A super short introduction to webmapping with Leaflet

Numa Gremling, Maptime Berlin, 21.02.2017

Slides and Data

http://geosysnet.de

http://geosysnet.de/en/downloads.html

Getting started

The Holy Trinity

- To get started with webmapping you first need a basic knowledge of web development

- And that requires just three things:
  - HTML
  - CSS
  - JavaScript

- Actually… four things: you also need a text editor ☺
- There are a lot of editors to assist you specifically with web development. Such a software is called IDE (Integrated Development Environment)
- Avoid a complex IDE at first!
- We will use Notepad++ tonight

HTML, CSS, JavaScript – What they do

HTML

the elements that are shown in the webpage

CSS

the appearance (styling) of these elements

JavaScript

functionality

HTML, CSS, JavaScript – Examples

HTML

heading, table, list, button

CSS

colors, positioning, sizes

JavaScript

something happens

Example - HTML

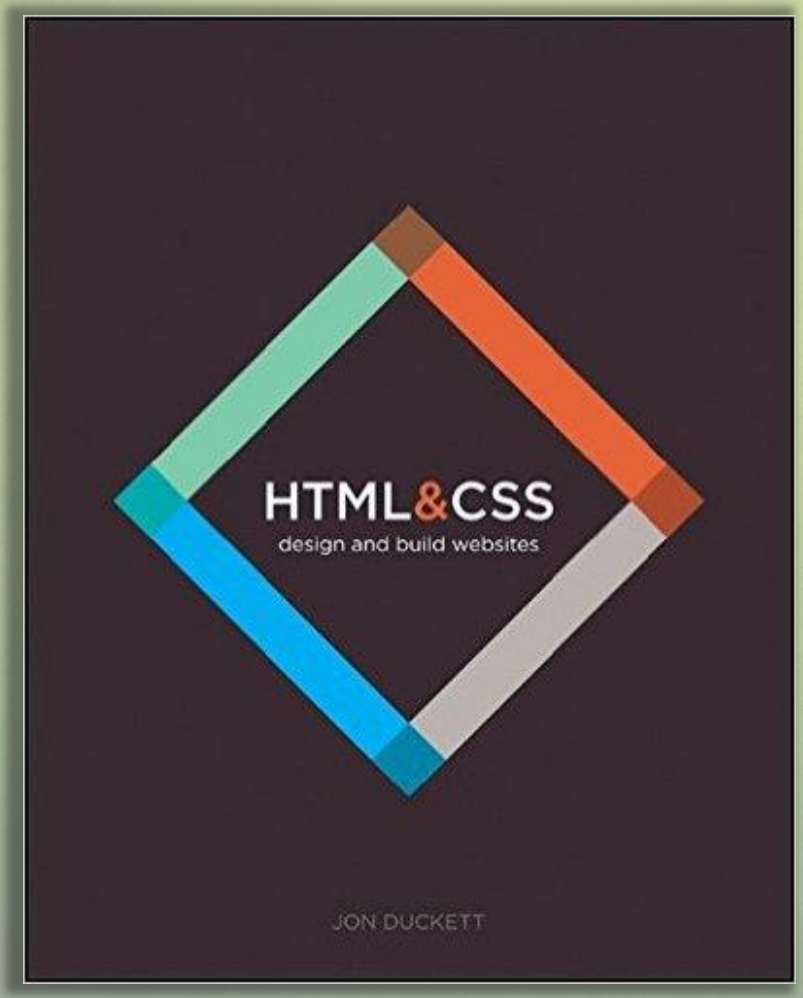
Example – HTML, CSS

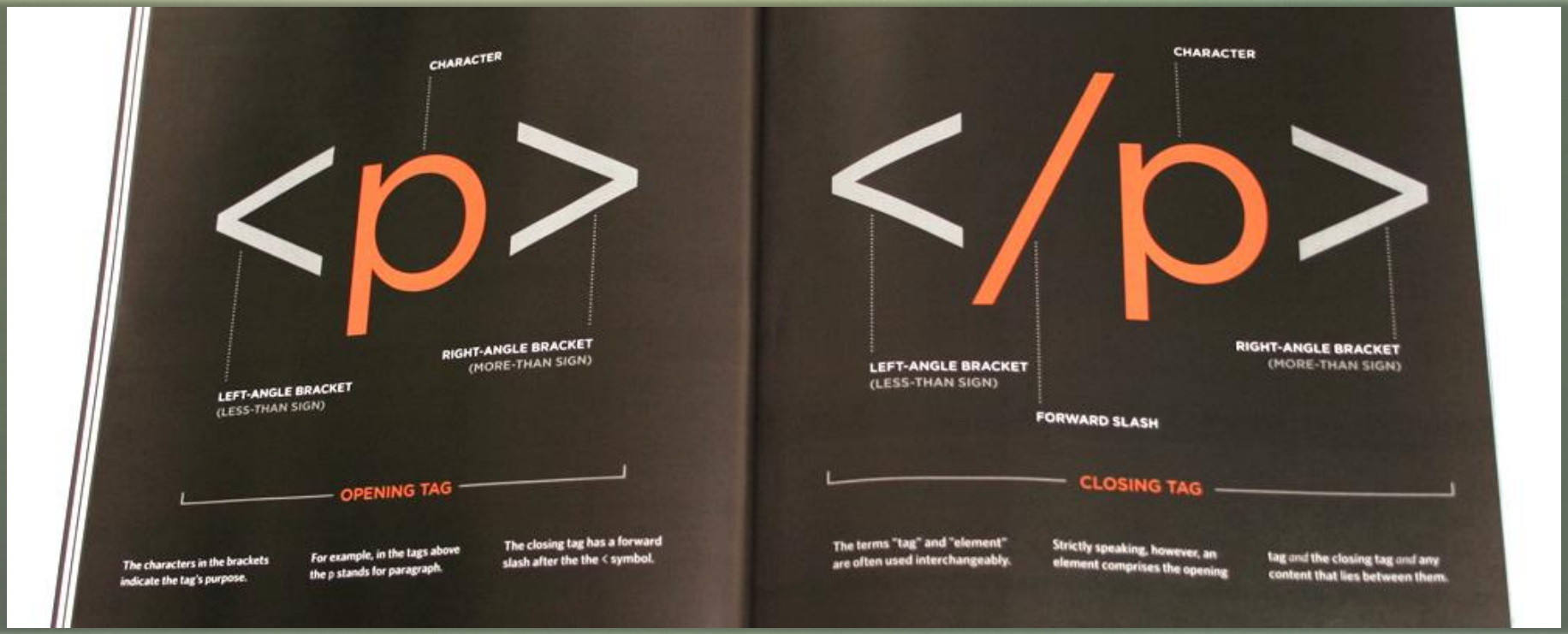
Example – HTML, CSS, JavaScript

click

Imagine that our little friend only appeared after we clicked the button ☺

Some resources – Books




CHARACTER

LEFT-ANGLE BRACKET
(LESS-THAN SIGN)
RIGHT-ANGLE BRACKET
(MORE-THAN SIGN)
OPENING TAG
The characters in the brackets indicate the tag's purpose.
For example, in the tags above the p stands for paragraph.
