

## 基于要素属性的样式

到目前为止，在此本章中，我们介绍了样式功能使用的 `setStyle` 方法在层级上 `ol.layer.Vector` 。我们还研究了根据单个特征的几何类型设置样式，然后使用类中的 `setStyle` 方法应用样式 `ol.Feature` 。

对于本食谱，我们将探讨在特征级别使用样式功能对样式进行样式设置的另一种方法 `ol.layer.Vector` 。向量图层类具有名为的属性 `style` ，该属性不仅接受的一个实例 `ol.style.Style` 或各种 `ol.style.Style` 实例的数组，而且还接受一个 `ol.style.StyleFunction` 方法。每当在地图上呈现要素时，都会调用此方法，并且此方法的结果必须返回 `ol.style.Style` 实例数组。

作为这项新样式设计技术的一部分，我们将基于要素属性确定如何应用某些样式。

我们将加载一个美国某些城市的GeoJSON文件，并提供这些功能的默认样式。用户将能够从输入字段中按名称搜索城市，并以此在地图上过滤城市，从而使匹配的城市脱颖而出，而与搜索查询不匹配的城市将为不太明显。

可以在中找到源代码 `ch06/ch06-styling-features-by-attribute` ，然后以类似于以下屏幕快照的形式结束：

怎么做...

遵循以下说明，学习如何基于要素属性设置要素样式：

- 1 首先创建一个新的HTML文件，然后添加OpenLayers依赖项和一个 `div` 用于保存 `map` 实例的元素。特别是，添加搜索表单的HTML，因为我们在JavaScript中对此进行了引用：

```
<form>
  <label for="js-search-query">Filter by city name</label>
  <input id="js-search-query" placeholder="Enter city name...">
</form>
```

复制

2 创建一个自定义JavaScript文件，并 `map` 使用 `view` 实例和卫星图块栅格图层初始化实例：

复制

```
var map = new ol.Map({
  view: new ol.View({
    zoom: 4, center: [-10987364, 4109254]
  }),
  target: 'js-map',
  layers: [
    new ol.layer.Tile({
      source: new ol.source.MapQuest({layer: 'sat'})
    })
  ]
});
```

3 缓存 搜索输入字段DOM元素并订阅 `keyup` 活动。让处理程序在向量层上触发change事件：

复制

```
var searchQueryElem = document.getElementById('js-search-query');
searchQueryElem.addEventListener('keyup', function() {
  vectorLayer.changed();
});
```

4 实现将应用于矢量层的样式功能：

复制

```
var styleFunction = function(feature) {
  var cityName = feature.get('name');
  var searchQuery = searchQueryElem.value;
  var opacity = 1;
  if (searchQuery &&
    cityName.search(new RegExp(searchQuery, 'i')) === -1) {
    opacity = 0.3;
  }
  return [
    new ol.style.Style({
      image: new ol.style.Circle({
        radius: 10,
        stroke: new ol.style.Stroke({
          color: [74, 138, 168, opacity], width: 2
        })
      })
    }),
    new ol.style.Text({
      text: cityName,
      fill: new ol.style.Fill({
```

5 创建 向量图层，指定样式功能该 `style` 物业，并有矢量源获取GeoJSON的文件。最后，将图层添加到地图：

复制

```
var vectorLayer = new ol.layer.Vector({
  source: new ol.source.Vector({
    url: 'cities.geojson',
    format: new ol.format.GeoJSON()
  }),
  style: styleFunction
});
map.addLayer(vectorLayer);
```

## 怎么运行的...

正如我们通常所做的那样，为简洁起见，此处省略了许多HTML和所有CSS。请查看随附的源代码，以全面了解其结构和样式：

复制

```
searchQueryElem.addEventListener('keyup', function() {
  vectorLayer.changed();
});
```

我们没有等待用户通过表单提交来提交搜索请求，而是选择了 `keyup` 从 `input` 元素中订阅事件，从而选择了更直接的响应。`keyup` 发布此事件后，我们将 `changed` 在矢量层上调用方法。该 `changed` 方法为该层增加内部修订计数器，并调度一个change事件，该事件依次调用 `ol.style.StyleFunction` 向量源中的每个要素。

强制从此发布的事件中重新渲染要素样式非常重要，因为我们的要素样式部分基于用户的城市过滤要求。

现在，让我们看一下其中的一些 `styleFunction` 实现，以继续学习大部分代码：

复制

```
var styleFunction = function(feature) {
  var cityName = feature.get('name');
  var searchQuery = searchQueryElem.value;
  var opacity = 1;
```

什么时候 OpenLayers在内部调用此函数 为了确定特征样式，它将传入 `feature` 问题作为其第一个参数。它还提供了第二个参数，代表视图的分辨率，由于我们没有使用它，因此我们将其遗漏了。

GeoJSON文件中的每个要素都在 `name` 属性中包含城市名称。我们使用 `get from` 的方法 `ol.Feature` 来检索该值并将其存储在 `cityName` 变量中。我们从输入字段中获取最新的搜索查询值，并将其存储在 `searchQuery` 变量中。我们还建立了一个默认 `opacity` 的 `1`，它得到的特征造型定制使用。

复制

```
if (searchQuery &&
    cityName.search(new RegExp(searchQuery, 'i')) === -1) {
    opacity = 0.3;
}
```

我们检查输入字段的值是否不为空（如果为空，则 `searchQuery` 变量为`false`。如果不为空，则推断用户已输入某种搜索查询，我们希望根据用户要求过滤城市。

这使我们进入下一个条件：`cityName` 对照来自输入字段的搜索查询，检查要素（）中的城市名称。搜索查询将转换为不区分大小写的正则表达式（`new RegExp(searchQuery, 'i')`）并传递给JavaScript `search` 方法。这将在字符串中搜索正则表达式的内容。例如，来自`feature`属性的城市名称“New York, NY”将与搜索的 `'york'` 正则表达式 `/york/i` 。

如果存在匹配项，则返回匹配开始的索引；否则，`-1` 将不返回任何匹配项。如果此功能与搜索查询不匹配，则将 `opacity` 变量设置为 `0.3` 。这将使用户看不到不匹配的城市，从而突出显示所有匹配的城市，这将保留的不透明度 `1` 。

样式函数的其余部分将返回一个数组，其中一个项目包含一个 `ol.style.Style` 实例，该实例可以满足所需的功能协定 `ol.style.StyleFunction` 。早期的食谱看起来很熟悉此配置，因此不需要进一步说明。您会注意到该 `opacity` 变量用于 `color` 属性数组值的alpha部分，例如 `color: [74, 138, 168, opacity]` 。这是城市或要素是否与搜索查询匹配的动态结果。

你也会 请注意，该 `cityName` 变量已分配给实例的 `text` 属性 `ol.style.Text` 。城市名称构成可见的几何标签。

实例化向量层后，我们将样式函数分配给该 `style` 属性。

也可以看看

#### ✦ 该造型层配方

#### ✦ 在添加文本标签的几何点配方在第3章 (/book/web\_development/9781785287756/3)，使用矢量图层

#### ✦ 第3章 (/book/web\_development/9781785287756/3)，使用矢量层中的将点特征用作标记配方 (/book/web\_development/9781785287756/3)

#### ✦ 该删除或克隆利用覆盖功能在配方第3章 (/book/web\_development/9781785287756/3)，工作与矢量图层

