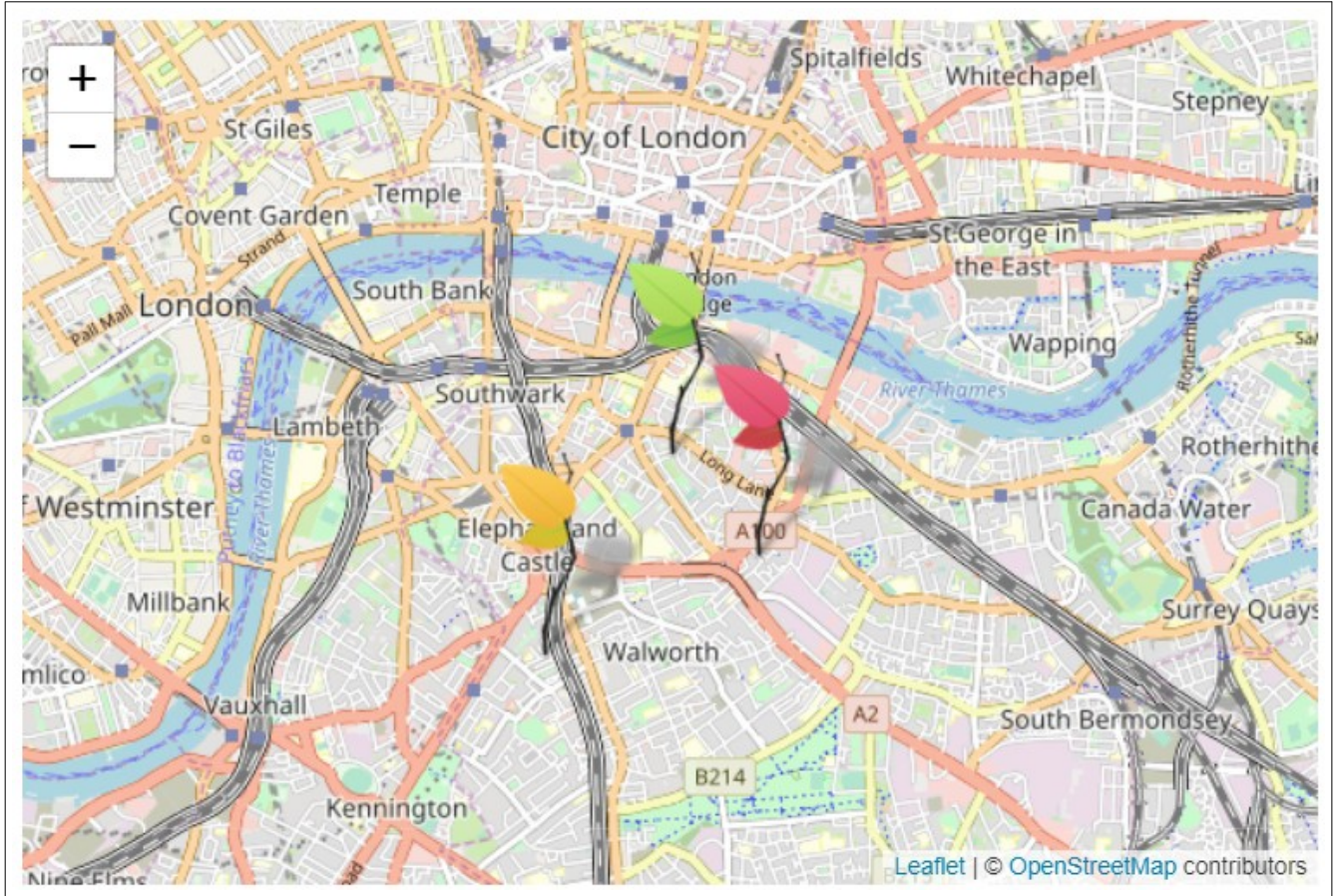


03. LEAFLET Custom Markers.

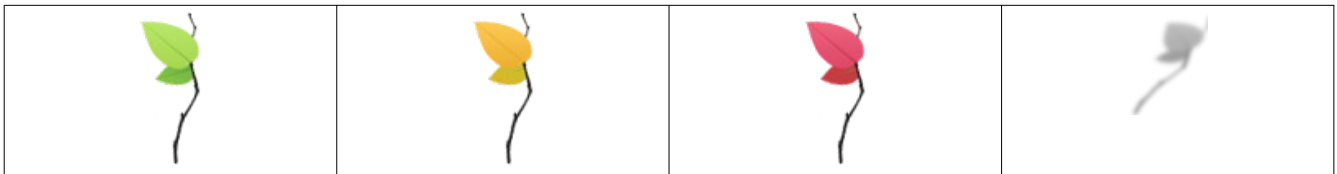
3.1. Markers with Custom Icons.

In this tutorial, you'll learn how to easily define your own icons for use by the markers you put on the map.



3.2. Preparing the images

To make a custom icon, we usually need two images — the actual icon image and the image of its shadow. For this tutorial, we took the Leaflet logo and created four images out of it — 3 leaf images of different colors and one shadow image for the three:



Note that the white area in the images is actually transparent.

