

10

Dessiner sur la carte

10.1. Cercles (de confinement), rectangles

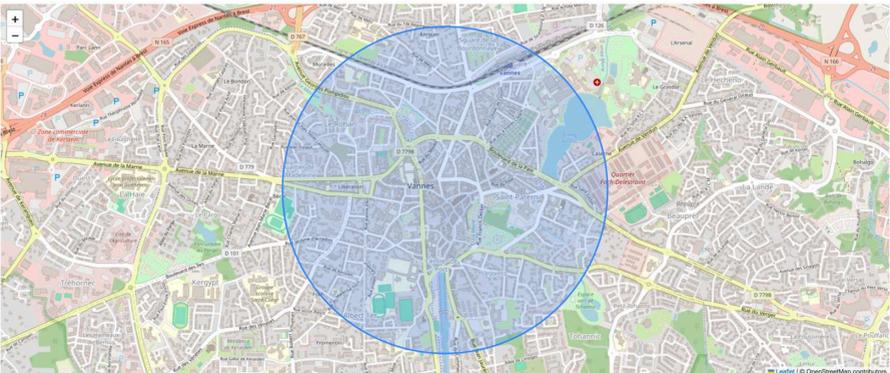
Tracer un cercle sur une carte est chose assez aisée, et également pratique pour indiquer une distance autour d'un point. Reprenons notre base habituelle d'une carte avec un centre, et rendons-nous à Vannes (ou Gwened si nous conservons les tuiles bretonnes).

```
let center = [47.65845, -2.75791];
let map = L.map("map").setView(center, 13);
let layer = L.tileLayer(
  "https://tile.osm.org/{z}/{x}/{y}.png",
  { attribution: `&copy; OpenStreetMap contributors` }
).addTo(map);
```

L'ajout du cercle va se faire en utilisant la bien nommée fonction `circle`, qui doit impérativement prendre un paramètre, le rayon, exprimé en mètres.

```
L.circle(center, { radius: 1000 }).addTo(map);
```

Figure 10.1 : Cercle bleu par défaut



Vous disposez de nombreuses options pour tracer le cercle, que nous allons mettre en œuvre au travers d'un souvenir, pas si ancien.

Nous avons vu pendant l'épidémie de COVID-19, lors du 2^e confinement (octobre/novembre 2020), la mise en place d'un certain nombre de restrictions. Parmi elles, la limitation de nos déplacements à un rayon de 1 km à vol d'oiseau autour du domicile (dans la limite d'1 heure par jour et avec une attestation). À partir du 5 avril 2021, le rayon était de 10 km (en France métropolitaine). Et au-delà des 10 km, une tolérance était autorisée jusqu'à 30 km pour des motifs tels que achats, établissement culturels ou cultuels (j'ai peut être sauté des étapes entre les divers confinements/déconfinement/reconfinement, mais le propos n'est pas là).

Bref, de nombreux sites sont apparus alors permettant de visualiser le fameux rayon autorisé ! Et beaucoup avec Leaflet... La plupart d'ailleurs associés à une géolocalisation automatique, ou une demande d'adresse... ce que nous verrons au chapitre [Géocodage](#). Pour l'instant restons à Vannes.

Si nous voulons placer nos cercles les uns sur les autres, il faut commencer par le plus large, puis empiler les plus petits. Pour passer en revue les options, chaque cercle aura des paramètres différents, qui sont assez explicites.

```
L.circle(center, {
  color: "red",           // couleur de contour
  fillColor: "transparent", // couleur de fond
  weight: 10,           // épaisseur du trait
  radius: 30000,       // rayon en mètres
  dashArray: "10,20",  // pointillés
}).addTo(map);

L.circle(center, {
  color: "white",
  fillColor: "#FFFFFF", // couleur de fond
  fillOpacity: 0.5,     // transparence entre 0 et 1
  radius: 10000,
}).addTo(map);
```

Notez que les cercles sont interactifs ; nous pouvons leur adjoindre un popup (ce que nous allons faire pour le dernier) et donc créer une variable associée.

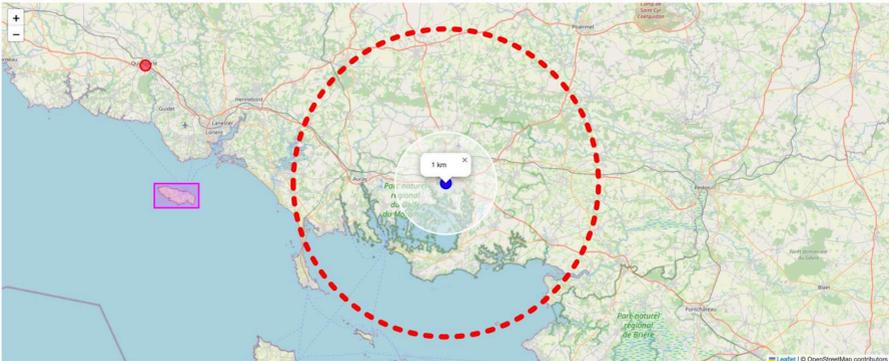
```
let circle = L.circle(center, {
  color: "blue",
  fillColor: "#30f",
  fillOpacity: 1,
  radius: 1000,
```

```
}).addTo(map);
circle.bindPopup("1 km").openPopup();
```

Les couleurs peuvent être nommées (en anglais) ou plus généralement exprimées en hexadécimal. L'option `dashArray` permet de faire des pointillés, en précisant le nombre de pixels colorés et le nombre de pixels sans couleur. On peut ainsi faire de pointillés complexes, en répétant les instructions:

```
// 10 pleins, 20 vides, 5 pleins, 40 vides...
dashArray: "10,20,5,40"
```

Figure 10.2 : Souvenirs de confinements... et un petit rectangle



Nous avons déjà croisé une fonction proche, `circleMarker` qui permet de créer un marqueur circulaire, ici en rouge sur Quimperlé, en haut à gauche.

```
L.circleMarker([47.86469, -3.54721], {
  radius: 10, // Rayon du cercle en pixels
  color: "red", fillColor: "#f03", fillOpacity: 0.5
}).addTo(map);
```

La principale différence est que `circle` aura un rayon en mètres, là où `circleMarker` sera en pixels. Autre différence, notable : au zoom, le `circle` deviendra de plus en plus petit, là où `circleMarker` restera à la même taille ; il sera donc plutôt utilisé pour marquer des points (un marqueur, donc), là où le cercle représentera des zones (d'influence ou de confinement).