The closing tag has a forward slash after the the < symbol.
CHARACTER

LEFT-ANGLE BRACKET
(LESS-THAN SIGN)
FORWARD SLASH
RIGHT-ANGLE BRACKET
(MORE-THAN SIGN)
CLOSING TAG
The terms "tag" and "element" are often used interchangeably.
Strictly speaking, however, an element comprises the opening tag and the closing tag and any content that lies between them.

Some resources – Websites

- http://www.w3schools.com/
- https://www.codecademy.com

Leaflet

- By Vladimir Agafonkin
- Since 2010
- Leaflet: Past, Present, Future

https://www.youtube.com/watch?v=_P2SaCPbJ4w

Why pick Leaflet?

- Not as complex as other libraries
- Easy to learn
- Very well documented
- Large community to help you:
  - Google Group
    https://groups.google.com/forum/#!forum/leaflet-js
  - GIS Stack Exchange
    http://gis.stackexchange.com/questions/tagged/leaflet
  - Stack Overflow
    https://stackoverflow.com/questions/tagged/leaflet

What you will learn tonight

- Including Leaflet
- Creating a template
- Adding the map container
- Adding a basemap
- Adding markers
- Adding a polygon
- Adding a GeoJSON

Example 1

Including Leaflet

- Let’s first create an empty HTML page:

```
<!DOCTYPE html>
<html>
    <head>
        ...
    </head>

    <body>
        ...
    </body>
</html>
```

```
<!DOCTYPE html>
<html>
    <head>
        ...
    </head>

    <body>
        ...
    </body>
</html>
```

metadata

The actual content of the page; what you will see in the browser!

