depiction of the point that it would produce. Copy and paste these classes into the "Stylesheet" field of the "Styles" tab of your Neatline exhibit, and change the class names to match the tags you'd like to use:


```
.blue {
  fill-color: blue;
  fill-color-select: blue;
  stroke-color: black;
  fill-opacity: .75;
  fill-opacity-select: .75;
  min-zoom: 11;
  max-zoom: 20;
}
```



```
.green {
  fill-color: green;
  fill-color-select: green;
  stroke-color: black;
  fill-opacity: .75;
  fill-opacity-select: .75;
  min-zoom: 11;
  max-zoom: 20;
}
```



```
.purple {
  fill-color: purple;
  fill-color-select: purple;
  stroke-color: black;
  fill-opacity: .75;
  fill-opacity-select: .75;
  min-zoom: 11;
  max-zoom: 20;
}
```

```
.gray {
  fill-color: gray;
  fill-color-select: gray;
  stroke-color: black;
  fill-opacity: .75;
  fill-opacity-select: .75;
  min-zoom: 11;
  max-zoom: 20;
}
```