3.3. Creating an icon

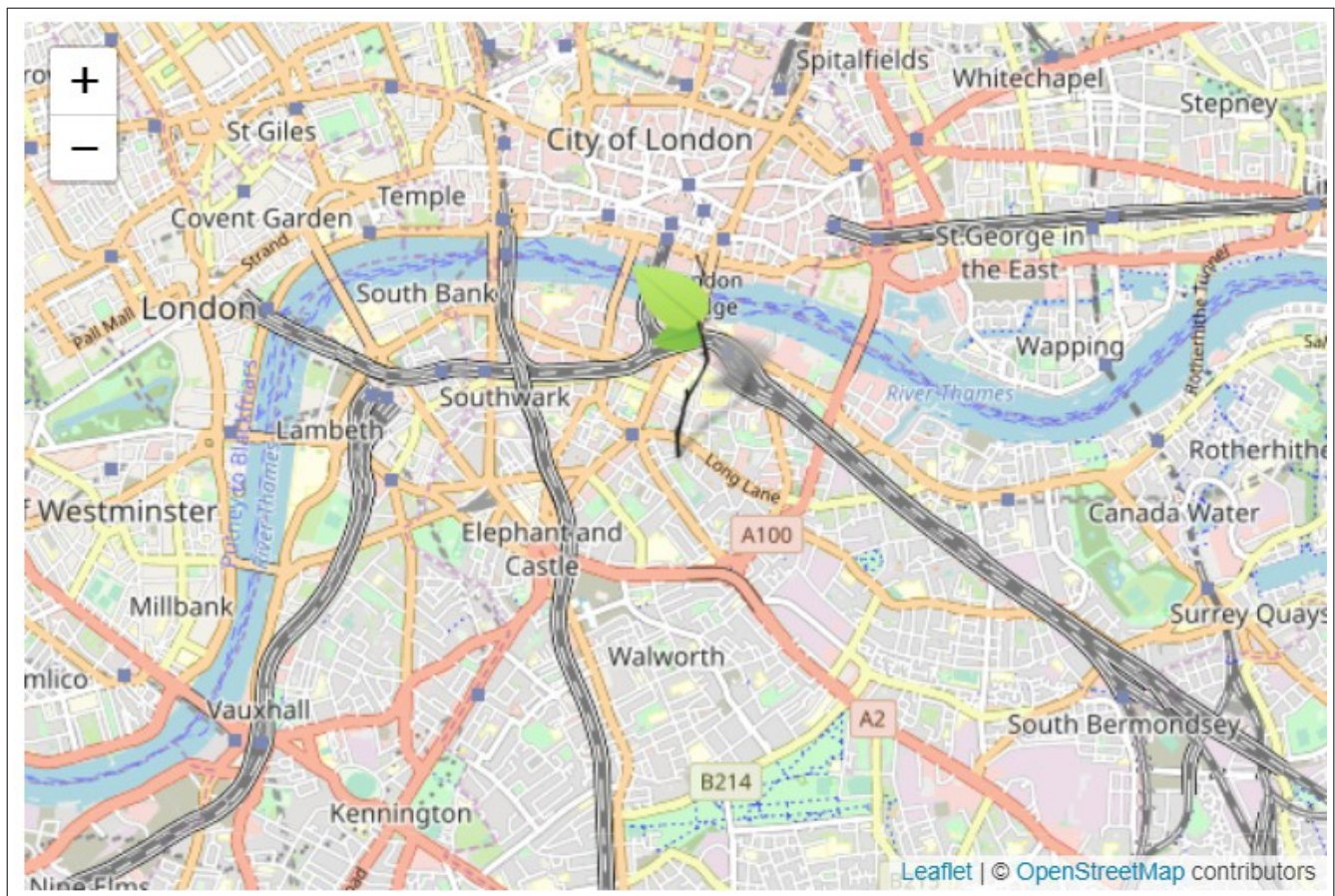
Marker icons in Leaflet are defined by L.Icon objects, which are passed as an option when creating markers. Let's create a green leaf icon: 2.3. Setting up the Map

```
var greenIcon = L.icon({
  iconUrl: 'leaf-green.png',
  shadowUrl: 'leaf-shadow.png',

  iconSize: [38, 95], // size of the icon
  shadowSize: [50, 64], // size of the shadow
  iconAnchor: [22, 94], // point of icon matches marker's location
  shadowAnchor: [4, 62], // the same for the shadow
  popupAnchor: [-3, -76] // point from popup should open relative
});
```

Now putting a marker with this icon on a map is easy:

```
L.marker([51.5, -0.09], {icon: greenIcon}).addTo(map);
```



2.4. Defining an Icon Class

What if we need to create several icons that have lots in common? Let's define our own icon class containing the shared options, inheriting from **L.Icon**! It's really easy in Leaflet:

Example:

```
var LeafIcon = L.Icon.extend({
  options: {
    shadowUrl: 'leaf-shadow.png',
    iconSize: [38, 95],
    shadowSize: [50, 64],
    iconAnchor: [22, 94],
    shadowAnchor: [4, 62],
    popupAnchor: [-3, -76]
  }
});
```

Now we can create all three of our leaf icons from this class and use them:

Example:

```
var greenIcon = new LeafIcon({iconUrl: 'leaf-green.png'}),
    redIcon = new LeafIcon({iconUrl: 'leaf-red.png'}),
    orangeIcon = new LeafIcon({iconUrl: 'leaf-orange.png'});
```

You may have noticed that we used the new keyword for creating LeafIcon instances. So why do all Leaflet classes get created without it? The answer is simple: the real Leaflet classes are named with a capital letter (e.g. L.Icon), and they also need to be created with new, but there are also shortcuts with lowercase names (L.icon), created for convenience like this:

Example:

```
L.icon = function (options) {
  return new L.Icon(options);
};
```

You can do the same with your classes too. OK, lets finally put some markers with these icons on the map:

Example:

```
L.marker([51.5, -0.09], {icon: greenIcon}).  
  addTo(map).bindPopup("I am a green leaf.");  
  
L.marker([51.495, -0.083], {icon: redIcon}).  
  addTo(map).bindPopup("I am a red leaf.");  
  
L.marker([51.49, -0.1], {icon: orangeIcon}).  
  addTo(map).bindPopup("I am an orange leaf.");
```