Getting and including Leaflet

- One way to get Leaflet:

http://leafletjs.com/download.html

- Two ways to use Leaflet:
  - Use a CDN
  - Download and use locally

Way 1: CDN

Using a Hosted Version of Leaflet

The latest stable Leaflet release is hosted on a CDN — to start using it straight away, place this in the `head` of your HTML code:

```
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.0.3/dist/leaflet.css" />
<script src="https://unpkg.com/leaflet@1.0.3/dist/leaflet.js"></script>
```

To get started with Leaflet we include these two lines in our <head>:

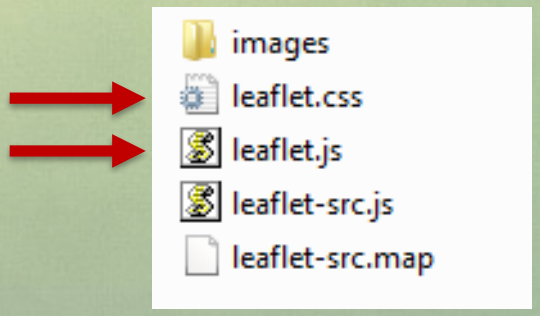
```
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.0.3/dist/leaflet.css" />
<script src="https://unpkg.com/leaflet@1.0.3/dist/leaflet.js"></script>
```

```
<!DOCTYPE>
<html>
    <head>
        <link rel="stylesheet" href="https://unpkg.com/leaflet@1.0.3/dist/leaflet.css" />
         <script src="https://unpkg.com/leaflet@1.0.3/dist/leaflet.js"></script>
    </head>

    <body>

    </body>
</html>
```

That’s it!
That is all it takes to use Leaflet!
Easy, isn’t it? ☺

Way 2: Download

- Download
- Unzip
- Include both the JS and CSS files, like we did in the previous step



Example 2

Creating our first map

The map container

- The map needs to be displayed *somewhere*
- We create a container that is dedicated to the map
- We use the HTML element
- We assign an ID to that div, so we can reference the div to add the map to it
- Let's call that ID *myMapContainer*

The map container

```
<!DOCTYPE html>
<html>
    <head>
        <link rel="stylesheet" href="https://unpkg.com/leaflet@1.0.3/dist/leaflet.css" />
        <script src="https://unpkg.com/leaflet@1.0.3/dist/leaflet.js"></script>
    </head>

    <body>

        <div id='myMapContainer'></div>

    </body>
</html>
```

1. We add our map container

Some styling

```
<!DOCTYPE html>
<html>
    <head>
        <link rel="stylesheet" href="https://unpkg.com/leaflet@1.0.3/dist/leaflet.css" />
        <script src="https://unpkg.com/leaflet@1.0.3/dist/leaflet.js"></script>
        <style>
            #myMapContainer {
                border: red 1px solid;
                width: 500px;
                height: 500px;
            }
        </style>
    </head>

    <body>
        <div id='myMapContainer'></div>
    </body>
</html>
```

2. We assign a style to that container

Sidenote:
IDs are selected using the pound sign (#).
So the ID *myMapContainer* is selected like this: *#myMapContainer*

IMPORTANT: in order for the map to be displayed its container must have a **height**!!!

The script tag

```
...

<body>

    <div id='myMapContainer'></div>

    <script>

    </script>

</body>
</html>
```

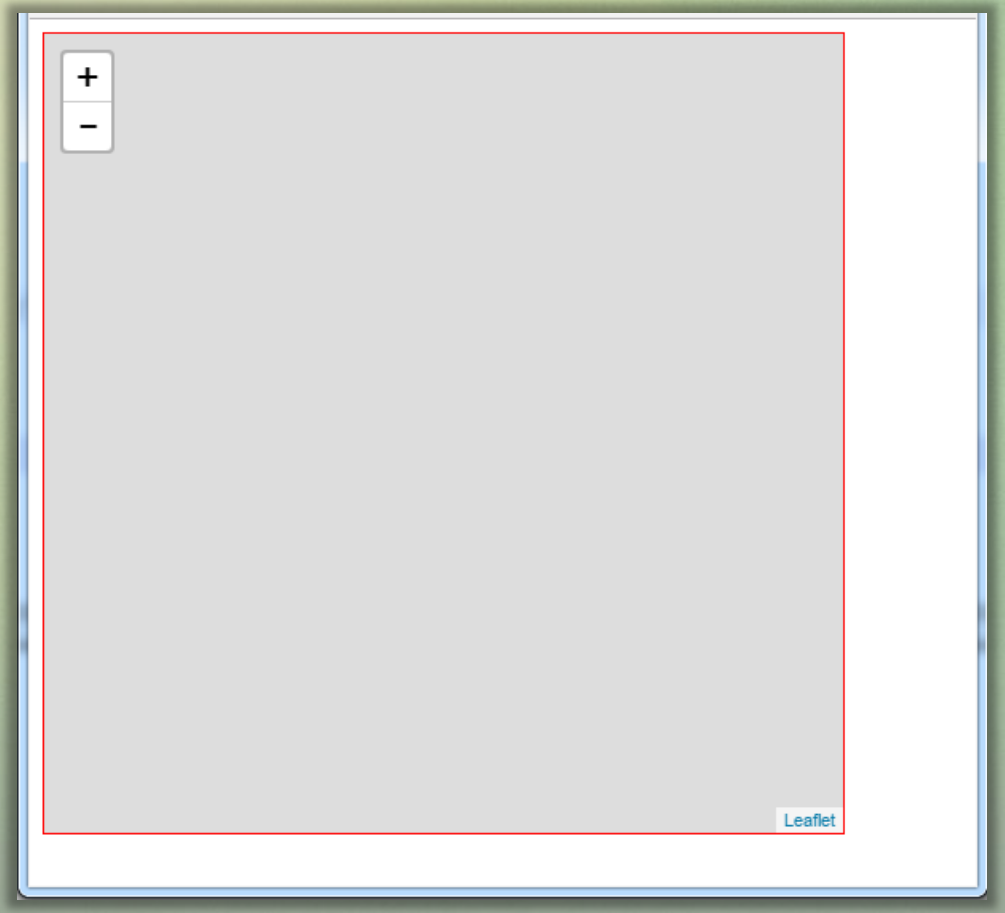
3. We add a script tag at the end of our body, so we can execute JavaScript

Creating the map

```
...

<body>

    <div id='myMapContainer'></div>

    <script>
        var myMap = L.map('myMapContainer');
    </script>

</body>
</html>
```

Where is our map?

- L.map creates a map container; it does not add an actual basemap or any data: we need to do that



- Let's add some options so our map loads in Berlin
- We need:
  - A center (latitude: 52.494 - longitude: 13.446)
  - A zoom level (10)
- Options are added with {curly braces}
- Available map options are shown here:

http://leafletjs.com/reference-1.0.3.html#map-factory

We need the options *center* and *zoom*

Adding a center and a zoom level

```
...
<body>

    <div id='myMapContainer'></div>

    <script>
        var myMap = L.map('myMapContainer', {
                                center: [52.494, 13.446],
                                zoom: 10
        });
    </script>

</body>
</html>
```

the **div** we created:

variable:
Lets us use the map later on

Leaflet Map **method**

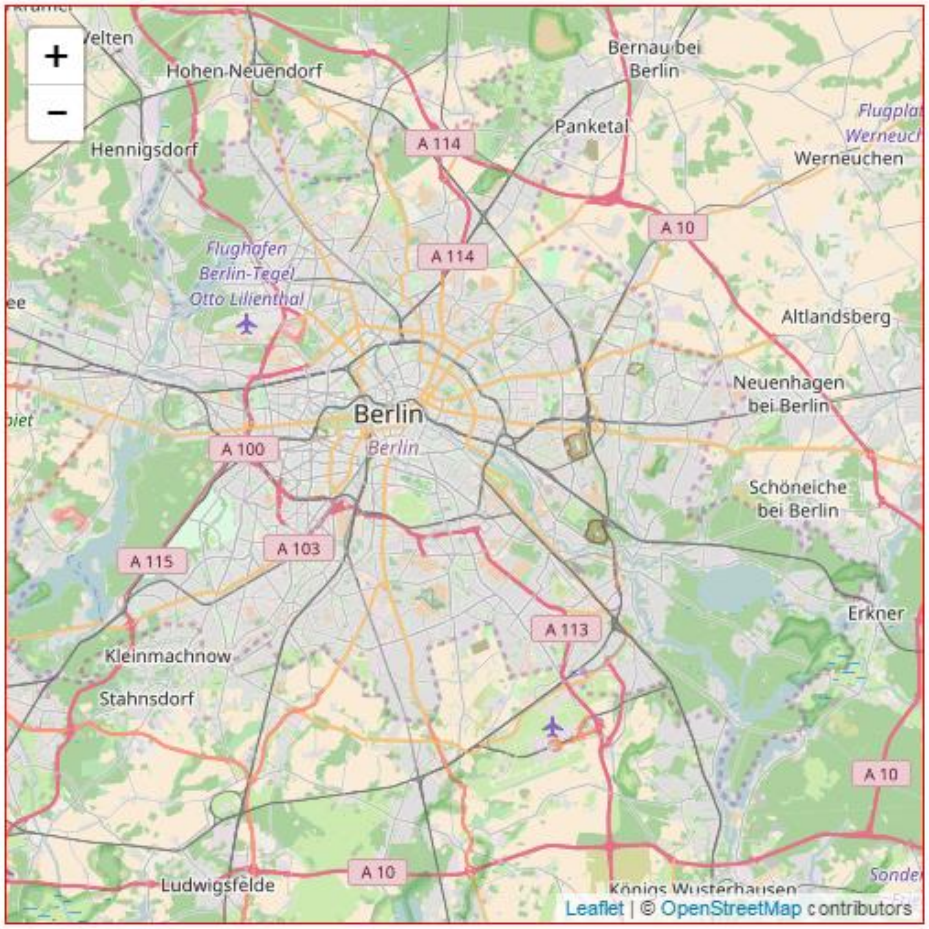
```
var myMap= L.map('myMapContainer', {
    center: [52.494, 13.446],
    zoom: 15
});
```

the **option** *center*

the **option** *zoom*

Adding a basemap

```
...
        <script>
            var myMap = L.map('myMapContainer', {
                                    center: [52.494, 13.446],
                                    zoom: 10
                    });

            var basemap = L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {
                                attribution: '&copy; <a href="http://osm.org/copyright">OpenStreetMap</a> contributors'
                        });

            basemap.addTo(myMap);
        </script>

  </body>
</html>
```


Hohen-Neuendorf
Bernau bei Berlin
Panketal
Hennigsdorf
A 114
Werneuchen
A 10
Flughafen Berlin-Tegel Otto Lilienthal
A 114
Altlandsberg
Neuenhagen bei Berlin
Berlin
Berlin
A 100
Schöneiche bei Berlin
A 103
A 115
Erkner
A 113
Kleinmachnow
Stahnsdorf
A 10
A 10
Ludwigsfelde
Leaflet | © OpenStreetMap contributors

```
<!DOCTYPE html>
<html>
	<head>
		<link rel="stylesheet" href="https://unpkg.com/leaflet@1.0.3/dist/leaflet.css" />
		<script src="https://unpkg.com/leaflet@1.0.3/dist/leaflet.js"></script>

		<style>
			#myMapContainer {
				border: red 1px solid;
				width: 500px;
				height: 500px;
			}
		</style>

	</head>

	<body>

		<div id='myMapContainer'></div>

		<script>
			var myMap = L.map('myMapContainer', {
								center: [52.494, 13.446],
								zoom: 10
				});

			var basemap = L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {
								attribution: '&copy; <a href="http://osm.org/copyright">OpenStreetMap</a> contributors'
				});

			basemap.addTo(myMap);
		</script>

	</body>
</html>
```

- More basemaps:

https://leaflet-extras.github.io/leaflet-providers/preview/

- Pick a basemap and copy the code:

Adding geometries

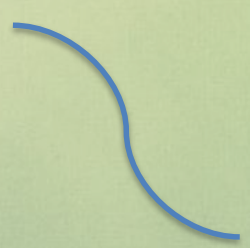
- Reminder: there are only three types of geometries in GIS:

Point Line Polygon

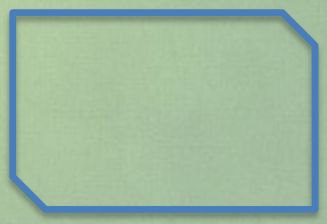
Leaflet jargon

- In Leaflet these geometries are called:

Marker



PolyLine



Polygon

Example 3

Adding markers

- Let’s add two famous Berlin landmarks:
  - Fernsehturm
    - Latitude: 52.520861
    - Longitude: 13.409564
  - Brandenburger Tor:
    - Latitude: 52.516312
    - Longitude: 13.377624

- A point is called marker in Leaflet
- To add it we use L.marker([latitude,longitude]);

<u>Brandenburger Tor</u>:

L.marker([52.516312, 13.377624]);

<u>Fernsehturm</u>:

L.marker([52.520861, 13.409564]);

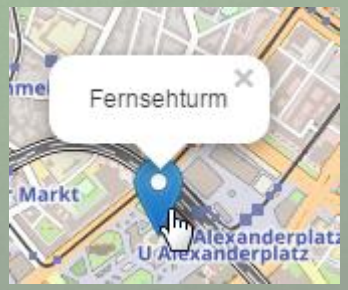
Adding the markers

- We assign the markers to variables
- Then we add the markers to the map

```
// Creating markers
var brandenburgerTor = L.marker([52.516312, 13.377624]);
var fernsehturm = L.marker([52.520861, 13.409564]);

// Adding the marker to the map
fernsehturm.addTo(myMap);
brandenburgerTor.addTo(myMap);
```

Adding popups

```
// Creating markers
var brandenburgerTor = L.marker([52.516312, 13.377624]);
var fernsehturm = L.marker([52.520861, 13.409564]);

// Adding the marker to the map
fernsehturm.addTo(myMap);
brandenburgerTor.addTo(myMap);

// Adding popups
fernsehturm.bindPopup('Fernsehturm');
brandenburgerTor.bindPopup('Brandenburger Tor');
```



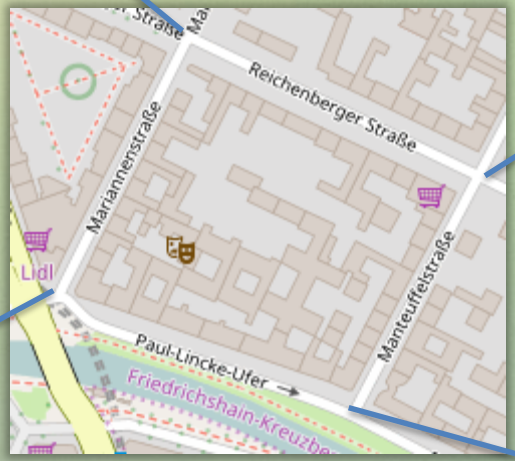
```
<script>
        var myMap = L.map('myMapContainer', {
                                        center: [52.52050, 13.39791],
                                        zoom: 14
        });

        var basemap = L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {
                                attribution: '&copy; <a href="http://osm.org/copyright">OpenStreetMap</a> contributors'
                        });

        basemap.addTo(myMap);

        // Creating markers
        var brandenburgerTor = L.marker([52.516312, 13.377624]);
        var fernsehturm = L.marker([52.520861, 13.409564]);

        // Adding the marker to the map
        fernsehturm.addTo(myMap);
        brandenburgerTor.addTo(myMap);

        // Adding popups
        fernsehturm.bindPopup('Fernsehturm');
        brandenburgerTor.bindPopup('Brandenburger Tor');

</script>
```

Example 4

Adding a polygon

- Let’s add the most important block in Berlin: the „Maptime block“ ☺

52.498107, 13.421515



52.497225, 13.424503

52.496493, 13.420244

52.495729, 13.423194

```
// Creating a polygon
var block = L.polygon(
                [
                        [52.498107, 13.421515],
                        [52.497225, 13.424503],
                        [52.495729, 13.423194],
                        [52.496493, 13.420244]
                ]
        );

// Adding the polygon to the map
block.addTo(myMap);
```

Example 5

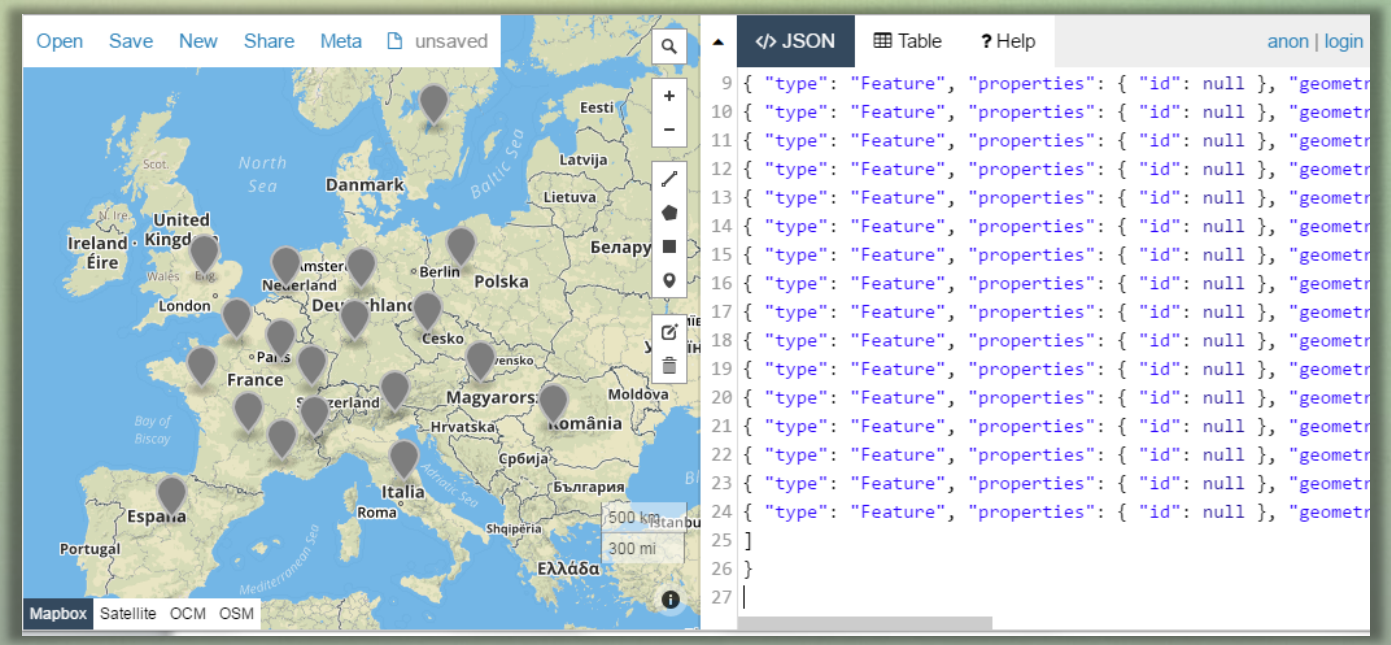
Adding a GeoJSON

- **Data**: we have a Shapefile of Berlin districts (Bezirke)
- **Goal**: we'd like to add it to our map
- **Problem**: Leaflet does not support shapefiles
- **Solution**: we convert the shapefile to a data format that Leaflet understands

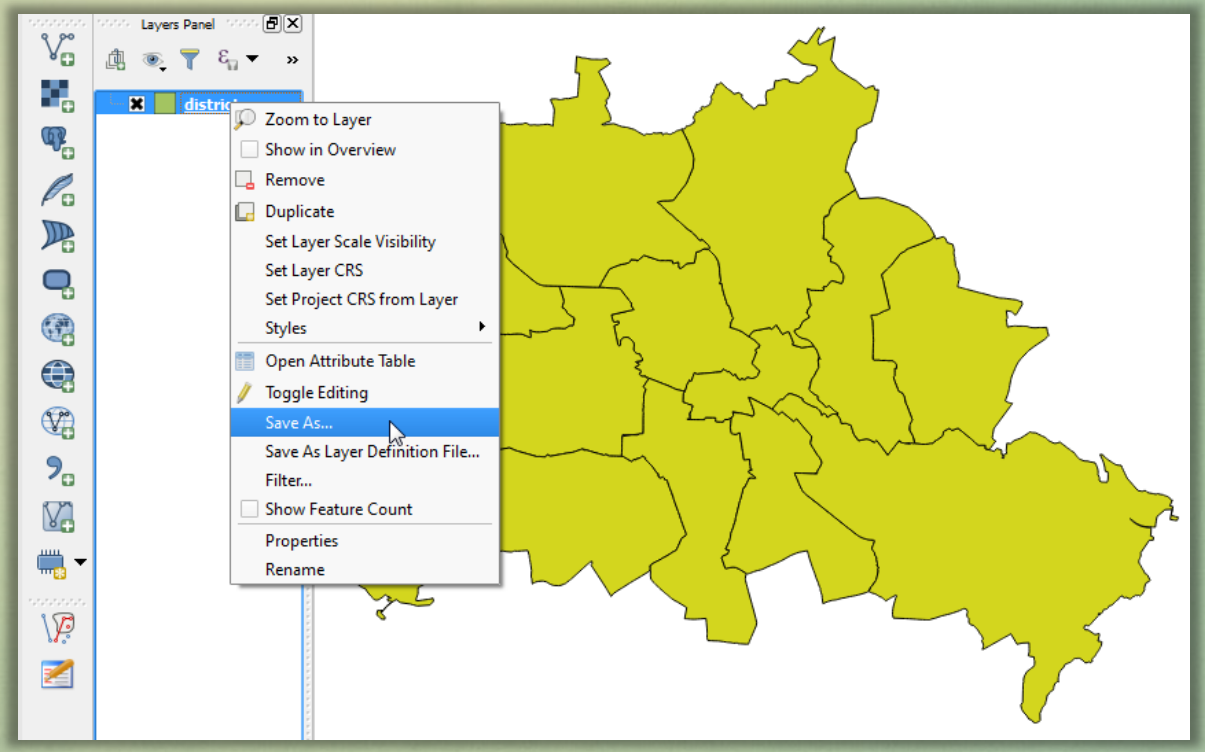
➔ We convert the shapefile to a GeoJSON

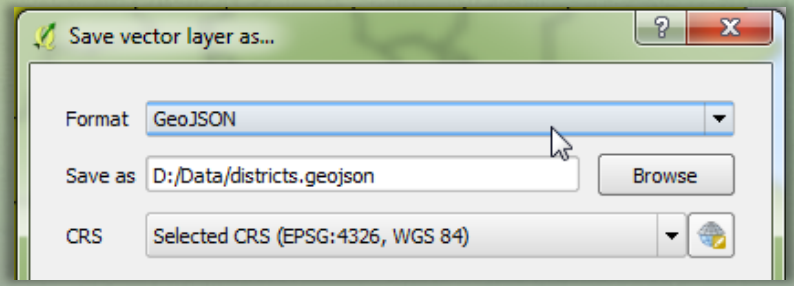
Shapefile to GeoJSON

- There are many ways to convert a shapefile to a GeoJSON, such as:
  - http://geojson.io/
  - ogr2ogr
  - QGIS



Using QGIS as a conversion tool





Always use EPSG:4326 (WGS 84)

Having a look at our QGIS export


```
{
"type": "FeatureCollection",
"crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:OGC:1.3:CRS84" } },

"features": [
{ "type": "Feature", "properties": { "ID": "01", "Name": "Mitte" }, "geometry": { "type": "
[ [ [ 13.360376473686552, 52.500257464788056 ], [ 13.3600135516529, 52.500368980656766 ], [
52.50097976721959 ], [ 13.357650337439214, 52.501054705260287 ], [ 13.355501607041377, 52.5
13.355368798820084, 52.501616554472022 ], [ 13.353684738096486, 52.501996404674834 ], [ 13.
52.502138655624456 ], [ 13.352267474646215, 52.502331930000089 ], [ 13.352252972395277, 52.
13.350139351067776, 52.50283112844226 ], [ 13.349889920351897, 52.502919513091513 ], [ 13.3
52.502984408267956 ], [ 13.349270634279481, 52.503042977621618 ], [ 13.34615360021364, 52.5
13.345732376739214, 52.503887464169082 ], [ 13.344623419382419, 52.50412297275161 ], [ 13.3
52.504236795219612 ], [ 13.343096841454232, 52.504488675987915 ], [ 13.34260866924555, 52.5
13.341576245149099, 52.504836400697606 ], [ 13.340856397704753, 52.505007544175868 ], [ 13.
52.505009000106455 ], [ 13.340778224037052, 52.505033080515567 ], [ 13.340474384566676, 52.
13.340136844888921, 52.505148355942623 ], [ 13.339887803135635, 52.505166623911329 ], [ 13.
52.505159708772005 ], [ 13.339328390467749, 52.505132916490368 ], [ 13.339060653672355, 52.
13.339046413490692, 52.505084919877063 ], [ 13.338891991311751, 52.505438300213314 ], [ 13.
52.505930188805813 ], [ 13.33720213215674, 52.505760393715001 ], [ 13.336890017386891, 52.5
13.336633775743197, 52.505741928950165 ], [ 13.335802651481149, 52.505818073407617 ], [ 13.
52.505839212941297 ], [ 13.335447155742678, 52.505903469184716 ], [ 13.334005417168726, 52.
13.3340712967436, 52.506617209767654 ], [ 13.333777297078392, 52.506755776253826 ], [ 13.33
52.508136067298054 ], [ 13.334885264698324, 52.508245796623306 ], [ 13.334887317929661, 52.
```

Assigning the GeoJSON to a variable

- We can now paste that GeoJSON into our script and assign it to a variable, so we can use it later on to create a „Leaflet GeoJSON“

```
var districtsRaw =  {
    "type": "FeatureCollection",
    "crs": { "type": "name", "pro

    "features": [
    { "type": "Feature", "propert
    "coordinates": [ [ [ 13.36037
```

Creating a Leaflet GeoJSON

```
// Creating a Leaflet GeoJSON
var districtsFinal = L.geoJson(districtsRaw);

// adding the GeoJSON
districtsFinal.addTo(myMap);
```

Thanks, Contact, Questions

console.log('Thank you! :-)!');

- gis.stackexchange.com/users/23263/britishsteel StackExchange
- numa.gremling@geosysnet.de
- twitter.com/Gremling89

- geosysnet.de
- gis-trainer.de
- twitter.com/geoSYSnet
- twitter.com/gis_trainer

New: instagram.com/gis_trainer/

Slides and data: http://geosysnet.de/en/downloads.html

Imprint

gis-trainer.de / geoSYS
Pflügerstr. 56
12047 Berlin, Germany

Telefon: 030/82070659
Telefax: 030/82070658

Email: info@gis-trainer.de / info@geosysnet.de
Website: www.gis-trainer.de / www.geosysnet.de